



Natural for UNIX

エディタ

バージョン 8.4.1

2017 年 10 月

ADABAS & NATURAL

このマニュアルは Natural バージョン 8.4.1 およびそれ以降のすべてのリリースに適用されます。

このマニュアルに記載される仕様は変更される可能性があります。変更は以降のリリースノートまたは新しいマニュアルに記述されます。

Copyright © 1992-2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Software AG およびその子会社が所有する登録商標および特許の詳細については、<http://documentation.softwareag.com/legal/> を確認してください。

本ソフトウェアの一部にはサードパーティ製製品が含まれています。サードパーティの著作権表示およびライセンス規約については『License Texts, Copyright Notices and Disclaimers of Third-Party Products』を参照してください。このドキュメントは製品ドキュメントセットの一部であり、<http://documentation.softwareag.com/legal/> 上、またはライセンス製品のルートインストールディレクトリ内にあります。

本ソフトウェアの利用は、Software AGのライセンス規約に則って行われるものとします。ライセンス規約は製品ドキュメントセット内、<http://documentation.softwareag.com/legal/> 上、またはライセンス製品のルートインストールディレクトリ内にあります。

ドキュメント IDは: NATUX-NNATEDITORS-841-20200614JA

目次

前書き	vii
1	1
表記規則	2
オンライン情報	2
データ保護	3
I プログラムエディタ	5
2 プログラムエディタの起動	7
3 プログラムエディタの終了	9
Exit 機能	10
4 上部の情報行	13
5 エディタコマンド行	15
6 接頭辞エリア	17
行コマンドの入力	18
7 編集エリア	19
テキストの挿入	20
テキストのコピーおよび貼り付け	21
データウィンドウを使用したテキストのコピーまたは移動	22
テキストの検索および置換	26
ソースコードのフォーマット	27
ソースコードのチェック	28
8 画面分割モード	29
9 エディタコマンド	33
ADVANCE	35
AORDER	36
AUTOSAVE	36
BNDS	36
CANCEL	37
CAPS	37
CENTER	37
CHANGE	38
COLS	41
CWINDOW	41
DELETE	41
DWINDOW	43
DX, DY, DX-Y	43
EMPTY	44
EX, EY, EX-Y	44
EXCLUDE	44
EXIT	46
FIND	47
FLIP	49
HEX	49
HOME	50

INCLUDE	50
JLEFT	50
JRIGHT	51
JUSTIFY	52
LABEL	52
LC	53
LIMIT	55
LOCATE	55
LOG	56
MASK	56
MWINDOW	57
NEXT	57
ORDER	58
POINT	58
POWER	58
PROF	59
PROFILE	59
PROTECT	59
RCHANGE	60
RESET	60
RFIND	60
SET TYPE	61
SHIFT	61
SORT	62
SPLIT	62
SWAP	63
TABS	63
UC	67
UNDO	67
WINDOW	68
X	68
XSWAP	69
Y	69
共通のコマンドオプション	69
10 スクロール用エディタコマンド	73
11 行コマンド	75
12 エディタプロファイル	81
プロファイル設定の表示および非表示	82
一時的に使用するためのプロファイル設定の変更	83
永続的に使用するためのプロファイル設定の変更	84
13 エディタバッファプール設定	91
複数のエディタセッション	92
14 ソースの保存およびカタログ	93
II データエリアエディタ	95
15 データエリアエディタ	97

データエリアエディタの起動	98
編集モードおよびコマンドモード	101
編集エリア	102
行コマンド	111
エディタコマンドおよびファンクションキー	114
データエリアの保存とカタログ	118
ヘルプ情報および選択オプション	119
III マップエディタ	121
16 マップエディタ	123
マップの作成	124
マップエディタの起動	125
Map Editor メニュー	127
テキスト定数の作成	130
ユーザー定義変数の作成	131
ユーザー定義変数の変更 - フィールド編集	132
データ定義の選択	146
パラメータまたはローカルデータ定義用フィールドの定義	149
マッププロファイル	151
事後割り当て	155
フィールドに基づいた処理	155
IV DDM サービス	159
17 操作の原理	161
DDM の保存 - FDDM システムファイル	162
Restrictions of Use	163
18 DDM サービスの起動と終了	165
19 DDM メンテナンス機能の使用	167
メンテナンス機能のリストと実行	168
メンテナンス機能の説明	169
20 DDM の作成	171
DDM のコピー	172
[<CREATE>] を使用した、Adabas からの作成	174
[<CREATE>] を使用した、SQL からの作成	177
SQL からの複数の DDM の作成	181
[<CREATE>] を使用した、Tamino からの作成	182
21 DDM エディタの起動と終了	187
DDM メンテナンスを使用したエディタの起動	188
EDIT によるエディタの起動	189
エディタの終了	190
22 DDM エディタの使用	193
DDM ヘッダー情報	194
フィールド属性列	197
編集および機能実行のためのコマンド	202
拡張フィールド属性の指定	208
エディタの設定 - サービスプロファイル	214
23 DDM の保存とカタログ	219

24 DDM のリスト	221
[DDM Maintenance] メニューによる DDM のリスト	222
LIST による DDM のリスト	224
25 異なる環境での DDM の管理	227
26 Adabas または RDBMS のデータ変換	229
Adabas	230
Adabas D	230
Adabas SQL サーバー	231
DB2	231
Informix	232
Oracle	233
Sybase	234
Microsoft SQL Server	235
関連トピック	236
27 Tamino のデータ変換	237
組み込み Tamino XML スキーマ言語のデータタイプ	238
Tamino XML スキーマ構造	240
Tamino XML スキーマ言語の多重度	241

前書き

このドキュメントでは、Natural で使用できるすべてのエディタについて説明します。

エディタの使用に関するチュートリアルについては、『ファーストステップ』ドキュメントを参照してください。

Natural エディタでサポートされている Unicode とコードページの詳細については、『*Unicode とコードページのサポート*』ドキュメントの「*SPoD 環境のエディタ*」を参照してください。

-  **注意:** ソースコードの作成にはNaturalエディタのみ使用することをお勧めします。Natural エディタ以外のエディタを使用して Natural の管理外で作成されたソースは、Natural では読み込みおよび解釈できません。Natural エディタ以外のエディタは、そのエディタが Natural プログラムエディタの代わりとして、Natural プロファイルパラメータ `EDITOR` を使用して起動された場合にのみ、使用できます。このパラメータを使用すると、ソースコードが正しく解釈されます。

『エディタ』ドキュメントは次の項目で構成されています。

- | | |
|----------------------------|--|
| プログラムエディタ | Naturalのプログラム、サブプログラム、サブルーチン、クラス、コピーコード、ヘルプルーチン、関数およびテキストオブジェクトを作成および変更する場合に使用する、プログラムエディタについて説明します。 |
| データエリアエディタ | ローカル、グローバル、パラメータの各データエリアを作成および変更する場合に使用する、データエリアエディタについて説明します。 |
| マップエディタ | マップ（画面レイアウト）を作成および変更する場合に使用する、マップエディタについて説明します。 |
| DDM サービス | Naturalデータ定義モジュール（DDM）を作成、管理、および削除する場合に使用する、DDM サービスについて説明します。 |

1

■ 表記規則	2
■ オンライン情報	2
■ データ保護	3

表記規則

規則	説明
太字	画面上の要素を表します。
モノスペースフォント	<i>folder.subfolder:service</i> という規則を使用して webMethods Integration Server 上のサービスの保存場所を表します。
大文字	キーボードのキーを表します。同時に押す必要があるキーは、プラス記号 (+) で結んで表記されます。
斜体	独自の状況または環境に固有の値を指定する必要がある変数を表します。本文で最初に出現する新しい用語を表します。
モノスペースフォント	入力する必要があるテキストまたはシステムから表示されるメッセージを表します。Program code.
{}	選択肢のセットを表します。ここから1つ選択する必要があります。中カッコの内側にある情報のみを入力します。{}記号は入力しません。
	構文行で相互排他的な2つの選択肢を区切れます。いずれかの選択肢を入力します。 記号は入力しません。
[]	1つ以上のオプションを表します。大カッコの内側にある情報のみを入力します。[]記号は入力しません。
...	同じ種類の情報を複数回入力できることを示します。情報だけを入力してください。実際のコードに繰り返し記号 (...) を入力しないでください。

オンライン情報

Software AG マニュアルの Web サイト

マニュアルは、Software AG マニュアルの Web サイト (<http://documentation.softwareag.com>) で入手できます。このサイトでは Empower クレデンシャルが必要です。Empower クレデンシャルがない場合は、TECHcommunity Web サイトを使用する必要があります。

Software AG Empower 製品のサポート Web サイト

もしまだ Empower のアカウントをお持ちでないのなら、こちらへ empower@softwareag.com 電子メールにて あなたのお名前、会社名、会社の電子メールアドレスをお書きの上、アカウントを請求してください。

いったんアカウントをお持ちになれば、Empower <https://empower.softwareag.com/> の eService セクションにて サポートインシデントをオンラインで開くことができます。

製品情報は、Software AG Empower 製品のサポート Web サイト (<https://empower.softwareag.com>) で入手できます。

機能および拡張機能に関するリクエストの送信、製品の可用性に関する情報の取得、[製品](#)のダウンロードを実行するには、Products に移動します。

修正に関する情報を取得し、早期警告、技術論文、Knowledge Base の記事を読むには、[Knowledge Center](#) に移動します。

もしご質問があれば、こちらのhttps://empower.softwareag.com/public_directory.asp グローバルサポート連絡一覧の、あなたの国の電話番号を選んで、わたくし共へご連絡ください。

Software AG TECHcommunity

マニュアルおよびその他の技術情報は、Software AG TECHcommunity Web サイト (<http://techcommunity.softwareag.com>) で入手できます。以下の操作を実行できます。

- TECHcommunity クレデンシャルを持っている場合は、製品マニュアルにアクセスできます。TECHcommunity クレデンシャルがない場合は、登録し、関心事の領域として [マニュアル] を指定する必要があります。
- 記事、コードサンプル、デモ、チュートリアルにアクセスする。
- Software AG の専門家によって承認されたオンライン掲示板フォーラムを使用して、質問したり、ベストプラクティスを話し合ったり、他の顧客が Software AG のテクノロジをどのように使用しているかを学んだりすることができます。
- オープンスタンダードや Web テクノロジを取り扱う外部 Web サイトにリンクできます。

データ保護

Software AG 製品は、EU一般データ保護規則(GDPR)を尊重した個人データの処理機能を提供します。該当する場合、適切な手順がそれぞれの管理ドキュメントに記載されています。

I プログラムエディタ

Natural プログラムエディタを使用すると、タイプがプログラム、サブプログラム、サブルーチン、ヘルプルーチン、コピーコード、テキスト、クラス、または機能である Natural オブジェクトのソースコードを作成および変更できます。

関連トピック：

Natural エディタでサポートされている Unicode とコードページの詳細については、『*Unicode とコードページのサポート*』ドキュメントの「*SPoD 環境のエディタ*」を参照してください。

[プログラムエディタの起動](#)

[プログラムエディタの終了](#)

[上部の情報行](#)

[エディタコマンド行](#)

[接頭辞エリア](#)

[編集エリア](#)

[画面分割モード](#)

[エディタコマンド](#)

[スクロール用エディタコマンド](#)

[行コマンド](#)

[エディタプロファイル](#)

[エディタのバッファプール設定](#)

[ソースの保存およびカタログ](#)

2 プログラムエディタの起動

▷手順 2.1. プログラムエディタを起動するには

- 『システムコマンド』ドキュメントの説明に従って、システムコマンド EDIT を使用します。

プログラムエディタが起動されると、次の例のようなエディタ画面が表示されます。

```
>> -----Columns 001 072 << Program SAGDEMO  Lines 14      User SAG      ←
Command ===>                                         Mode   Struct  Lib  SAGTEST ←
***** ***** top of data *****
000010 ** Example 'SAGDEMO': DISPLAY
000020 *****
000030 DEFINE DATA LOCAL
000040 1 VIEWEMP VIEW OF EMPLOYEES
000050 2 PERSONNEL-ID
000060 2 NAME
000070 2 BIRTH
000080 2 JOB-TITLE
000090 END-DEFINE
000100 *
000110 READ (3) VIEWEMP BY BIRTH
000120 DISPLAY PERSONNEL-ID NAME JOB-TITLE
000130 END-READ
000140 END
***** ***** bottom of data *****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Save  Exit  Run   Rfind  Stow  -      +      Check  Home  Undo  Canc
```

エディタ画面には、上から順に、[上部の情報行](#)、[エディタコマンド行](#)、[接頭辞エリア](#)、[編集エリア](#)、およびPFキー行（「[PA/PFキーの割り当ての表示および変更](#)」で説明）があります。これらの項目については、以降のセクションで説明します。

3 プログラムエディタの終了

■ Exit 機能	10
-----------------	----

➤手順 3.1. プログラムエディタを終了するには

- コマンド行（[Command ==>]）で、次のいずれかのエディタコマンドを入力します。

EXIT

または

CANCEL

または

.

（ピリオド）

または：

エディタ画面で PF3 キーまたは PF12 キーを押します。

エディタプロファイル設定によっては、エディタを終了する前に、次のセクションで説明している **[EXIT Function]** ウィンドウが表示されます。

Exit 機能

エディタプロファイルオプション **[Prompt Window for Exit Function]** または **[Prompt Window for Cancel Function]** が [Y] に設定されている場合は、Exit 機能を実行するときにいつでも終了できます（PF3 キーまたは EXIT エディタコマンド）またはキャンセル機能（PF12 キーまたは CANCEL エディタコマンド）を実行するたびに、**[EXIT Function]** プロンプトウィンドウが表示されます。このウィンドウには、次のオプションが用意されています。

オプション	説明
Save and Exit	現在のソースコードに加えたすべての変更を保存した後、エディタを終了します。
Exit without Saving	最後の保存以降に現在のソースコードに加えた変更を保存せずにエディタを終了します。
Resume Function	エディタを終了せず、変更を保存しません。プロンプトウィンドウを閉じて、現在の関数を再開します。

[Prompt Window for Exit Function] または **[Prompt Window for Cancel Function]** が [N] に設定されている場合、対応する Exit 機能またはキャンセル機能によってエディタが直ちに終了します。このとき、プロンプトウィンドウは表示されません。ソースを最後に保存した後に現在のソースに加えられた変更内容は保存されません。

ソースの保存の詳細については、「ソースの保存およびカタログ」を参照してください。エディタプロファイルオプションの設定の詳細については、「エディタプロファイル」を参照してください。

4 上部の情報行

エディタ画面の上部の情報行は、二重の大なり記号 (>>) で示されます。この行には、左から順に次の項目が表示されます。

■ Columns

現在表示されている最初の列と最後の列です。72列目を超えてテキストを入力することは可能ですが、他のプラットフォームでもデータを確実に処理できるようにするために、列1～72のみを使用してください。

■ オブジェクトタイプ

現在ソースワークエリアに表示されているオブジェクトのタイプ。プログラムエディタを起動したときにオブジェクトタイプまたはオブジェクト名が指定されていなかった場合は、デフォルトでオブジェクトタイプ **Program** が表示されます。

オブジェクトタイプは、[SET TYPE](#) エディタコマンドを使用して変更できます。

■ Object Name

現在ソースワークエリアに表示されているオブジェクトの名前。ソースワークエリアが空の場合、または現在のソースコードがソースオブジェクトとしてまだ保存されていない場合、名前は表示されません（「[ソースの保存およびカタログ](#)」も参照）。

■ Lines

エディタで現在使用されている行の総数（ソース行および情報行）。

■ User

現在のユーザーの ID。

5 エディタコマンド行

エディタコマンド行は **[Command ==>]** で示されます。コマンド行では、次のいずれかのコマンドを入力できます。

- 任意の Natural システムコマンド。

例えば、次のようにになります。システムコマンド **CHECK** でソースコードの構文チェックを行ったり、**SAVE** でソースコードを保存したりします（「[ソースの保存およびカタログ](#)」も参照）。

オブジェクトソースの管理と使用に関連するその他のシステムコマンドについては、『システムコマンド』ドキュメントの「Natural オブジェクトによるアプリケーションの管理」を参照してください。

- エディタコマンド（「[エディタコマンド](#)」および「[スクロール用エディタコマンド](#)」を参照）。
- 実行する Natural プログラムの名前。

また、コマンド行には、左から順に次の項目が表示されます。

- **Mode**

現在有効なプログラミングモードとして、ストラクチャード（**[Struct]**）またはレポートティング（**[Report]**）が示されます。Natural オブジェクトがソースワークエリアに読み込まれると、モードは、そのオブジェクトが保存されたときに有効だったモードに設定されます（「[ソースの保存およびカタログ](#)」も参照）。

ストラクチャードモードとレポートティングモードの違いの詳細については、『プログラミングガイド』の「プログラミングモードの目的」セクションを参照してください。

モードをストラクチャードとレポートティングとで切り替えるには、**[Report]** の最初の桁に「**S**」を入力するか、**[Struct]** の最初の桁に「**R**」を入力し、**ENTER** キーを押します。

ストラクチャードモードが有効になったかどうかをチェックする場合、またはストラクチャードモードがオンに切り替える場合は、『プログラミングガイド』の「プログラミングモードの

「設定と変更」の説明に従って、現在のNaturalセッションのプログラミングモードも設定できます。

■ Lib

現在ログオンしているライブラリ。

6 接頭辞エリア

- 行コマンドの入力 18

エディタ画面の左端6桁は接頭辞エリアと呼ばれます。接頭辞エリアには次の内容が表示されます。

- ソースコードが含まれている行の6桁の行番号（000010など）。



注意: 技術的な理由から6桁表示されますが、内部的に処理されるのは4桁です。

- 空のソース行を示す一連のアポストロフィ（'……'）。挿入モードが有効の場合に表示されます。このような行にテキストを入力してENTERキーを押すと、アポストロフィが行番号に置き換わります。
- 空の編集エリアを示す一連のアスタリスク（*****）。この場合、関連セクションで説明しているI（挿入）行コマンドのいずれかを使用してテキストを入力するには、挿入モードを有効にする必要があります。
- 行の内容または状態を示すテキスト。例えば、次のようにになります。

=prof>：この行に現在のエディタプロファイル設定に関する情報が表示されていることを示します。

=col>：この行に現在の列位置が表示されていることを示します。

==chg>：この行で文字列が置換されたことを示します。

.x：この行がラベルでマークされていることを示します。

▷手順 6.1. 接頭辞エリアのテキストを非表示にし、保留状態のコマンドをリセットするには

- コマンド行で、次のエディタコマンドを入力します。

RESET

行コマンドの入力

接頭辞エリアは、ソース行またはソーステキストの番号の表示の他に、関連セクションで説明している[行コマンド](#)の入力に使用します。

接頭辞エリアを保護してソース行が上書きされないようにするには、「エディタコマンド」で説明している[PROTECT](#) エディタコマンドを使用します。

7 編集エリア

■ テキストの挿入	20
■ テキストのコピーおよび貼り付け	21
■ データウィンドウを使用したテキストのコピーまたは移動	22
■ テキストの検索および置換	26
■ ソースコードのフォーマット	27
■ ソースコードのチェック	28

編集エリアは、空であるか、または「[プログラムエディタの起動](#)」のプログラムの例に示すように、コマンド EDIT または READ を使用してソースワークエリアに最後に読み込まれたソースコードが表示されます。

既存のオブジェクトのソースを読み込むと、ソースコード全体がソースワークエリアにロードされ、編集できるようになります。ただし、ソースのサイズによっては、ソース行の一部のみが編集エリアに表示される場合があります。その場合は、ソースを下方にスクロールして（「[スクロール用エディタコマンド](#)」を参照）、表示または編集する行へ移動することが必要です。

また、[画面分割モード](#)（関連セクションを参照）を使用する場合、編集エリアに表示されるソースコード行数が少なくなります。

ソースコードの作成または変更には、いくつかの方法を使用できます。

- 関連するソース行で、テキストを直接入力または更新します。
- 関連セクションの説明に従って、1つ以上の[行コマンド](#)を使用します。

行コマンドは、現在のソースコード内で、行の挿入や削除、変数定義やフィールド定義のコピーなどに使用します。

- 「[スクロール用エディタコマンド](#)」、「[エディタコマンド](#)」、および「[PA/PF キーの割り当ての表示および変更](#)」の説明に従って、1つまたは複数のエディタコマンドや代替 PF キーを使用します。

エディタコマンドは、ソースのスクロール、テキスト文字列の検索と置換、編集操作の取り消し、エディタプロファイル機能の呼び出し、変数定義またはフィールド定義のコピーに使用するための別オブジェクトの表示などに使用します。

テキストの挿入

このセクションでは、行コマンドを使用して現在のソースにテキストを挿入する手順例について説明します。

➤手順 7.1. テキストをソースに挿入するには

- 1 目的のソース行の横にある[接頭辞エリア](#)の左端の列に、I（挿入）コマンド（「[行コマンド](#)」を参照）を入力し、接頭辞エリアの文字を一時的に上書きします。

例えば、次の行コマンドを入力します。

I5

挿入モードが有効になって一連のアポストロフィ（'……'）が表示され、行コマンドを入力した行の下に空のソース行が5行挿入されます。カーソルは、最初の空ソース行の先頭に配置されます。

- 2 空のソース行にテキストを入力します。ENTERキーを押すと、空行がソースから削除されます。

アポストロフィは行番号に置き換わり、新しいテキストがソースに組み込まれたことを示します。

ソース行番号の再設定

オブジェクトのソースコードに行を追加すると、追加された行に 10 単位の増分で番号が設定されます。ソースコードに行を挿入すると、最初の行を 000010 とする 10 単位の増分で、ソースコードのすべての行に番号が自動的に再設定されます。

増分は、『システムコマンド』ドキュメントで説明しているシステムコマンド `RENUMBER` を使用して変更できます。

『*Programming Guide*』の「ソースコード行番号参照の変更」も参照してください。

テキストのコピーおよび貼り付け

このセクションでは、コピー機能と貼り付け機能を使用して現在のソース内でテキストをコピーする手順について説明します。現在のソースに別の Natural オブジェクトからテキストをコピーする手順については、「[SPLIT を使用して定義を表示およびコピーするには](#)」を参照してください。

手順 7.2. テキストを切り取るかコピーしてソースに貼り付けるには

- 1 テキストを切り取るかコピーしてクリップボードに格納します。
- 2 クリップボードに切り取るかコピーしたテキストを配置するために必要な数のソース行を追加します。このことは、例えば、次の手順を実行します。
 - `I` 行コマンドを使用して行を挿入します。
 - 新しい行に何か文字を入力します。
 - 新しい行を `Rn` 行コマンドを使用してコピーします。`n` は繰り返し数です。
- 3 コマンド行で、次のように入力します。

`PROTECT ON`

このコマンドは接頭辞エリアを保護し、貼り付けられる行が 72 文字を超えている場合にソース行が上書きされないようにします。

- 4 テキストの貼り付け先となるソース行の先頭行の先頭列にカーソルを置きます。
- 5 端末エミュレーションの貼り付け機能を選択するか、`CTRL+V` キーを押します。

クリップボードに配置されたテキストが、カーソル位置からソースの末尾に向けて、ソース行に貼り付けられます。

貼り付けられた行が、ソースからコピーされた6桁の行番号で始まっている場合は、次の手順で行番号を削除できます。

- 6 コマンド行で、次のように入力します。

```
PROTECT OFF
```

接頭辞エリアが保護されなくなり、行コマンドの入力が可能になります。

- 7 貼り付けられた行のうち、コピーされた行番号を含む行の横で次の行コマンドを入力します。

```
(7
```

その行が7列左にシフトされ、行番号が削除されます。

データウィンドウを使用したテキストのコピーまたは移動

データウィンドウを使用すると、行の先頭や最後で始まったり終わったりしないテキストをコピーまたは移動できます。データウィンドウ機能は、行コマンドおよび／またはエディタコマンドを使用して実行できます。

ウィンドウを定義すると、そのウィンドウの開始から終了までの間にある画面上のすべてのテキストが、ウィンドウの一部となります。

データウィンドウの使用例

このセクションでは、行コマンドまたは対応するエディタコマンドを使用して、データウィンドウを持つテキストを定義または移動する例について説明します。

この例では、手順1に示すテキストを使用し、Note that when... (行 80) で始まる文全体を、...copy operations (行 30) で終わっている表示テキスト部分の最初の文の後ろに移動します。

➤手順 7.3. 行コマンドを使用してウィンドウを定義および移動するには

- 1 次に示すように、テキストを入力します。

```

>> -----Columns 001 072 <<      Text WINEX      Lines 10      User MMO      ←
Command ==>                      Mode      Report Lib  SAGTEST ←
***** ***** top of data *****
000010 Copy a Window with Text ←
000020 ←
000030 You can specify a window with text for move or copy operations. This ←
000040 allows you to copy or move text that does not start or end at the ←
000050 beginning or end of a line. This function can be performed using ←
000060 line commands and/or editor commands. ←
000070 ←
000080 Below are some examples of copying windows with text. Note that when ←
000090 you define a window, all text on your screen between start and end of ←
000100 the window become part of the window. Available line commands are: ←
***** ***** bottom of data *****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--- ←
      Help  Save   Exit   Run   Rfind Stow           Check Home  Undo  Canc ←

```

- 2 移動するテキストの先頭行である行80に行コマンド WS を入力し、対象とする列（単語 Note の N）にカーソルを置いて ENTER キーを押します。

行 80 の接頭辞エリアにメッセージ「WS55」が表示され、この列番号が選択されたことを示します。

```
>> -----Block is pending <<      Text WINEX      Lines 10      User MMO
Command ==>                               Mode   Report Lib  SAGTEST
***** **** top of data ****
000010 Copy a Window with Text
000020
000030 You can specify a window with text for move or copy operations. This
000040 allows you to copy or move text that does not start or end at the
000050 beginning or end of a line. This function can be performed using
000060 line commands and/or editor commands.
000070
WS55 Below are some examples of copying windows with text. Note that when
000090 you define a window, all text on your screen between start and end of
000100 the window become part of the window. Available line commands are:
***** **** bottom of data ****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help  Save  Exit  Run  Rfind  Stow           Check  Home  Undo  Canc
```

- 3 移動するテキストの最終行である行 100 に行コマンド WE を入力し、移動する最後の列
(window の後のピリオド (.)) にカーソルを移動して ENTER キーを押します。

行 100 の接頭辞エリアにメッセージ「WE37」が表示されます。

```

>> -----Block is pending <<      Text WINEX      Lines 10      User MMO
Command ===>                               Mode      Report Lib      SAGTEST
***** ***** top of data *****
000010 Copy a Window with Text
000020
000030 You can specify a window with text for move or copy operations. This
000040 allows you to copy or move text that does not start or end at the
000050 beginning or end of a line. This function can be performed using
000060 line commands and/or editor commands.
000070
WS55 Below are some examples of copying windows with text. Note that when
000090 you define a window, all text on your screen between start and end of
WE37 the window become part of the window. Available line commands are:
***** ***** bottom of data *****

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
 Help Save Exit Run Rfind Stow Check Home Undo Canc

- 4 行コマンド WM を行 30 に入力し (テキストは次の行である行 40 に移動) 、行 30 を分割する列 (単語 This の前の空白) にカーソルを移動します。ENTER キーを押します。

指定されたテキストセクションが移動します。

>> -----Columns 001 072 << Text WINEX Lines 13 User MMO
Command ==> Mode Report Lib SAGTEST
***** ***** top of data *****
000010 Copy a Window with Text
000020
000030 You can specify a window with text for move or copy operations.
000040 Note that when
000050 you define a window, all text on your screen between start and end of
000060 the window become part of the window.
000070 This
000080 allows you to copy or move text that does not start or end at the
000090 beginning or end of a line. This function can be performed using
000100 line commands and/or editor commands.
000110
000120 Below are some examples of copying windows with text.
000130 Available line commands are:
***** ***** bottom of data *****

次に示すコマンドシーケンスを使用すると、上記の手順と同じ結果が得られます。

›手順 7.4. エディタコマンドを使用してウィンドウを定義および移動するには

- 1 上記の手順1に示すテキストを入力します。
 - 2 コマンド行で、次のように入力します。

WINDOW 80 100 55 37 ;MWINDOW 30 64

指定されたテキストセクションが、上記の手順4に示すように移動します。

テキストの検索および置換

このセクションでは、現在のソースで文字列を検索および置換する手順について説明します。

➤手順 7.5. ソースに含まれている文字列を検索および置換するには

- 1 次のエディタコマンドを使用して、文字列を検索します。

FIND '*string*'

string は任意の英数字文字列です。

指定した文字列の最初のオカレンスが検索されます。

- 2 次のコマンドを使用して、検索された文字列を置換します。

```
CHANGE 'string' 'new-string'
```

または:

PF5 キーを押して、次のオカレンスを検索します。

使用するコマンドに応じて、文字列の最初のオカレンスが置換されるか、文字列の次のオカレンスがあればそれが検索されます。

- 3 PF17 キーを押して、検索された文字列を置換します。

または:

PF5 キーを押して、次のオカレンスを検索します。

使用する PF キーに応じて、文字列の次のオカレンスが置換されるか、文字列の次の次のオカレンスがあればそれが検索されます。

ソースコードのフォーマット

ソース行を段落付けすることにより、ソースコードをフォーマットできます。段落付けは、レポートイングモードで作成されたソースの場合とストラクチャードモードで作成されたソースの場合では、異なる方法で実行されます。

▷手順 7.6. ソースコードをフォーマットするには

- コマンド行で、次のシステムコマンドを入力します。

```
STRUCT
```

ソースコードの行が段落付けされます。

詳細については、『システムコマンド』ドキュメントの STRUCT の説明を参照してください。

ソースコードのチェック

▷手順 7.7. 現在のソースコードの構文エラーをチェックするには

- エディタ画面で PF9 キーを押します。

または:

コマンド行で、次のように入力します。

CHECK

ソースワークエリアに現在表示されているソースコードの構文エラーがチェックされます。構文エラーが検出されるとエラーメッセージが表示され、エラーがある行番号と定義が示されます。

8 画面分割モード

画面分割モードでは、画面の半分をオブジェクトの編集に使用し（編集セクション）、同時にもう半分を他のNaturalオブジェクトの表示に使用できます（表示セクション）。また、表示セクションに表示されている定義をコピーできます。

画面分割モードでは、次のタイプのNaturalオブジェクトを表示できます。

DDM（データ定義モジュール）、データエリア、プログラム、サブプログラム、サブルーチン、ヘルプルーチン、コピーコード、テキスト、マップ、クラス、機能、およびPredictプログラム記述。

次の手順で、別のNaturalオブジェクトの定義を表示し、それを現在編集中のオブジェクトにコピーする方法を示します。

➤手順 8.1. SPLIT を使用して定義を表示およびコピーするには

- 1 現在のソースで、**SPLIT**エディタコマンド（「エディタコマンド」で説明）を例えば次のように入力し、テキストのコピー元となるオブジェクトのソースを表示します。

```
SPLIT P LDATEST1
```

画面分割モードが設定され、次の例に示すように、エディタ画面の編集セクション（上半分）に現在のソース（次の例ではプログラム PGMTEST1）が表示され、表示セクション（下半分）にローカルデータエリア（LDATEST1）が表示されます。

画面分割モード

```
>> -----Columns 001 072 << Program PGMTEST1 Lines 187 User SAG ↵
Command ===>                                         Lib SYSLIB ↵
001720 IF (RC NE 0)                                ↵
001730 then                                         ↵
001740 reset CMD_LINE_2                            ↵
001750 IF (s_prog ne ' ')                         ↵
001760 then                                         ↵
001770 assign CMD_LINE_PGM = S_PROG                ↵
001780 assign CMD_LINE_LIB = S_LIB                  ↵
001790 END-IF                                         ↵
001800 END-IF                                         ↵
>> -----Columns 001 072 << Local LDATEST1 Lines 158 User SAG ↵
Command ===>                                         Lib SYSLIB ↵
***** ***** ***** ***** ***** ***** top of data ↵
***** ***** ***** ***** ***** ***** ↵
000001 DEFINE DATA LOCAL                         ↵
000002 1 NPF_F_READ(A1)                         ↵
000003 CONST                                     ↵
000004 <'R'>                                    ↵
000005 1 NPF_F_WRITE(A1)                        ↵
000006 CONST                                     ↵
000007 <'W'>                                    ↵
↵
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---  

      Help  Save  Exit  Run  Rfind  Stow  -  +  Check  Home  Undo  Canc
```



注意:

1. 表示セクションでは定義を変更できないため、一部のエディタコマンドは使用できません。PFキーは編集セクション用に予約されています。このため、表示セクションを対象とするコマンドはコマンド行で発行する必要があります。
 2. **SWAP** エディタコマンドを使用すると、表示セクションのコマンド行と編集セクションのコマンド行との間でカーソルを移動できます。
- 2 表示セクションで、**LDATEST1** からコピーする行のブロックをマークします。例えば、行コマンド **CC** を目的のブロックの先頭行と最終行の横に入力します。
- 3 **PGMTEST1** の編集セクションで、行のブロックを配置する行の 1 つ前の行の横に、行コマンド **A** を入力します。
- マークされた行のブロックが、**PGMTEST1** の、行コマンドを入力した行の下にコピーされます。
- 4 必要に応じて、**SPLIT END** エディタコマンドを入力して画面分割モードを終了します。

9 エディタコマンド

■ ADVANCE	35
■ AORDER	36
■ AUTOSAVE	36
■ BNDS	36
■ CANCEL	37
■ CAPS	37
■ CENTER	37
■ CHANGE	38
■ COLS	41
■ CWINDOW	41
■ DELETE	41
■ DWINDOW	43
■ DX, DY, DX-Y	43
■ EMPTY	44
■ EX, EY, EX-Y	44
■ EXCLUDE	44
■ EXIT	46
■ FIND	47
■ FLIP	49
■ HEX	49
■ HOME	50
■ INCLUDE	50
■ JLEFT	50
■ JRIGHT	51
■ JUSTIFY	52
■ LABEL	52
■ LC	53
■ LIMIT	55
■ LOCATE	55
■ LOG	56
■ MASK	56
■ MWINDOW	57

■ NEXT	57
■ ORDER	58
■ POINT	58
■ POWER	58
■ PROF	59
■ PROFILE	59
■ PROTECT	59
■ RCHANGE	60
■ RESET	60
■ RFIND	60
■ SET TYPE	61
■ SHIFT	61
■ SORT	62
■ SPLIT	62
■ SWAP	63
■ TABS	63
■ UC	67
■ UNDO	67
■ WINDOW	68
■ X	68
■ XSWAP	69
■ Y	69
■ 共通のコマンドオプション	69

ADVANCE | AORDER | AUTOSAVE | BNDS | CANCEL | CAPS | CENTER | CHANGE | COLS | CWINDOW | DELETE | DWINDOW | DX | DY | DX-Y | EMPTY | EX | EY | EX-Y | EXCLUDE | EXIT | FIND | FLIP | HEX | HOME | INCLUDE | JLEFT | JRIGHT | JUSTIFY | LABEL | LC | LIMIT | LOCATE | LOG | MASK | MWINDOW | NEXT | ORDER | POINT | POWER | PROF | PROFILE | PROTECT | RCHANGE | RESET | RFIND | SET TYPE | SHIFT | SORT | SPLIT | SWAP | TABS | UC | UNDO | WINDOW | X | XSWAP | Y | 共通のコマンドオプション

このセクションでは、ソースの変更に使用できるすべてのエディタコマンドについて説明するとともに、一般的に使用されるエディタコマンドオプションの概要を示します。上記のエディタコマンドに加えて、[エディタコマンドを使用してソースをスクロールし、また行コマンド](#)を使用して単一または複数のソース行を操作できます。

エディタコマンドの使用

- エディタコマンドは、エディタ画面の[コマンド行](#)に入力します。インストールの設定によっては、エディタコマンドを小文字で入力できます。ただし、このセクションでは、コマンドであることがわかるように、すべてのコマンドを大文字で示しています。
- エディタコマンドをセミコロン (;) で区切ると、一度の入力操作で複数のコマンドを入力できます。
- このセクションで使用されている構文記号の説明については、『システムコマンド』ドキュメントの「システムコマンド構文」を参照してください。
- 頻繁に使用される一部のコマンドは、このセクションに示しているように [PFキー](#)を使用して発行できます。

ADVANCE

ADVANCE	[ON]
		OFF	
		PAGE	

このコマンドは、行の更新後にカーソルを次の行に自動的に移動するかどうかを指定するために使用します。

ON	カーソルは更新後に次の行に移動します。
OFF	カーソルは更新後に次の行に移動しません。
PAGE	更新後に、カーソルのある行は編集エリアの一番上に配置されます。

パラメータを指定せずに発行する ADVANCE コマンドは、ADVANCE ON と同じ効果があります。

AORDER

```
AORDER [ ON ] [ OFF ]
```

このコマンドは、設定されている境界内でテキストの位置を自動的に調整するかどうかを指定するために使用します。

パラメータを指定せずに発行する AORDER コマンドは、AORDER ON と同じ効果があります。プロファイルを編集すると、基本設定を変更できます。

AUTOSAVE

```
{ AUTOSAVE } [ ON ] [ OFF ]
```

このコマンドは、EXIT コマンドを発行したときに自動的に SAVE コマンドを実行するかどうかを指定するために使用します。

パラメータを指定せずに発行する AUTOSAVE コマンドは、AUTOSAVE ON と同じ効果があります。

BNDS

```
BNDS [ n m ]
```

このコマンドは、特定のコマンドの影響を特定の列の範囲に制限するために使用します。

これらの境界は、[FIND](#)、[CHANGECENTER](#)、[ORDER](#)、[LEFT](#)、[JRIGHT](#) の各エディタコマンド、および [TC](#)、[T0](#)、[LJ](#)、[RJ](#) などの対応する行コマンド（存在する場合）に適用されます。

<u>n</u>	左境界を配置する列の番号。
<u>m</u>	右境界を配置する列の番号。

n と m が省略されると、編集エリアの最初の列と最後の列に境界が設定されます。

現在の境界設定を表示するには、[BNDS](#) 行コマンドを発行します。

CANCEL

CANCEL

代替 PF キー：PF12

このコマンドは、ソースを最後に保存した後に行われたすべての変更内容を取り消し（「[ソースの保存およびカタログ](#)」も参照）、エディタを終了します。エディタプロファイル設定によっては、変更内容を保存するか保存せずに終了するかを確認するプロンプトが表示されます（「[Exit 機能](#)」も参照）。

CAPS

CAPS
$$\begin{bmatrix} \text{ON} \\ \text{OFF} \\ \text{PGM} \end{bmatrix}$$

このコマンドは、大文字変換のオンとオフを切り替えるために使用します。このコマンドは、このコマンドの発行後に作成または変更された行にのみ適用されます。

ON	行に含まれるテキストが大文字に変換されます。
OFF	行に含まれるテキストは入力されたままで、変換されません。
PGM	行に含まれるテキストが大文字に変換されます（ただし、コメントは入力されたままで変換されません）。

パラメータを指定せずに発行する CAPS コマンドは、CAPS ON と同じ効果があります。

CENTER

CENTER
$$\begin{Bmatrix} \text{ALL} \\ n \\ n \ m \end{Bmatrix}$$

このコマンドは、テキストを中央揃えにするために使用します。

エディタコマンド

ALL	すべての行のテキストを中央揃えにします。
<i>n</i>	行 <i>n</i> から最終行までのテキストを中央揃えにします。
<i>n m</i>	行 <i>n</i> から行 <i>m</i> までのテキストを中央揃えにします。

CENTER コマンドは、BNDS エディタコマンドで設定された水平方向の境界内にのみ適用されます。

中央揃えには、行コマンド TC および TCC を使用することもできます。

CHANGE

CHANGE	CHG	$\left[\begin{array}{c} * \\ [T]'string1' \\ C'string1' \\ X'string1' \\ P'string1' \end{array} \right] \left\{ \begin{array}{c} * \\ [X]'string2' \end{array} \right\}$	$\left[\begin{array}{c} .X \\ .X .Y \end{array} \right] \left[\begin{array}{c} n \\ n m \end{array} \right] \left[\begin{array}{c} ALL \\ NEXT \\ PREV \\ FIRST \\ LAST \end{array} \right] \left[\begin{array}{c} CHARS \\ WORD \\ PREFIX \\ SUFFIX \end{array} \right] \left[\begin{array}{c} NX \\ X \end{array} \right]$
---------------	------------	---	---

このコマンドを使用すると、文字列 (string1) を別の文字列 (string2) に置換できます。

アポストロフィを string1 または string2 の一部として使用するには、2つのアポストロフィを入力する必要があります。

置換する文字列 (string1) は、次のセクションで説明している方法で指定できます。

T' string1 '	文字列が大文字か小文字かに関係なく、string1 を置換します。これがデフォルトです。
' string1 '	T' string1 ' も同様です。
C' string1 '	指定した文字列に完全に一致する場合にのみ string1 を置換します。
X' string1 '	指定した 16 進数文字列 string1 に対応する文字列を置換します。この文字列を 16 進数文字列 string2 で置換します。
P' string1 '	次のワイルドカード文字を含む string1 を置換します。 = 任意の文字

	§	英文字
	#	数字
	\$	特殊文字
	^	空白以外の文字
	-	数字以外の文字
	<	小文字
	>	大文字
*		前のコマンド (FIND、CHANGE、EXCLUDE など) で指定した文字列を使用します。
.X		詳細については、「 行指定 」を参照してください。
.X .Y		
n		詳細については、「 列指定 」を参照してください。
n m		
ALL		詳細については、「 操作の方向 」を参照してください。
NEXT		
PREV		
FIRST		
LAST		
CHARS		詳細については、「 特殊なオカレンス 」を参照してください。
WORD		
PREFIX		
SUFFIX		
NX		詳細については、「 表示されている行または表示されていない行 」を参照してください。
X		

このセクションでは、次のトピックについて説明します。

■ [CHANGE と他のコマンドの併用](#)

■ CHANGE コマンドの例

CHANGE と他のコマンドの併用

CHANGE コマンドの繰り返し実行には、RCHANGE コマンドを使用できます。

ソースコード全体から文字列を検索し、別の文字列で置き換えるかどうかをオカレンスごとに判断する場合は、「[テキストの検索および置換](#)」の説明に従って、[FIND](#) コマンド、[CHANGE](#) コマンド、および PF キーを、[RFIND](#) コマンドおよび [RCHANGE](#) コマンドと組み合わせて使用できます。

CHANGE コマンドの例

例 1

```
CHG 'LOW' 'HIGH'
```

このコマンドは、LOW の最初のオカレンスを HIGH に置換します（大文字か小文字かは考慮されません）。

例 2

```
CHG C'OPS' 'SPF' .X .Y 28 32 ALL
```

このコマンドは OPS (完全に一致する必要があります) を SPF に変更します。.X および .Y のラベルが付けられた行のブロックの列 28~32 で、すべてのオカレンスが変更されます。

例 3：

```
CHG C'NAME' 'APPL' .X .Y ALL PREFIX NX
```

このコマンドは、.X および .Y とラベル付けされた行のブロック内のすべての表示されている行で、NAME (完全に一致する必要があります) で始まる文字列のすべてのオカレンスを APPL に変更します。

例 4

```
CHG * 'NEW'
```

このコマンドは、最後の CHANGE コマンドで指定された文字列の次のオカレンスを、文字列 NEW で置換します。

例 5

```
CHG 'OLD' *
```

このコマンドは、文字列 OLD の次のオカレンスを、最後の CHANGE コマンドで指定されたものと同じ新しい文字列で置換します。

COLS

```
COLS [ ON ]
      [ OFF ]
```

このコマンドは、編集エリア上部に行を表示して列位置を示します。

列位置の表示には、行コマンド `COLS` も使用できます。

CWINDOW

```
CWINDOW [ n ]
      [ n m ]
```

このコマンドは、コマンドパラメータに従ってデータウィンドウをコピーするために使用します。

<i>n</i>	データウィンドウを挿入する行の番号。
<i>m</i>	データウィンドウを挿入する列の番号。

「[データウィンドウを使用したテキストのコピーまたは移動](#)」も参照してください。

DELETE

```
DELETE [ * ]
      [ T ] 'string'
      'string'
      C 'string'
      X 'string'
      P 'string'
      [ .X .Y ]
      [ n m ]
      [ ALL ]
      [ NEXT ]
      [ PREV ]
      [ FIRST ]
      [ LAST ]
      [ CHARS ]
      [ WORD ]
      [ PREFIX ]
      [ SUFFIX ]
      [ NX ]
      [ X ]
```

このコマンドは、行を削除するために使用します。

次のセクションで説明しているように、指定した文字 *string* を含む行だけを削除するように指定できます。

エディタコマンド

T' <i>string</i> '	小文字か大文字かに関係なく、 <i>string</i> を含む行を削除します。 これがデフォルトです。																
' <i>string</i> '	T' <i>string</i> ' も同様です。																
C' <i>string</i> '	指定した文字列に完全に一致する <i>string</i> を含む行を削除します。																
X' <i>string</i> '	指定した 16 進数文字 <i>string</i> に対応する文字列を含む行を削除します。																
P' <i>string</i> '	次のワイルドカード文字を持つ <i>string</i> を含む行を削除します。 <table border="1" data-bbox="301 559 1068 982"> <tr><td>=</td><td>任意の文字</td></tr> <tr><td>§</td><td>英文字</td></tr> <tr><td>#</td><td>数字</td></tr> <tr><td>\$</td><td>特殊文字</td></tr> <tr><td>^</td><td>空白以外の文字</td></tr> <tr><td>-</td><td>数字以外の文字</td></tr> <tr><td><</td><td>小文字</td></tr> <tr><td>></td><td>大文字</td></tr> </table>	=	任意の文字	§	英文字	#	数字	\$	特殊文字	^	空白以外の文字	-	数字以外の文字	<	小文字	>	大文字
=	任意の文字																
§	英文字																
#	数字																
\$	特殊文字																
^	空白以外の文字																
-	数字以外の文字																
<	小文字																
>	大文字																
*	前のコマンド (FIND、CHANGE、EXCLUDE など) で指定した検索文字列を使用します。																
.X	詳細については、「 行指定 」を参照してください。																
.X .Y																	
<i>n</i>	詳細については、「 列指定 」を参照してください。																
<i>n m</i>																	
ALL NEXT PREV FIRST LAST	詳細については、「 操作の方向 」を参照してください。																
CHARS WORD PREFIX SUFFIX	詳細については、「 特殊なオカレンス 」を参照してください。																

NX	詳細については、「 表示されている行または表示されていない行 」を参照してください。
X	

パラメータを指定せずに **DELETE** コマンドを入力すると、現在の行が削除されます。

行の削除には、行コマンド **D**、**Dn**、または **DD** も使用できます。

例 1

```
DEL C'NAME' 1 20 ALL PREFIX NX
```

このコマンドは、NAME が 1 列から 20 列の間にある場合に、単語の接頭辞として文字列 NAME (ここで入力したとおりの大文字) を含むすべての表示されている行を削除します。

例 2

```
DEL C'Abc' .X .Y 10 30 ALL
```

このコマンドは、.X と .Y でラベル付けされた行のブロック内で列 10~30 にある文字列 Abc (完全に一致する必要があります) を含むすべての行を削除します。

DWINDOW

```
DWINDOW
```

このコマンドは、最後に定義したデータウィンドウを削除するために使用します。

DX, DY, DX-Y

```
DX
DY
DX-Y
```

これらのコマンドは、マークされた行を削除するために使用します。

- DX コマンドは、.X ラベルでマークされた行を削除します。
- DY コマンドは、.Y ラベルでマークされた行を削除します。
- DX-Y コマンドは、.X ラベルと .Y ラベルの間にあるすべての行を削除します。

EMPTY

```
EMPTY [ ON ]
```

このコマンドは、空行の削除を制御します。

OFF	空行は削除されません。
ON	空行は削除されます。

パラメータを指定せずに発行する EMPTY コマンドは、EMPTY ON と同じ効果があります。

EX, EY, EX-Y

```
EX  
EY  
EX-Y
```

これらのコマンドは、ソース内の行を削除する場合に使用します。

- EX コマンドは、.X ラベルでマークされた行の前にあるすべての行を削除します。
- EY コマンドは、.Y ラベルでマークされた行の後にあるすべての行を削除します。
- EX-Y コマンドは、.X ラベルの前および .Y ラベルの後にあるすべての行を削除します。

EXCLUDE

```
EXCLUDE [ *  
          [T]'string'  
          'string'  
          C'string'  
          X'string'  
          P'string' ] [ .X .Y ] [ n m ] [ ALL  
                                         NEXT  
                                         PREV  
                                         FIRST  
                                         LAST ] [ CHARS  
                                         WORD  
                                         PREFIX  
                                         SUFFIX ]
```

このコマンドは、行が表示されないようにするために使用します。

指定した文字 *string* を含む行だけを非表示にするように指定できます。次のセクションを参照してください。

T' <i>string</i> '	小文字か大文字かに関係なく、 <i>string</i> を含む行を非表示にします。これがデフォルトです。																
' <i>string</i> '	T' <i>string</i> ' も同様です。																
C' <i>string</i> '	指定した文字列に完全に一致する <i>string</i> を含む行を非表示にします。																
X' <i>string</i> '	指定した 16 進数文字 <i>string</i> に対応する文字列を含む行を非表示にします。																
P' <i>string</i> '	次のワイルドカード文字を持つ <i>string</i> を含む行を非表示にします。 <table border="1" style="margin-left: 20px;"> <tr><td>=</td><td>任意の文字</td></tr> <tr><td>§</td><td>英文字</td></tr> <tr><td>#</td><td>数字</td></tr> <tr><td>\$</td><td>特殊文字</td></tr> <tr><td>^</td><td>空白以外の文字</td></tr> <tr><td>-</td><td>数字以外の文字</td></tr> <tr><td><</td><td>小文字</td></tr> <tr><td>></td><td>大文字</td></tr> </table>	=	任意の文字	§	英文字	#	数字	\$	特殊文字	^	空白以外の文字	-	数字以外の文字	<	小文字	>	大文字
=	任意の文字																
§	英文字																
#	数字																
\$	特殊文字																
^	空白以外の文字																
-	数字以外の文字																
<	小文字																
>	大文字																
*	前のコマンド (EXCLUDE、FIND、CHANGE など) で指定した検索文字列を使用します。																
.X	詳細については、「 行指定 」を参照してください。																
.X .Y																	
<i>n</i>	詳細については、「 列指定 」を参照してください。																
<i>n m</i>																	
ALL	詳細については、「 操作の方向 」を参照してください。																
NEXT																	
PREV																	
FIRST																	
LAST																	
CHARS	詳細については、「 特殊なオカレンス 」を参照してください。																
WORD																	
PREFIX																	

エディタコマンド

SUFFIX	
--------	--

パラメータを指定せずに EXCLUDE コマンドを入力すると、現在の行が非表示になります。

非表示になった行を再表示するには、[INCLUDE](#) エディタコマンドを使用します。

例 1

```
EXCLUDE .X .Y
```

このコマンドは、.X というラベルの行から .Y というラベルの行までの行を非表示にします。

例 2

```
EXCLUDE C'NAME' ALL PREFIX
```

このコマンドは、NAME (ここに入力されているとおりの大文字) で始まる文字列が含まれているすべての行を非表示にします。

EXIT

EXIT

代替 PF キー：PF3

このコマンドは、エディタを終了するために使用します。ソースを最後に保存した後に変更を加えた場合（「[ソースの保存およびカタログ](#)」も参照）、エディタプロファイルの設定（「[Exit 機能](#)」も参照）によっては、変更内容を保存するか保存せずに終了するかを確認するプロンプトが表示されます。

 **注意:** [AUTOSAVE](#) が ON に設定されている場合は、変更内容が自動的に保存されます。このとき、セッションの終了前にこのプロンプトは表示されません。

FIND

FIND	*	[T]'string'	'string'	[.X .Y]	[n m]	[ALL NEXT PREV FIRST LAST]	[CHARS WORD PREFIX SUFFIX]	[NX X]
------	---	-------------	----------	-----------	---------	------------------------------	------------------------------	----------

このコマンドは、特定の文字 *string* を検索するために使用します。カーソルは、最初に検出した *string* の開始位置に配置されます。*string* を含む行が非表示にされている場合は、検出時に表示されます。

アポストロフィを *string* の一部として使用するには、2つのアポストロフィを入力する必要があります。

string は、次のセクションで説明している方法で指定できます。

T' string'	小文字か大文字かに関係なく、 <i>string</i> を検索します。これがデフォルトです。																
' string'	T' string' も同様です。																
C' string'	指定した文字列に完全に一致する <i>string</i> を検索します。																
X' string'	指定した 16 進数文字 <i>string</i> に対応する文字列を検索します。																
P' string'	次のワイルドカード文字を含む <i>string</i> を検索します。 <table border="1" style="margin-left: 20px;"> <tr> <td>=</td> <td>任意の文字</td> </tr> <tr> <td>§</td> <td>英文字</td> </tr> <tr> <td>#</td> <td>数字</td> </tr> <tr> <td>\$</td> <td>特殊文字</td> </tr> <tr> <td>^</td> <td>空白以外の文字</td> </tr> <tr> <td>-</td> <td>数字以外の文字</td> </tr> <tr> <td><</td> <td>小文字</td> </tr> <tr> <td>></td> <td>大文字</td> </tr> </table>	=	任意の文字	§	英文字	#	数字	\$	特殊文字	^	空白以外の文字	-	数字以外の文字	<	小文字	>	大文字
=	任意の文字																
§	英文字																
#	数字																
\$	特殊文字																
^	空白以外の文字																
-	数字以外の文字																
<	小文字																
>	大文字																
*	前のコマンド (FIND、DELETE、EXCLUDE など) で指定した <i>string</i> を検索します。																
.X	詳細については、「 行指定 」を参照してください。																

エディタコマンド

.X .Y	
n	詳細については、「 列指定 」を参照してください。
n m	
ALL	詳細については、「 操作の方向 」を参照してください。
NEXT	
PREV	
FIRST	
LAST	
CHARS	詳細については、「 特殊なオカレンス 」を参照してください。
WORD	
PREFIX	
SUFFIX	
NX	詳細については、「 表示されている行または表示されていない行 」を参照してください。
X	

例 1

```
F C'NAME' .X .Y ALL PREFIX X
```

このコマンドは、.X および .Y でラベル付けされた行のブロック内の非表示の行で、単語の接頭辞として NAME (完全に一致する必要があります) を含むオカレンスをすべて検索します。

例 2

```
F C'HILITE' X PREV
```

このコマンドは、除外された行で HILITE (完全に一致する必要があります) の前の該当箇所を検索します。

FIND コマンドの繰り返し実行には、[RFIND](#) コマンドを使用できます。

例 3：

```
F P'RCV#' .X .Z 20 30
```

このコマンドは、RCV で始まり 4 番目の文字が数値である 4 文字の文字列を検索します。.X および .Z でラベル付けされた行のブロックの列 20~30 内を検索します。

例 4

```
F X'6C' SUFFIX NX
```

このコマンドは、16進表記が 6C である文字を検索します。単語の末尾にある文字の該当箇所だけが検出されます。この検索は、非表示にされていない行についてのみ有効です。

例 5

```
F ' ' ' w '
```

このコマンドは、次の文字列を検索します：'w

例 6

```
F ' r ' ' w '
```

このコマンドは、次の文字列を検索します：r'w

例 7

```
F ' '
```

このコマンドはアポストロフィ (') を検索します。

FLIP

```
FLIP
```

このコマンドは、PF1～PF12 と PF13～PF24 の PF キーの表示を切り替えます。

HEX

```
HEX [ ON ]  
[ OFF ]
```

このコマンドは、16進表示モードのオン／オフを切り替えます。

HOME

HOME

このコマンドは、次に ENTER キーを押した後でカーソルをコマンドフィールドに戻します。

INCLUDE

INCLUDE	$\begin{bmatrix} * \\ [T]'string' \\ 'string' \\ C'string' \\ X'string' \\ P'string' \end{bmatrix} \begin{bmatrix} .X \\ .X \end{bmatrix} \begin{bmatrix} n \\ .Y \end{bmatrix} \begin{bmatrix} n \\ m \end{bmatrix} \begin{bmatrix} ALL \\ \underline{NEXT} \\ PREV \\ FIRST \\ LAST \end{bmatrix} \begin{bmatrix} \underline{CHARS} \\ WORD \\ PREFIX \\ SUFFIX \end{bmatrix}$
---------	--

このコマンドは、[EXCLUDE](#) コマンドによって非表示にされた行を再表示するために使用します。
このコマンドは [EXCLUDE](#) コマンドと同じパラメータをとります。

パラメータを指定せずに [INCLUDE](#) コマンドを入力すると、非表示の行のブロックの最初の行が含まれます。

JLEFT

JLEFT	$\begin{Bmatrix} ALL \\ \{ n \\ n \ m \} \end{Bmatrix}$
-------	---

このコマンドは、テキストを左詰めに整列するために使用します。

ALL	すべての行のテキストを整列します。
<i>n</i>	<i>n</i> 行から最後の行までのテキストを整列します。
<i>n m</i>	<i>n</i> 行から <i>m</i> 行までのテキストを整列します。

[JLEFT](#) コマンドは、[BNDS](#) コマンドで設定された水平方向の境界内にのみ適用されます。

左詰めには、行コマンド [LJ](#) および [LJJ](#) も使用できます。

[JRIGHT](#) コマンドの説明も参照してください。

例：

```
BNDS 10;JLEFT 15 20
```

行 15～20 で列 10 と画面右端の列の間にあるテキストを、列 10 に左詰めにします。

JRIGHT

JRIGHT $\left\{ \begin{array}{l} \text{ALL} \\ n \\ n \ m \end{array} \right\}$	
---	--

このコマンドは、テキストを右詰めに整列するために使用します。

ALL	すべての行のテキストを整列します。
n	n 行から最後の行までのテキストを整列します。
$n \ m$	n 行から m 行までのテキストを整列します。

JRIGHT コマンドは、[BNDS](#) コマンドで設定された水平方向の境界内にのみ適用されます。

右詰めには、行コマンド [RJ](#) および [RJJ](#) も使用できます。

[JLEFT](#) コマンドの説明も参照してください。

例 1

```
BNDS 4 40;JRIGHT 6 18
```

行 6～18 で列 4～40 の間にあるテキストを列 40 に右詰めにします。

例 2

```
BNDS 10;JRIGHT 15
```

行 15 から最後の行までの列 10 から右にあるテキストが、編集画面の右端の列に対して右詰めになります。

JUSTIFY

```
JUSTIFY {  
    LEFT  
    RIGHT  
    BOTH  
}
```

このコマンドは、行コマンド [T0](#) および [T00](#) に対して位置合わせモードを設定します。

[T0](#) および [T00](#) は、ソース行を次の行と連結するために使用します。どちらのコマンドも、[BNDS](#) コマンドで設定された水平方向の境界内にのみ適用されます。

LEFT	テキストを左境界に揃えます。
RIGHT	テキストを右境界に揃えます。
BOTH	テキストを左右両方の境界に揃えます。

例：

列10と列60に水平方向の境界を設定し、左詰めの整列を有効にするには、次のコマンドを使用します。

```
BNDS 10 60;JUSTIFY LEFT
```

[T0](#) 行コマンドを使用して行をマークする（または [T00](#) 行コマンドを 2 回使用して行のブロックをマークする）と、マークされた行（または行間）で列10～60のテキストが列10に左詰めになります。

LABEL

```
LABEL .label
```

このコマンドは、指定された *label* で現在の行（編集エリアの上部に現在表示されている行）をマークするために使用します。

label は、1～4 文字の英数字文字列です。

例：

現在の行を [.X](#) でラベル付けするには、次のコマンドを使用します。

```
LABEL .X
```

行のブロックに 2 つのラベルを付けることもできます。例えば、ラベル .X および .Y でブロックをマークするには、上記の例で示すように .X で現在の行をマークしてから（ブロックの最初の行をマークすることを前提とする）、ブロックの最後の行が現在の行になるまでスクロールし、LABEL .Y コマンドを発行してその行を .Y でマークします。

行のラベル付けには、[.label](#) 行コマンドも使用できます。

LC

LC	$ \begin{bmatrix} * \\ [T]'string' \\ 'string' \\ C'string' \\ X'string' \\ P'string' \end{bmatrix} \begin{bmatrix} .X \\ .X .Y \end{bmatrix} \begin{bmatrix} n \\ n m \end{bmatrix} \begin{bmatrix} ALL \\ NEXT \\ PREV \\ FIRST \\ LAST \end{bmatrix} \begin{bmatrix} CHARS \\ WORD \\ PREFIX \\ SUFFIX \end{bmatrix} \begin{bmatrix} NX \\ X \end{bmatrix} $
----	---

このコマンドは、1 行以上の行を小文字に変更するために使用します。

指定した文字 *string* を含む行だけを小文字に変更するように指定できます。アポストロフィを *string* の一部として使用するには、2 つのアポストロフィを入力する必要があります。

string は、次のセクションで説明している方法で指定できます。

T' string'	小文字か大文字かに関係なく、 <i>string</i> を含む行を変更します。これがデフォルトです。																		
' string'	T' string' も同様です。																		
C' string'	指定文字列に完全に一致する <i>string</i> を含む行を変更します。																		
X' string'	指定した 16 進数文字 <i>string</i> に対応する文字列を含む行を変更します。																		
P' string'	次のワイルドカード文字が指定された <i>string</i> を含む行を変更します。 <table border="1" style="margin-left: 20px;"> <tr><td></td><td></td></tr> <tr><td>=</td><td>任意の文字</td></tr> <tr><td>§</td><td>英文字</td></tr> <tr><td>#</td><td>数字</td></tr> <tr><td>\$</td><td>特殊文字</td></tr> <tr><td>^</td><td>空白以外の文字</td></tr> <tr><td>-</td><td>数字以外の文字</td></tr> <tr><td><</td><td>小文字</td></tr> <tr><td>></td><td>大文字</td></tr> </table>			=	任意の文字	§	英文字	#	数字	\$	特殊文字	^	空白以外の文字	-	数字以外の文字	<	小文字	>	大文字
=	任意の文字																		
§	英文字																		
#	数字																		
\$	特殊文字																		
^	空白以外の文字																		
-	数字以外の文字																		
<	小文字																		
>	大文字																		

エディタコマンド

*	前のコマンド (LC、DELETE、EXCLUDE など) で指定した <i>string</i> を含む行を変更します。
.X	詳細については、「 行指定 」を参照してください。
.X .Y	
<i>n</i>	詳細については、「 列指定 」を参照してください。
<i>n m</i>	
ALL	詳細については、「 操作の方向 」を参照してください。
NEXT	
PREV	
FIRST	
LAST	
CHARS	詳細については、「 特殊なオカレンス 」を参照してください。
WORD	
PREFIX	
SUFFIX	
NX	詳細については、「 表示されている行または表示されていない行 」を参照してください。
X	

パラメータを指定せずに「LC」コマンドを入力すると、現在の行が小文字に変更されます。

例：

```
LC C'NAME' .X .Y ALL PREFIX NX
```

このコマンドは、.X および .Y というラベルの行のブロック内の、表示されているすべての行のうち、単語の接頭辞として文字列 NAME (ここに入力されているとおりの大文字) を含む行を小文字に変更します。

LIMIT

`LIMIT [n]`

このコマンドでは、[FIND](#) または [RFIND](#) コマンドで検索する行の最大数を指定します。パラメータ *n* は、検索する行数です。

LOCATE

`[LOCATE] { 0
n
.label }`

このコマンドは、特定の行を編集エリアの上部にスクロールする（つまり現在の行にする）ために使用します。

このコマンドでは、次のオプションを使用できます。

0	ソースコードの最初の行を現在の行にします。
<i>n</i>	<i>n</i> 行を現在の行にします。
<i>.label</i>	<i>.label</i> というラベルの行を現在の行にします。

例：

`LOC 32`

行番号 32 が編集エリアの最上位に配置されます。

`32`

上記と同様です。

`LOC .X`

`.X` というラベルが付けられた行が編集エリアの最上位に配置されます。

LOG

```
LOG [ ON ]  
      [ OFF ]
```

このコマンドは、内部ログファイルを有効または無効にします。

ログファイルは、セッションが開始してからエディタで行ったすべての変更の履歴です。ログファイルが有効になっているときは、ENTERキーを押すごとに、前回ENTERキーを押してから加えられた変更がログファイルに記録されます。UNDOコマンドを使用すると、エディタセッションの開始以降に加えられた変更を連続的に取り消すことができます。

! **重要:** ソースワークエリアをクリアした場合、別のNaturalオブジェクトをソースに読み込んだ場合、またはプログラムエディタセッションを終了した場合は、ログファイルのすべてのエントリがクリアされます。

MASK

```
MASK [ ON ]  
      [ OFF ]
```

このコマンドは、マスク機能を有効または無効にします。マスク機能がアクティブになっていると、エディタで行を挿入するたびに、定義済みのテキスト行が空行の代わりに入力されます。マスク行の定義には、次の段落で説明するように、MASK行コマンドを使用します。マスク機能は、同一または非常に似た複数行のコードを記述する必要があるときに役に立ちます。

➤手順 9.1. マスク行を定義および使用するには

1 任意のソース行に MASK 行コマンドを入力し、ENTERキーを押します。

コマンドを入力した行の上に [=mask] で示される空行が表示されます。

2 マスク行として定義するテキストをこの空行に入力し、ENTERキーを押します。

このマスク行は、新しいマスク行でマスクを更新するか、またはマスク機能を無効にするまで、現在のソースで有効です。

3 MASK ON エディタコマンドを入力します。

マスク機能が有効になります。定義されたマスク行は、行挿入操作で追加されたすべての行に表示されます。

4 行挿入コマンドを入力します。例：I2

新しい行が2行、マスク行のテキストが指定されてソースに挿入されます。マスク行のテキストは、挿入コマンドを使用して追加されたすべての行に指定されます。

- 5 新しい行のテキストを変更します。テキストを変更しなかった場合、挿入された行は次回 ENTER キーを押したときに削除されます。

MASK OFF コマンドはマスク機能を無効にしますが、マスク行の内容は削除しません。

MWINDOW

MWINDOW $\left[\begin{smallmatrix} n \\ n \ m \end{smallmatrix} \right]$

このコマンドは、コマンドパラメータに従ってデータウィンドウを移動するために使用します。

<i>n</i>	データウィンドウを挿入する行の番号。
<i>m</i>	データウィンドウを挿入する列の番号。

「[データウィンドウを使用したテキストのコピーまたは移動](#)」も参照してください。

NEXT

NEXT $\left[\begin{smallmatrix} * \\ object-name \end{smallmatrix} \right]$

このコマンドは、次回の同時編集セッションを表示するために使用します。複数の編集セッションが同時に稼動しており、プロファイルパラメータ EDTRB (『パラメータリファレンス』ドキュメントを参照) が設定されていることが前提となります。次のコマンドパラメータはオプションです。

*	現在稼動中のすべてのセッションの選択リストを表示します。
<i>object-name</i>	現在稼動中の編集セッションを名前で直接呼び出します。

ORDER

```
ORDER { ALL
          n
          n m }
```

このコマンドは、ソース行を連結するために使用します。

ALL	すべての行を連結します。
n	n 行から最後の行までを連結します。
n m	n 行から m 行までを連結します。

ORDER コマンドは、BNDS コマンドで設定された水平方向の境界内にのみ適用されます。

設定されている境界内で行が連結され、可能な限り多くの文字が挿入されます。1行に収まらない単語は次の行に自動的に配置されます。

ソース行の連結には、行コマンド [TF](#)、[TO](#)、および [T00](#) も使用できます。

POINT

```
POINT
```

このコマンドは、行コマンド [NZ](#) でマークされた行を編集エリアの最上位に配置します。

POWER

```
POWER
```

このコマンドは、エディタを挿入モードに切り替えます。1行以上のテキスト行を入力できる空白画面が表示されます。入力後に ENTER キーを押すと、テキストは編集エリアの最初の行に挿入されます。

PROF

PROF[*n*]

このコマンドは、エディタプロファイルをエディタ画面の最上位に表示します。

n を使用して、表示する追加の行を指定します。*n* の設定可能値を以下に示します。

6	エディタプロファイルおよびすべてのタブ位置 (TABS コマンドで指定) を表示します。
7	6 と同様のものと、さらにマスク行 (MASK コマンドで指定) を表示します。
8	7 と同様のものと、さらに境界 (BNDS コマンドで指定) を表示します。
9	8 と同様のものと、さらに列番号 (COLS コマンドで指定) を表示します。

PROFILE

PROFILE

このコマンドは、エディタプロファイル機能を呼び出します。現在のセッションおよび今後のセッション用にエディタのデフォルトを変更可能にします。エディタプロファイル機能の詳細については、「[永続的に使用するためのプロファイル設定の変更](#)」セクションを参照してください。

PROTECT

PROTECT [INS]
[ON]
[OFF]

このコマンドは、[接頭辞エリア](#)を保護するために使用します。

エディタコマンド

INS	行挿入コマンドを使用して追加した行の接頭辞エリアを保護します。
ON	保護を有効にします。
OFF	保護を無効にします。

RCHANGE

RCHANGE

このコマンドは最後の [CHANGE](#) コマンドを繰り返します。

RESET

RESET

このコマンドは、保留状態のエディタコマンドおよび行コマンドをすべてリセットし、行ラベルをすべて削除します。

RFIND

RFIND

代替 PF キー：PF5

このコマンドは最後の [FIND](#) コマンドを繰り返します。

SET TYPE

SET TYPE	CLASS	}
	4	
	COPYCODE	
	FUNCTION	
	7	
	HELPROUTINE	
	PROGRAM	

SET TYPE	SUBPROGRAM	}
	N	
	SUBROUTINE	
	TEXT	

このコマンドは、ソースワークエリアに現在表示されているオブジェクトのタイプを変更します。

SHIFT

SHIFT	[<i>n</i>]	[RIGHT]
		LEFT

このコマンドは、.X ラベルと .Y ラベルの間の行のブロックを *n* 列右または左に（または最後の空白以外の文字まで）移動します。移動のデフォルトは、5 列右です。

<i>n</i>	行を移動する列数（デフォルト値は 5）。
RIGHT	行のブロックを右に移動します（デフォルト）。
LEFT	行のブロックを左に移動します。

SORT

```
SORT [n m] [ .X ] [ .X .Y ] [ A ] [ D ]
```

SORT コマンドは、エディタ内の行を昇順または降順にソートします。パラメータ指定なしで SORT を入力すると、オブジェクトに含まれるすべてのテキストが昇順にソートされます。

<i>n m</i>	列 <i>n</i> から <i>m</i> までをソートします。
.X	.X というラベルの行からソースの最後までをソートします。
.X .Y	.X というラベルの行から .Y というラベルの行までソートします (.X および .Y は 4 文字以内の任意の文字列)。
A	テキストを昇順 (A から Z) にソートします。
D	データを降順 (Z から A) にソートします。

SPLIT

```
SPLIT { PROGRAM object-name [library-name] |
        VIEW    object-name [SHORT] |
        END }
```

このコマンドは、[画面分割モード](#)を設定し、別のオブジェクトのソースをエディタ画面に表示します。

PROGRAM	プログラム、サブプログラム、サブルーチン、ヘルプルーチン、データエリア（グローバル、ローカル、パラメータ）、コピーコード、テキスト、マップ、クラス、または機能を表示します。
VIEW	ビュー (DDM, Predict または SYSDDM で定義) を表示します。SHORT を指定すると、ビューが短縮形で表示され (つまり、Adabas ショートネームおよび対応する Natural フィールド名のみ表示)、フィールドヘッダーやフィールド編集マスク情報は表示されません。
END	画面分割モードを終了します。

PROGRAM または VIEW を指定する場合、*object-name* にアスタリスク (*) を指定すると、使用可能なすべてのオブジェクトのリストを表示できます。1 文字以上の文字の後にアスタリスク (*) を付けると、これらの文字で始まる名前のオブジェクトのみが表示されます。

SPLIT の使用例については、「[SPLIT を使用して定義を表示およびコピーするには](#)」を参照してください。

SWAP

SWAP

SWAP コマンドは、[画面分割モード](#)（関連セクションを参照）で 2 つのオブジェクトを切り替えます。この操作中、カーソルは 1 つのオブジェクトから別のオブジェクトに切り替ります。

TABS

TABS	[ON [<i>tab-character</i>] OFF [LEFT RIGHT DECIMAL] [<i>tab-character</i>] [<i>column...</i>]]
------	--

このコマンドは、タブの設定を制御するために使用します。

コマンド TABS ON または TABS OFF を使用して、論理的または物理的なタブ機能を有効または無効にすることができます。タブ機能は、タブ設定を変更する任意のコマンドで有効にすることもできます。

例えば、次のコマンドは、論理タブ機能を有効にし、論理タブ文字としてアンパサンド記号 (&) を使用します。

TABS &

タブ位置は、TABS コマンドを使用して設定します。例えば、次のコマンドは列 10、20、および 30 にタブを設定します。

TABS 10 20 30

論理タブ文字をテキストの前に挿入すると、テキストを入力して特定のタブ位置に自動的に移動できます。タブ文字を 1 つ付けるとテキストは次のタブ位置に移動し、タブ文字を 2 つ付けると 2 番目のタブ位置に移動します。

エディタコマンド

現在の TABS コマンド設定を表示するには、[PROF](#) コマンドを発行します。

現在のタブ位置を表示するには、[TABS](#) 行コマンドを発行します。

タブ位置とは別に、[TABS](#) コマンドで次のパラメータを指定できます。

LEFT	タブ位置で左詰めにテキストを配置します。
RIGHT	タブ位置で右詰めにテキストを配置します。
DECIMAL	テキスト内の小数点がタブ位置になるように、テキストを配置します。

特定の列でテキストのタブ設定を行うために、複数のタブ文字を使用できます。このためには、[TABS](#) 行コマンドを発行し、タブ位置を示す各アスタリスク (*) の位置に別の特殊文字を入力します。これらの特殊文字に続く入力は、対応する列でタブ設定されます。各タブ文字の後ろに L (LEFT) 、 R (RIGHT) または D (DECIMAL) を入力すると、タブ位置のテキストの配置を指定できます。

次のタブ機能の例では、タブ文字としてアンパサンド (&) を使用しています。列位置の表示には [COLS](#) 行コマンドが発行されています。

- [例 1 - タブ位置](#)
- [例 2 - TABS RIGHT](#)
- [例 3 - TABS DECIMAL](#)
- [例 4 - 混在した位置揃え](#)
- [例 5 - 複数のタブ記号](#)
- [例 6 - タブ設定記号としての空白の使用](#)

例 1 - タブ位置

次のコマンドを実行します。

```
TABS 10 20 40 LEFT
```

タブ列 10、20、および 40 で左詰めの論理タブを有効にします。ENTER キーを押すと、入力テキスト行

```
&abc &def &ghi
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      abc      def      ghi
```

例 2 - TABS RIGHT

コマンド

```
TABS RIGHT
```

右詰めの論理タブを有効にします。ENTER キーを押すと、入力テキスト行

```
&abc &def &ghi
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      abc      def      ghi
```

例 3 - TABS DECIMAL

コマンド

```
TABS DECIMAL
```

タブ位置に小数点を揃える論理タブを有効にします。ENTER キーを押すと、入力テキスト行

```
&15.27$ &16.3 EUR &13 IS
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      15.27$    16.3 EUR      13 IS
```

例 4 - 混在した位置揃え

以下のコマンドを発行します。

```
TABS 10 20 30 40 50
```

次に、TABS 行コマンドを発行します。これにより、現在のタブ位置が次のように表示されます。

エディタコマンド

```
=tabs> * * * *
```

必要に応じて、`L`、`R`、または`D`を各タブ位置の横に入力します（タブ位置を指定しなければ、最後の TAB コマンドの値が使用されます）。

```
=tabs> *R *D *D *L
```

ENTER キーを押すと、入力テキスト行

```
&start &0.01 &0.02 &0.03 &end
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      start      0.01      0.02      0.03      end
```

例 5・複数のタブ記号

`[=tabs>]` 行のアスタリスクを他の特殊文字に置き換え、それを次のように左詰めに指定します。

```
=tabs> ]L &L $L =L
```

ENTER キーを押すと、入力テキスト行

```
=first$second#third&fourth]fifth
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      first
      second
      third
      fourth
      fifth
```

例 6・タブ設定記号としての空白の使用

次のコマンドを発行します。

```
TABS ' '
```

1つの空白をタブ文字として有効にします。つまり、1つの空白で分割された単語がタブ設定されます。ENTER キーを押すと、入力テキスト行

```
this is a blank tabulation
```

は、次のように表示されます。

```
=cols> -----1-----2-----3-----4-----5-----6
      this      is      a      blank      tabulation
```

UC

UC	$\begin{bmatrix} * \\ [T]'string' \\ 'string' \\ C'string' \\ X'string' \\ P'string' \end{bmatrix} \begin{bmatrix} .X & .Y \end{bmatrix} \begin{bmatrix} n & m \end{bmatrix} \begin{bmatrix} ALL \\ NEXT \\ PREV \\ FIRST \\ LAST \end{bmatrix} \begin{bmatrix} CHARS \\ WORD \\ PREFIX \\ SUFFIX \end{bmatrix} \begin{bmatrix} NX \\ X \end{bmatrix}$
----	--

UC コマンドは、1行以上の行を大文字に変換します。LC コマンドと同じパラメータを使用します。パラメータを指定せずに UC コマンドを入力すると、現在の行が大文字に変更されます。

UNDO

UNDO	$\begin{bmatrix} ALL \\ n \end{bmatrix}$
------	--

代替 PF キー：PF11

ログファイルが有効になっている場合 (LOG コマンドを参照) 、UNDO コマンドは、前回 ENTER キーを押してから加えられたすべての変更を取り消します。UNDO コマンドを繰り返し使用すると、連続的に行った変更が逆の順序で取り消されます。このように、セッション開始時の元の状態にメンバを戻すまで、1つずつすべての変更内容を取り消すことができます。

エディタコマンド

⚠ 重要: ソースワークエリアをクリアした場合、別の Natural オブジェクトをソースに読み込んだ場合、またはプログラムエディタセッションを終了した場合は、ログファイルのすべてのエントリがクリアされます。

UNDO コマンドでは、次のパラメータを指定できます。

ALL	現在のエディタセッションで行われたすべての変更をバックアウトします。
<i>n</i>	最後の <i>n</i> 個の変更をバックアウトします。

WINDOW

```
WINDOW { line1 line2
          line1 line2 column1
          line1 line2 column1 column2 }
```

このコマンドは、コピーまたは移動するデータウィンドウを定義するために使用します。ウィンドウの開始行と開始列および終了行と終了列は、コマンドパラメータで指定します。少なくとも *line1* および *line2* が必要です。

<i>line1 line2</i>	<i>line1</i> の列 1 で開始し <i>line2</i> の最後の列で終了するウィンドウを定義します。
<i>line1 line2 column1</i>	<i>line1</i> の <i>column1</i> で開始し <i>line2</i> の最後の列で終了するウィンドウを定義します。
<i>line1 line2 column1 column2</i>	<i>line1</i> の <i>column1</i> で開始し <i>line2</i> の <i>column2</i> で終了するウィンドウを定義します。

「[データウィンドウを使用したテキストのコピーまたは移動](#)」も参照してください。

X

X

このコマンドは、行コマンド [.X](#) でマークされた行を編集エリアの最上位に配置します。

XSWAP

XSWAP

このコマンドは、表示されている行と非表示の行を入れ替えるために使用します。EXCLUDE コマンドを使用すると行が非表示になります。

Y

Y

このコマンドは、行コマンド .Y でマークされた行を編集エリアの最上位に配置します。

共通のコマンドオプション

複数のエディタコマンドに使用できるオプションがいくつかあります。これらのオプションについて、次の各セクションで説明します。

- 行指定
- 列指定
- 表示されている行または表示されていない行
- 操作の方向
- 特殊なオカレンス

行指定

次のオプションは、エディタコマンドの適用対象を1行または .X 行コマンドと .Y 行コマンドでラベル付けされた行のブロックに制限するために使用できます。

.X	エディタコマンドは、.X でラベル付けされた行にのみ影響します。例外：SORT エディタコマンド
.X .Y	エディタコマンドは、.X というラベルの行から .Y というラベルの行までにのみ影響します。このオプションを使用する場合は、コマンドにパラメータ ALL も指定する必要があります。



注意: .X および .Y には、1~4 文字の任意の英文字のラベルを指定できます (LABEL エディタコマンドおよび .label 行コマンドの説明も参照)。

列指定

次のオプションは、エディタコマンドの影響を特定の列範囲に限定するために使用できます。これらの列番号は実際のソースコードの列であり、ソースコードの前にある行番号はカウントされません。したがって、コマンドで列1を指定すると、物理的には画面の8番目の列になるかもしれません、実際には編集しているソースコードの最初の列になります。

<i>n</i>	コマンドは、指定された文字列が列 <i>n</i> から始まる行にのみ影響します（つまり、文字列の最初の文字が列 <i>n</i> にあることが必要）。
<i>n m</i>	コマンドは、指定された文字列が列 <i>n</i> と <i>m</i> の間の任意の位置にある行にのみ影響します。

表示されている行または表示されていない行

次のオプションは、表示されていない行のみまたは表示されている行にのみコマンドが影響するようにするために使用できます。

<i>NX</i>	コマンドは、非表示ではない行のみ、つまり現在表示されている行にのみ影響します。
<i>X</i>	コマンドは、非表示の行のみ、つまり EXCLUDE コマンドで指定された、現在表示されていない行にのみ影響します。表示されていない行は、エディタコマンド機能がその行に対して実行されても、非表示のままになります。

操作の方向

次のオプションは、エディタコマンドの操作方向を指定するために使用できます。

<i>NEXT</i>	コマンドは、指定された <i>string</i> が存在する次の行に影響します（カーソル位置から開始）。これはデフォルト設定です。
<i>PREV</i>	コマンドは、指定された <i>string</i> . が存在する前の行に影響します。
<i>FIRST</i>	コマンドは、指定された <i>string</i> が存在する最初の行に影響します。
<i>LAST</i>	コマンドは、指定された <i>string</i> が存在する最後の行に影響します。
<i>ALL</i>	コマンドは、指定された <i>string</i> が存在するすべての行に影響します。

特殊なオカレンス

次のオプションは、指定された *string* の特別なオカレンスのみにコマンドが影響するようにするため使用できます。

CHARS	コマンドは、指定された <i>string</i> が存在する任意の行に影響します。これはデフォルト設定です。
WORD	コマンドは、指定された <i>string</i> が 1 つの語を形成する行にのみ影響します。
PREFIX	コマンドは、指定された <i>string</i> が 単語の先頭にある行にのみ影響します。
SUFFIX	コマンドは、指定された <i>string</i> が単語の末尾にある行にのみ影響します。

10 スクロール用エディタコマンド

このセクションでは、編集エリアに現在表示されているソースコードのスクロールに使用できるエディタコマンドおよび PF キー（該当する場合）について説明します。

エディタコマンドは、エディタ画面のコマンド行に入力します。インストールの設定によっては、エディタコマンドを小文字で入力できます。ただし、このセクションでは、コマンドであることがわかるように、すべてのコマンドを大文字で示しています。

次の表で、テキストの下線部分はコマンドとして許容される省略形を表しています。

Command	関数
<u>BOTTOM</u> または ++	ソースの末尾までスクロールします。
<u>TOP</u> または --	ソースの先頭までスクロールします。
<u>DOWN</u> または + または ENTER キー または PF8	scroll mode で指定された分だけ前にスクロールします。
<u>DOWN</u> <i>n</i>	<i>n</i> 行前にスクロールします。
<u>+n</u>	<i>n</i> 行前にスクロールします。

スクロール用エディタコマンド

Command	関数
UP	scroll mode で指定された分だけ後ろにスクロールします。
または	
-	
または	
PF7	
UP <i>n</i>	<i>n</i> 行後ろにスクロールします。
- <i>n</i>	<i>n</i> 行後ろにスクロールします。
LEFT	画面に表示される列数だけ左にスクロールします。
LEFT <i>n</i>	<i>n</i> 列左にスクロールします。
LEFT MAX	最大列数スクロールして画面の左端へ移動します。
RIGHT	画面に表示される列数だけ右にスクロールします。
RIGHT <i>n</i>	<i>n</i> 列右にスクロールします。
RIGHT MAX	最大列数スクロールして画面の右端へ移動します。
FIX <i>n</i>	右向きにスクロールしたときに表示したままにしておく列（列1から開始）の数 <i>n</i> を指定します。

11 行コマンド

このセクションでは、プログラムエディタに用意されている行コマンドについて説明します。行コマンドは常に、その行コマンドを入力したソース行、または複数の行コマンドでマークされた行のブロックに適用されます。

インストールの設定によっては、行コマンドを小文字で入力できます。ただし、このセクションでは、コマンドであることがわかるように、すべてのコマンドを大文字で示しています。

➤手順 11.1. 行コマンドを実行するには

- 1 目的のソース行の横にある[接頭辞エリア](#)で、左端の桁から行コマンドを上書き入力し、ENTERキーを押します。

次のいずれかが該当する場合は、行コマンドを入力できません。

- 挿入モードが有効である（一連のアポストロフィ（'……'）で示されます）。この場合は、挿入モードを非挿入モードに切り替えてください。
- [接頭辞エリア](#)が保護されている。この場合は、[PROTECT](#)エディタコマンドを使用して、保護を解除してください。

または：

コマンド行にコロン（:）に続けて行コマンドを入力し、コマンドを適用する行にカーソルを置き、ENTERキーを押します。例えば、次のようになります。

コマンド行に「:I」と入力し、カーソルを先頭行に置き、ENTERキーを押します。これにより、先頭行の下に新しい行が挿入されます。

- 2 必要に応じて、[RESET](#)エディタコマンドを入力して、保留状態の行コマンドをリセットし、行ラベルをすべて削除します。

行コマンド

Command	関数
)	この行を 2 列右に移動します。
)n	行の他のテキストに関係なく、この行を n 列右に移動します。移動された行のテキストが失われる場合があります。
))n	右に n 列移動する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには))n がもう 1 つ必要です。この機能は、2 番目の))n が指定されたときに実行されます。ブロックの他のテキストに関係なく、ブロックが移動されます。移動されたブロックのテキストが失われる場合があります。
(この行を 2 列左に移動します。
(n	他のテキストに関係なく、この行を n 列左に移動します。移動された行のテキストが失われる場合があります。
((n	左に n 列移動する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには ((n がもう 1 つ必要です。この機能は、2 番目の ((n が指定されたときに実行されます。
<	この行のテキストを左に 2 列移動します。
>	この行のテキストを右に 2 列移動します。
>n	この行のテキストを右に n 列移動します。または、最後の空白以外の文字まで移動します。テキストは失われません。
>>n	右に n 列または最後の空白以外の文字まで移動する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには >> がもう 1 つ必要です。この機能は、2 番目の >> が指定されたときに実行されます。
<n	この行のテキストを左に n 列移動します。または、最初の空白以外の文字まで移動します。
<<n	左に n 列または最初の空白以外の文字まで移動する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには << がもう 1 つ必要です。この機能は、2 番目の << が指定されたときに実行されます。
A	移動行コマンド (M、Mn、MM) またはコピー行コマンド (C、Cn、CC) の対象行をマークします。移動またはコピーされた行は、この行の後ろに挿入されます。
B	移動行コマンド (M、Mn、MM) またはコピー行コマンド (C、Cn、CC) の対象行をマークします。移動またはコピーされた行は、この行の前に挿入されます。
BNDS	この行の境界位置を表示します。BNDS エディタコマンドの説明も参照してください。
C	行コマンド A、B、または 0 で示された位置にこの行をコピーします。
Cn	行コマンド A、B、または 0 で示された位置にこの行と次の $n-1$ 行をコピーします。
CC	コピーする行のブロックの最初の行をマークします。コピーするブロックの最後の行をマークするには CC コマンドがもう 1 つ必要です。この機能は、2 番目の CC が指定されたときに実行されます。行は、行コマンド A、B、または 0 で示された位置にコピーされます。
CX	.X というラベルの行をコピーします。テキストをこの行の後ろに挿入します。
CY	.Y というラベルの行をコピーします。テキストをこの行の後ろに挿入します。
CX-Y	.X というラベルの行から .Y というラベルの行に、行のブロックをコピーします。テキストをこの行の後ろに挿入します。
COLS	この行の列位置を表示します。

Command	関数
D	この行を削除します。
D n	現在の行および次の $n-1$ 行を削除します。
DD	削除するブロックの最初の行をマークします。削除するブロックの最後の行をマークするには DD コマンドがもう 1 つ必要です。2 番目の DD を入力した後に削除が実行されます。
DX	.X というラベルの行を削除します。
DY	.Y というラベルの行を削除します。
DX-Y	.X というラベルの行から .Y というラベルの行に、行のブロックを削除します。
F	除外された最初の行を含めます。
F n	除外された最初の n 行を含めます。
I	1 行を挿入します。エディタは、[.....] で示される挿入モードに切り替わります。つまり、新しい行にテキストを入力するか空白を入力して ENTER キーを押すと、新しい行が自動的に挿入され、その位置にカーソルが置かれます。 挿入された行に新しいテキストを入力しないで ENTER キーを押すと、挿入モードが終了して空行が削除されます。 挿入した行に定義済みの内容を埋め込むこともできます（ MASK エディタコマンドの説明も参照）。
I n	n 行を挿入します。新しい行にテキストを入力できます。ENTER キーを押すと、使用されていない行が削除されますが、カーソルのある 1 行の空行は残ります（エディタは挿入モードのままで）。
.I(obj, pos, n)	現在のシステムファイル内の現在のライブラリに含まれている、任意のオブジェクトをエディタ画面に挿入します。 注意: このコマンドは、接頭辞エリアではなく、ソース行の先頭列に入力します。 pos エントリは、挿入処理を開始する行を示します。例えば、 pos を 20 に設定すると、ソースの 20 行目から挿入が開始されます。 n エントリは、挿入する行数を示します。 オブジェクトが Natural マップの場合は、すべての変数が定義済みの INPUT USING MAP ステートメント（『ステートメント』ドキュメントの「INPUT 構文2 - 定義済みマップレイアウトの使用」を参照）が現在の行に自動的に含まれます。 オブジェクトがデータエリアの場合は、コメント行以外の全データエリアが含まれます。編集エリアに表示されているソースに含めることができるのは、STOW システムコマンドを使用して保存およびカタログされたデータエリアのみです。 オブジェクトがアダプタの場合は、すべての変数が定義済みの PROCESS PAGE USING（『ステートメント』ドキュメントの「構文2 - PROCESS PAGE USING」を参照）が現在の行に自動的に含まれます。
.I(*)	現在のライブラリにあるオブジェクトの選択リストを表示します。

行コマンド

Command	関数
J	次の行をこの行に連結します。行の分割箇所にカーソルを置いて ENTER キーを押すことにより、連結する次の行の文字数を指定できます。行全体を連結するには、連結する行の外にカーソルを置きます。このコマンドは、 TJ コマンドと同一です。
Ln	除外された最後の <i>n</i> 行を含めます。
LC	この行の文字を小文字に変更します。
LC <i>n</i>	この行と次の <i>n-1</i> 行の文字を小文字に変更します。
LCC	すべての文字を小文字に変更する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには LCC がもう 1 つ必要です。この機能は、2 番目の LCC が指定されたときに実行されます。
LJ	この行で設定されている境界内のテキストを左境界に揃えます。
LJJ	左に揃える設定済み境界内の行のブロックの最初の行をマークします。揃えるブロックの最後の行をマークするには LJJ コマンドがもう 1 つ必要です。2 番目の LJJ コマンドが発行された後、行揃えが実行されます。
M	行コマンド A 、 B 、または 0 で示された位置にこの行を移動します。
M <i>n</i>	行コマンド A 、 B 、または 0 で示された位置にこの行と次の <i>n-1</i> 行を移動します。
MM	移動する行のブロックの最初の行をマークします。移動するブロックの最後の行をマークするには MM コマンドがもう 1 つ必要です。この機能は、2 番目の MM が指定されたときに実行されます。行は、行コマンド A 、 B 、または 0 で示された位置に移動されます。
MASK	マスクを定義できる空行を編集エリアに挿入します。この行は、 In 行コマンドを使用して 1 行以上の新しい行を作成するたびに挿入されます。 MASK エディタコマンドの説明および「マスク行を定義および使用するには」も参照してください。
MX	.X というラベルの行を移動します。この行の後ろに挿入します。
MY	.Y というラベルの行を移動します。この行の後ろに挿入します。
MX-Y	.X というラベルの行から .Y というラベルの行に、行のブロックを移動します。この行の後ろに挿入します。
N	この行で行われた変更は、ENTER キーを押したときに有効なりません。
NZ	POINT エディタコマンドが発行されたときに、この行を編集エリアの最上位に配置します。
0	移動行コマンド (M 、 Mn 、 MM) またはコピー行コマンド (C 、 Cn 、 CC) の対象行としてこの行をマークします。移動またはコピーされた行はこの行とマージされます。つまり、この行の空白文字は上書きされます。
0 <i>n</i>	移動行コマンド (M 、 Mn 、 MM) またはコピー行コマンド (C 、 Cn 、 CC) の対象行として、この行および次の <i>n-1</i> 行をマークします。移動またはコピーされた行はこれらの行とマージされます。つまり、これらの行の空白文字は上書きされます。
00	移動行コマンド (M 、 Mn 、 MM) またはコピー行コマンド (C 、 Cn 、 CC) 用に、対象となる行のブロックの最初の行をマークします。対象行のブロックの最後の行をマークするには 00 コマンドがもう 1 つ必要です。
	移動またはコピーされた行はこれらの行とマージされます。つまり、これらの行の空白文字は上書きされます。
R	この行を 1 回繰り返します。

Command	関数
R n	この行を n 回繰り返します。
RR	繰り返す行のブロックの最初の行をマークします。繰り返すブロックの最後の行をマークするには RR コマンドがもう 1 つ必要です。繰り返し処理は、2 番目の RR を入力したときに実行されます。
RR n	行のブロックを n 回繰り返します。
RJ	この行で設定されている境界内のテキストを右境界に揃えます。
RJJ	右に揃える設定済み境界内の行のブロックの最初の行をマークします。揃えるブロックの最後の行をマークするには RJJ コマンドがもう 1 つ必要です。行揃えは、2 番目の RJJ を入力したときに実行されます。
S	この行をカーソル位置で始まる 2 行に分割します。コマンドを入力して行を分割する位置にカーソルを移動し、ENTER キーを押します。
T	ソースをスクロールして、マークした行を最上行にします。
TABS	この行のタブ位置を表示します。TABS エディタコマンドの説明も参照してください。
TC	この行で設定されている境界内のテキストを中央揃えにします。
TCC	中央揃えにする設定済み境界内のテキストのブロックの最初の行をマークします。中央に配置したブロックの最後の行をマークするには TCC コマンドがもう 1 つ必要です。中央揃えは、2 番目の TCC コマンドを発行したときに実行されます。
TE	この行の下のすべての行を非表示にし、挿入モードに切り替えます。非表示の行は、ENTER キーを押すと再表示されます。
TF	次の空行まで、この行を後に続く行に連結し続けます。境界設定を使用して、影響を受ける列を制限できます (BNDS エディタコマンドの説明を参照)。
TF n	この行コマンドには、右境界を指定する数値を指定できます。例えば、行コマンド TF5 はテキストを列 5 に揃えます。
TI	現在の行および設定されている境界内のすべての文字の順序を逆にします。
TII	設定されている境界内で順序を逆にする行のブロックの最初の行をマークします。ブロックの最後の行をマークするには TII がもう 1 つ必要です。この機能は、2 番目の TII が指定されたときに実行されます。
TJ	次の行をこの行に連結します。J 行コマンドと同じです。
T0	この行を次の行に連結します。
T00	連結する設定済み境界内の行のブロックの最初の行をマークします。連結するブロックの最後の行をマークするには T00 コマンドがもう 1 つ必要です。この機能は、2 番目の T00 が指定されたときに実行されます。
TS	この行をカーソル位置で 2 行に分割します。空行も自動的に挿入されますが、使用されなければ削除されます (S 行コマンドと同じです)。
UC	この行のすべての文字を大文字に変更します。
UC n	この行と次の $n-1$ 行のすべての文字を大文字に変更します。
UCC	すべての文字を大文字に変更する行のブロックの最初の行をマークします。ブロックの最後の行をマークするには UCC がもう 1 つ必要です。この機能は、2 番目の UCC が指定されたときに実行されます。
W	1 行のウィンドウを開きます。

Command	関数
Wn	n 行のウィンドウを開きます。
WC	データウィンドウをコピーします。カーソル位置は、コピーされたテキストを挿入するためにこの行が分割される列を示します。
WCn	この行を列 n で分割し、分割した行の間にテキストをコピーします。
WE	データウィンドウの終わりをマークします。WS と同じように機能します。ウィンドウの開始行と終了行が同じ場合は、WS コマンドを WE コマンドで置換します。設定されたウィンドウの確認として、エディタにより接頭辞エリアにメッセージとして「WW」が表示されます。 「 データウィンドウを使用したテキストのコピーまたは移動 」も参照してください。
WM	データウィンドウを移動します。WC と同様に機能しますが、元のテキストはコピー操作後に削除されます。 「 データウィンドウを使用したテキストのコピーまたは移動 」も参照してください。
WMn	この行を列 n で分割し、分割した行の間にテキストを移動します。
WS	データウィンドウの始まりをマークします。カーソル位置は、テキストの読み込み元である列を示します。コマンドが入力される行にカーソルがない場合、列1が使用されます。 「 データウィンドウを使用したテキストのコピーまたは移動 」も参照してください。
WSn	データウィンドウがこの行の列 n で開始されることを指定します。
X	この行を非表示にします。
Xn	この行および次の n 行を非表示にします。
XX	非表示にする行のブロックの最初の行をマークします。ブロックの2行目をマークするには XX がもう 1 つ必要です。この機能は、2 番目の XX が指定されたときに実行されます。
.X	この行を .X でマークします。 後述の .label も参照してください。
.Y	この行を .Y でマークします。 後述の .label も参照してください。
.label	この行を .label でマークします。label は、1~4 文字の英数字文字列です。 例えば、行コマンド .X はこの行に .X という名前を付け、行コマンド .Y はこの行に .Y という名前を付けます。 LABEL エディタコマンドの説明も参照してください。

12 エディタプロファイル

■ プロファイル設定の表示および非表示	82
■ 一時的に使用するためのプロファイル設定の変更	83
■ 永続的に使用するためのプロファイル設定の変更	84

各ユーザーには、個々のニーズに合わせてパラメータを設定できるエディタプロファイルがあります。エディタを最初に起動するときは、管理者が決めたデフォルト設定が使用されます。

デフォルトのエディタプロファイル設定を表示および変更するために使用できるオプションについて、次の各セクションで説明します。

プロファイル設定の表示および非表示

▶手順 12.1. 現在のエディタプロファイル設定を表示または非表示にするには

- 1 エディタのコマンド行で、次のコマンドを入力します。

PROF

(PROF エディタコマンドの説明も参照)

編集エリアの上部に、次の行が表示されます。

```
>> -----Columns 001 072 << Program SAGDEMO Lines 19 User SAG      ↵
Command ==>                                         Mode  Struct Lib  SAGTEST
***** **** top of data ****
=prof> date: 30/01/08 16:07:57 user: MMO      init size: 14 size: 14
=prof> var - 250,..recovery off (100 0)...autosave off... empty line off
=prof> mask off.caps on .hex off nulls on std.autoren on std auto order off
=prof> log off .mso on .fix off .escape on + . tabs off
=prof> advance on .protect off.limit off
```

- 2 プロファイル設定の表示を非表示にするには、エディタのコマンド行に次のコマンドを入力します。

RESET

プロファイル設定が非表示になります。現在のプロファイル設定は保持されます。

現在のエディタプロファイルの各項目と、項目を一時的に変更するため（変更できる場合）に使用できるエディタコマンドについては、次のセクションで説明します。

一時的に使用するためのプロファイル設定の変更

現在のエディタセッションに限って、または設定を再び変更するまで、デフォルトのエディタプロファイル設定を変更することができます。

▶手順 12.2. 現在のセッションに限ってエディタプロファイル設定を変更するには

- 変更するプロファイル項目について、次の表に示す該当するエディタコマンドを使用します。例えば、テキストを大文字に変換する場合は CAPS OFF を入力します。

エディタコマンドの構文で使用されている縦棒は、選択肢の区切りです。リストされているエディタコマンドの詳細については、「[エディタコマンド](#)」を参照してください。

プロファイル項目	説明	エディタコマンド
date	現在の日時。 この項目は変更できません。	なし
user	現在のログオンユーザー。 この項目は変更できません。	なし
init size	エディタ起動時のソースコード行数。 この項目は変更できません。	なし
size	情報行（プロファイル行およびメッセージ行など）を除く、ソースコード内の現在の行番号。 この項目は変更できません。	なし
var	現在の行の長さ。 この項目は変更できません。	なし
autosave	EXIT エディタコマンド発行時の自動保存を有効または無効にします。	AUTOSAVE
empty line	空白文字のみを含む行を自動的に削除するかどうかを指定します。	EMPTY ON OFF
mask	マスク行機能を有効または無効にします。	MASK ON OFF
caps	テキストを大文字に変換するかどうかを指定します。	CAPS ON OFF PGM
hex	文字を 16 進形式で表示するかどうかを指定します。	HEX ON OFF
autoren	ソース行番号の自動再設定を有効にすることを指定します。 この項目は変更できません。	なし
auto order	定義された境界内でテキストを自動的に調整します。	AORDER ON OFF

プロファイル項目	説明	エディタコマンド
log	ログファイルを有効または無効にします。有効の場合、 UNDO エディタコマンドを使用して最後の変更をバックアウトできます。	LOG ON OFF
mso	複数セッション操作が可能であることを示します。複数セッション操作とは、オブジェクト間でテキストをコピーするなど、 画面分割モード で2つの編集セッション間でテキストを交換する操作です。 この項目は変更できません。	なし
fix	固定数の列を表示するかどうか、および固定する列の数を指定します。	FIX ON OFF n
tabs	タブ機能を有効または無効にします。	TABS ON OFF
advance	行の更新後にカーソルを次の行に自動的に移動するかどうかを指定します。	ADVANCE ON OFF PAGE
protect	行番号の保護を指定します。	PROTECT ON OFF INS
limit	FIND エディタコマンドまたは RFIND エディタコマンドで検索する行の最大数を指定します。	LIMIT n

永続的に使用するためのプロファイル設定の変更

プログラムエディタプロファイル機能を使用すると、現在のセッションのプロファイル設定（前述を参照）だけでなく、今後のエディタセッションで使用するプロファイル設定も変更できます。エディタプロファイル機能を使用して変更した設定は、新しいNaturalセッションごとに、または再び変更するまで、有効です。これらの設定は、「[一時的に使用するためのプロファイル設定の変更](#)」で説明しているエディタコマンドを使用して、現在のセッションに限って上書きできます。

➤手順 12.3. 永続的に使用するためにエディタプロファイル設定を変更するには

- プログラムエディタのコマンド行で、次のコマンドを入力します。

PROFILE

プログラムエディタのプロファイル機能の **[Main Menu]** が開き、次のオプションが表示されます。

オプション	関数
Save	現在のプロファイルを保存します。
Modify	PA/PF キーおよびエディタコマンドのデフォルト設定を変更するための入力画面が呼び出されます。「 デフォルトのエディタ設定の変更 」を参照してください。
Read	【Profile Name】フィールドで指定する別のユーザーのエディタプロファイルを読み込みます。外部プロファイルの設定を、自分のプロファイル名で保存することができます。
Technical Info	現在編集中のオブジェクトおよびシステム環境に関する情報を表示します。「 技術情報の表示 」を参照してください。

このセクションでは、次のトピックについて説明します。

- [エディタプロファイルコマンド](#)
- [デフォルトのエディタ設定の変更](#)

エディタプロファイルコマンド

このセクションでは、エディタプロファイル機能で使用できるダイレクトコマンドおよび代替 PF キーについて説明します。ダイレクトコマンドは、エディタプロファイル下部のコマンド行（【Command ==>】）に入力します。

Command	PF キー	関数
CANCEL	PF12	現在の変更をキャンセルし、以前の画面を表示します。プロファイルを変更しても、現在のセッションには影響しません。
EXIT	PF3	現在のセッションを終了し、以前の画面に戻るか、または【EXIT Function】プロンプトウィンドウ（「 Exit 機能 」も参照）を呼び出します。
FLIP	なし	PF1～PF12 と PF13～PF24 とで切り替えます。
READ	PF6	【Profile Name】フィールドに現在含まれているユーザー ID のプロファイルパラメータを読み込みます。これまでに加えた未保存の変更はすべて上書きされます（【Main Menu】でのみ有効）。
SAVE	PF5	現在のセッションおよびデータベースの両方に関して、現在有効なプロファイルパラメータをすべて保存します。ただし、現在の機能は終了しません（【Main Menu】でのみ有効）。

デフォルトのエディタ設定の変更

【Main Menu】で【Modify】を選択するか PF4 キーを押すと、【Modify Defaults】画面が開き、次のオプションが表示されます。

オプション	関数
PA/PF-Keys	PA/PF キーの割り当てを変更します。「PA/PF キーの割り当ての表示および変更」を参照してください。
Commands	コマンドのデフォルトを変更します。「コマンドのデフォルトの変更」を参照してください。
Find	FIND コマンドのパラメータ設定を変更します。「FIND コマンドのデフォルトの変更」を参照してください。
General	【EXIT Function】プロンプトウィンドウの有効または無効の指定、およびエラー行の前に表示される行数の変更を行います。「全般的なデフォルト設定の変更」を参照してください。

PA/PF キーの割り当ての表示および変更

【Main Menu】で【PA/PF-Keys】を選択するか PF2 キーを押すと、現在の PA/PF キーの割り当てを示す【Modify PF/PA-Keys】画面が表示されます。PF キーの割り当てを変更するには、既存のコマンド（エディタコマンド、システムコマンドまたはプログラム名）の名前を、別のコマンド（最大 5 文字）で置き換えます。

▷手順 12.4. PA キーまたは PF キーの割り当てを変更するには

- 目的の PA キーまたは PF キーの横の入力フィールドに、有効なエディタコマンドまたはシステムコマンドの名前を入力するか、既存のプログラムの名前を入力します。最大で 5 文字まで入力できます。

エディタのデフォルトの PA/PF キー割り当ては次のとおりです。

PF キー	Command	関数
PF1	HELP	Natural オンラインヘルプ機能を呼び出します。
PF2	SAVE	システムコマンド SAVE を実行して、現在のシステムファイル内の現在のライブラリに、ソースワークエリアに現在表示されているソースコードを、ソースオブジェクトとして保存します。「ソースの保存およびカタログ」も参照してください。
PF3	EXIT	エディタを終了します。「Exit 機能」も参照してください。
PF4	RUN	編集エリアのプログラムをチェックおよび実行します。
PF5	RFIND	最後の FIND コマンドを繰り返します。
PF6	STOW	システムコマンド STOW を実行して、現在ソースワークエリアに表示されているソースコードを保存およびカタログします。「ソースの保存およびカタログ」も参照してください。

PF キー	Command	関数
PF7	UP または -	上方にスクロールします。
PF8	DOWN または +	下方にスクロールします。
PF9	CHECK	ソースワークエリアに現在表示されているソースコードに構文エラーがないかどうかをチェックします。
PF10	HOME	カーソルをコマンド行に置きます。
PF11	UNDO	ソースに対して最後に行われた変更をバックアウトします。
PF12	CANCEL	エディタを終了します。『 Exit 機能 』も参照してください。
PF13	PROFILE	エディタプロファイル機能を呼び出します。
PF14	RESET	保留状態のエディタコマンドおよび行コマンドをすべてリセットし、行ラベルをすべて削除します。
PF15	SWAP	画面分割モードが設定されている場合に、分割画面の上半分と下半分との間でカーソルを移動します。
PF16	LAST	システムコマンド LAST (『システムコマンド』ドキュメントを参照) を実行して、最後に発行されたコマンドを再び呼び出してコマンド行に配置します。
PF17	RCHANGE	最後の CHANGE エディタコマンドを繰り返します。
PF18	FLIP	PF キーの表示を PF1～PF12 と PF13～PF24 とで切り替えます。
PF19	TOP または --	ソースの先頭までスクロールします。
PF20	BOTTOM または ++	ソースの末尾までスクロールします。

PF キーの割り当てを変更するには、既存のコマンド（エディタコマンド、システムコマンドまたはプログラム名）の名前を、別のコマンド（最大 5 文字）で置き換えます。

コマンドのデフォルトの変更

[Main Menu] で [Commands] を選択するか PF5 キーを押すと、選択したエディタコマンドの有効または無効の指定およびエディタコマンドのデフォルト文字の指定ができます。以下の表では、画面に表示される入力フィールドについて説明します。特定のフィールドの詳細については、フィールドで疑問符 (?) を入力し、ENTER キーを押してください。

フィールド	説明
aorder	autoorder 機能を有効または無効にします。
autosave	autosave 機能を有効または無効にします。
caps	テキストを大文字に変換するかどうかを指定します。
cols	現在の列位置を示す行を表示するかどうかを指定します。
decimal character	タブで使用する数値の小数点を表す文字を指定します。 TABS エディタコマンドの「 例3 - TABS DECIMAL 」も参照してください。
empty	空白文字のみを含む行を自動的に削除するかどうかを指定します。
fix	右にスクロールしたときに fixlen (次項を参照) に入力されている列数を表示し続けるかどうかを指定します。
fixlen	fix (前項を参照) が ON に設定されている場合のみ表示されます。 右向きにスクロールしたときに表示し続ける列 (列1から開始) の数を指定します。
hex	文字を 16 進形式で表示するかどうかを指定します。
justify	位置合わせの有効または無効、および位置合わせのタイプを指定します。
limit	FIND コマンドまたは RFIND コマンドで検索する行の最大数を指定します。
log	ログファイルを有効または無効にします。有効の場合、 UNDO エディタコマンドを使用して最後の変更をバックアウトできます。
mask line	マスク行機能を有効または無効にします。
message line	メッセージ出力を有効または無効にします。
mso	画面分割セッション間のコピーなど、複数セッションの操作が許可されていることを示します。この項目は変更できません。
scroll mode	スクロールの実行方法 (ページ全体、半ページ、カーソルまで) を指定します。
tabs	タブ機能を有効または無効にします。
tabulator character	入力を特定のタブ位置に自動的に移動するために使用する論理タブ文字を指定します。

FIND コマンドのデフォルトの変更

【Main Menu】で【Find】を選択するか PF8 を押すと、現在のパラメータ設定を示す画面が表示されます。これらの設定は、プログラムエディタのコマンド行でパラメータを指定せずに **FIND** エディタコマンドを発行するたびに使用されます。検索を実行する前に、これらの値をウィンドウで変更できます。

次のパラメータを変更できます。

パラメータ	関数
FIND string	検索する文字列を指定します。
search from/to col	検索する列範囲の開始列番号と終了列番号を指定します。
search from/to label	検索対象である、マークされている行のブロックの開始ラベルと終了ラベルを指定します。

全般的なデフォルト設定の変更

【Main Menu】で【General】を選択すると、現在のデフォルトを示す画面が表示されます。必要に応じて、これらの設定を変更できます。次に、各設定について簡単に説明します。

オプション	関数
Prompt Window for Exit Function	Exit 機能 (関連セクションを参照) を使用してプログラムエディタを終了するときにプロンプトウィンドウを表示するかどうかを指定します。
Prompt Window for Cancel Function	キャンセル機能を使用してプログラムエディタを終了するときにプロンプトウィンドウを表示するかどうかを指定します。【Exit 機能】も参照してください。
Lines displayed before error line	ページ上部で、構文エラーを含む行の前に表示する行数を指定します。0 (ゼロ) を指定すると、構文エラーを含む行が編集エリアの最初の行に配置されます。

技術情報の表示

【Main Menu】で【Technical Info】を選択するか PF7 キーを押すと、現在編集中のオブジェクトおよびコンピューティング環境に関する情報を示す画面が表示されます。次の項目が表示されます。

項目	説明
User ID	現在のユーザーの ID。
Current library	現在ログオンしているライブラリの名前。
Current program	現在編集エリアに表示されているオブジェクトの名前。
Current type	現在編集エリアに表示されているオブジェクトのタイプ (プログラムなど)。
Object type	現在のオブジェクトの内部タイプコード。

項目	説明
Mode	プログラミングモード（「Mode」も参照）。
Operating system	インストールされているオペレーティングシステムの名前。
Steplib	現在のライブラリに連結されているライブラリの名前。

13 エディタバッファプール設定

- 複数のエディタセッション 92

場合によっては、バッファプールのサイズをデフォルト設定である 400 KB より大きくする必要があります。『パラメータリファレンス』ドキュメントのプロファイルパラメータ `EDTBPSIZE` の説明を参照してください。

複数のエディタセッション

プロファイルパラメータ `EDTRB`（『パラメータリファレンス』ドキュメントを参照）が設定されていると、プログラムエディタから新しい編集セッションを開くときに、古いセッションはバッファプールに保存されます。キャンセル機能またはExit 機能（「[Exit 機能](#)」も参照）を使用して新しいセッションを終了すると、古いセッションがバッファプールから解放されます。[NEXT](#) エディタコマンドを使用すると、アクティブなセッションから別のセッションにジャンプできます。

`EDTRB` パラメータが OFF になっていると、リングバッファは使用されず、新しいセッションが現在のセッションに置き換わります。エディタプロファイルパラメータ `AUTOSAVE`（「[エディタコマンド](#)」を参照）がオンになっていると、該当するセッションの内容が `FUSER` に保存されます。

[LOG](#) エディタコマンドを使用する場合は、バッファプールのサイズを 1 MB に増やすことをお勧めします。通常、変更内容はバッファプールエディタに保存されます。

14 ソースの保存およびカタログ

現在ソースワークエリアに表示されているソースコードを、ソースオブジェクトとして、およびカタログ化オブジェクトとしても、保存できます。保存先は Natural システムファイル内の Natural ライブラリです。

ソースを保存またはカタログするときに適用される命名規則については、『Natural の使用』ドキュメントの「オブジェクトの命名規則」を参照してください。

➤手順 14.1. ソースコードをソースオブジェクトとして保存するには

- エディタ画面で、『システムコマンド』ドキュメントで説明している構文規則に従ってシステムコマンド SAVE を入力します。

または:

エディタ画面で PF2 キーを押します。

➤手順 14.2. ソースコードをカタログ化オブジェクトとして保存するには

- エディタ画面で、『システムコマンド』ドキュメントで説明している構文規則に従ってシステムコマンド CATALOG を入力します。

➤手順 14.3. ソースコードをソースオブジェクトおよびカタログ化オブジェクトとして保存するには

- エディタ画面で、『システムコマンド』ドキュメントで説明している構文規則に従ってシステムコマンド STOW を入力します。

または:

エディタ画面で PF6 キーを押します。

-  **注意:** 新しいソースを保存またはカタログする場合、または現在のソースをコピーする場合は、オブジェクト名を指定する必要があります。指定しない場合は、該当するメッセージが表示されるか、または名前の指定を求めるプロンプトが表示されます。

II データエリアエディタ

15 データエリアエディタ

■ データエリアエディタの起動	98
■ 編集モードおよびコマンドモード	101
■ 編集エリア	102
■ 行コマンド	111
■ エディタコマンドおよびファンクションキー	114
■ データエリアの保存とカタログ	118
■ ヘルプ情報および選択オプション	119

Natural データエリアエディタを使用すると、データエリアを作成および編集できます。データエリアは、グローバルデータエリア (GDA) 、ローカルデータエリア (LDA) 、またはパラメータデータエリア (PDA) タイプの Natural オブジェクトです。データエリアの使用方法の詳細については、『プログラミングガイド』の「データエリア」を参照してください。

データエリアには、データ定義モジュール (DDM) のデータビューで参照する、ユーザー定義変数、定数、データベースフィールドなどのデータ要素定義が含まれています。データ要素定義は、1つ以上の Natural オブジェクトによって使用されます。データエリアからコピーコードを作成することもできます。DDM のデータビューは PDA では定義できないことに注意してください。

関連トピック：

- 『Unicode およびコードページのサポート』ドキュメントの「SPoD 環境のエディタ」

データエリアエディタの起動

データエリアエディタは、『システムコマンド』ドキュメントで説明しているように、Natural システムコマンド EDIT を使用して起動します。

手順 15.1. 新しいデータエリア用にデータエリアエディタを起動するには

- 作成するデータエリアのタイプ (GLOBAL、LOCAL、または PARAMETER) を指定して、EDIT コマンドを発行します。

例えば、次のようになります。

```
EDIT LOCAL
```

LDA (画面上部の情報行に LOCAL と表示) に対しては、次の説明に示す例のようなエディタ画面が、編集エリアが空の状態で表示されます。

手順 15.2. 既存のデータエリア用にデータエリアエディタを起動するには

- 現在の Natural 環境にソースオブジェクトとして格納されているデータエリアの名前を指定して、EDIT コマンドを発行します。

例えば、次のようになります。

```
EDIT LDA1
```

次に示す例のようなエディタ画面が表示されます。この画面には、ローカルデータエリア LDA1 のソースが表示されています。

```

Press <ESC> to enter command mode

Mem: LDA1      Lib: SAGTEST  Type: LOCAL      Size: 1662  Line: 0 of: 36
C T      Comment

*   *** Top of Data Area ***

V 1 EMPLOYEES_VIEW                           EMPLOYEES
2 PERSONNEL-ID                               A      8
G 2 FULL-NAME
3 FIRST-NAME                                A      20
3 MIDDLE-I                                  A      1
3 NAME                                     A      20
2 MIDDLE-NAME                               A      20
2 MAR-STAT                                  A      1
2 SEX                                       A      1
2 BIRTH                                     D
2 N@BIRTH                                   I      2
G 2 FULL-ADDRESS
M 3 ADDRESS-LINE                            A      20 (1:191)
3 CITY                                      A      20
3 ZIP                                       A      10
3 POST-CODE                                 A      10
3 COUNTRY                                   A      3
G 2 TELEPHONE
F 1 HELP          F 2 CHOICE      F 3 QUIT      F 4 SAVE      F 5 STOW      F 6 CHECK
F 7 READ          F 8 CLEAR       F 9 MEM TYPE F10 GEN      F11 FLD TYPE F12

```

エディタ画面には、上から下に、次のセクションが含まれています。

セクション	説明												
コマンド行	コマンド行を使用すると、エディタコマンドまたはシステムコマンドを発行したり、プログラムを実行したりできます（「 編集モードおよびコマンドモード 」を参照）。コマンド行は、コマンドモードに設定されている場合にのみ、表示されます。それ以外の場合は、コマンドモードに関するメッセージが代わりに表示されます。												
情報行	<p>エディタ画面に表示中のデータエリアに関する次の情報（左から右へ）が含まれます。</p> <table border="1"> <tr> <td>Mem:</td><td>データエリア名が表示されます。SAVE あるいは STOW システムコマンドを使用してソースオブジェクトとして保存していない新しいデータエリアの場合は、empty と表示されます。</td></tr> <tr> <td>Lib:</td><td>現在ログオンしているライブラリ。</td></tr> <tr> <td>Type:</td><td>データエリアのタイプ：LOCAL、GLOBAL または PARAMETER。このタイプは、エディタコマンド SET TYPE を使用して変更できます。</td></tr> <tr> <td>Size:</td><td>現在のソースのサイズ（文字数）。</td></tr> <tr> <td>Line:</td><td>現在の（強調表示されている）ソース行の番号。</td></tr> <tr> <td>of:</td><td>ソースワークエリアに含まれる行の総数。</td></tr> </table>	Mem:	データエリア名が表示されます。SAVE あるいは STOW システムコマンドを使用してソースオブジェクトとして保存していない新しいデータエリアの場合は、empty と表示されます。	Lib:	現在ログオンしているライブラリ。	Type:	データエリアのタイプ：LOCAL、GLOBAL または PARAMETER。このタイプは、エディタコマンド SET TYPE を使用して変更できます。	Size:	現在のソースのサイズ（文字数）。	Line:	現在の（強調表示されている）ソース行の番号。	of:	ソースワークエリアに含まれる行の総数。
Mem:	データエリア名が表示されます。SAVE あるいは STOW システムコマンドを使用してソースオブジェクトとして保存していない新しいデータエリアの場合は、empty と表示されます。												
Lib:	現在ログオンしているライブラリ。												
Type:	データエリアのタイプ：LOCAL、GLOBAL または PARAMETER。このタイプは、エディタコマンド SET TYPE を使用して変更できます。												
Size:	現在のソースのサイズ（文字数）。												
Line:	現在の（強調表示されている）ソース行の番号。												
of:	ソースワークエリアに含まれる行の総数。												
編集エリア	<p>データエリアのソースが表示されます。新しいデータエリアの場合は、空となります。「編集エリアの使用」を参照してください。</p> <p>ソースはスクロールすることのみができます。ただし、編集モードに設定されている場合は、ソースを変更できます。「編集モードおよびコマンドモード」を参照してください。</p>												
ファンクションキー行	エディタコマンドを実行する場合に使用できるファンクションキー（F キー）を表示します。「 エディタコマンドおよびファンクションキー 」を参照してください。												
メッセージ行	エラーが発生すると表示されます。この場合は、メッセージ行により、ファンクションキー行の 1 行目が該当するエラーメッセージで一時的に上書きされます。												

編集モードおよびコマンドモード

データエリアエディタは、編集モードおよびコマンドモードという2つの異なるモードで動作します。

編集モードでは、現在のデータエリアのソースを上下にスクロールしたり、ソース行の作成または変更に必要な行コマンドを使用できます。また、画面で使用可能なすべてのFキーを押すこともできます。

コマンドモードでは、エディタコマンドまたはシステムコマンドを入力または選択したり、プログラムを実行したりできます。コマンドモードでは、データエリアのソースを変更できません。

デフォルトでは、データエリアエディタは起動時に編集モードになっています。

➤手順 15.3. 編集モードとコマンドモードを交互に切り替えるには

- ESCキーを押します。

コマンドモードが設定されると、エディタコマンド行が表示されます。エディタコマンド行は、エディタ画面左上隅の【Command:】という表示によって示されます。

現在のモードは Natural セッションの間、保持されます。

➤手順 15.4. ダイレクトコマンドまたはプログラム名を入力するには

- コマンド行に次のいずれかを入力します。

- 任意の Natural システムコマンド。

例えば、次のようにになります。システムコマンド CHECK でソースコードの構文チェックを行ったり、SAVE でソースコードを保存したりします（「[データエリアの保存とカタログ](#)」も参照）。

オブジェクトソースの管理と使用に関連するその他のシステムコマンドについては、『[システムコマンド](#)』ドキュメントの「Natural オブジェクトによるアプリケーションの管理」を参照してください。

- 実行する Natural プログラムの名前。
- エディタコマンド。使用可能なすべてのエディタコマンドについては、「[エディタコマンドおよびファンクションキー](#)」を参照してください。

➤手順 15.5. メニューからダイレクトコマンドを選択するには

- コマンド行で「M」（メニュー）を入力します。

次のコマンドメニューが表示されます。

■ Commands

このメニューを選択すると（デフォルトで選択）、ウィンドウが開き、最も頻繁に使用するコマンドの一部が表示されます。次のコマンドのいずれかを選択して実行できます。

[Check](#)、[Clear](#)、[Fld-type](#)、[Gen](#)、[Read](#)、[Save](#)、[Type](#) または [Stow](#)。これらのコマンドについては、「[エディタコマンドおよびファンクションキー](#)」の、対応するダイレクトコマンドを参照してください。

■ Direct Command Line

このメニューを選択すると、すべてのコマンドメニューが閉じ、コマンド行が表示されます。

■ Quit

このメニューを選択すると、データエリアエディタは終了します。最後の [SAVE](#) または [STOW](#) コマンドの実行後、現在のデータエリアに対して加えられた変更は保存されません。

編集エリア

編集エリアは、空であるか、「[データエリアエディタの起動](#)」の例のように、コマンド [EDIT](#) または [READ](#) を使用してソースワークエリアに最後に読み込まれたソースコードが表示されます。

既存のオブジェクトのソースを読み込むと、ソースコード全体がソースワークエリアにロードされ、編集できるようになります。ただし、ソースのサイズによっては、ソース行の一部のみが編集エリアに表示される場合があります。この場合は、「[ソース行の表示および選択](#)」で説明しているように、表示または編集する行へ移動するために、ソースを下方にスクロールすることが必要となります。

- [編集エリアの列](#)
- [ソース行の表示および選択](#)
- [ソース行の作成および修正](#)

- [\[Definition\] または \[Redefine\] ウィンドウの入力フィールド](#)

編集エリアの列

エディタ画面の編集エリアは列で構成されており、変数定義またはフィールド定義のすべての属性が1行で管理されます。

次のセクションでは、編集エリアに表示される列について説明します。列の内容は、変数またはフィールドを作成または変更するとき、[\[Definition\]](#) および [\[Redefine\]](#) ウィンドウに入力された値によって異なります（「[Definition または Redefine ウィンドウの入力フィールド](#)」を参照）。また、列の内容は、カウンタフィールド（C* 変数）がフィールドから作成されたかどうかによっても異なります。

 **注意:** 現在のソース行に含まれる変数またはフィールドのタイプに応じて、列見出しを変更したり非表示にしたりできます。

列見出し	説明			
C	コマンド列。ソース行の作成または変更に必要な行コマンドの1つを入力できます。「 行コマンド 」も参照してください。			
T	変数またはフィールドのタイプ。 指定可能なタイプは次のとおりです。 <table border="1" data-bbox="546 1087 1052 1172"> <tr> <td>B</td> <td> [Block] GDA 内のデータブロック。 </td> </tr> </table>		B	[Block] GDA 内のデータブロック。
B	[Block] GDA 内のデータブロック。			
C	[Constant] または [Counter Variable] ユーザー定義変数（PDAでは使用不可）またはカウンタフィールド（C* 変数）。カウンタフィールドは、ビュー（DDM）内のマルチプルバリューフィールドまたはピリオディックグループに対して使用されます。			
F	[Data Field]	：充填文字。 再定義するフィールドまたは変数内に指定できる充填バイト。		
G	[Group]	ビュー（DDM）内のグループ。		
M	[Multiple Field]	ビュー（DDM）内のマルチプルバリューフィールド。		
O	[Object Handle]	オブジェクトのハンドル。		

列見出し	説明	
	P	【Periodic Group】 ビュー (DDM) 内のピリオディックグループ。
	R	【Redefined Field】 変数またはフィールドの再定義。
	V	PDA では使用できません。 【View】 DDM から作成されたビュー定義。
	空白	【Data Field】 ユーザー定義変数、ユーザー定義フィールド、またはグループ構造 (ビュー外で有効)。
	*	【Comment】 コメントフィールド。
L	変数またはフィールドのレベル番号 (1~99)。 階層構造外の変数およびビュー定義にはレベル 1 を指定する必要があります。 レベル番号は、データブロック定義には使用できません。	
Name of variable-field-type または Comment	変数またはフィールドの名前。【T】列の説明で示されているように、この列見出しが、現在選択されている変数またはフィールドのタイプに応じて変わります。 ビューには、【Name of variable-field-type】の他に、【Name of DDM】列も表示されます。 ブロックの場合は、Name of variable-field-typeに加えて、Name of Parent Block 列が表示されます。	
F	変数またはフィールドの Natural データフォーマット。	
Length	変数またはフィールドの長さ。	
Index/Comment	変数またはフィールドの、配列の添字および／またはコメント。	
M	変数またはフィールドに編集マスク、ヘッダー、および／または初期値が定義されている場合は、【M】(Miscellaneous) 列に X が表示されます。 ビュー、グループ、ピリオディックグループ、またはビュー内のマルチプレバリューフィールドに対しては、この列は表示されません。	

ソース行の表示および選択

編集モードでは、現在のデータエリアを上下にスクロールして、ソースワークエリアに現在含まれているソースのすべての行を表示することができます。また、変更を加える行を選択したり、新しい行を挿入する行を指定したりできます。

手順 15.6. ソースをスクロールし、行を選択するには

- UP ARROW キーおよび DOWN ARROW キーを押すと、それぞれ上下に 1 行ずつスクロールします。

または:

HOME キーまたは END キーを押すと、ソースワークエリア内に含まれる最初の（先頭）行または最後の（最下端）行に移動します。

または:

PAGE UP キーまたは PAGE DOWN キーを押すと、一度に 1 画面分のページだけ現在のソース行を上または下に移動します。これらのキーのいずれかにアクセスできない場合は、ファイル *SAGtermcap* を調べてエントリを確認します。

カーソルが置かれている行が現在の行となります。この行は選択された状態となり（強調表示）、変更を加えることができます。

ソース行の作成および修正

編集モードでは、「[行コマンド](#)」で説明している適切な行コマンドを使用して、現在のソース行で、変数定義およびフィールド定義を作成または変更できます。

次の例は、行コマンドを使用して、单一または複数の行をソースに追加したり、行を変更したりする手順を示しています。

手順 15.7. ビュー、再定義、またはカウンタフィールド以外の、新しい定義の行を追加するには

- 1 新しい定義を配置する行の上の行を選択し、その横の [C] 列に、次の行コマンドを入力します。

I

データエリアが空である場合は、[Top of Data Area] の横に「I」を入力します。

選択ウィンドウが表示されます。

- 2 次のいずれかを選択します。

Data Field： ユーザー定義変数またはデータベースフィールド。

- Block** : GDA 内のデータブロック（『プログラミングガイド』を参照）。
- Constant** : ユーザー定義定数（『プログラミングガイド』の「ユーザー定義定数」を参照）。
- Handle** : オブジェクトハンドル（『プログラミングガイド』の「クラスとオブジェクトの使用」を参照）。
- Structure** : 階層グループ構造（ビュー外で有効）。
- Comment** : コメントフィールド。

選択された変数またはフィールドのタイプに応じて、必要な属性定義を入力できる

[Definition] ウィンドウが表示されます。このウィンドウに表示される入力フィールドの詳細については、「**Definition** または **Redefine** ウィンドウの入力フィールド」を参照してください。

➤手順 15.8. ビュー定義の行を追加するには

- 1 ビューを配置する行の上の行を選択し、その横の [C] 列に、次の行コマンドを入力します。

＼

データエリアが空である場合は、[Top of Data Area] の横に「＼」を入力します。

View Definition ウィンドウが表示されます。

- 2 作成するビューの名前、および、そのビューの作成元となるDDMの名前を入力します。指定したDDMは、現在のシステムファイル内の現在のNaturalライブラリに含まれている必要があります。

[DDM] 選択ウィンドウが開き、指定したDDMに定義されているすべてのフィールドのリストが表示されます。

- 3 方向（カーソル）キーを使用してリストをスクロールし、ビューに含める各DDMフィールドの横に「x」を入力します。

または:

すべてのDDMフィールドをビューに含める場合は、**[DDM]** 選択ウィンドウで、いずれかのDDMフィールドの横に「A」を入力します。

フィールドの選択を解除するには、xを空白文字で置き換えます。

ピリオディックグループまたはマルチプルバリューフィールドを選択しなかった場合は、データエリアにすべてのフィールドがコピーされます。

ピリオディックグループまたはマルチプルバリューフィールドを選択した場合は、**[Occurrences Definition]** ウィンドウが表示されます。

- 4 [From:] フィールドにはビューで使用される1次元または2次元の配列の下限を入力し、[To:] フィールドには上限を入力します。

ピリオディックグループまたはマルチプルバリューフィールドは、ビューネームがレベル1である階層構造を持つビュー定義の下位フィールドとして、指定したオカレンス数だけデータエリアにコピーされます。

手順 15.9. 行の内容を変更するには

- 選択した行の横の [C] 列に、次の行コマンドを入力します。

E

選択した行の変数定義またはフィールド定義に応じて、[Definition] ウィンドウ（または再定義の場合には [Redefine] ウィンドウ）が表示されます。

[Definition] または [Redefine] ウィンドウの入力フィールド

[Definition] ウィンドウ、または再定義の場合には [Redefine] ウィンドウで、変数またはフィールドを作成および変更できます。このウィンドウに表示される入力フィールドは、選択された変数またはフィールドのタイプによって異なります。次の表に、使用可能なすべての入力フィールドのリストおよび説明を示します。

[Definition] または [Redefine] ウィンドウで入力した定義は、構文エラーがチェックされます。

このセクションで説明する変数属性またはフィールド属性の詳細については、『プログラミングガイド』の「ユーザー定義変数」および「フィールドの定義」、および、『ステートメント』ドキュメントの「DEFINE DATA」も参照してください。DEFINE DATA ステートメントで使用される値は、データエリアで使用される値に対応しています。

入力フィールド	説明
Level	<p>変数またはフィールドのレベル番号（1～99）。</p> <p>階層構造外の変数またはフィールドにはレベル1を指定する必要があります。ビュー定義にはレベル1を指定する必要があります。レベル番号は、データブロック定義には使用できません。</p>
Name	<p>次のいずれかの名前を指定します。</p> <ul style="list-style-type: none"> ■ ユーザー定義変数またはデータベースフィールド。 ■ ユーザー定義定数。 ■ 構造（グループ定義、ビュー外で有効）。 ■ オブジェクトのハンドル。 ■ ブロックおよび親ブロック（GDAのみ）。 ■ ビュー（DDM）。

入力フィールド	説明
	<ul style="list-style-type: none"> ■ マルチプルバリューフィールド（ビューで有効）。 ■ ピリオディックグループ（ビューで有効）。 ■ ビュー内のマルチプルバリューフィールドまたはピリオディックグループに対するカウンタフィールド（C* 変数）。 <p>有効な名前については、『Natural の使用』ドキュメントの「ユーザー定義変数の命名規則」を参照してください。</p> <p>ユーザー定義変数については、『ステートメント』ドキュメントの CONSTANT の説明を参照してください。</p>
Format	<p>変数またはフィールドの Natural データフォーマット。</p> <p>有効なフォーマットについては、『プログラミングガイド』の「ユーザー定義変数のフォーマットおよび長さ」および「特殊フォーマット」を参照してください。</p> <p>カウンタフィールド（C* 変数）に対して、Natural データフォーマット／長さ I2 または I4 を指定できます（デフォルト設定は、フォーマット／長さなしで N3 です）。</p>
Length	<p>変数またはフィールドの長さ。</p> <p>有効な長さについては、『プログラミングガイド』の「ユーザー定義変数のフォーマットおよび長さ」を参照してください。</p> <p>Natural データフォーマット C、D、T、L には長さは許可されません。 [Length] フィールドで [DYNAMIC] を指定することで、ダイナミック変数を定義できます。</p> <p>カウンタフィールド（C* 変数）に対して、Natural データフォーマット／長さ I2 または I4 を指定できます（デフォルト設定は、フォーマット／長さなしで N3 です）。</p>
Arraydefinition	<p>変数またはフィールドの配列定義。</p> <p>1 次元または 2 次元の配列の下限と上限を定義できます（「配列の定義例」を参照）。</p>
Edit Mask	<p>PDA では使用できません。</p> <p>I/O ステートメントで変数またはフィールドを表示するときに使用される、変数またはフィールドの編集マスク。</p> <p>カッコまたはアポストロフィを使用せずに、有効な値を入力します（「編集マスクの定義例」を参照）。有効な入力値の詳細については、『パラメータリファレンス』ドキュメントで説明している、対応するセッションパラメータ EM を参照してください。</p>
Header Definition	<p>PDA では使用できません。</p> <p>DISPLAY ステートメントの変数またはフィールドに対して表示されるヘッダー。</p> <p>アポストロフィまたはカッコを使用せずに、英数文字列を入力します（「ヘッダーの定義例」を参照）。ヘッダーの詳細については、『パラメータリファレンス』ドキュメントで説明している、対応するセッションパラメータ HD を参照してください。</p>
Initialization	PDA では使用できません。

入力フィールド	説明
	変数またはフィールドに割り当てられる初期値、または、変数またはフィールドに定義された配列オカレンス。次のいずれかの初期化モードを指定できます。
F	<p>フリーフォームモード。</p> <p>「F」を入力すると、【Free Form Initialization】 ウィンドウが表示され、DEFINE DATA ステートメントの共通 Natural 構文定義に従って、初期値を入力できます。フィールドオカレンス全体に同じ初期値を一度に割り当てることができます。</p> <p>「フリーモードでの初期値の割り当て例」も参照してください。</p> <p>初期値の定義の詳細については、『ステートメント』ドキュメントの「初期値の定義」および「配列用の初期値／定数値」を参照してください。</p>
S	<p>単独値モード。</p> <p>「S」を入力すると、【Single Value Initialization】 ウィンドウが表示され、初期化の節ではなく、単独の初期値を入力できます。</p> <p>値は変数またはフィールドのタイプに基づいて入力します。必要な変数値またはフィールド値のみを入力します。値以外に必要となる指定（例：英数字の変数やフィールドのアポストロフィや、16進を示すH、データを示すD、時刻を示すTなどの値の接頭辞）は自動生成されます。例えば、バイナリ変数（B2）の初期値として H'F1F2' を指定するには、「F1F2」と入力します。データエディタで生成される内容：INIT <H'F1F2'></p> <p>変数またはフィールドが配列の場合は、配列のすべての要素がリストされます。各要素に値を入力できます（オプション）。</p>
	<p>注意: 初期化タイプを変更すると、事前に入力した初期化情報がすべて削除されます。</p> <p>『プログラミングガイド』の「初期値（およびRESET ステートメント）」および「配列の初期値」セクションも参照してください。</p>
Value Clause	<p>PDA にのみ適用されます。</p> <p>CALLNAT ステートメントにパラメータとして指定された変数値またはフィールド値をプログラムから呼び出されたオブジェクト（サブプログラムなど）に渡す方法を指定します。</p> <p>有効な入力値は次のとおりです。</p>

入力フィールド	説明									
	空白	パラメータ指定の参照による呼び出しを示します（デフォルト）。								
	V	パラメータ指定の値による呼び出しを示します。								
	R	パラメータ指定の値による呼び出しと結果を示します。								
詳細については、「パラメータデータ定義」のDEFINE DATAステートメントについて説明している、対応するオプションBY VALUEとBY VALUE RESULT、および『ステートメント』ドキュメントのCALLNATステートメントについて説明している <i>operand2</i> を参照してください。										
Optional Param	<p>PDAにのみ適用されます。</p> <p>〔Value Clause〕（上記を参照）での設定に従って、変数値またはフィールド値をパラメータとして渡すかどうかを指定します。</p> <table border="1"> <tr> <td></td><td></td></tr> <tr> <td>N</td><td>パラメータを渡す必要があります（デフォルト）。</td></tr> <tr> <td>Y</td><td>パラメータを任意で渡すことができます。</td></tr> <tr> <td></td><td></td></tr> </table>				N	パラメータを渡す必要があります（デフォルト）。	Y	パラメータを任意で渡すことができます。		
N	パラメータを渡す必要があります（デフォルト）。									
Y	パラメータを任意で渡すことができます。									
詳細については、「パラメータデータ定義」のDEFINE DATAステートメントについて説明している、対応するオプションOPTIONALおよび『ステートメント』ドキュメントのCALLNATステートメントについて説明している <i>operand2</i> を参照してください。										
Indexdefinition	<p>構造（グループ）タイプの変数の配列定義。</p> <p>この入力フィールドは、配列の上限および下限の定義、変数の初期値の指定、または変数の編集マスクの指定に使用します。</p> <p>「配列の定義例」も参照してください。</p>									
Comment	コメントのテキスト。									

配列の定義例

```
(2,2)      /* 2 dimensions, 2 occurrences
(2,2,2)    /* 3 dimensions, 2 occurrences
(1:10,2)
(-1:3,2)
```

編集マスクの定義例

```
999.99
XXX...XX
MM.DD.YY
```

ヘッダーの定義例

```
HEADER TEXT
```

フリーモードでの初期値の割り当て例

```
INIT<3>
INIT<'ABC'>
INIT<H'F1F2'> /* binary variable (B2)
CONST<12>
INIT ALL<'ABC'>
```

行コマンド

編集モードが設定されている場合は、ソース行の [C] 列に行コマンドを入力できます。入力されたコマンドは、列に表示されない場合があります。

F2 キーを押すと、すべての使用可能な行コマンドの選択リストが表示されます。

次の表では、データエリアエディタで使用できるすべての行コマンドについて説明します。表中で使用されている「編集ブロック」という表現は、マークされているソース行のブロックを表します。

Command	関数
C (コピー)	1行以上の行をコピーします。編集ブロックで「C」を入力すると、このブロック内のすべての行がコピーされます。 「 編集ブロックをコピーまたは切り取って貼り付けるには 」も参照してください。
D (削除)	1行以上の行を削除します。 編集ブロックに「D」を入力すると、このブロック内のすべての行が削除されます。詳細については、行コマンド C (コピー) および P (貼り付け) を参照してください。 「 編集ブロックをコピーまたは切り取って貼り付けるには 」も参照してください。
E	選択した変数またはフィールドのタイプに応じて、[Definition] または [Redefine] ウィンドウを開きます (Definition または Redefine ウィンドウの入)

Command	関数								
(編集)	<p>カフィールドも参照)。このウィンドウでは、その行に含まれる変数またはフィールドの属性を変更できます。</p> <p>ビュー定義内のフィールドについては、以下のみを変更できます。 Level (ピリオディックグループおよびマルチプルバリューフィールドを除く)、Edit Mask、Header Definition、Indexdefinition および Comment。</p>								
H (ブロックのマーク解除)	現在の編集ブロックからマークを解除します。このコマンドは、編集ブロック内から発行する必要があります。								
I (行の挿入)	新しい定義用の行を挿入します (「 ビュー、再定義、またはカウンタフィールド以外の、新しい定義の行を追加するには 」を参照)。								
M (コメントの追加)	選択した行またはマークした編集ブロックにコメントマークを追加します。コンパイラチェックでは、これらの行は無視されます。このコマンドは、テストのために使用すると便利です。								
N (コメントの削除)	選択した行またはマークした編集ブロックからコメントマークを削除します。								
P (貼り付け)	<p>現在の行の後のデータエリアに 1 行以上の行を貼り付けます。</p> <p>「編集ブロックをコピーまたは切り取って貼り付けるには」も参照してください。</p>								
R (再定義)	<p>行に含まれる変数またはフィールドを、单一変数または変数グループとして再定義します。</p> <p>次のいずれかの再定義オプションを選択できるウィンドウが表示されます。</p> <table border="1"> <tr> <td>Data Field</td><td>変数またはフィールドを单一のユーザー定義変数として再定義します。</td></tr> <tr> <td>Filler</td><td>充填オプション (nX) を使用して、変数またはフィールドを再定義します。充填オプションを使用して、再定義するフィールドまたは変数に n 充填バイトを指定できます。n には、最大 10 行 (1 GB) の値を指定できます。末尾の充填バイトは任意指定です。</td></tr> <tr> <td>Structure</td><td>変数またはフィールドを構造 (グループ) として再定義します。</td></tr> <tr> <td>Comment</td><td>変数またはフィールドをコメントとして再定義します。</td></tr> </table>	Data Field	変数またはフィールドを单一のユーザー定義変数として再定義します。	Filler	充填オプション (nX) を使用して、変数またはフィールドを再定義します。充填オプションを使用して、再定義するフィールドまたは変数に n 充填バイトを指定できます。 n には、最大 10 行 (1 GB) の値を指定できます。末尾の充填バイトは任意指定です。	Structure	変数またはフィールドを構造 (グループ) として再定義します。	Comment	変数またはフィールドをコメントとして再定義します。
Data Field	変数またはフィールドを单一のユーザー定義変数として再定義します。								
Filler	充填オプション (nX) を使用して、変数またはフィールドを再定義します。充填オプションを使用して、再定義するフィールドまたは変数に n 充填バイトを指定できます。 n には、最大 10 行 (1 GB) の値を指定できます。末尾の充填バイトは任意指定です。								
Structure	変数またはフィールドを構造 (グループ) として再定義します。								
Comment	変数またはフィールドをコメントとして再定義します。								

Command	関数
	<p>選択した再定義オプションに応じて、[Redefine] または [Definition] ウィンドウが表示され、必要な定義を入力できます（「Definition または Redefine ウィンドウの入力フィールド」も参照）。</p> <p>データエリアエディタでは、再定義に利用できる空きバイト数を追跡します。空きバイトがない場合は、再定義機能が終了します。</p> <p>使用する再定義オプションに応じて、データエリアエディタによって、再定義される変数またはフィールドの上に、再定義に必要となる行が自動的に追加されます。</p>
S (表示)	<p>[Definition] または [Redefine] ウィンドウが開き、行に含まれている変数またはフィールドのすべての定義が表示されます（「Definition または Redefine ウィンドウの入力フィールド」も参照）。</p> <p>変更できません。</p> <p>また、初期化を指定した場合は、初期化情報を含む別のウィンドウが表示されます。情報をスクロールすることができます。</p>
V (ビューディレクティブの挿入)	ビューディレクティブを挿入します（「 ビューディレクティブの行を追加するには 」を参照）。
X (ブロックの先頭のマーク)	編集ブロックの先頭をマークします。
Y (ブロックの最後のマーク)	編集ブロックの最後をマークします。
Z (グループのマーク)	<p>グループ構造全体を編集ブロックとしてマークします。</p> <p>編集ブロックは現在の行から開始し、現在の行よりレベルの小さいすべての連続する行が含まれます。例えば、ビューディレクティブに「Z」を入力すると、ビューディレクティブ全体が編集ブロックとしてマークされます。</p>
*	<p>PDA では使用できません。</p> <p>行に含まれるマルチプルバリューフィールドまたはピリオディックグループからカウンタフィールド (C* 変数) を生成します。カウンタフィールドは、コマンドを入力した行の上の行に配置されます。</p> <p>カウンタフィールドは、マルチプルバリューフィールドのオカレンス数またはピリオディックグループを取得するために使用します。『プログラミングガイド』の「データベース配列の内部カウントの参照 (C* 表記)」も参照してください。</p>

次に、行コマンドを使用して、ソース内の行ブロックまたは單一行を移動またはコピーする手順を示します。

▷手順 15.10. 編集ブロックをコピーまたは切り取って貼り付けるには

- 1 切り取るかまたはコピーする行ブロック（または單一行）の最初の行の横に、次の行コマンドを入力します。

X

行がマークされます。

- 2 行ブロックの最後の行まで上または下にスクロールして、次の行コマンドを入力します。

Y

單一行の場合は、行コマンド「X」を入力したのと同じ行に、行コマンド「Y」を入力します。

ブロックのすべての行がマーク（強調表示）され、編集ブロックが示されます。

- 3 編集ブロック内で、次のいずれかの行コマンドを入力します。

C

行をコピーします。

D

行を削除し（切り取り）ます。

- 4 行を貼り付ける位置の上の行にカーソルを置き、次の行コマンドを入力します。

P

ソースに行が貼り付けられます。

- 5 編集ブロックのマークを解除するには、編集ブロック内で次の行コマンドを入力します。

H

編集ブロックのマークは解除され、現在の行のみが強調表示されます。

エディタコマンドおよびファンクションキー

このセクションでは、コマンドモードでコマンド行に入力したり、 [Commands] メニューから選択したりできる、エディタコマンドおよびエディタ固有のシステムコマンドについて説明します。

また、編集モードで使用できる、代替 F キーについても説明します。コマンドモードでは、 F1 キーのみを使用できます。

エディタコマンドで使用されている構文記号の説明については、『システムコマンド』ドキュメントの「システムコマンド構文」を参照してください。

Command	F キー	関数												
n/a	F1	<p>エディタ機能のヘルプ情報を表示します。「ヘルプ情報および選択オプション」も参照してください。</p> <p>注意: 『システムコマンド』ドキュメントのシステムコマンド HELP で説明しているように、「HELP」と入力すると、エラーメッセージのヘルプ機能を呼び出します。</p>												
n/a	F2	選択ウィンドウが表示され、カーソルが置かれている入力フィールドに対して有効な入力値が表示されます。												
CATALOG [object-name]	n/a	<p>現在のデータエリア定義をチェックしてカタログするシステムコマンド CATALOG を実行します。</p> <p>新しいデータエリア定義をカタログする場合、または現在のデータエリアをコピーする場合は、コマンドにオブジェクト名を指定する必要があります。指定しなかった場合、該当するメッセージが表示されます。</p> <p>「データエリアの保存とカタログ」も参照してください。</p>												
CHECK	F6	現在のデータエリア定義の構文をチェックするシステムコマンド CHECK を実行します。構文チェックを処理中であるというウィンドウが表示されます。構文エラーが検出された場合は、エラーを含む行が現在の行になり、メッセージ行にエラーが表示されます。エラーが見つからなければ、対応するメッセージが表示されます。												
CLEAR	F8	ソースワークエリアをクリアするシステムコマンド CLEAR を実行します。ソースワークエリアに現在含まれているデータエリアに対する変更内容のうち、保存済みでないものは失われます。												
FLD TYPE	F11	<p>FLD TYPEは [Commands] メニューからのみ使用できます（コマンド行には入力できません）。</p> <p>ウィンドウが表示され、そのウィンドウで現在の変数またはコメントのタイプを変更できます。次のタイプのいずれかを選択できます。</p> <table border="1" data-bbox="620 1478 1468 1752"> <tr><td>D</td><td>データフィールド</td></tr> <tr><td>B</td><td>ブロック</td></tr> <tr><td>C</td><td>定数</td></tr> <tr><td>H</td><td>ハンドル</td></tr> <tr><td>S</td><td>Structure</td></tr> <tr><td>*</td><td>コメント</td></tr> </table> <p>ビュー定義内のフィールドタイプは変更できません。</p>	D	データフィールド	B	ブロック	C	定数	H	ハンドル	S	Structure	*	コメント
D	データフィールド													
B	ブロック													
C	定数													
H	ハンドル													
S	Structure													
*	コメント													

Command	Fキー	関数
		注意: タイプを変更すると、変数またはコメントの属性定義の一部が失われる場合があります。
GENERATE <i>object-name</i>	F10	<p>コピーコードタイプの Natural オブジェクトを現在のデータエリア定義から生成します。したがって、ソースワークエリアの内容は上書きされます。最後の SAVE または STOW 以降にデータエリアに変更を加えた場合、その変更内容は失われます。</p> <p>コピーコードオブジェクトは、コマンドを使用して指定された <i>object-name</i> で、ソースオブジェクトとして、現在のシステムファイル内の指定された Natural ライブラリに保存されます。</p>
QUIT または . (ピリオド)	F3	<p>データエリアエディタを終了します。</p> <p>最後の SAVE または STOW 以降に変更を加えた場合、その変更内容はすべて失われます。</p>
READ <i>object-name</i>	F7	既存のデータエリア定義をソースワークエリアに読み込むシステムコマンド READ を実行します。データエリアは、指定した <i>object-name</i> で、ソースオブジェクトとして、現在のシステムファイル内の現在の Natural ライブラリに保存する必要があります。
SAVE [<i>object-name</i>]	F4	<p>現在のデータエリア定義を保存するシステムコマンド SAVE を実行します。</p> <p>新しいデータエリア定義を保存する場合、または現在のデータエリアをコピーする場合は、オブジェクト名を指定する必要があります。指定しなかった場合、該当するメッセージまたは入力ウィンドウが表示されます。F4 キーを押すと、常に、データエリアおよびライブラリの名前の指定を求めるプロンプトが表示されます。現在のオブジェクトおよびライブラリの名前が正しい場合、入力は必要ありません。</p> <p>「データエリアの保存とカタログ」も参照してください。</p>
SCAN <i>scan-value</i>	n/a	<p>前に SET SCAN コマンドが実行されたかどうかに応じて、データエリアで、エディタ画面の [Name of variable-field-type] 列（デフォルト）および／または [M] 列を文字列 (<i>scan-value</i>) でスキャンします。</p> <p><i>scan-value</i> が検出された行は強調表示されます。<i>scan-value</i> の最初のインスタンスが強調表示されている場合、ENTER キーを押すと、次のインスタンスに移動できます。</p> <p>ソースワークエリアの最初から最後の行までスキャンされ、最後の行に到達すると先頭に戻ります。</p>

Command	F キー	関数						
		<p>注意: SCAN コマンドは、指定した <i>scan-value</i> に対して完全一致検索を実行します。DBCS (ダブルバイト文字セット) 文字を検索するときは、このことを考慮する必要があります。</p>						
SET ABS [ON OFF]	n/a	<p>絶対モードと非絶対モードのどちらで SCAN コマンドを実行するかを指定します。</p> <table border="1"> <tr> <td>ON</td><td>SCAN コマンドは絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要はありません。</td></tr> <tr> <td>OFF</td><td>SCAN コマンドは非絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要があります。</td></tr> </table> <p>デフォルトは OFF です。</p>	ON	SCAN コマンドは絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要はありません。	OFF	SCAN コマンドは非絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要があります。		
ON	SCAN コマンドは絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要はありません。							
OFF	SCAN コマンドは非絶対モードで動作します。つまり、スキャンする値を空白または特殊文字で区切る必要があります。							
SET SCAN [COMMENT NAME LEVEL]	n/a	<p>SCAN コマンドが <i>scan-value</i> を検索する列を指定します。</p> <table border="1"> <tr> <td>COMMENT</td><td>【Index/Comment】列がスキャンされます。</td></tr> <tr> <td>NAME</td><td>【Name of variable-field-type】 (または [Comment]) 列がスキャンされます。</td></tr> <tr> <td>LEVEL</td><td>階層的なグループ構造内がスキャンされます。</td></tr> </table> <p>デフォルトは NAME です。</p>	COMMENT	【Index/Comment】列がスキャンされます。	NAME	【Name of variable-field-type】 (または [Comment]) 列がスキャンされます。	LEVEL	階層的なグループ構造内がスキャンされます。
COMMENT	【Index/Comment】列がスキャンされます。							
NAME	【Name of variable-field-type】 (または [Comment]) 列がスキャンされます。							
LEVEL	階層的なグループ構造内がスキャンされます。							
STOW [object-name]	F5	<p>現在のデータエリア定義を保存してカタログするシステムコマンド STOW を実行します。</p> <p>新しいデータエリア定義を STOW する場合、または現在のデータエリアをコピーする場合には、オブジェクト名を指定する必要があります。</p> <p>「データエリアの保存とカタログ」も参照してください。</p>						
SET TYPE G L A	F9	<p>現在のデータエリアのタイプを変更します。</p> <table border="1"> <tr> <td>G</td><td>グローバルデータエリア</td></tr> <tr> <td>L</td><td>ローカルデータエリア</td></tr> <tr> <td>A</td><td>パラメータデータエリア</td></tr> </table>	G	グローバルデータエリア	L	ローカルデータエリア	A	パラメータデータエリア
G	グローバルデータエリア							
L	ローカルデータエリア							
A	パラメータデータエリア							

データエリアの保存とカタログ

Natural プログラムまたは別のオブジェクトでデータエリアを使用できるようにするには、現在の Natural 環境で、ソースオブジェクトおよび／またはカタログ化オブジェクトとして、そのデータエリアを保存する必要があります。

次のセクションでは、現在のデータエリア定義を保存およびカタログする場合に使用するコマンドについて説明します。

新しいデータエリア定義を保存またはカタログする場合、または現在のデータエリアをコピーする場合は、オブジェクト名を指定する必要があります。指定しない場合は、該当するメッセージが表示されるか、または名前の指定を求めるプロンプトが表示されます。データエリアを保存またはカタログするときに適用される命名規則については、『Natural の使用』ドキュメントの「オブジェクトの命名規則」を参照してください。

▶手順 15.11. データエリアをソースオブジェクトとして保存するには

- 『システムコマンド』ドキュメントで説明している構文規則に従って、システムコマンド **SAVE** を入力します。

または:

F4 キーを押します。

または:

[Commands] メニューで、[Save] を選択します。

データエリアのソースは、ソースオブジェクトとして、現在のシステムファイルの指定された Natural ライブラリに保存されます。データエリアの構文エラーはチェックされません。

▶手順 15.12. データエリアをカタログ化オブジェクトとして保存するには

- 『システムコマンド』ドキュメントで説明している構文規則に従って、システムコマンド **CATALOG** を入力します。

データエリアのソースの構文エラーはチェックされます。エラーが見つからなかった場合、ソースは、現在のシステムファイルで指定された Natural ライブラリに、カタログ化オブジェクトとして保存されます。

➤手順 15.13. データエリアをソースオブジェクトおよびカタログ化オブジェクトとして保存するには

- 『システムコマンド』ドキュメントで説明している構文規則に従って、システムコマンド STOW を入力します。

または:

F5 キーを押します。

または:

【Commands】メニューで、【Stow】を選択します。

データエリアのソースの構文エラーはチェックされます。エラーが見つからなかった場合、ソースは、現在のシステムファイルで指定された Natural ライブラリに、ソースオブジェクトおよびカタログ化オブジェクトとして保存されます。

ヘルプ情報および選択オプション

ヘルプシステムを使用すると、エディタ機能および入力フィールドに関するヘルプ情報を参照できます。編集モードの場合は、ヘルプシステムを使用して有効な入力値または行コマンドをリストから選択できます。

➤手順 15.14. ヘルプ情報を表示するには

- 1 編集モードの場合、使用できるすべてのエディタ機能の概要を参照するには、【C】列にカーソルを置いて F1 キーを押します。

または:

フィールドの使用方法の詳細を参照するには、そのフィールドにカーソルを置いて F1 キーを押します。

ウィンドウが開き、ヘルプ情報が表示されます。

方向（カーソル）キーを使用すると、ウィンドウ内を上下にスクロールできます。

- 2 ヘルプウィンドウを閉じるには、ENTER キーまたは ESC キーを押します。

➤手順 15.15. 有効な値または適切な行コマンドを選択するには

- フィールドに対して有効な入力値を選択するか、または行コマンドを実行する場合は、該当するフィールドにカーソルを置き、F2 キーを押します。

該当する場合は、選択ウィンドウが表示され、そのウィンドウで入力値または行コマンドを選択できます。

III マップエディタ

16 マップエディタ

■ マップの作成	124
■ マップエディタの起動	125
■ Map Editor メニュー	127
■ テキスト定数の作成	130
■ ユーザー定義変数の作成	131
■ ユーザー定義変数の変更 - フィールド編集	132
■ データ定義の選択	146
■ パラメータまたはローカルデータ定義用フィールドの定義	149
■ マッププロファイル	151
■ 事後割り当て	155
■ フィールドに基づいた処理	155

Naturalマップエディタを使用して、タイプマップのNaturalオブジェクトを作成します。マップとは、INPUT USING MAPステートメント（入力マップ用）またはWRITE USING MAPステートメント（出力マップ用）のいずれかを使用して、プログラムなどのNaturalオブジェクトで参照できる画面レイアウトです。

マップには、テキストフィールドおよびデータフィールドが含まれます。テキストフィールドはリテラルの文字列であり、データフィールドは変数です。データフィールドには、ユーザー定義変数またはNaturalシステム変数のいずれかを使用できます。

マップを作成すると、ソースオブジェクトおよびカタログ化オブジェクトとして、Naturalシステムファイルのライブラリに保存できます。

 **注意:** Windows環境でNaturalスタジオを使用する場合、マップエディタはUnicodeフォーマットおよびUnicode文字列のフィールドをサポートしています。ただし、ローカルのUNIX、OpenVMS、またはメインフレーム環境で、Unicodeマップのソースをマップエディタの編集エリアに読み込むと、すべてのUnicode文字列がソースから削除されます。

マップの作成

マップの作成には、主に次の4つの手順が関係しています。

1. マッププロファイルの定義（つまり、使用するフィールドデリミタ、フォーマット設定、コンテキスト設定、および充填文字）。必要な項目を選択するメニューが提供されます。
2. マップの定義。マップは、2つの異なる方法で定義できます。

■ 最初にプロトタイプマップを定義して、次にマップを参照するオブジェクトで対応するデータ定義を作成し、その後でマップをアプリケーションに統合します。

フィールドは、マップ編集画面で直接定義できます。各フィールドにはデフォルト名が割り当てられます。続けて、対応するデータ定義が各オブジェクトで作成されたら、これらのデータ定義をマップフィールドに割り当てることができます（[事後割り当て](#)）。

■ 既存のデータ定義を使用してマップを定義します。

データ定義がマップを参照するオブジェクトにすでに存在する場合、マップフィールドは、このオブジェクトに含まれているデータ定義を使用することによって作成できます。この場合、データ定義のすべての特性はマップにコピーされます。

3. マップで使用するフィールドの定義。マップフィールドは、マップ編集画面でフィールド定義を直接入力するか、または別のNaturalオブジェクトからデータ定義を選択して作成できます（「[データ定義の選択](#)」を参照）。
4. マップ定義の保存またはカタログ、あるいはその両方。前の手順で説明したようにマップを定義した後は、ソースオブジェクトまたはカタログ化オブジェクトとして、現在のライブラリおよびNaturalシステムファイルに保存できます。ソースオブジェクトとして保存した後

は、続くマップ編集セッション中にマップを読み込んだり変更したりできます。カタログ化オブジェクトとしてマップを保存すると、そのマップを Natural プログラムから呼び出すことができます。

マップエディタの起動

次のセクションでは、Natural マップエディタを起動して新規マップの作成または既存マップの編集を行う方法について説明します。

- 新規マップの作成
- 既存マップの編集

新規マップの作成

Natural のメインメニューで **[Direct]** を選択して ENTER キーを押し、**[Direct Command]** ウィンドウを呼び出します。



注意: また、Natural のメインメニューでライブラリを選択したときに表示されるライブラリ選択リストで、**[<DIRECT COMMAND>]** を選択して **[Direct Command]** ウィンドウを呼び出すこともできます。

Direct Command window に次のコマンドを入力します。

EDIT MAP

または、短縮形では次のようにになります。

E M

ENTER キーを押します。 **[MAP EDITOR]** メニューが表示されます。



既存マップの編集

Naturalのメインメニューで **[Direct]** を選択して ENTER キーを押し、 **[Direct Command]** ウィンドウを呼び出します。

このウィンドウで、次のコマンドを入力します。

```
EDIT map-name
```

または、短縮形では次のようになります。

```
E map-name
```

編集するマップの名前を覚えていない場合は、Naturalのメインメニューで **[Library]** を選択し、ENTER キーを押します。使用可能なすべてのライブラリのリストが表示されます。

カーソルを使用して必要なライブラリをリストから選択し、ENTER キーを押します。このライブラリにあるすべてのオブジェクトのリストが表示されます。

 **注意:** ファイル指定は列 **[Type]** の下にリストされます。

カーソルキーを使用して、目的のマップがリストに表示されるまでリストをスクロールします。次に、オブジェクト名の前の列でマップを **E** (編集) でマークし、ENTER キーを押します。マップエディタが起動して、選択したマップが表示されます。

 **注意:** **F2** キーを押すと、この特定のオブジェクトに有効な機能 (**C=Check**、**D=Read**、**E>Edit** など) がリストされたウィンドウが項目 **[Ops. Map]** に表示されます。

呼び出されたマップに関係なく、画面の最下部にプロンプトが表示され、カーソルキーを使用してフィールドを選択し、ENTER キーを押すよう求められます。選択したフィールドは強調表示されます。

 **注意:** 最下行の右端に、強調表示されたフィールドの列番号および行番号が表示されます。

ESC キーを押すと、 **[MAP EDITOR]** メニューが呼び出されます。

 **注意:** 既存マップのプログラミングモード (ストラクチャード/レポーティング) を変更するには、 **[Direct Command]** ウィンドウに次のコマンドを入力します。

```
READ map-name
```

```
GLOBALS SM=ON/OFF
```

```
SAVE map-name
```

Map Editor メニュー

マップエディタのメインメニューは上記の **[MAPEDITOR]** メニューです。次のセクションでは、メニュー項目について説明します。

- [Create](#)
- [変更](#)
- [Erase](#)
- [Drag](#)
- [Info ON/OFF](#)
- [行](#)
- [オプションマップ](#)
- [Quit](#)

Create

ENTER キーを押すと、次の項目を含むリストが表示されます。

- [A - パラメータデータエリア](#)
- [G - グローバルデータエリア](#)
- [H - Helproutine](#)
- [L - ローカルデータエリア](#)
- [M - マップ](#)
- [N - サブプログラム](#)
- [P - プログラム](#)
- [S - サブルーチン](#)
- [T - テキスト定数](#)
- [U - ユーザ一定義](#)
- [V - ビュー \(DDM\) 定義](#)
- [1 - パラメータ定義](#)
- [2 - ローカル定義](#)

これらの項目を使用して、このクラスのオブジェクトのフィールド／変数をオブジェクトから選択します。また、「C」およびオブジェクトクラスの省略形（キー依存）を入力して、オブジェクトクラスを起動することもできます。詳細については、「[データ定義の選択](#)」セクションを参照してください。

変更

〔Modify〕で選択したフィールドを変更できます。選択したフィールドとは現在のフィールドのことです。このフィールドは強調表示されます。

フィールド属性（拡張フィールド編集）を表示するウィンドウが表示されます。このウィンドウで、これらの属性の内容を変更できます。

Erase

〔Erase〕で現在のフィールドを削除できます。次のメッセージが表示されます。

Delete field (Y/N)?

すべての応答はキーに依存するため、注意してください。

現在のフィールドが配列フィールドの場合は、（フィールドだけでなく）配列全体が削除されます。

Drag

〔Drag〕で画面上で未使用の位置に現在のフィールドを移動できます。選択したフィールドは、カーソルを使用して制限なく移動できます。

フィールドを配置して ENTER キーを押すと、このフィールド位置が有効になります。

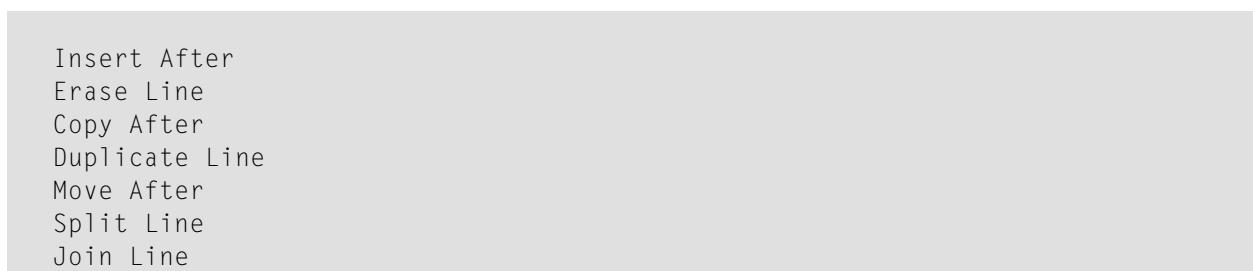
Info ON/OFF

〔Info ON/OFF〕は、〔Extended Field Editing〕 ウィンドウの表示の ON と OFF の切り替えに使用されます（トグルスイッチ）。〔OFF〕はデフォルト値です。

ON と OFF の切り替えには、ENTER キーまたは I (キー依存) を使用します。

行

〔Lines〕を選択すると、次の行固有機能を選択できる選択リストが表示されます。



これらの機能を選択すると、（单一のフィールドではなく）マップの行全体に操作を実行できます。すべての操作は現在の行に対して実行されます。

 **注意:** これらの機能の動作はその機能名が示しています。各項目を選択すると、画面の最下部にプロンプトが表示されます。

オプションマップ

[Ops. Map] (マップ操作) を選択すると、次の選択リストが表示されます。

```
C Check Map
E Edit Map
V Reverse Map
K Key Rules
L List Map
P Prof. Map
R Read Map
S Save Map
T Test Map
W Stow Map
```

リストには、次の項目が含まれています。

項目	説明
Check Map	構文をチェックし、ソースコードを生成します。
Edit Map	マップ編集画面を起動し、既存のマップ定義を変更します。マップエディタによって、編集セッションが新たに開始されます。
Reverse Map	<p>この機能は、デフォルトコードページが、右から左 (RTL) に書く言語（ヘブライ語など）をサポートしている場合にのみ適用されます。</p> <p>画面のマップの方向に関して、左から右（デフォルト）を右から左に、またはその逆に反転します。これは視覚的な変更にすぎません。マップソースで定義された物理的なフィールドの位置は維持されます。</p> <p>画面の方向は、[Reverse Map] を再度選択するまで反転したままとなります。</p>
Key Rules	ファンクションキーに関連した処理ルールの編集を起動します。
List Map	ソースコードを生成し、そのソースコードを表示します。
Prof. Map	マッププロファイルウィンドウを起動します。このウィンドウについては、「 マッププロファイル 」で説明します。
Read Map	マップ編集画面を起動し、既存のマップ定義を読み込みます。
Save Map	ソースコード生成チェックを実行してからマップを保存します。マップ定義はソース形式で Natural ライブラリに保存されます。
Test Map	現在のマップ定義をテストして、正しく実行できることを確認します。このテストには、すべての処理ルールおよびヘルプ機能のテストが含まれています。

項目	説明
Stow Map	ソースコード生成チェックを実行し、マップ定義を保存およびカタログします。マップ定義はカタログされ、ソース形式で現在の Natural ライブラリに保存されます。

Quit

〔Quit〕 でマップエディタセッションを終了します。

マップを編集して保存していない場合に、 〔Quit〕 を選択して ENTER キーを押すと、次のプロンプトが表示されます。

Modifications have not been saved, quit anyway Y/N ?

 **注意:** 応答はキーに依存するため、注意してください。

エディタセッションを終了すると、Natural のメインメニューが表示されます。

テキスト定数の作成

〔Create〕 リストから 〔Text Constant〕 を選択し、ENTER キーを押します。

呼び出されたマップタイプに応じた画面が表示されます。

画面の最下部に、カーソルを配置してテキストを入力することを求めるプロンプトが常に表示されます。

空フィールドの先頭にカーソルを配置します。

 **注意:** 既存のフィールドは上書きできません。

テキストを入力します。最初の文字を入力すると、その行が強調表示されます。強調表示は、テキストを入力するスペースの空きが最大であることを示しています。ENTER キーを押すまで文字を入力または削除できます。

 **注意:** ESC キーを押すと、テキスト入力がキャンセルされます。

入力したテキストに使用する属性および色を選択するには、PF2 キーを押します。UP-ARROW および DOWN-ARROW キーを使用してスクロールし、使用可能な属性／色の 1 つを選択するか、または対応する省略形 (B=Blinking または RE=Red など) を使用して ENTER キーを押します。

LEFT-ARROW および RIGHT-ARROW キーを使用すると、属性定義とカラーリ定義を切り替えることができます。

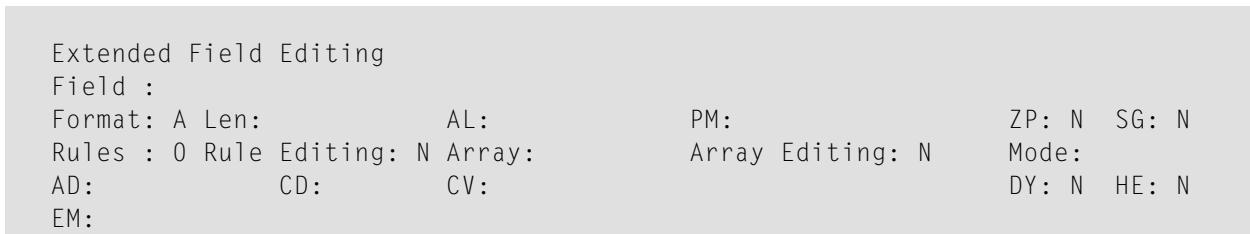
テキストの入力を終了します。

-  **注意:** テキストを定義した後にデータ変数を作成または定義、あるいはその両方を行う場合は、「マップでの *Natural* システム変数の使用」セクションを参照してください。

ユーザー定義変数の作成

フィールドリストから **[User Defined]** を選択し、ENTER キーを押します。

以下の例のような **[Extended Field Editing]** ウィンドウが表示されます。



画面の最下部にメッセージが表示され、次の操作を実行するように求められます：Position cursor and press Enter or format char.

フィールド位置の開始位置にカーソルを配置して ENTER キーを押します。

選択リストが開き、英数字のデータタイプの **A** など、有効な Natural データフォーマットがすべて表示されます。

-  **注意:** Natural データフォーマットがわかる場合は、カーソルを配置したフィールド位置に直接入力できます。したがって、使用可能なデータフォーマットは表示されません。

必要な Natural データフォーマットを選択して ENTER キーを押します。

[Extended Field Editing] ウィンドウに Natural データフォーマットが入力され、デフォルト名 (#1 など) がフィールドに割り当てられます。

この画面には、2 つの長さフィールドが表示されます。

1. **[AL]** (英数字長) : 表示上の長さ
2. **[Len]** : フィールドの内部的な長さ

ユーザー定義変数の長さは、次の手順を実行して定義します。

1. 最初のフィールドの長さを入力します (AL など)。
2. フィールド長の定義を完了した場合は ENTER キーを押します。
3. 変数にフィールド名を入力するように求められます。適切なフィールド名を選択し、ENTER キーを押します。

これで、最初のフィールドの定義機能が完了します。

2番目の長さフィールドにカーソルが自動的に移動します (Len)。フィールド長の定義を変更するか、またはTABキーを使用します。定義中のフィールドで使用する他の関連情報をすべて定義するまで、この処理を続行します。

これらのフィールドの詳細については、「[ユーザー定義変数の変更- フィールド編集](#)」を参照してください。

この定義が完了した場合は ENTER キーを押します。

マップでの Natural システム変数の使用

Natural システム変数はマップ定義でも指定できます。

Natural システム変数は、 [Create] > [User Defined] フィールドを使用して選択できます。

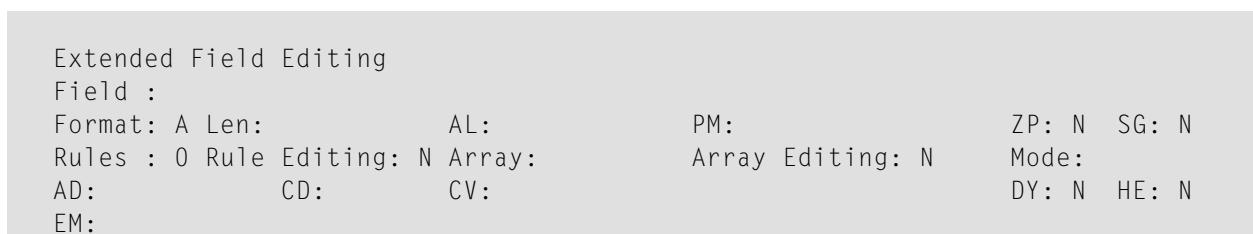
フィールドリストから [*] を選択し、表示されたリストからシステム変数を選択します。

指定したシステム変数の Natural データフォーマットがフィールド定義フォームに挿入されます。

ユーザー定義変数の変更 - フィールド編集

マップエディタを使用して、フィールドおよびそのすべての属性を定義します。

[MAP EDITOR] メニューから [Create] または [Modify] を選択し、ENTER キーを押します。次の例のような [Extended Field Editing] ウィンドウが表示されます。このウィンドウには、現在のフィールドの属性がすべて表示されます。



[Extended Field Editing] ウィンドウで、選択した任意のフィールドを変更できます。選択したフィールドとは現在のフィールドのことです。このフィールドは強調表示されます。

次の表で、 [Extended Field Editing] ウィンドウに表示されるフィールド属性について説明します。

フィールド属性	説明								
Field	<p>フィールド名。フィールド名の割り当ては、そのフィールドを最初に定義した方法に関係します。</p> <p>別の Natural オブジェクトのデータ定義からフィールドを取得した場合は、このオブジェクトのフィールド定義と同じ名前が割り当てられます。</p> <p>Natural システム変数としてフィールドを指定した場合は、指定した変数の名前が割り当てられます。</p> <p>フィールドが上のどちらでもない場合は、ダミーの名前が割り当てられます。このようなフィールドでは、名前を割り当ててからマップを実行する必要があります。</p> <p>フィールドの名前は変更できます。ただし、事前に接頭辞が割り当てられていないフィールドには、接頭辞を使用しないでください。接頭辞が付けられたフィールド名を取得するには、他の Natural オブジェクトのデータ定義からフィールドを選択します。名前の入力を求められます。変更が完了した場合は、ENTER キーを押して続行する必要があります。完了していない場合は、TAB キーを使用してフィールド属性を移動できます。</p> <p>注意: 重複するフィールド名は、出力専用フィールドとして定義されたフィールドについてのみ許可されています。</p>								
Format	フィールドの Natural データフォーマット。これらは現在の入力を上書きすることにより変更できます。								
Len	変数の内部的なプログラム長。								
〔AL〕 または 〔NL〕 または 〔FL〕 または 〔DF〕	<p>フィールドを表示するときに使用する長さ。フィールドラベルは、入力した Natural データフォーマットによって異なります。</p> <table border="1"> <tr> <td>AL</td> <td>フォーマット A (英数字)、L (論理)、および T (時刻) の場合</td> </tr> <tr> <td>NL</td> <td>フォーマット B (バイナリ)、I (整数)、N (数値)、および P (パック型数値) の場合</td> </tr> <tr> <td>FL</td> <td>フォーマット F (浮動小数点) の場合</td> </tr> <tr> <td>DF</td> <td>フォーマット D (日付) の場合</td> </tr> </table>	AL	フォーマット A (英数字)、L (論理)、および T (時刻) の場合	NL	フォーマット B (バイナリ)、I (整数)、N (数値)、および P (パック型数値) の場合	FL	フォーマット F (浮動小数点) の場合	DF	フォーマット D (日付) の場合
AL	フォーマット A (英数字)、L (論理)、および T (時刻) の場合								
NL	フォーマット B (バイナリ)、I (整数)、N (数値)、および P (パック型数値) の場合								
FL	フォーマット F (浮動小数点) の場合								
DF	フォーマット D (日付) の場合								
PM	<p>出力モード。</p> <p>この入力フィールドはセッションパラメータ PM に対応します。このフィールドと有効な入力値の詳細については、『パラメータリファレンス』の「PM-出力モード」を参照してください。</p>								
ZP	<p>ゼロ出力。ZP に値を入力できるのは、フィールドが数値または時間システム変数の場合のみです。</p> <p>この入力フィールドはセッションパラメータ ZP に対応します。このフィールドと有効な入力値の詳細については、『パラメータリファレンス』の「ZP-ゼロ出力」を参照してください。</p>								

フィールド属性	説明
SG	<p>記号の位置。SGに値を入力できるのは、フィールドが数値または時間システム変数の場合のみです。</p> <p>この入力フィールドはセッションパラメータ SG に対応します。このフィールドと有効な入力値の詳細については、『パラメータリファレンス』の「SG - 符号の位置」を参照してください。</p>
Rules	フィールドに対して現在定義されている処理ルールの数。
Rule Editing	<p>処理ルールの編集。次のプロンプトが表示されます。</p> <p>Inline Processing Rule Editing (Y, <PF2>=EDIT)?</p> <p>注意: 処理ルールの定義に使用したソースコードは、Natural プログラムエディタを使用する場合と同じ方法で入力または編集します。</p>
Array	フィールドが配列かどうかを示します（配列でない場合は空白）。
Array Editing	<p>配列の編集。次のプロンプトが表示されます。</p> <p>Array Editing (Y, <PF2>=EDIT)?</p>
Mode	<p>フィールドが作成された方法を示します。</p> <p>[Data] - フィールドは別のNaturalオブジェクト（DDMを除く）のデータ定義からコピーされています。</p> <p>[Sys] - フィールドはシステム変数です。</p> <p>[Undef] - フィールドは画面に直接作成されており、ダミー名が付けられています。</p> <p>[User] - フィールドの名前は拡張フィールド編集で作成されています。</p> <p>[View] - フィールドは DDM のフィールド定義からコピーされています。</p>
AD	<p>フィールド属性。</p> <p>このフィールドはセッションパラメータ AD に対応します。このフィールドと有効な入力値の詳細については、『パラメータリファレンス』ドキュメントの「AD - 属性の定義」で関連セクション（次を参照）を参照してください。</p> <p>次のプロンプトが表示されます。</p> <p>Attribute definitions (<PF2>=EDIT)</p> <p>次の項目を含む [Attribute Definition] ウィンドウが表示されます。</p> <ul style="list-style-type: none"> ■ Representation 「フィールド表現」を参照してください。 ■ Alignment 「フィールド揃え」を参照してください。

フィールド 属性	説明
	<ul style="list-style-type: none"> ■ I/O Characteristics 「フィールド入力／出力特性」を参照してください。 ■ Mandatory Characters 「必須入力」を参照してください。 ■ Length Characteristics 「入力値の長さ」を参照してください。 ■ Upper/Lower Case 「フィールド大文字／小文字特性」を参照してください。 ■ 充填文字 「充填文字」を参照してください。 <p>この機能にはトグル機能が組み込まれています。表示されている属性定義文字が正しい場合は、ESCキーを使用します。文字は変更されません。フィールドの属性特性を変更するには、目的の属性を選択してENTERキーを押します。AD定義に変更が挿入されます。この機能を終了するにはESCキーを押します。選択した項目ごとに属性ウィンドウが表示されます。このウィンドウの説明はそれぞれのウィンドウを参照してください。</p>
CD	<p>カラー定義。次のプロンプトが表示されます。</p> <p>Color definitions (<PF2>=EDIT <CSR-UP/DN>=Select (Esc=Cancel Enter=OK))</p> <p>カラー定義を編集するにはPF2キーを押します。色を選択するにはカーソルを使用してENTERキーで確定します。</p>
CV	<p>ダイナミックフィールド属性の制御変数。次のプロンプトが表示されます。</p> <p>(<PF2>=Edit Rank if Array)</p> <p>このフィールドに使用する属性を含む変数の名前。プログラムでは、この変数をNaturalデータフォーマットCで定義する必要があります。</p> <p>また、制御変数にはMODIFIEDデータタグが含まれています。このデータタグは、マップ実行後にフィールドが変更されたかどうかを示しています。</p> <p>単一の制御変数を複数のマップフィールドに適用できます。その場合、制御変数を参照しているフィールドのいずれかが変更されると、MODIFIEDデータタグが設定されます。</p>
DY	<p>ダイナミック STRING 属性。次のプロンプトが表示されます。</p> <p>(Y,<PF2>=Edit)</p> <p>このパラメータを使用して、英数字変数のテキスト文字列に格納される特定の文字を定義し、属性設定を制御します。有効な入力値の詳細については、『パラメータリファレンス』の「DY - ダイナミック属性」を参照してください。</p>

フィールド属性	説明
HE	<p>〔HE〕 オプションは、ヘルプルーチンまたはヘルプマップをマップフィールドに割り当てるために使用します。ヘルプルーチンまたはヘルプマップは、その後ヘルプ要求がマップフィールドに対して行われると、実行時に呼び出されます。詳細については、『パラメータリフレンス』の「HE ヘルプルーチン」でHEセッションパラメータの説明を参照してください。</p> <p>〔HE〕 フィールドに「Y」を入力するか、またはフィールドにカーソルを配置して PF2 キーを押すと、ヘルプルーチンまたはヘルプマップの名前、およびこのヘルプルーチンまたはヘルプマップに渡されるパラメータを入力するウィンドウが表示されます。</p> <p>〔HE〕 フィールドに指定する名前およびパラメータに適用される構文は、『パラメータリフレンス』の「HE パラメータの構文」に記載されている HE セッションパラメータの構文に対応しています。そこに記載されている構文の説明に加えて、マップエディタを使用する場合、次のものが適用されます。</p> <p><i>operand1</i>:</p> <ul style="list-style-type: none"> ■ マップフィールドの名前に対応する変数名が指定されている場合は、このフィールドは Natural データフォーマット／長さ A8 である必要があります。 ■ マップフィールドがまだ存在しない変数名が指定されている場合、その名前のマップパラメータが、Natural データフォーマット／長さフィールド A8 で自動的に定義されます。 <p><i>operand2</i>:</p> <ul style="list-style-type: none"> ■ マップフィールドがまだ存在しない変数名が指定されている場合、その名前のマップパラメータが、Natural データフォーマット／長さ N7 で自動的に定義されます。 <p>〔HE〕 フィールドからパラメータを削除することは、パラメータがマップフィールドでない場合、あるいはヘルプパラメータまたは [Starting from] 値として他のマップフィールドに関連付けられていない場合、パラメータがマップからも削除されることを示しています。</p>
EM	フィールドに対して使用する編集マスク。

上記の 3 つの項目は、次の内容と特に関連しています。

- ルール編集 - 処理ルール
- Array Editing

■ AD - 属性定義

ルール編集 - 処理ルール

フィールド関連の処理ルール

次の3つのタイプの処理ルールを定義できます。

- インライン処理ルール
- Predict フリールール
- Predict 自動ルール

インライン処理ルールはマップソース内で定義され、名前は割り当てられません。インラインルールでは、Predict を使用できなくても構いません。

 **注意:** フィールド関連のインライン処理ルールは、フィールド単位でも実行できます。詳細については、「[フィールドに基づいた処理](#)」セクションを参照してください。

Predict フリールールには名前が割り当てられ、Predict ディクショナリに保存されます。既存の Predict フリールールは変更できません（変更は Predict のみで可能）。ただし、フリールールでは、エディタへの読み込み、変更、および別名での保存による新規フリールールの作成を行うことができます。

ローカルまたはリモートの Predict 環境に保存されている Predict フリールールにアクセスできます。

UNIX、OpenVMS、またはメインフレームホストのリモート環境に保存した Predict フリールールには、Natural RPC サーバーを使用してアクセスできます。

Predict サーバーに接続する方法の詳細については、『パラメータリファレンス』のプロファイルパラメータ USEDIC、および『コンフィグレーションユーティリティ』ドキュメントの「コンフィグレーションファイルのパラメータの概要」セクションの「ディクショナリサーバー割り当て」を参照してください。

ルール名を割り当てると、インラインルールを Predict フリールールにできます。削除すると Predict フリールールをインラインルールにできます。

Predict の自動ルールはデータベースフィールドに適用され、Predict 管理者によって定義されます。他の Natural オブジェクトのデータ定義からフィールドを作成し、そのフィールドがデータベースフィールドである場合は、そのフィールドのすべての自動ルールがマップ定義にリンクされます。すべての自動ルールは連結され、単一のマップルールとして処理されます。

自動ルールのランクはマッププロファイル設定に定義されています（デフォルトは 1）。

自動ルールは、マップエディタを使用して変更することはできません。ただし、コマンド `P=n` を使用するか、または古いランクを上書きして、異なるランクを割り当てることができます。

処理ルールのソースコード内のアンパサンド (&) は、そのルールを定義したフィールドの完全修飾名によって、ダイナミックに置き換えられます。配列の添字はこの置き換えの影響を受けません。次の例に示すように、アンパサンド (&) の後に明示的に添字表記を指定する必要があります。

例：

```
IF &      = ' ' THEN REINPUT 'ENTER NAME' MARK *&    /* For a scalar field
IF &(1) = ' ' THEN MOVE 'X' TO &(*)                 /* For an array field
```

処理ルールのソースコード内でフィールド名表記 `&.field-name` を使用すると、ビューネームを持つフィールドを明示的に指定することなく、データベースフィールド間の値の完全性を照合する DDM 固有のルールを定義できます。`field-name` として、DDM に定義されたデータベースフィールドの名前を指定します。コンパイル時には、対応するビューネームでアンパサンド (&) が置き換えられ、フィールドがダイナミックに指定されます。これにより、フィールドを取得するビューに関係なく、特定のフィールドに同じ処理ルールを使用できます。

ファンクションキー関連の処理ルール

次の 2 種類のファンクションキー関連の処理ルールを定義できます。

- インライン処理ルール
- Predict フリールール

ファンクションキー関連の処理ルールを使用すると、プログラム依存のファンクションキーにマップ処理中の動作を割り当てるすることができます。すでにプログラムでコマンドが割り当てられているファンクションキーの場合、ルールは処理されずにこのコマンドが実行されます。

例：

```
IF *PF-KEY = 'PF3'
  ESCAPE ROUTINE
END-IF
```

このルールが実行されると、それ以上ルール処理が行われずに、マップ処理が終了します。

処理ルールのランク

1 つのフィールドには、最大で 100 の処理ルール（ランク 0~99）を定義できます。マップ実行時に、処理ルールはフィールドのランクおよび画面位置によって昇順に実行されます。PF キー処理ルールは、常に最初の画面位置にあるものとみなされます。

最適なパフォーマンスを得るために、処理ルールにランクを割り当てるときは、次の割り当てをお勧めします。

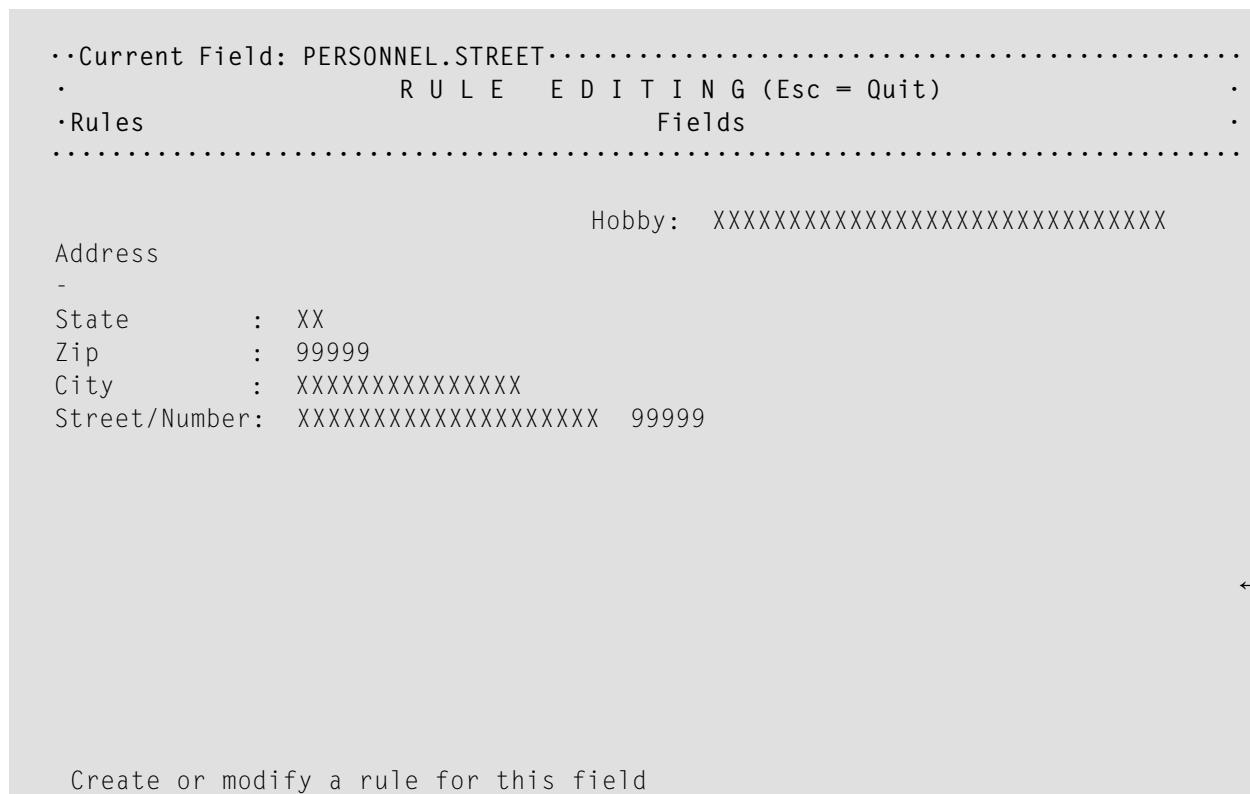
ランク	処理ルール
0	終了ルール
1~4	自動ルール
5~24	フォーマットチェック
25~44	個別フィールドの値チェック
45~64	フィールド間での値の照合
65~84	データベースアクセス
85~99	特殊な用途

処理ルールの編集

 **注意:** 処理ルールを変更または追加する場合は、変更内容を有効にするために、そのルールを参照しているマップを再カタログする必要があります。

フィールド関連の処理ルールを編集するには、**【Extended Field Editing】** ウィンドウで **【Rule Editing】** を選択します。ファンクションキー関連の処理ルールを編集するには、**【Ops. Map】** 選択リストから **【Key Rules】** を選択します。

オプション **【Rules】** および **【Fields】** を含む次のウィンドウが表示されます。



カーソルを使用してこれらのオプションの1つを選択し、ENTERキーを押します。

[Fields] を選択すると、現在のマップで使用されている変数のリストが表示されます（参照目的のみ）。

[Rules] を選択すると、現在のフィールドの既存ルールの選択リストが表示されます。

各リスト上で、Predict ルールは名前によって、インライнулルールは最初の 3 行のソースコードによって識別されます。

フィールドまたはキーに処理ルールを定義するには、次の方法があります。

- 新規処理ルールの作成
 - 新規ルールの定義
 - 既存ルールを変更して新しい名前で保存
- 既存ルールの割り当て
 - 編集
 - 移動
 - コピー
 - ルールのリンク解除

▷手順 16.1. 新規処理ルールを定義するには

- 1 [Create] を選択します。
空のルールエディタが表示されます。
- 2 ルールを入力します。（ソースコードは、Natural プログラムエディタと同じ方法で使用します。）
- 3 このルールを Predict フリールールにする場合は、ルールに名前を付けます。このルールをインライнулルールにする場合は、名前を付けないでルールを保存します。
- 4 ルールを保存します。「[処理ルール編集のコマンド](#)」セクションを参照してください。



重要: ルールを保存したら、Predict で変更のみできます。

▷手順 16.2. 既存の処理ルールを変更するには

- 1 ルールエディタヘッダーで、[Rule] フィールドに Predict ルールの名前を入力して ENTER キーを押します。
ルールエディタにルールが表示されます。
- 2 ルールを変更します。（ソースコードは、Natural プログラムエディタと同じ方法で使用します。）

- ルールの名前を変更して保存します。「[処理ルール編集のコマンド](#)」セクションを参照してください。



重要: ルールを保存したら、Predict で変更のみできます。

手順 16.3. 既存のルールを割り当てるには

- リストから既存のルールを選択します。
- ENTER キーを押します。

ウィンドウが開き、オプション Edit、Move、Copy、Unlink が表示されます。

■ 編集

ルールを変更するには、[Edit] を選択します。ルールの名前および内容（フリールールの場合）がエディタに表示されます。セクションの手順2を参照し、既存の処理ルールを変更します。

■ Move

ルールのランクを変更するには、[Move] を選択します。ENTER キーを押すと、新しいランクを選択できるリストが表示されます。

■ Copy

ルールをコピーして新しいランクを割り当てるには、[Copy] を選択します。ENTER キーを押すと、新しいランクを選択できるリストが表示されます。

■ リンク解除

フィールドからルールを削除するには、[Unlink] を選択します。

■ ESC で終了します。

 **注意:** データベースステートメントを参照するルールが書かれている場合は、行参照番号ではなくラベルを使用する必要があります。

必要なフィールド処理ルールを入力した後、コマンドを発行します。

P=nn

ここで、nn は処理ルールのランクです。

このコマンドにより、ルールが自動的に保存されます。

処理ルール編集のコマンド

処理ルールエディタでは、エディタコマンド行で次のコマンドを使用して、編集する処理ルールを選択できます。

Command	関数
P nn	ランク nn のルールを選択します。
P*	選択リストからルールを選択します。
P	フィールドに定義された次のルールに進みます。
P=nn	現在のランクのルールにランク nn を割り当て、ルールを自動的に保存します。
U	リンクを解除（削除）します。
.	処理ルールの編集を終了し、ルールを保存します。

Array Editing

配列の編集を起動するには、TAB キーを使用して、フィールド属性ウィンドウの項目 **[Array Editing]** を選択します。文字 N を Y (キー依存) で置き換えるか、または PF2 キーを押すと、**[Array Definition]** ウィンドウが表示されます。

·Array Definition.....			
Name #1	Upper Bnds 1____ 1____ 1____		
Dimensions	Occurrences	Starting from	Spacing
1 . Index vertical	1__	_____	0 Lines
0 . Index horizontal	1__	_____	1 Columns
0 . Index (H/V) V	1__	_____	0 Cols/Ls

このウィンドウには、マップに定義されたフィールドが表示されます。変更が不要な場合は、TAB キーを押して次のフィールドに進みます。前のフィールドに戻る場合は、SHIFT+TAB キーを押します。

 **注意:** マップエディタでは、X-array (拡張可能な配列) はサポートされていません。

この **[Array Definition]** ウィンドウには、次のエントリが含まれています。

エントリ	説明
Upper Bnds	配列の上限を示します。つまり、(左から右に) 1 次元、2 次元、および 3 次元で最も高いオカレンスです。 プログラムで定義されたフィールドがマップ配列を定義するために使用される場合、プログラムで定義されている、そのフィールド (ユーザー定義変数またはデータベースフィールド) の上限が使用されます。これらは配列定義画面で上書きすることはできません。 マップ配列が他の Natural オブジェクトのデータ定義から生成されている場合は、そのマップ配列の次元が、このフィールドに表示されている次元を超えてはいけません。 マップ配列が他の Natural オブジェクトのデータ定義から生成されていない場合は、そのマップ配列の次元が、Natural プログラムで定義されている次元を超えてはいけません。

エントリ	説明
Dimensions	配列は 3 次元まで保持できます。配列の次元をマップレイアウトにマップする順序は、[Index] オペランドの左側に入力された値によって決まります。使用される短縮形は、H=Horizontal および V=Vertical です。
Occurrences	1 次元に対して定義されるオカレンス数。
Starting from	1 次元に対する開始インデックス値。実際の値がマップ定義を呼び出す Natural プログラムで提供されることを示すために、数値を使用することも、変数名を使用することもできます。
Spacing	各次元のオカレンス間に挿入する空行（垂直方向の次元）または空列（水平方向の次元）の数。

必要な情報を入力して ENTER キーを押します。フィールド属性ウィンドウにリストされている次の項目 (AD) に戻ります。

配列定義の例

例 1

10 個の垂直方向のオカレンスから構成される 1 次元配列。各オカレンス間に 2 つの空行が挿入されています。

```

·Array Definition.....  

Name #1                                Upper Bnds 10____ 1____ 1____  

-----  

Dimensions          Occurrences      Starting from      Spacing  

1 . Index vertical    10____          _____           2 Lines  

0 . Index horizontal   1____          _____           1 Columns  

0 . Index (H/V) V      1____          _____           0 C1s/Ls  

.....
```

例 2

水平方向の配列となること以外は例 1 と同じです。

```

·Array Definition.....  

Name #1                                Upper Bnds 10____ 1____ 1____  

-----  

Dimensions          Occurrences      Starting from      Spacing  

0 . Index vertical    1____          _____           0 Lines  

1 . Index horizontal   10____          _____           1 Columns  

0 . Index (H/V) V      1____          _____           0 C1s/Ls  

.....
```

例 3：

2次元配列。最初の次元は10個の垂直方向のオカレンスから構成されています。各オカレンス間に1つの空行が挿入されています。2つ目の次元は5個の水平方向のオカレンスから構成されています。各オカレンス間に2つの空列が挿入されています。

```

·Array Definition.....  

Name #1                                Upper Bnds 10____ 5____ 1____  

-----  

Dimensions          Occurrences  Starting from  Spacing  

1 . Index vertical    10_          _____      1  Lines  

2 . Index horizontal   5_          _____      2  Columns  

0 . Index (H/V) V      1_          _____      0  Cls/Ls  

.....
```

例 4

次元の順序が逆であること以外は例3と同じです。

```

·Array Definition.....  

Name #1                                Upper Bnds 5____ 10____ 1____  

-----  

Dimensions          Occurrences  Starting from  Spacing  

2 . Index vertical    10_          _____      1  Lines  

1 . Index horizontal   5_          _____      2  Columns  

0 . Index (H/V) V      1_          _____      0  Cls/Ls  

.....
```

例 5

3次元配列。最初の次元は3個の垂直方向のオカレンスから構成されています。各オカレンス間に1つの空行が挿入されています。2つ目の次元は5個の水平方向のオカレンスから構成されています。各オカレンス間に2つの空列が挿入されています。3つ目の次元は2個のオカレンスから構成され、最初の次元の各オカレンス内に垂直方向に展開されています。

```

·Array Definition.....  

Name #1                                Upper Bnds 3____ 5____ 2____  

-----  

Dimensions          Occurrences  Starting from  Spacing  

1 . Index vertical    3_          _____      1  Lines  

2 . Index horizontal   5_          _____      2  Columns  

3 . Index (H/V) V      2_          _____      0  Cls/Ls  

.....
```

例 6

Starting from を使用した例。最初の次元はインデックス 1 から始まる 10 個の垂直方向のオカレンスから構成されています。1 は、デフォルトでは Natural データフォーマット／長さ N7 でマップエディタに定義されています。2 つ目の次元はインデックス 3 から始まる 5 個の水平方向のオカレンスから構成されています。

```

·Array Definition.....  

Name #1                                Upper Bnds 10____ 5____ 1____  

-----  

Dimensions          Occurrences  Starting from  Spacing  

1 . Index vertical    10____    1____      1  Lines  

2 . Index horizontal  5____    3____      2  Columns  

0 . Index (H/V) V      1____    _____    0  C1s/Ls  

.....
```

例 7

1 次元配列から 2 次元表示を作成する例。配列は 40 個の要素から構成されています。それぞれ 20 行ある 2 列に表示されます。このためには、水平方向インデックスとして 0 を指定します。

```

·Array Definition.....  

Name #1                                Upper Bnds 40____ 1____ 1____  

-----  

Dimensions          Occurrences  Starting from  Spacing  

1 . Index vertical    20____    _____    0  Lines  

0 . Index horizontal  2____    _____    10  Columns  

0 . Index (H/V) V      1____    _____    0  C1s/Ls  

.....
```

AD - 属性定義

属性定義の編集は、次のように実行します。TAB キーを使用して、フィールド属性ウィンドウの [AD] 項目に移動します。

PF2 キーを押すと、次の [Attribute Definition] ウィンドウが表示されます。

```
Children      : 99          Years-Educ: 99
Family Status: XXXXXXXXXXXX Years-Comp: 99
Sex          : X            Vacation-D: 99
                           Sick-Days : 99

                           Hobby: XXXXXXXXXXXXXXXXXXXXXXXXX

Address
-
State       : XX          ..Attribute Definition...
Zip         : 99999         ..Representation        .
City        : XXXXXXXXXXXX ..Alignment          .
Street/Number: XXXXXXXXXXXX ..I/O Characteristics  .
                           ..Mandatory Characters  .
                           ..Length Characteristics .
                           ..Upper/Lower Case     .
                           ..Filler Character     .
..Extended Field Editing..... .
·Field= PERSONNEL.STREET .
·Format= A Len= 20   Alphan. Len= 20   PM=
·Rules: 0 Rule Editing? N Array:           Array Editing? N      Mode= Data .
·AD= FHWOIL      CD=      CV=           DY= ->   HE= -> .
·EM=           .
.....
```

カーソルキーを使用して、必要な項目を選択します。

ENTER キーを押すと、選択した各項目の追加選択ウィンドウが表示されます。

 **注意:** 表示されている属性定義文字が正しい場合は、ESC キーを使用します。この文字は変更されません。ESC キーを押すと、AD 編集が終了します。

データ定義の選択

マップフィールドは、別のNaturalオブジェクトからデータ定義を選択することにより定義できます。データ定義は、Natural DDM（データ定義モジュール）内のフィールド定義、または変数定義のいずれかです。

データ定義を選択するには、まずオブジェクトクラスを選択します。有効なオブジェクトクラスは次のとおりです。

オブジェクトクラス	説明
A	パラメータデータエリア
G	グローバルデータエリア
H	Helproutine
L	ローカルデータエリア
M	マップ
N	Subprogram
P	プログラム
S	サブルーチン
V	ビュー (DDM)

プログラム、サブルーチン、サブプログラム、およびヘルプルーチンは、`DEFINE DATA`ステートメントが含まれている場合にのみ使用できます。

デモ用には、タイプ DDM の Natural オブジェクトを使用します。[MAP EDITOR] メニューから [Create] を選択し、続いて [View Defined] を選択します。または、(Create を表す) C を入力し、続いて (View Defined を表す) V を入力します。

開いたポップアップウィンドウから、必要な DDM を含むライブラリを選択します。

画面に次のライブラリリストが表示されます。

マップエディタ

必要な DDM (PERSONNELなど) を選択して ENTER キーを押します。選択した DDM がウィンドウに表示されます。

```
*** Personnel Data Detail Display Function ***

Person Data                                Employment Data
-
Name      : XXXXXXXXXXXXXXXXXXXXXXXX
Initial   : X
First-Name : XXXXXXXXXXXXXXXXXXXX
Children  : 99
Family Status: XXXXXXXXXX
Sex       : X

Job       : XXXXXXXXXXXXXXXXXXXXXXXX
Salary    : 999999
Commission: 999999
Years-Educ: 99
Years-Comp: 99
Vacation-D: 99
Sick-Days : 99

Hobby: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Address
-
...PERSONEL.....
· 1 AA PERSONNEL-NUMBER          N 8.0   D
· 1 AA PERSONAL-NUMMER          N 8.0   D
· 1 AA NUMERO-PERSONNEL        N 8.0   D
·G 1 G1 PERSON                  ·
· 2 BA NAME                     A 20 N D
.....
HD=PERSONNEL/NUMBER
```

強調表示されているフィールドの場合、このウィンドウには画面の最下部に追加の情報行が3行表示されます。

こうした行では、現在のフィールドについて次の情報が表示されます。

- 編集マスク
- Header
- コメント

フィールド名は、ピリオドでフィールド名と連結したDDM名で構成されます。次に例を示します。

personal.personnel-number

マップに追加するフィールドを選択して ENTER キーを押します。

ライブラリウィンドウが開いて選択したフィールドが [Field=] 行に表示され、次のようなプロンプトが表示されます。

Position cursor and press Enter.

フィールド位置の開始位置にカーソルを配置して ENTER キーを押します。

「[ユーザー定義変数の作成](#)」の説明に従って、フィールドの変更を続行します。

この手順を繰り返して、マップに必要なフィールドをすべて定義します。

パラメータまたはローカルデータ定義用フィールドの定義

〔Create〕メニューから項目〔Parm Defined〕または〔Local Defined〕を選択して ENTER キーを押すと、新規パラメータまたは新規ローカル変数を作成したり、既存のパラメータまたは変数を変更したりできます。

以下では次のトピックについて説明します。

- [パラメータ定義](#)
- [ローカルデータ定義](#)

パラメータ定義

〔Parm Defined〕項目を選択すると、次のウィンドウが表示されます。

...PARAMETER.....
·<CREATE>
·TEST1	L 001:003
·TEST2	L 001:003,001:004
·TEST3	L 001:003,001:004,001:005
.....

この機能を使用すると、新規パラメータの追加および既存パラメータの変更を実行できます。

既存パラメータ（TEST3 など）を変更する場合は、カーソルを使用してパラメータを選択し、ENTER キーを押します。ウィンドウが表示され、必要なアクションの選択を求められます。選択したパラメータの **Edit** および **Delete**、またはアクションの **Cancel** を実行できます。

〔Edit〕を選択すると、次のウィンドウが表示されます。このウィンドウでは、選択したパラメータを編集できます。

```
..PARAMETER.....  
·Name.....: TEST3  
·Format....: L  
·  
·Dimension: 3  
·      Lower Bnds  :  Upper Bnds  
·1. Index.:      1          3  
·2. Index.:      1          4  
·3. Index.:      1          5  
.....
```

新規パラメータを作成する場合は、[<CREATE>] オプションを選択します。同様のウィンドウが表示されますが、この時点ではウィンドウが空であるため、作成するパラメータを指定できます。

上述したウィンドウを終了すると別のウィンドウが表示され、変更／指定内容を保存するか、アクションをキャンセルするかを選択するように促されます。

ローカルデータ定義

[Local Defined] 項目を選択すると、次のウィンドウが表示されます。

```
..LOCAL.....  
·<CREATE>  
·TESTA          L      001:003  
·TESTB          L      001:003,001:004  
·TESTC          L      001:003,001:004,001:005  
.....
```

この機能を使用すると、新規ローカル変数の追加および既存の変数の変更を実行できます。ローカル変数を使用すると、ある処理ルールから別の処理ルールに値を渡すことができます。

既存ローカル変数 (TESTCなど) を変更する場合は、カーソルを使用して変数を選択し、ENTERキーを押します。ウィンドウが表示され、必要なアクションの定義を求められます。選択した変数の編集および削除、または機能のキャンセルを実行できます。

アクション [Edit] を定義すると、次のウィンドウが表示されます。このウィンドウでは、選択したローカル変数を編集できます。

```
..LOCAL.....  
·Name.....: TESTC  
·Format....: L  
·  
·Dimension: 3  
·           Lower Bnds : Upper Bnds  
·1. Index.:      1          3  
·2. Index.:      1          4  
·3. Index.:      1          5  
.....
```

新規ローカル変数を作成する場合は、[<CREATE>] オプションを選択します。同様のウィンドウが表示されますが、このときはウィンドウが空であるため、作成する変数を指定できます。

上述したウィンドウを終了すると別のウィンドウが表示され、変更／指定内容を保存するか、関数をキャンセルするかを選択するように促されます。

マッププロファイル

このセクションでは、マップにマップ設定（プロファイル）を定義する処理について説明します。

マッププロファイルは、[Ops. Map] 選択リストから [Prof. Map] を選択して呼び出します。次の画面が表示されます。

このマッププロファイル画面は、マッププロファイル設定および充填文字で構成されます。

以下では次のトピックについて説明します。

- マッププロファイル設定
 - 充填文字

マッププロファイル設定

次のマッププロファイル設定を使用できます。

エンtries	説明
Page Size	編集するマップの行数 (1~250)。 [Std Keys] (該当エンtriesを参照) が "Y" に設定されている場合、行数は 3~250 に制限されます。WRITE ステートメントを使用して出力されるマップの場合は (マップ設定の [Context] 列のエンtries「WRITE ステートメント」を参照)、マップのサイズではなく、WRITE ステートメントで出力される論理ページの行数を指定します。そのため、マップを 1 ページに複数回出力できます。
Line Size	編集するマップ列の数 (5~249)。
Layout dynamic	事前定義済みレイアウトを含むマップソース定義の名前。 Y-レイアウトをダイナミックと指定します。ダイナミックに使用されるレイアウトは、コンパイル時にマップの固定部にならず、ランタイムで実行されます。このため、レイ

エントリ	説明
	<p>アウトマップに対するその後の変更は、そのレイアウトマップを使用するすべてのマップに対して有効になります。</p> <p>レイアウトマップにユーザー定義変数が含まれている場合、レイアウトマップを使用して、マップでこれらのパラメータを定義する必要があります。レイアウトマップの入力フィールドおよび変更可能なフィールドは、ランタイムには開かれません。[Field and Variable Definitions] 機能で F9 キーを押すと、パラメータを追加できます。</p> <p>N- レイアウトをスタティックと指定します。マップが初期化されるときに、スタティックレイアウトはソースエリアにコピーされます。充填文字は転送されません。デフォルト設定は "N" です。</p>
Zero Print	<p>Y - すべてゼロのフィールド値をゼロ 1 つのみで表示します。</p> <p>N - ゼロ値を空白で表示します。</p> <p>新しいフィールドを作成すると、この値はフィールド定義にコピーされます。個々のフィールドの値は、拡張フィールド編集機能を使用して変更できます。</p>
Upper Case	<p>Y - マップ実行時に、フィールドに入力されたすべての入力が大文字に変換されることを示します。</p> <p>N - 小文字から大文字への変換が実行されないことを示します。</p> <p>新しいフィールドを作成すると、この値はフィールド定義にコピーされます。個々のフィールドの値は、拡張フィールド編集機能を使用して変更できます。</p>
Manual Skip	<p>Y - 実行時、現在のフィールドに完全に入力した場合でも、マップの次のフィールドにカーソルが自動的に移動しません。</p> <p>N - 実行時、現在のフィールドに完全に入力した場合は、マップの次のフィールドにカーソルが自動的に移動します。デフォルト設定は "N" です。</p>
Decimal Char	10 進文字として使用される文字。この文字は GLOBALS コマンドでのみ変更することができます。
Standard Keys	<p>Y - 実行時、ファンクションキー指定ができるように、マップの最後の 2 行が空白のままになります。</p> <p>N - すべての行がマップに使用されます。</p>
Right Justify	<p>フィールド桁揃えのタイプ。他の Natural オブジェクトのデータ定義から取得した数値フィールドおよび英数字フィールドに使用されます。</p> <p>Y - 右詰め N - 左詰め</p>
Print Mode	<p>変数のデフォルトの出力モードは、以下のとおりです。</p> <p>C - 代替文字セットを使用することを示します（特殊文字テーブル）。</p> <p>I - 出力方向を逆にすることを示します。</p> <p>IC - 標準の出力方向を示します。この値は、新しいフィールドを作成するとフィールド定義にコピーされます。</p>

エントリ	説明
Control Var	制御変数の名前。この内容によって、属性定義が AD=Y または (Y) に設定されているフィールドおよびテキストの属性特性が決定します。制御変数の最大長は、8 文字に制限されています。マップで参照される制御変数は、そのマップを使用するプログラムで定義する必要があります。
Field Sensitive	<p>Y - マップフィールドに付加された処理ルールがフィールド単位で実行されることを示します。つまり、指定したフィールドを離れた直後に実行されます。</p> <p>N - マップにフィールド感知を定義しないことを示します。つまり、必要な値をマップフィールドにすべて入力し、ENTERキー（または任意のFキー）を押すまでマップは処理されません。</p>
WRITEステートメント	<p>Y - このフィールドを Y でマークすると、マップ定義プロセスの終了時に WRITE ステートメントが生成されます。この生成されたマップは、WRITE USING FORM ステートメントを使用して Natural プログラムから呼び出すことができます。マップを 1 ページに複数回出力できるように、マップの最後にある空の行は自動的に削除されます。</p> <p>N - このフィールドを N でマークすると、マップ定義プロセスの結果、INPUT ステートメントが生成されます。この生成されたマップは、INPUT ステートメントを使用して Natural プログラムから呼び出すことができます。</p>
Helproutine	このマップに対してヘルプ機能を呼び出したとき、ランタイム時に呼び出されるヘルプルーチンの名前（マップのグローバルヘルプ）。構文の詳細については、「 ユーザー定義変数の変更 - フィールド編集 」セクションのパラメータ Help を参照してください。
Help Parameter	<p>ヘルプ機能を呼び出したとき、実行時に呼び出されるヘルプパラメータ。</p> <p>注意: 最大で 20 個のヘルプパラメータを指定できます。20 を超えるヘルプパラメータを入力すると、その値は無視されます。</p>
as field default	<p>Y - マップのヘルプルーチンが、マップの個々のフィールドにデフォルトとして適用されることを示します。つまり、各フィールドの名前が個別にヘルプルーチンに渡されます。</p> <p>N - マップの名前をヘルプルーチンに渡すことを示します。</p>
ヘルプテキスト	Y - このマップが実際はヘルプテキストであることを示します。デフォルトは "N" です。
Position Line Column	実行時にヘルプマップが表示される画面上の位置。
自動ルールランク	フィールド定義時にマップにリンクされるときに、Predict 自動ルールに割り当てられるランク（プライオリティ）。デフォルトは 1 です。

充填文字

充填文字を割り当てるとき、フィールドの情報が必須かどうか、フィールドを完全に満たす必要があるかどうかを示すことができます。

フィールドタイプ	説明
Optional Partial	任意。完全に埋める必要はありません。
Required Partial	必須。完全に埋める必要はありません (AD=E)。
Optional Complete	任意。完全に埋める必要があります (AD=G)。
Required Complete	必須。完全に埋める必要があります (AD=EG)。

拡張フィールド編集機能を使用して、個々のフィールドに充填文字を定義することもできます。フィールドタイプの定義については、セッションパラメータ AD を参照してください。

事後割り当て

以前にマップの画面レイアウトで定義されたフィールドに、DDM フィールド定義または DEFINE DATA 定義のフィールド名およびフィールド属性を割り当てることができます。

 **注意:** 重複するフィールド名は、「出力専用フィールド」として定義されたフィールドについてのみ許可されています。

DDM から作成されているマップフィールドは、この DDM または DEFINE DATA 定義から適切な DDM フィールド定義を使用して再定義する必要があります。ただし、これは同じデータベース フィールドである場合にのみ可能です。

DDM フィールド定義の事後割り当てでは、マップから個々のフィールド定義を削除して、新しい DDM フィールド定義を DDM または DEFINE DATA 定義から選択することによってのみ実行できます（「[データ定義の選択](#)」を参照）。

その配列の 1 つ以上の次元がレイアウトの配列の次元より小さい場合は、(DDM フィールド定義または DEFINE DATA 定義の) ビュー配列に事後割り当てを使用することはできません。

フィールドに基づいた処理

マッププロファイルでは、必要な値をマップフィールドにすべて入力して ENTER キーを押した後にマップを処理するか、またはマップフィールドに付加された処理ルールをフィールド単位で処理するかどうかを指定できます。

フィールド単位で処理する場合、フィールドに付加された処理ルールは、このフィールドに完全に入力するか、またはカーソルを別の位置に移動した直後に実行されます。実行される順序は、フィールド感知がない場合と同様です。

以下では次のトピックについて説明します。

- [フィールドに基づいた処理の利点](#)
- [Field-Sensitive としてマップを定義](#)

フィールドに基づいた処理の利点

フィールドに基づいた処理の利点については、次のセクションで説明します。

- [フィールドのダイナミックな充填（ユーザー入力に基づく）](#)
- [ユーザーガイダンスの改善](#)
- [ユーザー入力に基づいたセキュリティ](#)
- [迅速なデータ入力](#)

 **注意:** フィールドに基づいた処理の利点を十分活用するには、既存の処理ルールをフィールドに基づいて調整する必要があります。処理ルールの詳細については、「[ルール編集 - 処理ルール](#)」を参照してください。

フィールドのダイナミックな充填（ユーザー入力に基づく）

フィールドに基づいた処理をしない場合、アプリケーションは、すべてのフィールドへの入力が完了してENTERキーを押すまで待機してから、入力されたデータが有効かどうかをチェックします。最悪の場合、エラーが通知されるまでにフィールドへの入力に費やした時間と手間が、すべて無駄になることがあります。

一方、フィールドに基づいた処理を行っている場合は、フィールドに完全に入力するか、ENDキーまたはSHIFT+ENDキーを押してフィールドを離れるか、またはカーソル移動キーのいずれかを使用してカーソルを移動してフィールドを離れると、データ入力が直ちにチェックされます。他のフィールドは、前のフィールドに入力された値に基づくデータを使用して、事前に（REINPUT FULLを使用して）自動的に入力されます。また、この値に応じて、これらのデータがデータベースクエリの結果となる場合があります。

どのような状態でフィールドを離れたかに関する情報は、*PF-KEYシステム変数から取得できます。この変数には、"FULL"（フィールドに完全に入力した場合）または次の表に示す最後に押したキーの名前が含まれています。

押されたキー	*PF-KEY
←	LEFT
→	RIGT
↑	UP
↓	DOWN
Page Up	PGUP
Page Down	PGDN
TAB	TAB
HOME	HOME
END	END
Backtab	BTAB
ENTER	ENTR
フィールドが一杯	FULL

ユーザーガイダンスの改善

フィールドに基づいた処理を行うと、入力した値に従ってフィールドからフィールドへと誘導されます。カーソルは事前に入力されているフィールドをスキップし、次の入力フィールドへと移動します。

フィールドをチェックした直後にメッセージが表示されるため、対応方法についての通知がより迅速かつ正確になります。

ユーザー入力に基づいたセキュリティ

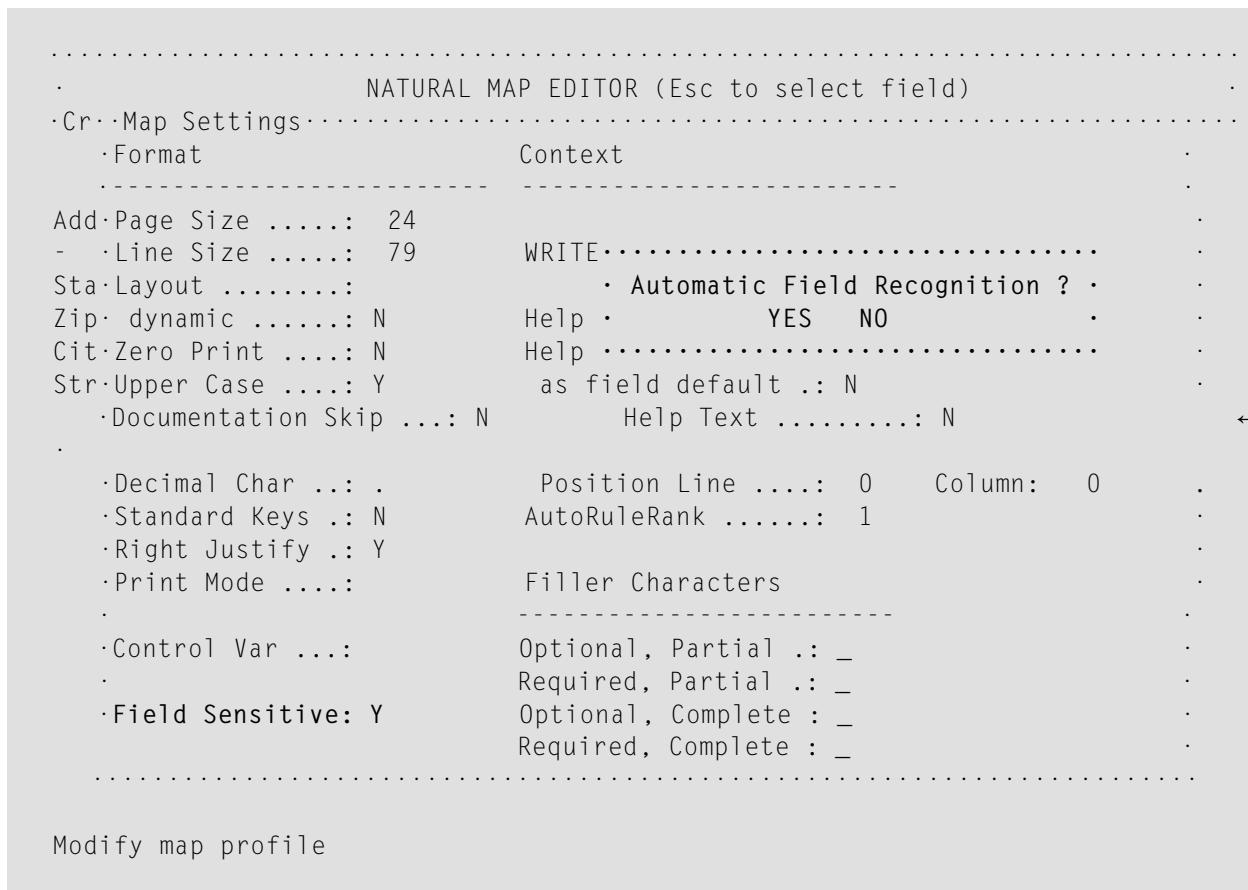
特定のフィールドに入力した値に応じて、プロンプトが表示されることがあります。これには例えば、要求した情報にアクセスするための特別なパスワードなどがあります。

迅速なデータ入力

上記で説明したすべてのメカニズムによって、より早く、より効率的にアプリケーションと対話できます。

Field-Sensitive としてマップを定義

フィールドに基づいた処理を使用できるようにするには、マッププロファイルで [Field Sensitive] を "Y" に設定し、フィールドに基づいてマップを処理するように定義する必要があります。フィールド感知での自動フィールド認識の有効または無効を確認するウィンドウが表示されます。



Natural マップ内のフィールド感知は、2 つの異なる方法で指定できます。

1. 自動フィールド認識あり。この場合は、適切な処理ルールを有効にするための条件をコーディングする必要がありません。



注意: このようなマップをメインフレームにアップロードした場合、Natural メインフレームマップエディタでは、新しい構文拡張のためにアップロードされたマップの構文を理解できません。

2. 自動フィールド認識なし。この場合は、最後のキー入力で離れたフィールドを通知する必要があります。Natural プログラムでシステム変数 *CURS-COL および *CURS-LINE を使用すると、このことを行うことができます。



注意: このようなマップをメインフレームにアップロードする場合は、メインフレーム上でマップを STOW すると、同じソースコードが生成されます。

IV DDM サービス

DDM サービスは、Natural データ定義モジュール (DDM) を作成、管理、および削除するためには使用します。

操作の原理

[DDM サービスの起動と終了](#)

[DDM のメンテナンス機能の使用](#)

[DDM の作成](#)

[DDM エディタの起動と終了](#)

[DDM エディタの使用](#)

[DDM の保存とカタログ](#)

[DDM のリスト](#)

[異なる環境での DDM の管理](#)

[Adabas または RDBMS のデータ変換](#)

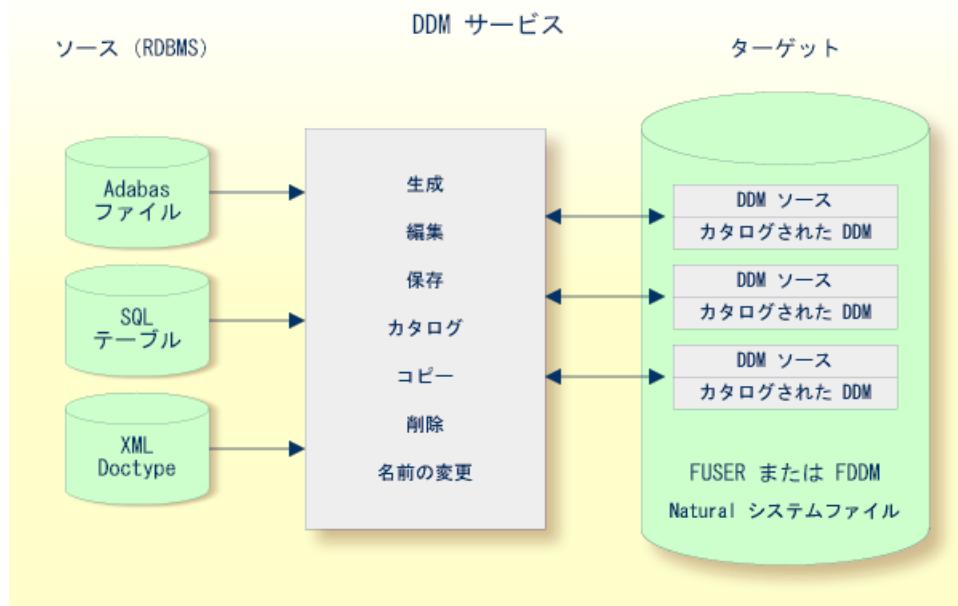
[Tamino のデータ変換](#)

17 操作の原理

■ DDM の保存 - FDDM システムファイル	162
■ Restrictions of Use	163

DDM サービスを使用して、データベースファイルまたは別の DDM から Natural DDM を作成します。プログラムやデータエリアなどの Natural オブジェクトは、データベースファイルに対し、対応する DDM がソースオブジェクトおよびカタログ化されたオブジェクトとして作成および保存されている場合にのみ、そのデータベースファイルにアクセスできます。Natural DDM の詳細については、『プログラミングガイド』の「データ定義モジュール - DDM」を参照してください。

次の図は、DDM サービスによる DDM 処理の主な機能と操作の基本原理を示しています。



DDM の保存 - FDDM システムファイル

DDM は、個別の Natural ユーザーライブラリまたはシステムライブラリのいずれかに存在します。システムライブラリは、システムファイル FUSER、FNAT、または FDDM に格納されています。FDDM システムファイルは、Natural システム環境内で有効なすべての DDM が格納される、独立したコンテナです。FDDM システムファイルは、NATPARM パラメータファイルに Natural プロファイルパラメータ FDDM を設定すると有効になります。

注意: FDDM システムファイルを有効にすると、FDDM システムファイル内の DDM の保存およびアクセスのみが実行可能になります。システムファイル FNAT または FUSER のライブラリに格納されている DDM にはアクセスできなくなります。

DDM は、Adabas データベースに保存されているファイルにも使用できます。『DDM サービス』ドキュメントに記載されているオプションには、Adabas のみに適用され、他のデータベース管理システムで使用しても無視されるものがあります。

関連トピック：

- 『オペレーション』ドキュメントの「システムファイル FDDM」
- 『パラメータリファレンス』ドキュメントの「FDDM-DDM 用の *Natural* システムファイル」

Restrictions of Use

The section below describes the restrictions that can apply to using DDM Services functions under:

- Predict
- Natural Security

Predict

To guarantee data integrity for DDMs defined in Predict, the Predict administrator can restrict the use of DDM Services functions for editing, copying, creating, deleting and renaming, for which equivalent functions are provided by Predict.

In principle, we strongly recommend that you do not use *any* DDM Services functions that can alternatively be performed by Predict.

For further information, contact your Predict system administrator.

Natural Security

Access to DDMs can be restricted when Natural Security has been installed. Within the DDM security profile, there may be a definition of whether a DDM may be modified only by specific users (DDM modifiers) or the owners of the security profile.

For further information, see *Protecting DDMs On UNIX, OpenVMS And Windows* in the *Natural Security* documentation.

18 DDM サービスの起動と終了

このセクションでは、DDM サービスを起動および終了する方法について説明します。

➤手順 18.1. DDM サービスを起動するには

- Natural メインメニューの **[Services]** を選択し、選択ウィンドウの **[DDM Services]** を選択するか、または「D」と入力します。

または:

Natural メインメニューの **[Direct]** を選択し、**[Direct Command]** ウィンドウで次のように入力します。

SYSDDM

以下の例に示すように、**[DDM Services]** 画面が表示されます。

```
2004-09-09          DDM Services          Library: SYSTEM
17:15:21          V 6.1.1 P1 9  Software AG 2004  DBID   :
User: SAG          FNR   :
+-----+
| Library      DDM Maintenance      Services Profile      Quit
|             +-----+
Logon to DDM library
```

[DDM Services] 画面で使用できるメニューについては、『DDM サービス』ドキュメントの該当セクションを参照してください。

➤手順 18.2. DDM サービスを終了するには

- [DDM Services] 画面で、[Quit] を選択します。

Natural メインメニューが表示されます。

19 DDM メンテナンス機能の使用

■ メンテナンス機能のリストと実行	168
■ メンテナンス機能の説明	169

DDM メンテナンスウィンドウの機能を使用して、DDM の作成（コピー）、表示、編集、名前の変更、削除を行います。

メンテナンス機能のリストと実行

このセクションでは、DDM メンテナンスウィンドウにリストされる DDM に対するメンテナンス機能をリストおよび実行する方法について説明します。

【<CREATE>】メニューのオプションについては、「[DDM の作成](#)」を参照してください。

手順 19.1. DDM のメンテナンス機能をリストおよび実行するには

- 1 DDM メンテナンスウィンドウ上で、処理する DDM にカーソルを移動して F2 キーを押します。

使用できる機能と対応するファンクションコードがリストされた選択ウィンドウが表示されます。

```
+-----+
| C Copy   |
| D Delete |
| E Edit   |
| L List   |
| N Rename |
+-----+
```

機能の詳細については、「[メンテナンス機能の説明](#)」を参照してください。

- 2 次のいずれかの方法を使用して、処理する DDM に対するファンクションコードを入力します。

■ 選択ウィンドウ上で、実行する機能に対応するファンクションコードにカーソルを移動して ENTER キーを押します。

■ または、

ENTER キーを押して選択ウィンドウを終了し、処理が必要な DDM の隣に有効なファンクションコードを入力して、再度 ENTER キーを押します。

DDM メンテナンスウィンドウ上の、処理対象の DDM の隣にファンクションコードが入力されます。

入力されたファンクションコードによって、対応する機能が即座に実行されたり、アクションの確認または DDM 名の入力を要求するウィンドウが表示されたりします。

また、複数のファンクションコードを指定することもできます。この場合、すべてのファンクションコードを入力して ENTER キーを押すと、1 つの DDM が順に処理されます。

機能を実行する方法の例については、「[DDM のコピー](#)」を参照してください。

メンテナンス機能の説明

DDM メンテナンスウィンドウの機能は、現在の Natural ライブラリおよび／またはシステムファイルに含まれている DDM に適用されます。

次の表は、DDM メンテナンスウィンドウで入力できる機能とファンクションコードの説明を示しています。

ファンクションコード	関数
C	DDM のコピー。 DDM のソースおよびカタログ化オブジェクトをコピーします。 Predict がインストールされている場合：FDIC システムファイルの DDM 用に XREF エントリが存在するのであれば、さらに XREF データをコピーします。
D	DDM の削除。 DDM のソースおよびカタログ化オブジェクトを削除します。
E	DDM の編集。 DDM エディタを起動し、DDM ソースを編集エリアに読み込みます。「 DDM エディタの使用 」も参照してください。
L	DDM のリスト。 [List] 画面を起動し、DDM ソースを表示します。この機能では、DDM エディタは起動されません。 [List] 画面に表示されるフィールド列については、「 フィールド属性列 」を参照してください。
R	DDM の名前の変更。 DDM のソースおよびカタログ化オブジェクトの名前を変更します。 Predict がインストールされている場合：FDIC システムファイルの DDM 用に XREF エントリが存在するのであれば、さらに XREF データをコピーします。 DDM の名前を変更するときに適用される命名規則については、『Natural の使用』ドキュメントの「オブジェクトの命名規則」を参照してください。

20 DDM の作成

■ DDM のコピー	172
■ [<CREATE>] を使用した、Adabas からの作成	174
■ [<CREATE>] を使用した、SQL からの作成	177
■ SQL からの複数の DDM の作成	181
■ [<CREATE>] を使用した、Tamino からの作成	182

このセクションでは、DDM をコピーするか、またはデータベースのフィールド定義から DDM を直接作成することによって、DDMを作成する方法を説明します。さらに、SQLデータベースから複数の DDM を生成する方法についての情報が提供されます。

DDM のコピー

このセクションでは、既存の DDM から新しい DDM を作成する方法について説明します。

ライブラリ間、データベースファイル間、またはハードウェアプラットフォーム間でDDMをコピーする場合は、「[異なる環境での DDM の管理](#)」も参照してください。

Predict のインストール

FDIC システムファイルに DDM に対する XREF エントリが存在する場合、それらのデータも新規 DDM のためにコピーされます。

▷手順 20.1. DDM をコピーするには

1 [DDM Services] 画面で、[DDM Maintenance] を選択します。

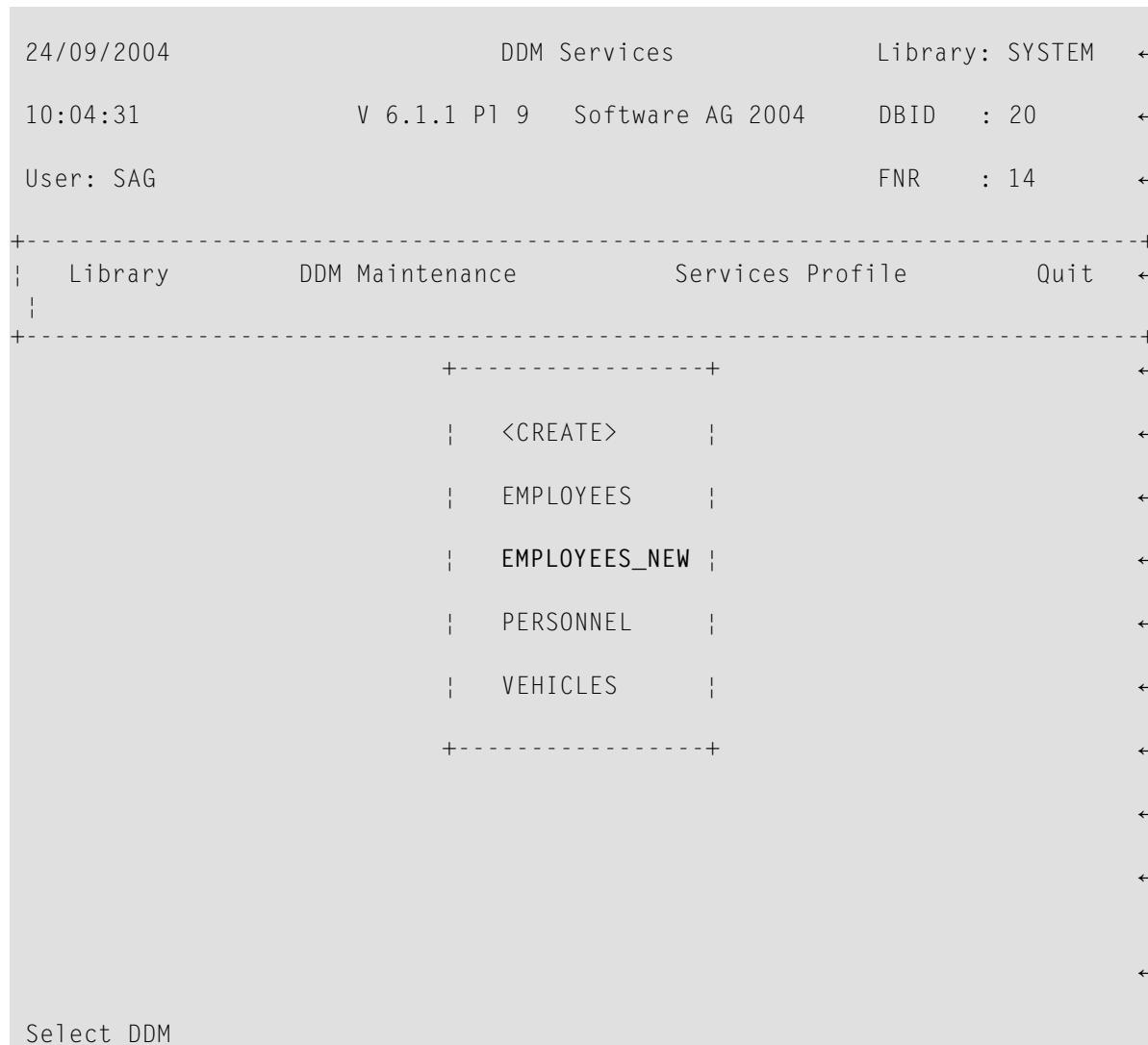
有効なすべての DDM がリストされた、DDM メンテナンスウィンドウが表示されます。

2 DDM メンテナンスウィンドウ上で、コピーする DDM にカーソルを移動し（次の例では EMPLOYEES）、ファンクションコード C を入力して ENTER キーを押します。

次の例のように、コピーする DDM の名前が事前設定された [Copy DDM to] ウィンドウが表示されます。

- 3 [Copy DDM to] ウィンドウ上で、表示されている名前を新しい名前に置き換え、ENTERキーを押します。

新しい DDM (ここではEMPLOYEES_NEW) がソースおよびカタログ化されたオブジェクトとして保存され、DDM のリストに追加されます。



[<CREATE>] を使用した、Adabas からの作成

このセクションでは、 [<CREATE>] メニューオプションを使用して、Adabas データベースから新しい DDM を作成する方法について説明します。

➤手順 20.2. Adabas データベースから DDM を作成するには

1 [DDM Services] 画面で、 [DDM Maintenance] を選択します。

DDM メンテナンスウィンドウが表示されます。

2 DDM メンテナンスウィンドウ上部の [<CREATE>] を選択します。

以下の例に示すように、 [Select Database] ウィンドウが表示されます。

```
24/09/2004          DDM Services          Library: SYSTEM ↵
10:04:31          V 6.1.1 P1 9  Software AG 2004  DBID   : ↵
User: SAG          FNR   : ↵
+-----+           ↵
| Library      DDM Maintenance      Services Profile      Quit ↵
|           ↵
+-----+           ↵
|           <CREATE>           |           ↵
+-----+ Select Database +-----+           ↵
| DBID      1           |           ↵
+-----+           ↵
|           VEHICLES           |           ↵
+-----+           ↵
Enter database number (0 - 65535 except 255) ↵
```

- 3 DDMを作成する Adabas ファイルのデータベース ID (DBID) を入力します。有効な値は 0~65535 (255を除く) です。0 (ゼロ) を入力した場合は、NATPARM パラメータファイルの Natural プロファイルパラメータ UDB (『パラメータリファレンス』ドキュメントを参照) で指定されているデータベース ID が使用されます。

指定したDBIDによって Adabas データベースが識別されると、 [Create ADABAS DDM] ウィンドウが表示されます。

DDM の作成

- 4 DDM を作成するデータベースファイルのファイル番号 (FNR) を入力し (有効な値: 1-5000) DDM に割り当てる名前を入力します。Adabas ファイルへのアクセスが保護されている場合、パスワードの入力が要求されます。
 - 5 ENTER キーを押します。

指定したデータベースとファイルが使用可能な場合は、DDMエディタが起動し、そのデータベースファイルに含まれているフィールドが編集エリアに読み込まれます。

If the specified database is not active or cannot be accessed or if the file does not exist, a corresponding error message is issued. Nevertheless, if you press **ENTER**, you can still open

an empty DDM editor screen, enter new field attribute definitions and save the DDM source. However, in this case you cannot check any definitions against the database file description.

- 6 必要に応じて、DDM を編集します。 「[DDM エディタの使用](#)」 を参照してください。
- 7 編集後、ESC キーを押して、 [MISC] メニューから [EXIT (with STOW)] を選択します。 「[DDM の保存とカタログ](#)」 も参照してください。

DDM ソースの構文がチェックされ、DDM はソースおよびカタログ化オブジェクトとして保存されます。

[<CREATE>] を使用した、SQL からの作成

このセクションでは、 [<CREATE>] メニューオプションを使用して、SQL データベースから DDM を作成する方法について説明します。

➤手順 20.3. SQL データベースから DDM を作成するには

- 1 [DDM Services] 画面で、 [DDM Maintenance] を選択します。
DDM メンテナンスウィンドウが表示されます。
- 2 DDM メンテナンスウィンドウ上部の [<CREATE>] を選択します。
以下の例に示すように、 [Select Database] ウィンドウが表示されます。

DDM の作成

- 3 DDMを作成するSQLテーブルのデータベースID（DBID）を入力します。有効な値は0～65535（255を除く）です。0（ゼロ）を入力した場合は、NATPARMパラメータファイルのNaturalプロファイルパラメータ UDB（『パラメータリファレンス』ドキュメントを参照）で指定されているデータベースIDが使用されます。

指定した DBID によって SQL データベースが識別されると、[Select SQL Table] ウィンドウが表示されます。

```

04/10/2004          DDM Services          Library: SYSTEM
10:49:05          V 6.1.1 P1 9  Software AG 2004  DBID  :
User: SAG          FNR  :
+-----+
| Library      DDM Maintenance      Services Profile      Quit  |
|           |
+-----+
|           | <CREATE> |           |
|           | EMPLOYEES |           |
|           | PERSONNEL |           |
+-----+ Select SQL Table +-----+
| Table Owner: * |           |
| Table Name : * |           |
+-----+
Enter SQL database Table Owner (case sensitive)

```

- 4 DDM を作成するテーブルのオーナー名とテーブル名を入力します。

範囲を指定する場合：すべてのテーブルをリストするにはデフォルト設定のアスタリスク (*) を、特定のテーブルをリストするにはアスタリスク (*) 表記を使用します。例えば、AB* と指定すると、名前が AB で始まるすべての SQL テーブルが選択されます。

SQL データベースの設定に応じて、セッション内で最初にこの SQL データベースにアクセスすると、**Database Logon** が表示されます。

データベースに指定されたユーザーIDおよびパスワードを入力し、ENTERキーを押します。

指定したテーブルが存在する場合、そのテーブルに対する DDM が作成されます。

該当するテーブルが存在しない、あるいはテーブルオーナーおよび／またはテーブル名を範囲指定した場合、次の例のように、必要なSQLテーブルを選択できる [Import SQL Table Contents] ウィンドウが表示されます。

```
04/10/2004          DDM Services          Library: SYSTEM ←
11:18:15          V 6.1.1 P1 9  Software AG 2004  DBID   : ←
User: SAG          FNR   : ←
+-----+          Library      DDM Maintenance      Services Profile      Quit ←
|       |          +-----+ ←
+-----+          +-----+ Import SQL Table Contents +-----+ ←
| Table Owner          Table Name          Type | ←
| ======+ ←
| DEMO          AUTOMOBILES          T | ←
| DEMO          EMPLOYEES          T | ←
| DEMO          SALARY          T | ←
| SAG          YACHT          T | ←
| SAG          ALldata          T | ←
+-----+ ←
Select SQL table
```



注意: ODBC インターフェイス経由でアクセスされている SQL データベースには、テーブルカタログは表示されません。この場合、テーブル名のみ指定できます。テーブルオーナーは指定できません。

5 ENTERキーを押します。

DDM エディタが起動し、選択されたテーブルから生成された DDM が編集エリアに読み込まれます。

DDMに対して名前が自動的に生成されます。これはテーブルオーナーとテーブル名の組み合わせで、変更できません。例えば、テーブルオーナーの名前が SAG で、テーブル名が TEST の場合、DDM 名は SAG-TEST です。

If the specified database is not active or cannot be accessed or if the file does not exist, a corresponding error message is issued. Nevertheless, if you press ENTER, you can still open an empty DDM editor screen, enter new field attribute definitions and save the DDM source. However, in this case you cannot check any definitions against the database file description.

- 6 必要に応じて、DDM を編集します。 「[DDM エディタの使用](#)」を参照してください。
- 7 編集後、ESC キーを押して、 [MISC] メニューから [EXIT (with STOW)] を選択します。 「[DDM の保存とカタログ](#)」も参照してください。

DDM ソースの構文がチェックされ、DDM はソースおよびカタログ化オブジェクトとして保存されます。

SQL からの複数の DDM の作成

The Natural program DDMGEN (supplied in the Natural system library SYSTEM) provides the option to generate multiple DDMs simultaneously from SQL tables without using the DDM editor.

➤手順 20.4. To execute DDMGEN

- 1 Enter the following direct command:

DDMGEN

The **SQL DDM Generation** screen appears where you can fill the fields required to generate a DDM from an SQL table as demonstrated in the example below:

```
SQL DDM Generation
=====
DDM Library  : DDMTEST
DDM DBID     : 210
Table Owner  : QA*
Table Name   : *
Replace (Y/N): N
```

Enter the name of the library where you want to create the DDMs and enter the name of a table and/or specify a range as described in the previous section.

- 2 Press ENTER to execute the program.

Status messages appear at the bottom of the screen that indicate which DDM is generated from which SQL table. The DDMs generated are saved as source and cataloged objects in the specified library.

[<CREATE>] を使用した、Tamino からの作成

このセクションでは、 [<CREATE>] メニューオプションを使用して、Tamino データベースから DDM を作成する方法について説明します。

➤手順 20.5. Tamino データベースから DDM を作成するには

- 1 [DDM Services] 画面で、 [DDM Maintenance] を選択します。
DDM メンテナンスウィンドウが表示されます。
- 2 DDM メンテナンスウィンドウ上部の [<CREATE>] を選択します。
以下の例に示すように、 [Select Database] ウィンドウが表示されます。

```
01/10/2004          DDM Services          Library: SYSTEM
15:50:58          V 6.1.1 Pl 9 Software AG 2004  DBID   :
User: SAG          FNR    :  
+-----+
| Library      DDM Maintenance      Services Profile      Quit
|  
+-----+
|           <CREATE>           |
+----- Select Database -----+
| DBID      1                  |
+-----+
|      EMPLOYEES_NEW  |
|      PERSONNEL        |
|      VEHICLES         |
+-----+
Enter database number (0 - 65535 except 255)
```

- 3 DDM を作成する Tamino doctype のデータベース ID (DBID) を入力します。

有効な値は 0～65535 (255 を除く) です。0 (ゼロ) を入力した場合は、NATPARM パラメータファイルの Natural プロファイルパラメータ UDB (『パラメータリファレンス』ドキュメントを参照) で指定されているデータベース ID が使用されます。

指定したDBIDによってTamino データベースが識別されると、 [Create XML DDM] ウィンドウが表示されます。

```
01/10/2004          DDM Services          Library: SYSTEM ↵
15:50:58          V 6.1.1 Pl 9  Software AG 2004  DBID   : ↵
User: SAG          FNR   : ↵
+-----+
| Library       DDM Maintenance       Services Profile       Quit ↵
| ↵
+-----+
+-----+
|   <CREATE>   | ↵
+-----+ Create XML DDM +-----+
| FNR       1       | ↵
| DDM Name | ↵
+-----+
|   PERSONNEL   | ↵
|   VEHICLES    | ↵
+-----+
```

4 DDM に割り当てる名前を入力します。

DDM のファイル番号 (FNR) は常に 1 です。ファイル番号は変更できません。

5 ENTER キーを押します。

次の例のように、doctype がリストされた [Select Doctype] ウィンドウが表示されます。

```

01/10/2004          DDM Services          Library: SYSTEM  ↵
10:55:38          V 6.1.1 P1 9  Software AG 2004  DBID  :  ↵
User: SAG          FNR  :  ↵
+-----+
| Library          DDM Maintenance      Services Profile      Quit  ↵
| ↵
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Doctype          Collection          Schema          |  ↵
| ======+=====+=====+=====+=====+=====+=====+=====+=====+  ↵
| SimpleContent    NATDemoData1        AttributeTest    |  ↵
| SimpleContentArray NATDemoData1        AttributeTest    |  ↵
| ComplexContent   NATDemoData1        AttributeTest    |  ↵
| DataType          NATDemoData1        DataType        |  ↵
| DataTypeDerived  NATDemoData1        DataTypeDerived  |  ↵
| Employee          NATDemoData1        Employee        |  ↵
| patient           NATDemoData1        patient        |  ↵
| NatArray           NATDemoData1        NatArray        |  ↵
| Array_Dim3        NATDemoData1        StoreMultiple  |  ↵
| Array_Dim2        NATDemoData1        StoreMultiple  |  ↵
| StoreTestSchema3  NATDemoData1        StoreMultiple  |  ↵
| StoreTestSchema4  NATDemoData1        StoreMultiple  |  ↵
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Select Doctype
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Select Doctype
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

6 リストから doctype を選択して ENTER キーを押します。

DDM エディタが起動し、選択された doctype から生成された DDM が編集エリアに読み込まれます。

If the specified database is not active or cannot be accessed or if the file does not exist, a corresponding error message is issued. Nevertheless, if you press ENTER, you can still open an empty DDM editor screen, enter new field attribute definitions and save the DDM source. However, in this case you cannot check any definitions against the database file description.

- 7 必要に応じて、DDM ソースを編集します。セクション「[DDM エディタ画面の使用](#)」を参照してください。
- 8 編集後、ESC キーを押して、 [MISC] メニューから [EXIT (with STOW)] を選択します。

DDM ソースの構文がチェックされ、DDM はソースおよびカタログ化オブジェクトとして保存されます。「[DDM の保存とカタログ](#)」も参照してください。

21 DDM エディタの起動と終了

■ DDM メンテナンスを使用したエディタの起動	188
■ EDIT によるエディタの起動	189
■ エディタの終了	190

DDM エディタは、DDM のソースを編集するために使用します。

このセクションでは、DDM サービスの **DDM Maintenance** メニュー、または Natural のシステムコマンド **EDIT** のいずれかを使用して、既存の DDM エディタを起動および終了する方法について説明します。

DDM メンテナンスを使用したエディタの起動

›手順 21.1. DDM メンテナンスから DDM エディタを起動するには

- DDM メンテナンスウィンドウ上で、編集する DDM の隣にファンクションコード **E** を入力します。

指定された DDM のソースコードが DDM エディタの編集エリアに読み込まれます。

指定された DDM によって参照される Adabas データベースまたは SQL データベースが有効で、かつ指定されたファイル番号のデータベースファイルが存在する場合、次の例のような画面が表示されます。そうでない場合は、代わりに空の画面が表示されます。

```
17/09/2004          DDM Services
12:44:43          V 6.1.1 P1 9 Software AG 2004          Line: 1
DBID: 20          FNR: 14          DDM: EMPLOYEES          DEF.SEQ.:
C   T          L Name          F  Length  S  D
          1 PERSONNEL-ID          A     8          D
          * CNNNNNNNN
          G 1 FULL-NAME          F  Length  S  D
          2 FIRST-NAME          A     20          N
          2 MIDDLE-I          A     1          N
          2 NAME          A     20          D
          1 MIDDLE-NAME          A     20          N
          1 MAR-STAT          A     1          F
          * M=MARRIED          A     1          F
          1 SEX          A     1          F
          1 BIRTH          N     06.0          D
          1 N@BIRTH          I     2          D
          G 1 FULL-ADDRESS          F  Length  S  D
          M 2 ADDRESS-LINE          A     20          N
          2 CITY          A     20          N  D
          2 ZIP          A     10          N
          2 POST-CODE          A     10          N
          2 COUNTRY          A     3          N
F1  HELP  F2 CHOICE F3  STOW+EXIT          F10 STOW          F11 CHECK
F12 DB-SHORT-NAMES  F13 MODIFY HEADER          F14 SHOW EXT FIELD  F15
```

指定された DDM によって参照される Tamino データベースが有効で、指定されたファイル番号のデータベースファイルが存在する場合、次の例のような画面が表示されます。そうでない場合は、代わりに空の画面が表示されます。

```

01/10/2004          DDM Services
11:05:45          V 6.1.1 P1 9 Software AG 2004          Line: 1
DBID: 175  FNR: 1  DDM: DDM_TEST          TYPE: XML
C   T   L   Name          F          Length  D
G   1   EMPLOYEE
G   2   GROUP$1
      3   PERSONNEL-ID          A          8  D
G   2   GROUP$2
G   3   FULL-NAME
G   4   GROUP$3
      5   FIRST-NAME          A          20  D
      5   MIDDLE-NAME          A          20  D
      5   MIDDLE-I          A          20  D
      5   NAME          A          20  D
      3   MAR-STAT          A          1  D
      3   SEX          A          1  D
      3   BIRTH          A          10 D
G   3   FULL-ADDRESS
G   4   GROUP$4
      5   ADDRESS-LINE          A          20
      5   CITY          A          20  D
      5   ZIP          A          20  D
F1  HELP  F2  CHOICE F3  STOW+EXIT          F10 STOW          F11 CHECK
F12 DOCTYPE INFO          F13 MODIFY HEADER          F14 SHOW EXT FIELD          F15

```

DDMエディタで使用できるフィールドとコマンドの詳細については、「[DDMエディタの使用](#)」を参照してください。

EDITによるエディタの起動

DDMサービスの機能を使用する代わりに、Naturalのシステムコマンド `EDIT` を使用して DDM エディタを起動できます。

➤手順 21.2. EDIT を使って DDM エディタを起動するには

- Natural メインメニューの **[Direct]** を選択し、**[Direct Command]** ウィンドウで次のように入力します。

```
EDIT VIEW object-name
```

object-name は編集対象の DDM の名前です。

指定した DDM について DDM エディタが起動し、DDM ソースが編集エリアに読み込まれます。

EDIT で使用可能なすべてのオプションの詳細については、『システムコマンド』ドキュメントで関連するセクションを参照してください。

エディタの終了

このセクションでは、エディタセッションを終了して DDM メンテナنسウィンドウに戻る方法について説明します。

›手順 21.3. DDM エディタを終了するには

- 1 編集が終わったら、DDM ソースを保存してカタログし（「[DDM エディタの使用](#)」および「[DDM の保存とカタログ](#)」を参照）、DDM エディタの画面で ESC キーを押します。

次の例のように、DDM エディタメニューの **[COMMANDS]** 、 **[MISC]** 、および **[QUIT]** （コマンドに関するセクションも参照）が DDM エディタ画面上部に表示されます。

```
COMMANDS          MISC          QUIT
14:22:37          V 6.1.1 P1 7 Software AG 2004          Line: 1
DBID: 20          FNR: 14 DDM: EMPLOYEES          DEF. SEQ.:
C   T          L Name          F  Length  S  D
          1 PERSONNEL-ID          A      8      D
*          CNNNNNNNN
G          1 FULL-NAME          F  Length  S  D
          2 FIRST-NAME          A      20     N
          2 MIDDLE-I          A      1      N
          2 NAME          A      20     D
          1 MIDDLE-NAME          A      20     N
          1 MAR-STAT          A      1      F
*          M=MARRIED
          1 SEX          A      1      F
          1 BIRTH          N      06.0     D
          1 N@BIRTH          I      2      D
G          1 FULL-ADDRESS
M          2 ADDRESS-LINE          A      20     N
          2 CITY          A      20     N  D
          2 ZIP          A      10     N
          2 POST-CODE          A      10     N
          2 COUNTRY          A      3      N
F1  HELP  F2  CHOICE F3  STOW+EXIT          F10 STOW          F11 CHECK
Select a Command
```

- 2 「[QUIT] メニュー」の説明に従って [QUIT] メニューの [EXIT (with STOW)] または [QUIT (without STOW)] を選択し、DDM ソースの処理を終了します。

DDM メンテナンスウィンドウが表示されます。

22 DDM エディタの使用

■ DDM ヘッダー情報	194
■ フィールド属性列	197
■ 編集および機能実行のためのコマンド	202
■ 拡張フィールド属性の指定	208
■ エディタの設定 - サービスプロファイル	214

DDM エディタ画面はテーブルで構成されており、フィールド定義データは行および列に含まれています。DDM に定義されているフィールドに属するすべての属性は、1 行（ソースコード行）に含まれており、タブで区切られています。

このセクションでは、DDM エディタ画面の列について説明し、DDM フィールドの作成と変更、画面内の移動、DDM ソースの作成などを実行できるコマンドについて説明します。

DDM ヘッダー情報

このセクションでは、DDM エディタ画面上部のヘッダーに含まれるフィールドとその変更方法について説明します。また、Tamino 固有の doctype 情報を読み取り専用で表示できます。

- [DDM ヘッダーフィールドの説明](#)
- [Tamino の doctype 情報の表示](#)
- [DDM ヘッダーフィールドの変更](#)

DDM ヘッダーフィールドの説明

次の表は、DDM ヘッダーに含まれるフィールドとその説明を示しています。Tamino 固有の doctype フィールドについては、『プログラミングガイド』の「*Tamino XML スキーマ言語について*」も参照してください。

ヘッダーフィールド	説明
DBID	グローバルコンフィグレーションファイルで指定されているデータベース ID (DBID) です。DBID には、DDM によって参照されるデータベースファイルが含まれています。 有効範囲：0～65535 (255 を除く) 以下の項目も参照してください。『コンフィグレーションユーティリティ』ドキュメントの「DBMS 割り当て」および「データベース管理」 0 (ゼロ) を指定すると、NATPARM パラメータファイル内の UDB プロファイルパラメータで指定されているデフォルトの DBID が使用されます。 フィールドの内容を変更する方法については、「 DDM ヘッダーフィールドの変更 」を参照してください。
FNR	データベースで参照されるファイル番号です。 Tamino からの DDM のファイル番号は、常に 1 となります。この値は変更できません。 有効範囲：1～5000 フィールドの内容を変更する方法については、「 DDM ヘッダーフィールドの変更 」を参照してください。
DDM	DDM エディタのワークエリアに現在含まれている DDM の名前です。

ヘッダー フィールド	説明
Line	現在カーソルが置かれているソースコード行の番号です。
DEF. SEQ.	<p>Tamino では使用できません。</p> <p>Natural プログラムの READ LOGICAL ステートメントを使用してアクセスするときの、ファイル読み込みのデフォルト順序です。『ステートメント』ドキュメントの READ ステートメントの説明も参照してください。</p> <p>デフォルト順序は、2 文字のフィールドのショートネームを使用して指定します。ショートネームは、選択したファイル番号に基づいて検証されます。データベースがアクセス可能な場合、データベースファイルの対応するフィールドに対し、ショートネームがチェックされます。該当するフィールドがデータベースに存在しない場合、有効なショートネームの選択リストが表示されます。データベースにアクセスできない場合、選択リストは生成されません。</p> <p>フィールドの内容を変更する方法については、「DDM ヘッダーフィールドの変更」を参照してください。</p>
TYPE	Tamino データベースから作成された DDM の場合は、タイプ XML が表示されます。
Collection	<p>読み取り専用の Tamino 固有 doctype 情報です。</p> <p>Tamino データベース内で使用されているコレクションの名前</p> <p>doctype 情報を表示するには、「Tamino の doctype 情報の表示」を参照してください。</p>
Schema	<p>読み取り専用の Tamino 固有 doctype 情報です。</p> <p>Tamino データベース内で使用されている Tamino XML スキーマの名前</p> <p>doctype 情報を表示するには、「Tamino の doctype 情報の表示」を参照してください。</p>
Doctype	<p>読み取り専用の Tamino 固有 doctype 情報です。</p> <p>コレクション内の doctype の名前</p> <p>doctype 情報を表示するには、「Tamino の doctype 情報の表示」を参照してください。</p>
Namespace URI Prefix	<p>読み取り専用の Tamino 固有 doctype 情報です。</p> <p>doctype に対応する、名前空間 URI／接頭辞の組み合わせのリスト</p> <p>doctype 情報を表示するには、「Tamino の doctype 情報の表示」を参照してください。</p>

Tamino の doctype 情報の表示

➤手順 22.1. Tamino 固有の doctype 情報の表示／非表示を切り替えるには

- F12 キーを押します（トグル切り替え）。

または:

[MISC] メニューで、 [SHOW DOCTYPE INFO ON/OFF] を選択します。

表示設定になっている場合（デフォルト設定は非表示）、次の例のように、DDMエディタ画面の中央下部に [Doctype Information] セクションが表示されます。

```

01/10/2004          DDM Services
11:05:45          V 6.1.1 P1 9 Software AG 2004      Line: 1
DBID: 175  FNR: 1  DDM: DDM_TEST          TYPE: XML
  C  T  L  Name          F      Length  D
  G  1  EMPLOYEE
  G  2  GROUP$1
  G  3  PERSONNEL-ID      A          8  D
  G  2  GROUP$2
  G  3  FULL-NAME
  G  4  GROUP$3
  G  5  FIRST-NAME      A          20  D
  G  5  MIDDLE-NAME      A          20  D
~~~~~ Doctype Information ~~~~~
Collection: NATDemoData1
Schema    : Employee
Doctype   : Employee

Namespace URI          Prefix
-----
http://www.w3.org/2001/XMLSchema          xs

F1  HELP  F2  CHOICE  F3  STOW+EXIT      F10  STOW          F11  CHECK
F12  DOCTYPE  INFO      F13  MODIFY  HEADER  F14  SHOW  EXT  FIELD  F15

```

DDM ヘッダーフィールドの変更

➤手順 22.2. DDM ヘッダーフィールドの内容を変更するには (Tamino 固有の情報は読み取り専用です)

1 F13 キーを押します。

または:

[MISC] メニューで、[MODIFY DDM-HEADER] を選択します。

変更可能な最初のヘッダーフィールド (DBID) にカーソルが移動します。

2 TAB キーを押すと、変更可能な次のヘッダーフィールドに移動します。

フィールド属性列

このセクションでは、DDMエディタ画面の行や列に定義できるフィールド属性について説明します。

列見出し	フィールド属性
T	<p>フィールドのタイプ。</p> <p>空白 エレメンタリフィールド。 このタイプのフィールドは、データを保持できますが、他のフィールドを含むことはできません。 レコード内に値を1つのみ持つことができます。</p>
G	<p>グループ。</p> <p>グループは、共通のグループ名を使用して定義されたフィールドの集まりです。これにより、個別のフィールドすべての名前ではなく、グループ名を使用して、複数のフィールドをまとめて参照することができます。このようなフィールドはデータを保持することができず、他のフィールドのコンテナに過ぎません。</p> <p>注意: DDM に定義するグループは、必ずしもその DDM が参照する Natural オブジェクトのグループとして定義されている必要はありません。</p>
M	<p>Tamino では使用できません。</p> <p>マルチプルバリューフィールド。</p> <p>このタイプのフィールドは、レコード内に複数の値を持つことができます。 『プログラミングガイド』の「マルチプルバリューフィールド」も参照してください。</p>

列見出し	フィールド属性	
P		
	P	<p>Tamino では使用できません。 ピリオディックグループ。 1 つのレコードに複数の値を含めることのできるフィールドのグループ。 『プログラミングガイド』の「ピリオディックグループ」も参照してください。</p>
*		コメント行。
L	<p>フィールドに割り当てるレベル番号。</p> <p>レベルは、フィールド定義の構造およびグループ化を示すために使用します。このレベルは、ビューディクタ、再定義、およびフィールドグループのレベルと対応しています（『プログラミングガイド』の該当するセクションを参照）。</p> <p>有効なレベル番号は 1~7 です。</p> <p>Tamino の場合：有効なレベル番号は 1~99 です。</p> <p>レベル番号は、連続した昇順で指定する必要があります。</p>	
DB	<p>Tamino では使用できません。</p> <p>デフォルトでは、[DB] 列は非表示に設定されています。表示／非表示を切り替えるには、F12 キーを押します（トグル切り替え）。</p> <p>[DB] 列には、データベースファイル内の対応するフィールドの 2 文字のショートネームが表示されます（「[DB] 列の例」も参照）。</p> <p>フィールドの作成：</p> <p>新しい DDM フィールドを作成したときに [DB] 列が非表示に設定されている場合、DDM エディタによって、まだ別のフィールドで使用されていないショートネームが新しいフィールドに割り当てられます。これは、新しいフィールドについては、データベースファイルと DDM の間に相関性がないことを意味します。新しいフィールドのショートネームが確実にデータベースでチェックされるようにするには、「編集および機能実行のためのコマンド」の説明に従って、行コマンド I を使用してフィールドを作成します。</p>	
Name	<p>フィールドの名前。</p> <p>3~32 文字（Adabas のフィールドの場合）および 1~32 文字（SQL の列および Tamino doctype の場合）になります。</p> <p>名前を作成する規則はユーザ一定義変数の命名規則に準拠します（『Natural の使用』ドキュメントを参照）。ただし、名前の最初の文字は常に大文字のラテン文字（A~Z）にする必要があります。また、名前を L@ または N@ を持つ文字です。これらの接頭辞を使用して、次のセクションの説明のように、インジケータフィールド が識別されます。</p> <p>フィールド名は、別の Natural オブジェクト（プログラムなど）でフィールドを参照するために使用される名前です。</p>	

列見出し	フィールド属性								
	<p>フィールド名は、DDM 全体で一意です。</p> <p>Tamino の場合、フィールド名は必ずしも Tag Name と同じ名前である必要はありません（「<i>Tamino 固有の拡張フィールド属性</i>」を参照）。</p>								
F	<p>A (英数字) 、P (パック型数値) 、L (論理) などの、エレメンタリフィールドの Natural データフォーマットです。</p> <p>有効な Natural データフォーマットについては、『プログラミングガイド』の「ユーザー定義変数のフォーマットおよび長さ」を参照してください。</p>								
Length	<p>エレメンタリフィールドの標準長。</p> <p>この長さは、Natural プログラムで上書きできます。</p> <p>数値フィールド (Natural データフォーマット N) の場合、<i>nn.m</i> の形式で長さを指定します。<i>nn</i> は小数点の前の桁数、<i>m</i> は小数点の後の桁数を表します。</p> <p>〔Length〕 入力フィールドには、数値としてフィールド長を指定することも、フィールド長が変数であることを示すキーワード DYNAMIC を入力することもできます。</p> <p>詳細については、『プログラミングガイド』の「可変長の列に対する DDM の生成および編集」を参照してください。</p>								
S	<p>Tamino では使用できません。</p> <p>空値抑制オプション。</p> <table border="1"> <tr> <td>空白</td><td>Adabas の標準の抑制が使用されることを示します。英数字フィールドの末尾の空白、および数値フィールドの先行ゼロが抑制されます。</td></tr> <tr> <td>F</td><td>Adabas の固定ストレージオプションを使用してフィールドが定義されていることを示します。つまり、抑制は行われず、フィールドは圧縮なしで保存されます。</td></tr> <tr> <td>N</td><td>Adabas の空値抑制オプションを使用してフィールドが定義されていることを示します。これは、フィールドの空値がインバーテッドリストに保存されないため、FIND ステートメントの WITH 節、HISTOGRAM ステートメント、または READ LOGICAL ステートメントでフィールドが使用されたときに空値が返されないことを意味します。</td></tr> <tr> <td>M</td><td>SQL の空値オプション not null を使用してフィールドが定義されていることを示します。このフィールドの [Remark] フィールド（「<i>拡張フィールド属性の指定</i>」を参照）には、NN NC (not null, not counted) が使用されます。このフィールドの下には、対応する空値インジケータフィールドが表示されます。</td></tr> </table>	空白	Adabas の標準の抑制が使用されることを示します。英数字フィールドの末尾の空白、および数値フィールドの先行ゼロが抑制されます。	F	Adabas の固定ストレージオプションを使用してフィールドが定義されていることを示します。つまり、抑制は行われず、フィールドは圧縮なしで保存されます。	N	Adabas の空値抑制オプションを使用してフィールドが定義されていることを示します。これは、フィールドの空値がインバーテッドリストに保存されないため、FIND ステートメントの WITH 節、HISTOGRAM ステートメント、または READ LOGICAL ステートメントでフィールドが使用されたときに空値が返されないことを意味します。	M	SQL の空値オプション not null を使用してフィールドが定義されていることを示します。このフィールドの [Remark] フィールド（「 <i>拡張フィールド属性の指定</i> 」を参照）には、NN NC (not null, not counted) が使用されます。このフィールドの下には、対応する空値インジケータフィールドが表示されます。
空白	Adabas の標準の抑制が使用されることを示します。英数字フィールドの末尾の空白、および数値フィールドの先行ゼロが抑制されます。								
F	Adabas の固定ストレージオプションを使用してフィールドが定義されていることを示します。つまり、抑制は行われず、フィールドは圧縮なしで保存されます。								
N	Adabas の空値抑制オプションを使用してフィールドが定義されていることを示します。これは、フィールドの空値がインバーテッドリストに保存されないため、FIND ステートメントの WITH 節、HISTOGRAM ステートメント、または READ LOGICAL ステートメントでフィールドが使用されたときに空値が返されないことを意味します。								
M	SQL の空値オプション not null を使用してフィールドが定義されていることを示します。このフィールドの [Remark] フィールド（「 <i>拡張フィールド属性の指定</i> 」を参照）には、NN NC (not null, not counted) が使用されます。このフィールドの下には、対応する空値インジケータフィールドが表示されます。								

列見出し	フィールド属性										
D	<p>配列ではないエレメンタリフィールドの Adabas ディスクリプタタイプ。</p> <p>ディスクリプタは、READ ステートメントまたは FIND ステートメントを使用して実行するデータベース検索の基準として使用できます。例えば、[D] 列が [D] または [S] の Adabas データベースのフィールドは、READ ステートメントの BY 節に使用できます。READ ステートメントを使用してデータベースからレコードが読み込まれると、この列が [D] または [S] であるすべてのフィールドを DISPLAY ステートメントで参照できます。</p> <p>Tamino XML スキーマでは、全体の多重度が最大 1 の場合、つまり要素およびスキーマ内の先行要素の maxOccurs の値が 1 を超えない場合、DDM のディスクリプタとしてその要素はマークされます。</p> <p>ディスクリプタタイプは次のとおりです。</p> <table border="1"> <tr> <td>空白</td><td>非ディスクリプタ。 このフィールドはディスクリプタではありません。</td></tr> <tr> <td>D</td><td>エレメンタリディスクリプタ。 FIND ステートメントの検索条件および FIND ステートメントのソートキーとして使用する、または READ ステートメントでの論理順読み込みを制御するために、このフィールドの値リストが Adabas によって作成および管理されます。</td></tr> <tr> <td>H</td><td>Tamino では使用できません。 ハイパーディスクリプタ。 ハイパーディスクリプタは Adabas のユーザー出口です。Natural では、フォネティックディスクリプタと同じ機能を提供します（下記参照）。</td></tr> <tr> <td>N</td><td>Tamino では使用できません。 非ディスクリプタ。 非ディスクリプタはディスクリプタではありませんが、非ディスクリプタ検索で検索フィールドとして使用できます。</td></tr> <tr> <td>P</td><td>Tamino では使用できません。 フォネティックディスクリプタ。 フォネティックディスクリプタによって、フィールドのフォネティック検索（人の名前など）を実行できます。フォネティック検索では、発音が検索値に類似するすべての値が返されます。</td></tr> </table>	空白	非ディスクリプタ。 このフィールドはディスクリプタではありません。	D	エレメンタリディスクリプタ。 FIND ステートメントの検索条件および FIND ステートメントのソートキーとして使用する、または READ ステートメントでの論理順読み込みを制御するために、このフィールドの値リストが Adabas によって作成および管理されます。	H	Tamino では使用できません。 ハイパーディスクリプタ。 ハイパーディスクリプタは Adabas のユーザー出口です。Natural では、フォネティックディスクリプタと同じ機能を提供します（下記参照）。	N	Tamino では使用できません。 非ディスクリプタ。 非ディスクリプタはディスクリプタではありませんが、非ディスクリプタ検索で検索フィールドとして使用できます。	P	Tamino では使用できません。 フォネティックディスクリプタ。 フォネティックディスクリプタによって、フィールドのフォネティック検索（人の名前など）を実行できます。フォネティック検索では、発音が検索値に類似するすべての値が返されます。
空白	非ディスクリプタ。 このフィールドはディスクリプタではありません。										
D	エレメンタリディスクリプタ。 FIND ステートメントの検索条件および FIND ステートメントのソートキーとして使用する、または READ ステートメントでの論理順読み込みを制御するために、このフィールドの値リストが Adabas によって作成および管理されます。										
H	Tamino では使用できません。 ハイパーディスクリプタ。 ハイパーディスクリプタは Adabas のユーザー出口です。Natural では、フォネティックディスクリプタと同じ機能を提供します（下記参照）。										
N	Tamino では使用できません。 非ディスクリプタ。 非ディスクリプタはディスクリプタではありませんが、非ディスクリプタ検索で検索フィールドとして使用できます。										
P	Tamino では使用できません。 フォネティックディスクリプタ。 フォネティックディスクリプタによって、フィールドのフォネティック検索（人の名前など）を実行できます。フォネティック検索では、発音が検索値に類似するすべての値が返されます。										

列見出し	フィールド属性
	<p>S</p> <p>Tamino では使用できません。 サブディスクリプタまたはスーパーディスクリプタ。 サブディスクリプタ/スーパーディスクリプタにマルチプルバリューフィールド、ピリオディックグループのフィールド、またはこれらのフィールドの一部が含まれている場合、このサブディスクリプタ/スーパーディスクリプタのフィールドタイプの列に M または P がマークされます。これにより、Natural では、このサブディスクリプタ/スーパーディスクリプタに対する検索アルゴリズムを適切に作成できます。</p>

[DB] 列の例

01/10/2004 DDM Services					
19:02:57 V 6.1.1 P1 9 Software AG 2004 Line: 1					
DBID:	20	FNR:	14	DDM:	EMPLOYEES DEF. SEQ.:
C	T	DB	L	Name	F Length S D
		AA	1	PERSONNEL-ID	A 8 D
*				CNNNNNNNN	
G	AB	1		FULL-NAME	
	AC	2		FIRST-NAME	A 20 N
	AD	2		MIDDLE-I	A 1 N
	AE	2		NAME	A 20 D
	AD	1		MIDDLE-NAME	A 20 N
	AF	1		MAR-STAT	A 1 F
*				M=MARRIED	
	AG	1		SEX	A 1 F
	AH	1		BIRTH	N 06.0 D
	AH	1		N@BIRTH	I 2 D
G	A1	1		FULL-ADDRESS	
M	AI	2		ADDRESS-LINE	A 20 N

インジケータフィールド

インジケータフィールドは、可変長フィールドの長さ、またはデータベースフィールドのデータの有意性 (NULL 値インジケータ) に関する情報を取得するために使用します。インジケータフィールドには、データベースフィールドの内容は格納されていません。

L@ または N@ で始まるデータベースフィールド名は、NATCONV.INI コンフィグレーションファイルに指定されているインジケータ (『オペレーション』ドキュメントの「異なる文字セットを使用する方法」にある、IDENTIFIER-VALIDATION の説明も参照) に従って、インジケータフィールドとして解釈されます。したがって、インジケータフィールドでないフィールドの場合、これらの文字列以外でデータベースフィールド名を始める必要があります。

DDM が最初に生成されるときは、次の処理が実行されます。

- [L@xxxxx] フィールドは、あらゆる可変長フィールドに自動的に追加されます。xxxxx は、関連付けられているフィールドの名前です。

これは、Adabas ファイルのロング英数字 (LA) フィールドとラージオブジェクト (LB) フィールドに適用されます。

長さインジケータが LA、LB、LOB の各フィールドに関連付けられている場合、Natural データフォーマット/データ長は I4 である必要があります。VARCHAR フィールドの場合、データフォーマット/データ長は I2 である必要があります。

- [N@xxxxx] フィールドは、NULL 値を保持できるフィールドに自動的に追加されます。xxxxx は、関連付けられているフィールドの名前です。

これは、SQL の空値オプションが定義されている Adabas フィールド、。NULL インジケータ フィールドの Natural データフォーマット/データ長は I2 である必要があります。

フィールド列のヘルプ

次のセクションでは、DDMエディタ画面で使用できるフィールド列に対するヘルプ機能を起動する方法について説明します。

手順 22.3. フィールド列のヘルプ情報を表示するには

- フィールドの有効値を入力する方法を確認するには、該当のフィールドにカーソルを移動して F1 キーを 1 回押します。

または:

フィールドの有効値をリストから選択するには、該当のフィールドにカーソルを移動して F2 キーを押します。

編集および機能実行のためのコマンド

このセクションでは、DDMエディタで使用できる位置調整コマンド、行コマンド、エディタコマンド、および Natural システムコマンドについて説明します。

位置調整コマンドは DDM エディタ画面内の移動に使用し、行コマンドは 1 行以上の DDM ソースコード行の操作に使用します。エディタコマンドはエディタ画面の表示モードの変更などに使用し、システムコマンドは DDM のソースやカタログ化オブジェクトの保存などに使用します。

- 位置調整コマンド
- 行コマンド

■ エディタおよびシステムコマンド

位置調整コマンド

DDM エディタ画面内を移動するには、次のキーを使用します。

キー	説明
DOWN-ARROW	1 行スクロールダウンします。
LEFT-ARROW	画面を左に移動します。
RIGHT-ARROW	画面を右に移動します。
UP-ARROW	1 行スクロールアップします。
TAB	ある入力列フィールドから次のフィールドに移動します。

行コマンド

DDM エディタで有効な行コマンドは、单一または複数の DDM ソースコード行のコピー、削除、挿入、移動に使用します。以下に説明されている行コマンドを DDM ソースに入力する代わりに、「ファンクションキーの割り当て」で説明されている、対応するファンクション (F) キーを使用できます。

➤手順 22.4. 行コマンドを実行するには

- DDM エディタ画面上で、コマンドを適用するソースコード行の列 C にカーソルを移動し、以下にリストされている行コマンドのいずれかを入力します。

行コマンド	説明
C	マークされている行ブロックをコピーします。 行ブロックを X および Y で区切り、行ブロックのコピー先の下の行にカーソルを移動して C を入力します。
D	行コマンドを入力した行、またはマークされている行ブロックを削除します。 行ブロックを X および Y で区切り、D を入力します。
H	行ブロックのマークを削除します。
I	行コマンドを入力した行の上に、空行が挿入されます。 データベースが有効な場合、[Select Database Field] ウィンドウが表示され、フィールドのショートネームを選択できます。

行コマンド	説明
	<pre>+----- Select Database Field -----+ AA AA-1 AB AB-1 AC AC-1 AE AE-1 AD AD-1 +-----+</pre> <p>「フィールド属性列」の DB の説明も参照してください。</p>
M	マークされている行ブロックを移動します。 行ブロックを X および Y で区切り、行ブロックの移動先の下の行にカーソルを移動して M を入力します。
X	コピー、削除、移動する行ブロックの最初の行をマークします。
Y	コピー、削除、移動する行ブロックの最後の行をマークします。

エディタおよびシステムコマンド

DDM エディタで使用できるエディタコマンドまたは Natural システムコマンドは、DDM エディタメニューの **[COMMANDS]** 、 **[MISC]** 、 **[QUIT]** 、またはファンクション (F) キーを使用して実行します。

›手順 22.5. メニューからエディタコマンドまたはシステムコマンドを実行するには

1 DDM エディタ画面で ESC キーを押します。

DDM エディタメニューの **[COMMANDS]** 、 **[MISC]** 、および **[QUIT]** が DDM エディタ画面上部に表示されます。

(「DDM エディタの起動と終了」の[画面例](#)も参照)

2 メニューを選択して ENTER キーを押します。

有効なコマンドのリストが表示されます。コマンドおよび対応する F キー（対応するキーがある場合）については、このセクションで後述します。

3 現在の DDM ソースに対して実行するコマンドを選択し、ENTER キーを押します。

次のセクションでは、DDM エディタメニューで使用できるエディタコマンドまたはシステムコマンド、および標準のファンクションキーの割り当てについて説明します。

- [\[COMMANDS\] メニュー](#)
- [\[MISC\] メニュー](#)
- [\[QUIT\] メニュー](#)

■ ファンクションキーの割り当て

[COMMANDS] メニュー

[COMMANDS] メニューで使用できるコマンドは、同じ名前の Natural システムコマンドに対応しています。これらのメニューは、DDMエディタの編集エリアにある DDMソースのチェックやスキャンに使用します。また、ソースおよび／またはカタログ化オブジェクトとして、現在の Natural ライブラリおよび／またはシステムファイルにソースを保存します。

次の表は、[COMMANDS] メニューオプションと対応する F キー（対応するキーがある場合）を示しています。

メニュー オプション	F キー	説明
CATALOG		『システムコマンド』ドキュメントのシステムコマンド CATALOG の説明に従って、DDM ソースをカタログ化オブジェクトとして保存します。
CHECK	F11	『システムコマンド』ドキュメントのシステムコマンド CHECK の説明に従って、DDM ソースの構文をチェックします。
SAVE		『システムコマンド』ドキュメントのシステムコマンド SAVE の説明に従って、DDM ソースをソースオブジェクトとして保存します。
SCAN		『システムコマンド』ドキュメントのシステムコマンド SCAN の説明に従って、DDM ソース内の文字列を検索します。別の文字列で文字列を置き換えるオプションも使用できます。
STOW	F10	『システムコマンド』ドキュメントのシステムコマンド STOW の説明に従って、DDM ソースの構文をチェックし、ソースおよびカタログ化オブジェクトの両方を保存します。

[MISC] メニュー

次の表は、[MISC] メニューオプションと対応する F キー（対応するキーがある場合）を示しています。

メニュー オプション	F キー	関数
DB-SHORT-NAMES (ON/OFF)	F12	Tamino では使用できません。 DDM エディタ画面の属性列 [DB] の表示／非表示を切り替えます。 「フィールド属性列」の DB の説明も参照してください。
SHOW DOCTYPE INFO (ON/OFF)	F12	Tamino にのみ適用されます。 DDM エディタ画面の [Doctype Information] セクションの表示／非表示を切り替えます。 「 Tamino の doctype 情報の表示 」を参照してください。

メニュー オプション	F キー	関数
MODIFY DDM-HEADER	F13	DDM エディタのヘッダーフィールド [DBID] 、 [FNR] 、および [DEF.SEQ.] 編集モードで表示します。 「 DDM ヘッダー情報 」も参照してください。
SHOW EXTENDED FIELDS (ON/OFF)	F14	DDM エディタ画面の [Extended Field Information] セクションの表示／非表示を切り替えます。「 拡張フィールド属性の指定 」を参照してください。
EDIT EXTENDED FIELDS	F15	DDM エディタ画面の [Extended Field Information] セクションのフィールドを、編集モードで表示します。「 拡張フィールド属性の指定 」を参照してください。
SHOW COUPLED FILES		Adabas ファイルを参照する DDM にのみ適用されます。 この DDM に対し、ファイルが物理的にカップリングされていることを示します。ファイルは Adabas ディスクリプタを使用してカップリングされます。 ファイルカップリングの詳細については、『 <i>Adabas</i> 』ドキュメントを参照してください。

[QUIT] メニュー

次の表は、[QUIT] メニューオプションと対応する F キー（対応するキーがある場合）を示しています。

メニュー オプション	F キー	説明
EXIT (with STOW)	F3	STOW コマンド（「 COMMANDS メニュー」を参照）を実行し、DDM エディタを終了します。
QUIT (without STOW)		変更内容の保存および STOW コマンド（「 COMMANDS メニュー」を参照）の実行を行わずに、DDM エディタを終了します。

ファンクションキーの割り当て

F キーが有効なコマンドでは、DDM エディタメニューまたは行コマンドの代用として F キーを使用できます。また、F キーを使用して、DDM エディタで有効なコマンドやフィールドのヘルプ情報を確認できます。

➤手順 22.6. 現在の F キーの割り当てをリストするには

- DDM エディタ画面で F1 キーを 2 回押します。

次の表は、メニュー オプションまたは行コマンドに対応する、すべての標準の F キー割り当てのリストを示しています。

ファンクションキー	説明												
F1	HELP 1 回押すと、カーソルが置かれているフィールドのヘルプ機能が表示されます。 2 回押すと、現在の F キー設定についてのヘルプ情報が表示されます。												
F2	CHOICE カーソルが置かれているフィールドの選択ウィンドウ（対応するウィンドウがある場合）が表示されます。このウィンドウから、値を選択できます。有効な選択オプションがない場合、代わりにヘルプ情報ウィンドウが表示されます。												
F3	STOW + EXIT [QUIT] メニューの EXIT (with STOW) に対応しています。												
F4～F9	F4～F9 キーは DDM エディタ画面には表示されません。これは、他のキーに再割り当てでできるためです。現在の F キーの設定を表示または変更するには、関連するセクションに記載されている、 [Services Profile] メニューの [Function Keys] オプションを使用します。 F4～F9 キーは次の行コマンドに対応しています。 <table border="1"> <tr> <td>F4</td> <td>行コマンド D に対応しています。</td> </tr> <tr> <td>F5</td> <td>行コマンド I に対応しています。</td> </tr> <tr> <td>F6</td> <td>行コマンド X および Y に対応しています。</td> </tr> <tr> <td>F7</td> <td>行コマンド H に対応しています。</td> </tr> <tr> <td>F8</td> <td>行コマンド C に対応しています。</td> </tr> <tr> <td>F9</td> <td>行コマンド M に対応しています。</td> </tr> </table> 「F キーを使用して行ブロックをコピーまたは移動するには」も参照してください。	F4	行コマンド D に対応しています。	F5	行コマンド I に対応しています。	F6	行コマンド X および Y に対応しています。	F7	行コマンド H に対応しています。	F8	行コマンド C に対応しています。	F9	行コマンド M に対応しています。
F4	行コマンド D に対応しています。												
F5	行コマンド I に対応しています。												
F6	行コマンド X および Y に対応しています。												
F7	行コマンド H に対応しています。												
F8	行コマンド C に対応しています。												
F9	行コマンド M に対応しています。												
F10	STOW [COMMANDS] メニューの STOW に対応しています。												
F11	CHECK [COMMANDS] メニューの CHECK に対応しています。												
F12	DB-SHORT-NAMES または DOCTYPE INFO スイッチを切り替えます。 Adabas : [MISC] メニューの DB-SHORT-NAMES (ON/OFF) に対応しています。 Tamino : [MISC] メニューの SHOW DOCTYPE INFO (ON/OFF) に対応しています。												
F13	[MISC] メニューの MODIFY DDM-HEADER に対応しています。												

ファンクションキー	説明
F14	SHOW EXT FIELD または HIDE EXT FIELD スイッチを切り替えます。 [MISC] メニューの SHOW EXTENDED FIELDS (ON/OFF) に対応しています。
F15	EDIT EXT FIELD [MISC] メニューの EDIT EXTENDED FIELDS に対応しています。

›手順 22.7. F キーを使用して行ブロックをコピーまたは移動するには

- 1 コピー、削除、または移動する行ブロックの最初の行にカーソルを移動して、**F6**キーを押します。
行がマークされます。
- 2 DOWN-ARROW キーまたは UP-ARROW キーを押して、次の行または前の行に移動します。
追加の行がそれぞれマークされます。
- 3 マークする行ブロックの最後の行で **F6** キーを押し、それ以上行がマークされないようにします。
- 4 行ブロックのコピー先または移動先の下の行にカーソルを移動して、**F8** キーまたは **F9** キーを押します。

拡張フィールド属性の指定

拡張フィールドの編集機能では、別のNaturalオブジェクト（プログラムなど）でフィールドが使用されるときに適用される、ヘッダー、編集マスク、および注釈に対するデフォルトのフィールド属性を指定するためのオプションが用意されています。

ヘッダー属性では、DISPLAYステートメントなどの出力時にフィールドの上部に表示される、デフォルトの列ヘッダーを指定します。ヘッダーが指定されていない場合、フィールド名が列ヘッダーとして使用されます。

編集マスク属性では、DISPLAYステートメントなどによるフィールドの出力時に使用される、デフォルトの編集マスクを指定します。編集マスクは、Natural構文規則に準拠し、フィールドのNaturalデータフォーマットおよびデータ長に対して有効な値が指定されている必要があります。

注釈属性では、フィールドについてのコメントを指定します。

Taminoの場合、拡張フィールド編集機能にも Tamino 固有の情報が追加されています。

関連トピック：

- DISPLAY および『ステートメント』ドキュメントの INPUT
- 『パラメータリファレンス』ドキュメントの「EM - 編集マスク」

以下のセクションでは、次のトピックについて説明します。

- [拡張フィールド属性の表示／非表示の切り替え](#)
- [拡張フィールド属性の編集](#)
- [Tamino 固有の拡張フィールド属性](#)
- [SQL 固有の拡張フィールド属性](#)

拡張フィールド属性の表示／非表示の切り替え

このセクションでは、拡張フィールド属性の表示／非表示を切り替える方法について説明します。

➤手順 22.8. 拡張フィールド属性の表示／非表示を切り替えるには

- F14 キーを押します。

または:

[MISC] メニューで、 [SHOW EXTENDED FIELDS (ON/OFF)] を選択します。

表示設定になっている場合（デフォルト設定は非表示）、次の例のように、DDMエディタ画面の中央下部に [Extended Field Information] セクションが表示されます。

Adabas からの拡張フィールド属性の例

```

23/09/2004          DDM Services
15:16:36          V 6.1.1 P1 9 Software AG 2004      Line: 8
DBID: 20      FNR: 14      DDM: EMPLOYEES      DEF. SEQ.:      ↵
C   T      L  Name      F  Length  S  D      ↵
      1 PERSONNEL-ID      A      8      D      ↵
*      CNNNNNNNN      ↵
G      1 FULL-NAME      ↵
      2 FIRST-NAME      A      20      N      ↵
      2 MIDDLE-I      A      1      N      ↵
      2 NAME      A      20      D      ↵
      1 MIDDLE-NAME      A      20      N      ↵
      1 MAR-STAT      A      1      F      ↵
*      M=MARRIED      ↵
      1 SEX      A      1      F      ↵
      1 BIRTH      N      06.0      D      ↵
      1 N@BIRTH      I      2      D      ↵
G      1 FULL-ADDRESS      ↵
M      2 ADDRESS-LINE      A      20      N      ↵
~~~~~ Extended Field Information ~~~~~
Header   : MARITAL/STATUS      ↵
Edit Mask:      ↵
Remark   : NC      ↵
F1  HELP  F2 CHOICE F3  STOW+EXIT      F10 STOW      F11 CHECK      ↵
F12 DB-SHORT-NAMES  F13 MODIFY HEADER  F14 HIDE EXT FIELD  F15 EDIT EXT FIELD

```

Tamino からの拡張フィールド属性の例

```

01/10/2004          DDM Services
13:52:41          V 6.1.1 P1 9 Software AG 2004          Line: 11
DBID: 102  FNR: 1          DDM: EMPLOYEES-TAMINO          TYPE: XML
  C  T  L  Name          F  Length  D
  G  4  GROUP$3
    5  FIRST-NAME          A  20  D
    5  MIDDLE-NAME          A  20  D
    5  MIDDLE-I          A  20  D
    5  NAME          A  20  D
    3  MAR-STAT          A  1  D
    3  SEX          A  1  D
    3  BIRTH          A  10  D
~~~~~ Extended Field Information ~~~~~
Header      : Marital/Status
Edit Mask   :
Remark      : xs:string
Tag Name    : Mar-Stat
XPath       : /Employee/Mar-Stat
Occurrence  :
Flags       : MULT_OPTIONAL
Default Value:
Fixed Value :
F1 HELP    F2 CHOICE F3 STOW+EXIT      F10 STOW          F11 CHECK
F12 DOCTYPE INFO   F13 MODIFY HEADER  F14 HIDE EXT FIELD F15 EDIT EXT FIELD

```

【Extended Field Information】セクションに表示されるフィールドの内容は、カーソルが置かれているフィールドによって変わります。上の例では、カーソルは MAR-STAT に置かれています。

拡張フィールド属性の編集

次のセクションでは、DDM エディタ画面の 【Extended Field Information】セクションに含まれるフィールド属性を編集する方法について説明します。 [Tamino 固有の拡張フィールド属性](#) (関連セクションを参照) は編集できないことに注意してください。

手順 22.9. 【Extended Field Information】セクションのフィールドを編集するには

- F15 キーを押します。

【Extended Field Information】セクションの [Header] 入力フィールドにカーソルが移動します。このフィールドは強調表示され、編集することができます。

次の入力フィールドに移動するには、DOWN-ARROW キーまたは TAB キーを押します。

前のフィールドに移動するには、UP-ARROW キーを押します。

➤手順 22.10. フィールドの変更を保存する／保存しないにかかわらず、編集を終了するには

- ENTER キーを押します。

【Extended Field Information】 セクションの外にカーソルが移動します。

Tamino 固有の拡張フィールド属性

Tamino 固有の拡張フィールド属性は、Tamino XML スキーマ定義から抽出されます。

【Header】、【Edit Mask】、【Remark】の各フィールドの他に、次の読み取り専用の Tamino 固有の属性が【Extended Field Information】セクションに表示されます。

属性	関数						
Tag Name	<p>Tamino doctype 内のフィールド名。</p> <p>この名前は XML ドキュメント全体で一意である必要はありません。Tag Name を持たないグループフィールドもあります。</p>						
XPath	<p>Tamino doctype 内のフィールドを参照する完全な XPATH。</p> <p>XPATH 情報は、指定された XML ドキュメント内のデータ要素を一意に識別するために、アプリケーションの実行時に使用されます。したがって、XPATH 情報は変更できません。</p> <p>XPATH を持たないグループフィールドもあります。</p>						
Occurrence	<p>最大および最小のオカレンス数です。</p> <p>Tamino では、Tamino XML スキーマから抽出されたフィールドの多重度を表します。フィールドの多重度は、Tamino XML スキーマの maxOccurs ファセットで表されます。</p>						
Flags	<p>フラグを使用して、Tamino グループ構造内のフィールドの階層構造を表します。フラグは、特別なグループ構造（つまり、要素タグの属性）や複数オカレンスを適切に認識するために内部的に使用されます。また、XML ドキュメント内で必須または任意の DDM フィールドを識別できます。</p> <p>フラグは、1つのフィールドに対して組み合わせて使用できます。</p> <p>次のフラグを表示できます。</p> <table border="1"> <tr> <td>ARRAY</td> <td>フィールドは配列です。つまり、maxOccurs は 1 より大きな値です。</td> </tr> <tr> <td>GROUP_ATTRIBUTES</td> <td>フィールドは、先行フィールドのサブフィールド属性を持つグループです。</td> </tr> <tr> <td>GROUP_ALTERNATIVES</td> <td>フィールドは、選択コンストラクタを表すグループです。選択要素がサブフィールドとして含まれます。</td> </tr> </table>	ARRAY	フィールドは配列です。つまり、maxOccurs は 1 より大きな値です。	GROUP_ATTRIBUTES	フィールドは、先行フィールドのサブフィールド属性を持つグループです。	GROUP_ALTERNATIVES	フィールドは、選択コンストラクタを表すグループです。選択要素がサブフィールドとして含まれます。
ARRAY	フィールドは配列です。つまり、maxOccurs は 1 より大きな値です。						
GROUP_ATTRIBUTES	フィールドは、先行フィールドのサブフィールド属性を持つグループです。						
GROUP_ALTERNATIVES	フィールドは、選択コンストラクタを表すグループです。選択要素がサブフィールドとして含まれます。						

属性	関数
	GROUP_SEQUENCE フィールドは、順序コンストラクタを表すグループです。順序要素がサブフィールドとして含まれます。
	GROUP_ALL フィールドは、すべてのコンストラクタを表すグループです。すべての要素がサブフィールドとして含まれます。
	ATTR_REQUIRED フィールドは、必須としてマークされている属性です。
	ATTR_OPTIONAL フィールドは、任意としてマークされている属性です。
	ATTR_PROHIBITED フィールドは、禁止としてマークされている属性です。
	MULT_OPTIONAL フィールドは、XML ドキュメントで複数回使用できますが、必須ではありません。
	MULT_REQUIRED フィールドは、XML ドキュメントで複数回使用する必要があります。
	MULT_ONCE フィールドは、XML ドキュメントで正確に一度だけ使用する必要があります。
	SIMPLE_CONTENT フィールドが simpleContent で complexType として定義されました。
Default Value	フィールドに割り当てられるデフォルト値です。この属性はまだ使用されていません。
Fixed Value	フィールドに割り当てられる固定値です。この属性はまだ使用されていません。

SQL 固有の拡張フィールド属性

〔Header〕、〔Edit Mask〕、および〔Remark〕の各フィールドの他に、次の読み取り専用の SQL 固有の属性が〔Extended Field Information〕セクションに表示されます。

属性	関数
SQLTYPE	Oracle データベースに含まれている場合に、データタイプ BLOB (バイナリラージオブジェクト) または CLOB (文字ラージオブジェクト) から生成される情報です。

エディタの設定・サービスプロファイル

DDM サービスの **[Services Profile]** メニューは、ファンクション (F) キー F4～F9 の割り当てや **[DB]** 列の表示モードに対する DDM エディタのプロファイル設定を表示したり変更したりするために使用します。

➤手順 22.11. サービスプロファイルを起動するには

- **[DDM Services]** 画面で、**[Services Profile]** を選択します。
次のようなメニューオプションが表示されます。



このセクションでは、 [Services Profile] メニュー オプションについて説明します。

■ ファンクションキー

- 他の定義

ファンクションキー

【Function Keys】 オプションでは、F4～F9の各キーを再割り当てできます。これらのFキーは、DDM エディタで行コマンド（関連セクションを参照）を実行するために使用します。

手順 22.12. F キーを再割り当てるには

- 1 [Services Profile] メニューで、【Function Keys】を選択します。

現在の F キーの割り当てがリストされた【Profile settings】ウィンドウが表示されます。

```
+--- Profile settings ---+  
| Delete      F4      |  
| Insert      F5      |  
| Mark Block  F6      |  
| Unmark Block F7      |  
| Copy        F8      |  
| Move        F9      |  
+-----+
```

- 2 【Profile settings】 ウィンドウ上で、再割り当てる機能の行にカーソルを移動し、その機能を割り当てる F キーを押します。他の DDM サービス機能にまだ割り当てられていない F キーのみ割り当てできます。

次の例のように、新しい F キー設定が【Profile settings】 ウィンドウに表示されます。

```
+--- Profile settings ---+  
| Delete      F21     |  
| Insert      F5      |  
| Mark Block  F6      |  
| Unmark Block F7      |  
| Copy        F8      |  
| Move        F9      |  
+-----+
```

他の定義

Tamino では使用できません。

【Other Definitions】 オプションでは、「フィールド属性列」の説明に従って、【DB】列の表示／非表示を指定できます。

➤手順 22.13. 【DB】列の表示モードを変更するには

1 【Services Profile】メニューで、【Other Definitions】を選択します。

【Profile settings】 ウィンドウが表示されます。

```
+----- Profile settings -----+
| Display Database short names N |
+-----+
```

2 常に【DB】列を表示する場合は [Y] (Yes) 、常に非表示にする場合は [N] (No) に、現在の値を変更します。

23 DDM の保存とカタログ

DDMは、ソースオブジェクトおよび／またはカタログ化オブジェクト（生成プログラム）として、現在のNaturalシステムファイルの現在のNaturalライブラリ（存在する場合）に保存できます（「[DDM の保存 - FDDM システムファイル](#)」も参照）。

オブジェクトに適用される命名規則については、『Natural の使用』ドキュメントの「オブジェクトの命名規則」を参照してください。

➤手順 23.1. DDM を保存またはカタログするには

- DDMエディタ画面で、「[\[COMMANDS\] メニュー](#)」の説明に従って、[COMMANDS]メニューの [SAVE] 、 [CATALOG] 、または [STOW] を選択します。

または:

DDMエディタ画面を終了するとき、「[\[QUIT\] メニュー](#)」の説明に従って、[QUIT] メニューの [SAVE] または [EXIT (with STOW)] を選択します。

24 DDM のリスト

-
- [DDM Maintenance] メニューによる DDM のリスト 222
 - LIST による DDM のリスト 224

このセクションでは、DDM サービスの **[DDM Maintenance]** メニューまたは Natural のシステムコマンド `LIST` のいずれかを使用して、DDM のリストを表示する方法について説明します。

[DDM Maintenance] メニューによる DDM のリスト

[DDM Maintenance] メニューを使用して、現在の Natural ライブラリおよび／またはシステムファイルで有効なすべての DDM をリストできます。

›手順 24.1. すべての DDM をリストするには

1 ログオンしているライブラリの DDM をすべてリストする場合は、[手順 3](#) に進みます。

ライブラリを切り替える場合は、次の手順を実行して、必要なライブラリにログオンします。

[DDM Services] 画面の **[Library]** メニューを選択します。

次の例のように、現在の Natural ライブラリおよび／またはシステムファイルに格納されている全ライブラリがリストされた選択ウィンドウが表示されます。

```
2004-09-10          DDM Services          Library: SYSTEM      ↵
17:05:48          V 6.1.1 P1 9  Software AG 2004  DBID   :      ↵
User: SAG          FNR   :      ↵
+-----+           +-----+           +-----+           +-----+
|  Library      DDM Maintenance      Services Profile      Quit      ↵
|  |           |           |           |           |           |           ↵
+-----+           +-----+           +-----+           +-----+
|  <LOGON>  |           |           |           |           |           ↵
|  DEMO      |           |           |           |           |           ↵
|  |           |           |           |           |           |           ↵
|  DEMO-1    |           |           |           |           |           ↵
|  |           |           |           |           |           |           ↵
|  ORD-EXAM  |           |           |           |           |           ↵
|  |           |           |           |           |           |           ↵
|  PRE        |           |           |           |           |           ↵
|  |           |           |           |           |           |           ↵
|  SYSEXPG   |           |           |           |           |           ↵
|  |           |           |           |           |           |           ↵
|  SYSEXSYN  |           |           |           |           |           ↵
|  SYSTEM     |           |           |           |           |           ↵
+-----+           +-----+           +-----+           +-----+
Logon to DDM library      ↵
```

- 2 選択ウィンドウのライブラリを選択するか、または [<LOGON>] を選択して [DDM Library] ウィンドウにライブラリ名を入力します。
 - 3 [DDM Services] 画面で、[DDM Maintenance] を選択します。

次の例のように、選択したライブラリで有効なすべての DDM が、DDM メンテナンスウィンドウにリストされます。

```
2004-09-15          DDM Services          Library: SYSTEM ←
16:57:44          V 6.1.1 Pl 9  Software AG 2004  DBID   : ←
User: SAG          FNR   : ←
+-----+
| Library       DDM Maintenance       Services Profile       Quit ←
| ←
+-----+
+-----+
|   <CREATE>          | ←
|   EMPLOYEES          | ←
|   PERSONNEL          | ←
|   VEHICLES           | ←
+-----+
←
←
Select DDM
```

DDM メンテナンスウィンドウでは、DDM を選択して更に処理を進めることも、
[<CREATE>] メニューオプションを選択して新規 DDM を追加することもできます。DDM
メンテナンスウィンドウで有効な機能については、「[DDM の作成](#)」および「[DDM のメン
テナンス機能の使用](#)」の各セクションを参照してください。

LIST による DDM のリスト

Natural のシステムコマンド `LIST` を使用して、現在の Natural ライブラリ、steplib、システム
ファイル FNAT および FUSER で有効なすべての DDM をリストします。

▷手順 24.2. LIST を使ってすべての DDM をリストするには

- Natural メインメニューの **[Direct]** を選択し、**[Direct Command]** ウィンドウで次のように入力します。

```
LIST VIEW *
```

LIST V* 画面が開き、現在の Natural ライブラリ、steplib、およびシステムファイル FNAT
と FUSER で有効なすべての DDM の選択リストが表示されます。

LIST で使用可能なすべてのオプションの詳細については、『システムコマンド』ドキュメントで関連するセクションを参照してください。

25 異なる環境での DDM の管理

異なるライブラリおよびシステムファイル間で DDM を転送して（コピー、移動など）、異なる環境で DDM 操作（削除、検索など）を実行するには、Natural ユーティリティ SYSMAIN を使用できます（『ツールおよびユーティリティ』ドキュメントを参照）。

異なるハードウェアプラットフォーム間（メインフレーム、UNIX、OpenVMS、および Windows）で DDM を転送するには、オブジェクトハンドラを使用できます（『ツールおよびユーティリティ』ドキュメントを参照）。

26

Adabas または RDBMS のデータ変換

■ Adabas	230
■ Adabas D	230
■ Adabas SQL サーバー	231
■ DB2	231
■ Informix	232
■ Oracle	233
■ Sybase	234
■ Microsoft SQL Server	235
■ 関連トピック	236

このセクションでは、Natural データフォーマットと対応する Adabas データベースまたはリレーショナルデータベース管理システム (RDBMS) のデータタイプの変換表を示します。

RDBMS から DDM を生成するには、RDBMS 固有のデータタイプから Natural データタイプに変換する必要があります。データアクセスおよびデータ変換の全般的な情報については、「[関連トピック](#)」の関連ドキュメントリストを参照してください。

ラージ変数／フィールドおよびダイナミック変数／フィールドの使用の詳細については、『[ステートメント](#)』ドキュメントの「可変長の列に対する DDM の生成および編集」を参照してください。

Adabas

データタイプ	Adabas データフォーマット	Natural データフォーマット／データ長
英数字	A (n)	An
バイナリ	B (n)	Bn
固定小数点	F (n) but: F8	In I4
float	G (n)	Fn
パック型	P (n)	P (2 * n - 1)
アンパック型	U (n)	Nn
ワイド文字 (Unicode)	W (n)	U (n/2 rounded down)

Adabas D

RDBMS データタイプ	Natural データフォーマット／データ長
boolean	L
char (n)	An
日付	A10
fixed (p,q)	Np-q, q
float	F8
integer	I4
long	A (DYNAMIC)
long varchar	A (DYNAMIC)
smallint	I2
string	An
時刻	A8

RDBMS データタイプ	Natural データフォーマット／データ長
timestamp	A26
varchar	A n

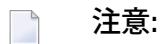
Adabas SQL サーバー

RDBMS データタイプ	Natural データフォーマット／データ長
char(5)	A5
char(253)	A253
decimal(5)	N5
decimal(10.4)	N6.4
double precision	N10.6
float(1...21)	N2.6
float(22...53)	N10.6
integer	I4
numeric(5)	N5
numeric(10.4)	N6.4
real	F4
smallint	I2

DB2

RDBMS データタイプ	Natural データフォーマット／データ長
日付	A10
blob	B (DYNAMIC)
clob	A (DYNAMIC)
dbclob	U (DYNAMIC)
decimal(5)	N5
decimal(10.4)	N6.4
fixed character(5)	A5
float	F n
graphic(n)	U n
longvar	A (DYNAMIC)
longvarg	A (DYNAMIC)

RDBMS データタイプ	Natural データフォーマット／データ長
large integer	I4
scientific notation	N10.6
small integer	I2
special data	A253
system date and time	A10
時刻	A8
timestamp	A26
varchar	A n
varg	2*A n
vargraphic(n)	U n

**注意:**

1. DB2 でデータタイプ blob、clob、dbblob、graphic および vargraphic を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。
2. DB2 では、データタイプ graphic および vargraphic は、CREATE DATABASE mydb USING CODESET UTF-8 TERRITORY US などのステートメントで生成されている場合にのみ使用できます。詳細については、ローカルの DB2 のドキュメントを参照してください。

Informix

RDBMS データタイプ	Natural データフォーマット／データ長
byte	A n
char(n)	A n
日付	A10
datetime	A26
decimal(p, q)	N $p-q, q$
float	F8
integer	I4
interval	A17
money	N14.2
nchar(2* n)	U n
nvarchar(2* n)	U n
real	F4
serial	I4
smallfloat	F4

RDBMS データタイプ	Natural データフォーマット／データ長
smallint	I2
テキスト	A n
varchar(n)	A n

 **注意:** Informix でデータタイプ nchar および nvarchar を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。

Oracle

RDBMS データタイプ	Natural データフォーマット／データ長
blob	B (DYNAMIC)
char (n)	A n
clob	A (DYNAMIC)
日付	A10
decimal (p, q)	N $p-q, q$
double precision	F8
float	F4
integer	I4
long	A (DYNAMIC)
long raw	B (DYNAMIC)
nchar(n)	U n
nclob	U (DYNAMIC)
number	N n
nvarchar2(n)	U n
raw (n)	B n
real	F4
rowid	A n
smallint	I2
timestamp	A26
varchar	A n
varchar2 (n)	A n

 **注意:**

- 同じテーブル内でデータタイプ long と long raw、および clob と blob を混同しないようにしてください。

2. データタイプ timestamp を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。timestamp のバリエントの timestamp with time zone および timestamp with local time zone はサポート対象外です。
3. Oracle でデータタイプ nchar、nvarchar2 および ncllob を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。

Sybase

RDBMS データタイプ	Natural データフォーマット／データ長
binary (n)	Bn
bit	N1
char (n)	An
datetime	A26
float	F8
int	I4
money	N15.4
real	F4
smalldatetime	A26
smallint	I2
smallmoney	N6.4
timestamp	B8
tinyint	I2
unichar(n)	Un
univarchar(n)	Un
varbinary (n)	Bn
varchar (n)	An



注意: Sybase でデータタイプ unichar および univarchar を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。これらのデータタイプは、Entire Access ctlib interface for Sybase を使用している場合にのみ使用できます。

Microsoft SQL Server

RDBMS データタイプ	Natural データフォーマット／データ長
binary (<i>n</i>)	B <i>n</i>
bit	N1
char (<i>n</i>)	A <i>n</i>
datetime	A26
float	F8
image	B (DYNAMIC)
int	I4
money	N15.4
nchar(2 ^{<i>n</i>})	U <i>n</i>
ntext	U (DYNAMIC)
nvarchar(2 ^{<i>n</i>})	U <i>n</i>
real	F4
smalldatetime	A26
smallint	I2
smallmoney	N6.4
テキスト	A (DYNAMIC)
timestamp	B8
tinyint	I2
varbinary (<i>n</i>)	B <i>n</i>
varchar (<i>n</i>)	A <i>n</i>



注意: Microsoft SQL Server でデータタイプ nchar、nvarchar、および ntext を使用するには、Entire Access バージョン 6.2.1 以降の使用が前提条件となります。また、これらのデータタイプには、Entire Access バージョン 6.2.1 以降の MSSQLODBC ドライバを使用する必要があります。

関連トピック

次のドキュメントの、Adabas または RDBMS からのデータ変換に関するセクションを参照してください。

- 『プログラミングガイド』
Adabas データベースのデータへのアクセス
SQL データベースのデータへのアクセス

27 Tamino のデータ変換

■ 組み込み Tamino XML スキーマ言語のデータタイプ	238
■ Tamino XML スキーマコンストラクタ	240
■ Tamino XML スキーマ言語の多重度	241

Tamino からの Natural DDM の生成は、Tamino XML スキーマ言語に基づいて行われます。Tamino XML スキーマ言語の基本概念、および Natural for Tamino との対話方法については、『プログラミングガイド』の「Tamino データベースのデータへのアクセス」を参照してください。

このセクションでは、Tamino データタイプの Natural データフォーマットへのマッピングについて説明します。

組み込み Tamino XML スキーマ言語のデータタイプ

Tamino XML スキーマ言語には、可能であれば、対応する Natural データフォーマットにマップされる組み込みデータタイプが多数用意されています。データタイプによっては、適切な Natural データフォーマットがない場合もあります。そのような Tamino データタイプは、最も汎用的な次の Natural データフォーマットにマップされます：U (DYNAMIC) スキーマ言語は長さが無制限の文字列を基本にしているため、データフォーマット U (DYNAMIC) はあらゆる Tamino XML スキーマの組み込みデータタイプを保持できます。

次の表は、Natural for Tamino でサポートされている組み込み Tamino 基本データタイプおよび派生データタイプと、これらのデータタイプがマップされる Natural データフォーマットを示しています。

- Tamino の基本データタイプ
- Tamino の派生データタイプ

Tamino の基本データタイプ

Tamino の基本データタイプ	Natural データフォーマット／データ長
xs:string	U (DYNAMIC)
xs:boolean	L
xs:decimal	P22.7
xs:float	F4
xs:double	F8
xs:duration	U (DYNAMIC)
xs:dateTime	U (DYNAMIC)
xs:time	T
xs:date	D
xs:gYearMonth	U (DYNAMIC)
xs:gYear	U (DYNAMIC)
xs:gMonthDay	U (DYNAMIC)
xs:gDay	U (DYNAMIC)

Tamino の基本データタイプ	Natural データフォーマット／データ長
xs:gMonth	U (DYNAMIC)
xs:hexBinary	U (DYNAMIC)
xs:base64Binary	U (DYNAMIC)
xs:anyURI	U (DYNAMIC)
xs:QName	U (DYNAMIC)
xs:NOTATION	U (DYNAMIC)

Tamino の派生データタイプ

Tamino の派生データタイプ	Natural データフォーマット／データ長
xs:normalizedString	U (DYNAMIC)
xs:token	U (DYNAMIC)
xs:language	U (DYNAMIC)
xs:NMTOKEN	U (DYNAMIC)
xs:NMTOKENS	U (DYNAMIC)
xs:Name	U (DYNAMIC)
xs:NCName	U (DYNAMIC)
xs:ID	U (DYNAMIC)
xs:IDREF	U (DYNAMIC)
xs:IDREFS	U (DYNAMIC)
xs:ENTITY	U (DYNAMIC)
xs:ENTITIES	U (DYNAMIC)
xs:Integer	P29
xs:nonPositiveInteger	P29
xs:negativeInteger	P29
xs:long	P19
xs:int	I4
xs:short	I2
xs:byte	I2
xs:nonNegativeInteger	P29
xs:unsignedLong	P19
xs:unsignedShort	I2
xs:unsignedByte	I2
xs:unsignedInt	I4
xs:positiveInteger	P29

Tamino XML スキーマコンストラクタ

Tamino XML スキーマコンストラクタは、ドキュメント構造を定義するために使用します。また、コンストラクタを使用することにより、既存のデータ型から新しいデータ型を派生させたり、ドキュメントのネスト構造を記述したりすることができます。

Tamino XML スキーマのデータタイプは、既存データタイプの一連の派生メソッドを使用して新規作成できます。派生データタイプを Natural データフォーマットにマップできない場合、代わりに最も汎用的なデータフォーマット U (DYNAMIC) が Natural によって使用されます。

次の表は、Natural によってサポートされている Tamino XML スキーマコンストラクタと各コンストラクタの属性を示しています。コメント列には、DDM が生成されるときに実行されるマッピングについての説明が記載されています。

XML スキーマコンストラクタの使用に関する制限事項については、『プログラミングガイド』の「*Tamino データベースのデータへのアクセス*」を参照してください。

コンストラクタ	属性	コメント
xs:all	minOccurs maxOccurs	Natural グループ構造にマップされます。
xs:attribute	名前 ref type form use	Natural グループ構造にマップされます。
xs:choice	minOccurs maxOccurs	Natural グループ構造にマップされます。
xs:complexType	名前 mixed=false (mixed=false のみサポート)	メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:element	名前 ref type form minOccurs maxOccurs	xs:element 副構造 (单一または複合タイプ定義) に応じて、Natural データタイプまたは Natural グループにマップされます。
xs:enumeration		タイプ U (DYNAMIC) にマップされます。
xs:extension	base	メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:fractionDigits	value	Natural データタイプの精度に影響します。

コンストラクタ	属性	コメント
xs:length	value	Natural データタイプの長さに影響します。unbounded の長さがタイプ U (DYNAMIC) にマップされます。
xs:maxInclusive xs:maxExclusive xs:minInclusive xs:minExclusive xs:minLength xs:pattern	value	マッピングに影響しません (つまり、base タイプは制限を受けません)。
xs:maxLength	value	Natural データタイプの長さに影響します。unbounded の長さがタイプ U (DYNAMIC) にマップされます。
xs:restriction	base	メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:schema	attributeFormDefault elementFormDefault targetNamespace	メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:sequence	minOccurs maxOccurs	Natural グループ構造にマップされます。
xs:simpleContent		メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:simpleType	名前	メタコンストラクタであるため、直ちに Natural データタイプにはなりません。
xs:totalDigits	value	Natural 数値データタイプの長さに影響します。

Tamino XML スキーマ言語の多重度

Tamino XML スキーマ言語の多重度は、適切なコンストラクタの属性 `maxOccurs` を使用して表現します。`maxOccurs` の値が 1 より大きい場合、Tamino から生成される Natural DDM は配列定義になります。`maxOccurs` の値に応じて、静的配列 (`maxOccurs` が数値に設定されている場合) または X-array (`maxOccurs` が `unbounded` に設定されている場合) が DDM に生成されます。通常と同様、配列定義は、DDM からビューを定義するときに上書きできます。

