

## Natural for Mainframes

**Natural の使用**

バージョン 4.2.5

October 2009

This document applies to Natural バージョン 4.2.5 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 1979-2009. All rights reserved.

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

# 目次

1 Natural の使用 .....	1
2 Natural セッションの呼び出しと終了 .....	3
Natural オンラインセッションの開始 .....	4
セッション開始後の手順 .....	4
Natural オンラインセッションの終了 .....	4
Natural バッチセッションの開始と終了 .....	6
3 コマンドとメニュー機能の使用 .....	7
Natural コマンドのカテゴリ .....	8
Natural コマンドの使用 .....	9
Natural メニューの使用 .....	12
4 Natural オンラインヘルプ .....	17
全般的な情報 - Natural ヘルプユーティリティ .....	18
エラーメッセージの詳細情報 .....	21
メニューとフィールドの詳細情報 .....	26
5 Natural ライブラリの使用 .....	27
ライブラリに含まれるオブジェクト .....	28
ライブラリタイプ .....	28
Steplib ライブラリ .....	29
オブジェクト実行の検索順序 .....	31
デフォルトのライブラリ割り当て .....	32
ライブラリへのログオン .....	33
ライブラリの作成 .....	34
ライブラリのリスト .....	34
ライブラリ内のオブジェクトのリスト .....	36
オブジェクトのリストの印刷 .....	38
ライブラリ内のオブジェクトの検索 .....	39
ライブラリのコピー、移動、および名前変更 .....	44
ライブラリ内のオブジェクトの削除 .....	46
6 Natural オブジェクトの管理と実行 .....	49
Natural オブジェクトについて .....	50
Natural エディタまたはユーティリティの使用 .....	52
オブジェクトの選択と表示 .....	55
オブジェクトの作成と編集 .....	56
オブジェクトのチェックとテスト .....	60
オブジェクトの保存とカタログ .....	62
オブジェクトディレクトリ情報の表示 .....	65
オブジェクトのコピー .....	66
オブジェクトの印刷 .....	68
オブジェクトの名前の変更 .....	69
オブジェクトの移動 .....	71
オブジェクトの削除 .....	72
プログラムの実行 .....	74
7 Natural のメインメニュー .....	77

---









Natural のメインメニューの表示またはクローズ .....	78
Development Functions .....	80
Development Environment Settings .....	85
Maintenance and Transfer Utilities .....	86
Debugging and Monitoring Utilities .....	87
Example Libraries .....	87
Other Products .....	88
8 出力ファイルとワークファイル .....	89
9 Natural 環境の構成 .....	91
プロファイルパラメータの使用 .....	92
セッションパラメータの使用 .....	95
パラメータ化のレベル .....	97
10 ルールと命名規則 .....	99
標準の文字設定 .....	100
オブジェクトの命名規則 .....	100
ライブラリの命名規則 .....	101
ユーザー定義変数の命名規則 .....	102
索引 .....	105


# 1 Natural の使用

このドキュメントでは、メインフレーム環境での Natural の使い方を紹介します。

このドキュメントで説明する手順と方法は、元の Natural ソフトウェアで提供するデフォルトの標準に合わせています。これらの説明は、Natural の全機能を包括的に説明するものではありません。すべてのオプションと機能の詳細な説明については、このドキュメントで示す関連トピックを参照してください。

『Natural の使用』ドキュメントに記載されているサンプル画面のレイアウト、およびここで説明している Natural の動作は、ユーザーの結果と異なる場合があります。例えば、コマンドまたはメッセージ行が画面の異なる位置に表示されたり、Natural コマンドの実行がセキュリティコントロールによって保護されている場合があります。環境のデフォルト設定は、Natural 管理者が設定したシステムパラメータによって異なります。

 <b>Natural</b> セッションの呼び出しと終了	Natural セッションの呼び出しと終了の方法について説明します。
 コマンドとメニュー機能の使用	Natural コマンドとメニュー機能の実行方法について説明します。
 <b>Natural</b> オンラインヘルプ	Natural のオンラインヘルプ機能と提供されるヘルプ情報のタイプについて説明します。
 <b>Natural</b> ライブラリの使用	Natural ライブラリの使い方と、ライブラリのアクセス、リスト、管理の方法について説明します。
 <b>Natural</b> オブジェクトの管理と実行	Natural オブジェクトの作成、編集、管理、削除、実行の方法について説明します。
 <b>Natural</b> のメインメニュー	開発機能、設定、ユーティリティ、サンプルライブラリにアクセスできる Natural のメインメニューと下位のメニューについて説明します。
 出力ファイルとワークファイル	Natural の出力ファイルとワークファイルについて説明します。
 <b>Natural</b> 環境の構成	Natural 環境のカスタマイズ、プロセスの標準化や自動化のために、Natural パラメータを指定する方法について説明します。

	ルールと命名規則	Natural 固有のルールと命名規則について説明します。
---	----------	-------------------------------

## 2 Natural セッションの呼び出しと終了

---

- Natural オンラインセッションの開始 ..... 4
- セッション開始後の手順 ..... 4
- Natural オンラインセッションの終了 ..... 4
- Natural バッチセッションの開始と終了 ..... 6

Naturalセッションは、オンラインモードまたはバッチモードで実行するために呼び出すことができます。

関連トピック：

- [コマンドとメニュー機能の使用](#)
- [Natural のメインメニュー](#)

## Natural オンラインセッションの開始

---

Naturalセッションを開始する方法は、ローカルのシステム環境によって異なります。詳細については、Naturalの管理者にお問い合わせください。Natural Securityがインストールされている場合、特定のライブラリへのアクセスとNatural機能の使用を制限することができます。

## セッション開始後の手順

---

使用環境のデフォルト設定に従ってサイトの手順でNaturalを起動すると、Naturalのメインメニュー、NEXTコマンドプロンプト、またはユーザー定義のスタートアップメニューが表示されます。

Naturalのメインメニューを使用する場合、Naturalの開発機能、環境設定、メンテナンスユーティリティ、およびサンプルプログラムがあるライブラリを選択することができます。Naturalのメインメニューおよび下位メニューの機能、Naturalのメインメニューを表示したり閉じたりする手順の詳細については、「[Natural のメインメニュー](#)」を参照してください。

NEXTコマンドプロンプトを使用する場合は、Naturalコマンドを入力するか、プログラムタイプのNaturalオブジェクトを実行します。詳細な手順については、「[Natural コマンドの使用](#)」を参照してください。

## Natural オンラインセッションの終了

---

Naturalのメインメニュー、Naturalコマンドプロンプト、またはNaturalオブジェクトからNaturalセッションを終了することができます。

### ▶手順 2.1. Natural オンラインセッションを終了するには

- 次のいずれかの方法を使用します。
  - PF3（終了）キーを押すか、Naturalのメインメニューで **[Exit Natural Session]** を選択して ENTER キーを押します。



- または：  
コマンド行（「[コマンド行](#)」を参照）で、次のいずれかのコマンドを入力します。

```
.
```

（ピリオド）

または

```
FIN
```

または

```
%%
```

さらに ENTER キーを押します。

- または：  
NEXT プロンプト（「[NEXT と MORE](#)」を参照）で、次のいずれかのコマンドを入力します。

```
FIN
```

または

```
%%
```

または

```
CLEAR
```

さらに ENTER キーを押します。

- または：  
MORE プロンプト（「[NEXT と MORE](#)」を参照）で、次のいずれかのコマンドを入力します。

```
FIN
```

または

```
%%
```

さらに ENTER キーを押します。

- または：  
Natural オブジェクト内で、TERMINATE ステートメントを実行します。

関連トピック：

- [Natural コマンドの使用](#)
- [Natural オブジェクトについて](#)

## Natural バッチセッションの開始と終了

---

Natural バッチインターフェイスには、バッチモードで Natural コマンドまたはプログラムタイプの Natural オブジェクトを実行するオプションがあります。

### ▶手順 2.2. Natural セッションをバッチで呼び出すには

- 『オペレーション』ドキュメントの「Natural セッションの開始」で説明している手順に従います。

Natural バッチモードセッションは、セッション中に次のいずれかを検出すると終了します。

- 入力データセット CMSYNIN の FIN コマンド。
- コマンド入力データセット CMSYNIN の入力終了条件。
- 実行中の Natural オブジェクト内の TERMINATE ステートメント。

関連トピック：

- バッチモードでの Natural - 『オペレーション』ドキュメント

# 3 コマンドとメニュー機能の使用

---

- Natural コマンドのカテゴリ ..... 8
- Natural コマンドの使用 ..... 9
- Natural メニューの使用 ..... 12

Natural のセッション中は、Natural コマンドまたはメニュー機能を使用して操作を実行できます。Natural コマンドを使用する場合、さまざまなメニューを表示しなくても操作を直接実行できます。

また「[プログラムの実行](#)」で説明するように、プログラムタイプの Natural オブジェクトは Natural コマンドを使用しなくても実行できます。

このセクションでは、Natural に備わっているコマンドのさまざまなカテゴリ、コマンドの実行方法、メニュー機能の使い方について説明します。

関連トピック：

- [Natural のメインメニュー](#)

## Natural コマンドのカテゴリ

---

このセクションでは、Natural コマンドのさまざまなカテゴリについて説明します。

- [システムコマンド](#)
- [端末コマンド](#)
- [エディタおよびユーティリティコマンド](#)

### システムコマンド

Natural システムコマンドは、Natural オブジェクトの作成、管理、または実行に必要な機能を実行します。また、Natural システムコマンドを使用して、Natural 環境をモニタしたり管理したりすることもできます。

『システムコマンド』ドキュメントの関連トピック：

- [システムコマンド \(概要\)](#)
- [機能別システムコマンド](#)
- [システムコマンド構文](#)

### 端末コマンド

例えば、Natural 端末コマンドは、次の場合に使用できます。

- PF キーとメッセージ行の位置決めや色の割り当てなどの画面表示およびレイアウトを配列します。
- 現在の環境に関するデバック情報を取得します。
- 現在の Natural 処理を中断します。

端末コマンドは、アプリケーションの実行中に呼び出すことができます。Natural コマンドプロンプト以外に、端末コマンドを任意の英数字入力フィールドに入力できます。端末コマンドは、Natural セッションパラメータを設定することによって指定できる制御文字で始まります。デフォルトの制御文字はパーセント (%) 記号です。

関連トピック：

- [画面設計](#) - 『プログラミングガイド』
- [画面からのデータのコピー](#) - 『プログラミングガイド』
- [機能別端末コマンド](#) - 『端末コマンド』ドキュメント
- [端末コマンド \(概要\)](#) - 『端末コマンド』ドキュメント
- [セッションパラメータの使用](#)

## エディタおよびユーティリティコマンド

Natural システムコマンドと Natural 端末コマンドに加えて、それぞれの Natural エディタと Natural ユーティリティには、その環境のみに適用される独自のコマンドがあります。これらのコマンドについては、エディタまたはユーティリティドキュメントの関連セクションで説明しています。

## Natural コマンドの使用

Natural システムコマンドは、コマンドプロンプトで入力します。Natural コマンドプロンプトには、次のようなタイプがあります。

- Natural のメインメニュー、Natural ユーティリティの画面、またはシステムコマンドの画面にあるコマンド行 (次の例も参照)。
- プログラムエディタのより大きい (>) 記号など、エディタコマンドプロンプト (次の例も参照)。
- NEXT プロンプトと MORE プロンプト。

一部のユーティリティでは、システムコマンドの前に、2つのスラッシュ (//) など特別な記号を指定する必要があります。詳細については、『ユーティリティ』ドキュメントの関連するセクションを参照してください。

ほとんどのシステムコマンドは、実行する操作を詳細に指定するために、1つ以上のパラメータやオペランドを指定して入力できます。「[システムコマンドの例](#)」も参照してください。

Natural 端末コマンドは、コマンドプロンプトまたは英数字の入力フィールドに入力します。

Natural エディタやユーティリティのコマンドは、コマンドプロンプト、またはエディタやユーティリティのコマンド行に入力します。

Natural コマンドの入力では、大文字と小文字は区別されません。Natural コマンドを入力したら、ENTER キーを押します。ENTER キーを押すと、操作が確認されてコマンドが実行されたり、コマンドの実行を明示的に確認する追加の確認ウィンドウが表示されたりします。

このセクションでは、以下のトピックについて説明します。

- コマンド行
- NEXT と MORE
- システムコマンドの例

### コマンド行

コマンド行は PF キー行の上であり、次のように表示されます。

```
Command ==>
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---  
      Help      Exit                                  Canc
```

### NEXT と MORE

起動されている Natural プログラム（例えば、Natural のメインメニューを表示するプログラム）がまだない場合に、NEXT プロンプトが表示されます。NEXT プロンプトは、Natural が次のコマンド入力を待っていることを示します。

MORE プロンプトはプログラムの実行時に表示され、追加的な出力があることを示します。ENTER キーを押すと、追加的な出力が表示されます。MORE プロンプトに応じてコマンドを入力すると、実行中のプログラムが終了して、コマンドが実行されます。

NEXT または MORE プロンプトは、通常、次の例に示すように画面左上または左下の端に表示されます。

```
NEXTLIB=TEST
```

## システムコマンドの例

次の手順は、オブジェクトを編集するために Natural システムコマンドを実行する方法の例です。

### ▶手順 3.1. Natural オブジェクトのために Natural エディタを起動するには

- 1 任意のコマンドプロンプトで、システムコマンド EDIT を入力し、必要な場合は 1 つ以上のオペランドも入力します。

例えば、次のように入力します。

```
EDIT P PROGX
```

P はオブジェクトタイプ（プログラム）であり、PROGX は編集するオブジェクトの名前です。

- 2 ENTER キーを押します。

Natural プログラムエディタが起動され、PROGX のソースコードが次の例のように編集エリアに表示されます。

```
>                                     > + Program   PROGX   Lib TEST
All  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7...
0010 READ (3) EMPLOYEES BY NAME
0020 DISPLAY NAME
0030 END
0040
0050
0060
0070
0080
```

「[メニュー機能の例](#)」で説明しているように、メニュー機能を使用する場合は、同等の入力コマンドも参照してください。

関連トピック：

- 例：コマンド構文 - 『システムコマンド』ドキュメント

## Natural メニューの使用

---

すべての Natural メニュー画面で、機能のリストが表示されます。メニューから機能呼び出す方法は、メニュー構造と提供されるオプションによって異なります。

Natural メニューの場合、次のようにメニュー機能の選択と呼び出しに利用できる別の方法があります。

- リストされた各機能に対応した個別の入力フィールド。
- 各機能に割り当てられた 1 文字または 2 文字のコードを入力できる **[Code]** フィールド。
- メニューの特定の機能に対応するファンクションキー（PF キー）。

PF キー行（通常は、画面の下部にある）は、どの機能がどのキーに割り当てられているかを示します。「[標準的な PF キー](#)」も参照してください。

ほとんどの Natural メニューには、機能に加えて、さらにオプションや選択条件を指定できるフィールドがあります。例えば、**[Development Functions]** メニューには、Natural オブジェクトの名前を入力できる **[Name]** フィールドがあります。これらのフィールドとその有効な入力値については、関連ドキュメントを参照するか、「[Natural オンラインヘルプ](#)」の説明に従ってヘルプ機能を使用してください。

このセクションでは、以下のトピックについて説明します。

- [メニュー機能の実行](#)
- [メニュー機能の終了](#)
- [標準的な PF キー](#)
- [メニュー機能の例](#)

### メニュー機能の実行

このセクションでは、メニュー機能の実行と終了のための手順を説明します。

#### ▶手順 3.2. メニュー機能の実行

- 目的のメニュー機能の横にある入力フィールドにカーソルを表示して、ENTER キーを押します。

または:

目的のメニュー機能の横にある入力フィールドにカーソルを表示して、任意の文字を入力します。

または:



[Code] フィールドに、目的の機能の前に表示されている 1 文字または 2 文字のコードを入力します。

または:

可能な場合は、PF キー行で目的の機能に対応している PF キーを押します（「標準的な PF キー」も参照）。

または:

目的の機能の横にある入力フィールドをダブルクリックします。

追加的な入力フィールドがメニューに表示される場合、必要な情報を入力します。入力フィールドが表示されない場合、有効な入力値を選択できるウィンドウが表示されるか、対応する Natural エラーメッセージが表示されます。

利用可能なフィールドエントリの説明については、目的のフィールドに疑問符 (?) を入力してヘルプ機能呼び出ししてください。

## メニュー機能の終了

次の手順では、別の Natural 画面で行った変更を保存するかどうかに関係なく、メニュー機能の終了に一般的に利用できる代替的な方法について説明します。

### ▶手順 3.3. 変更を保存しないで機能を終了するには

- ピリオド (.) を入力して、ENTER キーを押します。

または:

PF12 キーを押します。

### ▶手順 3.4. 変更を保存して機能を終了するには

- PF3 キーを押します。

## 標準的な PF キー

次のファンクションキー（PF キー）は、ほとんどの Natural メニューで次の機能に割り当てられています。

PF キー	PF キー名	説明
PF1	Help	オンラインヘルプ機能呼び出します。
PF2	Menu	Natural のメインメニューを表示します。
PF3	Exit	機能を終了します。
PF12	Canc	変更を適用しないで、機能を終了します。

## メニュー機能の例

次の手順は、オブジェクトを編集するために Natural メニューを使用する方法の例です。

### ▶手順 3.5. Natural オブジェクトのために Natural エディタを起動するには

1 [Development Functions] 画面で、次の操作を実行します。

[Code] フィールドに、[Edit Object] 機能に対応する 1 文字のコードを入力します。

[Type] フィールドに、Natural オブジェクトのタイプに対応する 1 文字のコードを入力します（次の例では、プログラムの P）。

[Name] フィールドに、Natural オブジェクトの名前を入力します（次の例では、PROGX）。

```

13:33:16          ***** NATURAL *****                               2007-08-14
User SAG          - Development Functions -                               Library TEST
                                                                Mode Structured
                                                                Work area empty

Code  Function

C    Create Object
E    Edit Object
R    Rename Object
D    Delete Object
X    Execute Program
L    List Object(s)
S    List Subroutines Used
?    Help
.    Exit

Code .. E      Type .. P
Name .. PROGX_____
    
```

```

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit                                     Canc
  
```

- 2 ENTER キーを押します。

Natural プログラムエディタが起動され、「システムコマンドの例」に示すように編集エリアに PROGX のソースコードが表示されます。



## 4 Natural オンラインヘルプ

---

■ 一般的な情報 - Natural ヘルプユーティリティ .....	18
■ エラーメッセージの詳細情報 .....	21
■ メニューとフィールドの詳細情報 .....	26

Natural では、次のようにいくつかのタイプのヘルプを提供しています。

- Natural ステートメント、コマンド、変数、エディタ、ユーティリティなど、Natural コンポーネントの全般情報。
- Natural システムエラーメッセージとユーザー定義メッセージの詳細情報。
- Natural ユーティリティとエディタに表示される Natural メニューと入力フィールドの詳細情報。

Natural コンポーネントの全般情報は、Natural ヘルプユーティリティに含まれており、システムコマンド `HELP` でアクセスできます。

Natural メッセージの詳細情報は、システムコマンド `HELP` および `LASTMSG` で表示されます。

メニューと入力フィールドの詳細情報は、個別のヘルプ機能で表示されます。

関連トピック：

- `HELP` - 『システムコマンド』ドキュメント
- `LASTMSG` - 『システムコマンド』ドキュメント

このセクションでは、提供されているヘルプトピックとそのアクセス方法について説明します。

## 全般的な情報 - Natural ヘルプユーティリティ

---

Natural ヘルプユーティリティでは、Natural ステートメント、コマンド、変数、エディタ、ユーティリティ、および Natural メッセージの全般情報を提供します。Natural ヘルプユーティリティとその下位のメニューを操作するか、特定のヘルプトピックに直接アクセスして、特定のヘルプトピックのオンラインヘルプを表示することができます。

このセクションでは、以下のトピックについて説明します。

- [Natural ヘルプユーティリティメニューの呼び出し](#)
- [ヘルプトピックへの直接的なアクセス](#)

### Natural ヘルプユーティリティメニューの呼び出し

#### ▶手順 4.1. ヘルプユーティリティのメニューを表示するには

- 1 次のシステムコマンドを入力します。

```
HELP
```

または

```
?
```

- 2 ENTER キーを押します。

次の例のようなNaturalヘルプユーティリティのメニューが表示され、提供されているヘルプトピックのリストが示されます。

```

10:50:05          ***** NATURAL HELP UTILITY *****          2003-02-07
                    - Menu -

-----

Natural Help provides information on:

S Natural Statements
V Natural System Variables
F Natural System Functions
C Natural System Commands
E Natural Editors
U Natural Utilities
P Natural Session Parameters
T Natural Terminal Commands
N Natural System Messages
M User-Defined Messages
. Exit

Code .. _

-----
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                                     Canc

```

- 3 [Code] フィールドに、目的のヘルプトピックに対応する1文字のコードを入力します。
- 4 ENTER キーを押します。

結果の画面が表示されて選択したヘルプトピックの情報が表示されるか、別のメニューが表示されて対象の絞り込みに役立つヘルプトピックがさらに示されます。

### ヘルプトピックへの直接的なアクセス

Natural ヘルプユーティリティの下位メニューを操作する方法とは別に、Natural コマンドまたは Natural プログラミング言語のヘルプ情報に直接アクセスすることもできます。

#### ▶手順 4.2. ヘルプトピックに直接アクセスするには

- 1 次のシステムコマンドを入力します。

```
HELP command
```

*command* には、システムコマンド EDIT など、ヘルプ情報を表示する対象のコマンドを指定します。

```
HELP EDIT
```

- 2 ENTER キーを押します。

Natural ヘルプユーティリティの [EDIT] 画面が表示され、次のサンプル画面のように要求したコマンドの情報が表示されます。

```
15:20:54          ***** NATURAL HELP UTILITY *****          2008-10-09
                    - EDIT -                                     Page 1
-----
Function: Invoke a Natural editor for the purpose of editing a Natural object.
Parameters:
  Object-type: Copycode(C), Description(D), Global(G), Helproutine(H),
              Local(L), Map(M), Parameter(A), Program(P), Subprogram(N),
              Subroutine(S), Text(T) or Class(4).
              -> Must be specified if object-name is not specified.
  Object-name: The name of the object to be edited, which will then be
              loaded in the edit work area.
  Library-ID : May only be specified if the object is contained in a library
              other than the one to which you are currently logged on.
              -> Must not start with 'SYS' except 'SYSTEM'
              -> Must not be specified if Natural Security is installed.
-----
                                                    More ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit                -      +                               Canc
```

ヘルプのテキストが画面に収まらない場合、画面の右上隅に [Page 1] が表示され、画面の右下隅には [More ...] が表示されます。

- 3 テキストをスクロールダウンするには、ENTER キーまたは PF8 キーを押します。

テキストの最終ページは、画面の右下隅に表示される [Last page] で示されます。



テキストをスクロールアップしたり最初のページ (Page 1) に戻ったりする場合は、PF7 キーを押します。

## エラーメッセージの詳細情報

このセクションでは、Natural システムエラーメッセージとユーザー定義メッセージについて説明します。

Natural が発行するシステムメッセージの場合、先頭に NAT と 4 桁の数字 *nnnn* が付けられます。

各 Natural エラーメッセージには、短文テキストと長文テキストがあります。

- 短文テキストはエラー発生時に表示される 1 行のメッセージです。
- 長文テキストは、エラーとその対処方法の詳細な説明です。

Natural でシステムメッセージが発行された場合、エラー番号と短文テキストのみが画面に表示されます。システムコマンド HELP を使用すると、メッセージの長文テキストも表示できます。

このセクションでは、以下のトピックについて説明します。

- [長文メッセージテキストの表示](#)
- [最後のエラーの情報](#)

### 長文メッセージテキストの表示

このセクションでは、Natural システムエラーメッセージまたはユーザー定義メッセージの長文メッセージテキストを表示する手順を説明します。また、最後に発生した Natural エラーの長文テキストを表示する[手順](#)も参照してください。

#### ▶手順 4.3. Natural システムメッセージの長文テキストを表示するには

- 1 次のシステムコマンドのいずれかを入力します。

```
HELP NATnnnn
```

または

```
? nnnn
```

*nnnn* は 4 桁のエラー番号です。

- 2 ENTER キーを押します。

Natural ヘルプユーティリティの [Natural System Message] 画面に、要求したエラーの長文テキストが表示されます。この画面は、次のセクションに示す Natural システムエラー NAT0082 の[サンプル画面](#)に似ています。

▶手順 4.4. ユーザー定義メッセージの長文テキストを表示するには

- 1 目的のユーザー定義メッセージが保存されているライブラリにログオンします。
- 2 次のシステムコマンドを入力します。

```
HELP USER nnnn
```

または

```
? U nnnn
```

*nnnn* は 4 桁のエラー番号です。

- 3 ENTER キーを押します。

Natural ヘルプユーティリティの **[User Message]** 画面に、要求したユーザー定義メッセージの長文テキストが表示されます。この画面は、次のセクションに示す Natural システムエラー NAT0082 の[サンプル画面](#)に似ています。

### 最後のエラーの情報

現在の Natural セッションで最後に発生したエラーに関して、短文または長文のエラーメッセージテキストを表示するには、以下で説明するように、コマンド `HELP ERROR` を使用するかカーソルを使用します。

最近発行された複数のエラーメッセージの短文テキスト、およびエラー状況に関する追加的な情報をリストするには、システムコマンド `LASTMSG` を使用します。表示される情報には、最後のメッセージの前に発生した関連のエラーメッセージも含まれます。

▶手順 4.5. 最後に発生したエラーの長文テキストを表示するには

- 次のいずれかの方法を使用します。

1. 次のシステムコマンドを入力します。

```
HELP ERROR
```

ENTER キーを押します。

Natural ヘルプユーティリティの [Natural System Message] 画面には、次のサンプル画面に示すように最後に発生した Natural エラーの長文テキストが表示されます。

```
15:18:27          ***** NATURAL HELP UTILITY *****          2003-02-25
Library SAG      - Natural System Message NAT0082 -          Page 1

      Invalid command, or Program ANTON does not exist in library.

Tx *** Short Text ***

      Invalid command, or ... .. does not exist in library.

Ex *** Explanation ***

      One of the following has occurred:

      - You entered a value in the command line which is neither a
        Natural command nor the name of a Natural program contained
        in the active library or in a library defined as a steplib.

      - An object which is required during execution of a program,
        subprogram, subroutine or helproutine is not contained in

                                                    More ...
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit          Print          +          Canc
```

2. または :

メッセージ行にカーソルを表示して PF1 (ヘルプ) キーを押します。この代替的な方法を実行できるのは、メニュー画面にメッセージ行が表示された場合のみです。


次の例のような [Current Natural Message] ウィンドウが表示されます。

```
11:04:30          ***** NATURAL *****          2006-12-21
User SAG          - Main Menu -          Library TEST

+----- Current Natural Message NAT0082 -----+
! Sh Invalid command, or Program ANTON does not exist in library.
!
!
! Tx Invalid command, or ... .. does not exist in library.
!
! Ex One of the following has occurred:
!
!   - You entered a value in the command line which is neither a
!     Natural command nor the name of a Natural program contained
!     in the active library or in a library defined as a steplib.
!
!   - An object which is required during execution of a program,
!     subprogram, subroutine or helproutine is not contained in
!     the active library or in a library defined as a steplib.
!
!   - Your Natural session is currently applying system files other
!     than those containing the object you specified.
!
! Ac Check to ensure that you entered a valid Natural command or the name
!   of an existing Natural object. Use the command SYSPROF to check
!   whether you are using the correct system file.
!
+-----+
```

```
NAT0082 Invalid command, or Program ANTON does not exist in library.
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exit                                                                    Canc
```

 **注意:** ユーザー出口ルーチン USR2002P (『オペレーション』ドキュメントを参照) では、[Current Natural Message] ウィンドウをカスタマイズできます。

#### ▶手順 4.6. 最後のエラーメッセージと詳細を表示するには

- 1 次のシステムコマンドを入力します。

```
LASTMSG
```

- 2 ENTER キーを押します。

[LASTMSG] ウィンドウが開いて、最後に発行されたエラーメッセージの短文テキストが表示されます。

- 3 エラーメッセージの詳細情報が表示される [Detailed Information] ウィンドウを開くには

[LASTMSG] ウィンドウで、追加的な情報を必要とする対象のメッセージの行にカーソルを表示して ENTER キーを押します。

次の例のような [Detailed Information] ウィンドウが表示されます。

```
+----- LASTMSG -----More: ++
! ERRST1  0080 NAT0917 Error 920 in COPYCODE ERRST2.      !
! ERRST2  0090 NAT0917 Error 920 in COPYCODE ERRST3.      !
! ERRST3  0070 NAT0920 Program HUGO cannot be loaded (00000004). !
! +--- Detailed Information for NAT0920 ---+                !
! ! Error Number .. 920                                     !
! ! Error Line ... 70                                       !
! ! Object ..... ERRST3                                     !
! ! Object Type ... Copycode                                !
! ! Level ..... 5                                           !
! ! Library ..... SYSEXV                                    !
! ! DBID/FNR ..... 10 / 410                                  !
! ! Error Class ... System                                  !
! ! Error Type ... Runtime                                  !
! ! Error Time ... 2003-02-27 15:58:01                       !
! +-----+                                                !
! ! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--- !
! ! Exit -- ++ << > Can !
```

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

## メニューとフィールドの詳細情報

---

Natural では、Natural ヘルプユーティリティのヘルプトピックに加えて、利用可能なすべての Natural メニューと入力フィールドに関して、個別にヘルプ情報を参照できます。

### ▶手順 4.7. メニューのヘルプを表示するには

- コマンド行または [Code] フィールドに、疑問符 (?) を入力して ENTER キーを押します。

または:

PF1 (ヘルプ) キーを押します。

### ▶手順 4.8. フィールドのヘルプを表示するには

- 1 目的のフィールドの先頭にカーソルを表示して疑問符 (?) を入力します。
- 2 ENTER キーを押します。

# 5 Natural ライブラリの使用

---

▪ ライブラリに含まれるオブジェクト .....	28
▪ ライブラリタイプ .....	28
▪ Steplib ライブラリ .....	29
▪ オブジェクト実行の検索順序 .....	31
▪ デフォルトのライブラリ割り当て .....	32
▪ ライブラリへのログオン .....	33
▪ ライブラリの作成 .....	34
▪ ライブラリのリスト .....	34
▪ ライブラリ内のオブジェクトのリスト .....	36
▪ オブジェクトのリストの印刷 .....	38
▪ ライブラリ内のオブジェクトの検索 .....	39
▪ ライブラリのコピー、移動、および名前変更 .....	44
▪ ライブラリ内のオブジェクトの削除 .....	46

Natural システムファイルの FNAT と FUSER に保存される Natural オブジェクトは、ライブラリと呼ばれる論理的な構造にグループ化されます。Natural ライブラリは、機能の基準に従ってオブジェクトを整理するために利用されます。例えば、ライブラリは、特定のタスクを実行するオブジェクトのセットで構成できます。Natural Security がインストールされている場合、ライブラリを使用して、管理者など、特定のユーザーグループにオブジェクトへのアクセスを制限することもできます。

Natural アプリケーションは、環境のセットアップ方法に基づいて、複数のライブラリにあるオブジェクトにアクセスできます。

ライブラリに対するすべての操作は、Natural メニュー機能またはその対応するコマンドで実行します。「[コマンドとメニュー機能の使用](#)」も参照してください。

関連トピック：

- [Natural システムファイル - 『Natural システムアーキテクチャ』ドキュメント](#)
- [システムファイル内のライブラリ - 『Natural システムアーキテクチャ』ドキュメント](#)
- [ライブラリ管理 - 『Natural Security』ドキュメント](#)

## ライブラリに含まれるオブジェクト

---

ライブラリには、Natural オブジェクトのうち、一部のタイプのオブジェクトモジュール（該当する場合は、ソースオブジェクトとカタログ化オブジェクト）が含まれます。含まれるタイプは、プログラム、サブプログラム、サブルーチン、コピーコード、ヘルプルーチン、クラス、テキスト、レコーディング、マップ、ローカルデータエリア、グローバルデータエリア、パラメータデータエリア、コマンドプロセッサソース、エラーメッセージ、ダイアログ、デバッグ環境、アダプタ、およびリソースです。

関連トピック：

- [カタログ化オブジェクト](#)
- [ソースオブジェクト](#)

## ライブラリタイプ

---

このセクションでは、提供される Natural ライブラリのタイプについて説明します。

- [ユーザーライブラリ](#)



## ■ システムライブラリ

### ユーザーライブラリ

ユーザーライブラリにはユーザーアプリケーションのオブジェクトが含まれます。これらのオブジェクトはそのアプリケーションに特有であり、そのアプリケーションの実行に必要なオブジェクトです。ユーザーライブラリは、FUSER システムファイルに含まれています。


### システムライブラリ

システムライブラリは、Software AG によってメンテナンスされます。このライブラリは FNAT システムファイルに含まれています。

システムライブラリには、Natural システムアプリケーションの実行に必要なすべてのオブジェクトが含まれています。

さらにプログラムタイプのオブジェクトを含むシステムライブラリがあります。これらは、デモンストレーション用に提供されています。「Natural のメインメニュー」の「[サンプルライブラリ](#)」を参照してください。

システムライブラリの SYSLIB と SYSLIBS にはログオンできないので注意してください。これらは、Software AG で内部使用するために予約されています。これらのライブラリにログオンしようとする、「ライブラリ SYSLIB および SYSLIBS は、システムが使用するためログオンできません。」というメッセージが表示されます。

 **重要:** ユーザー固有のオブジェクトは、Natural システムライブラリに保存しないでください。ユーザー固有のオブジェクトは、Software AG インストールデータセット（例えば、システムメンテナンスアップグレード、新バージョン、または修正テープ）をシステムファイルにロードするときに上書きされる可能性があります。

## Steplib ライブラリ

steplib は、Natural のユーザーライブラリまたはシステムライブラリで、現在のユーザーライブラリまたはシステムライブラリと連結されます。steplib は、さまざまなアプリケーションで共有する複数のオブジェクトを一元的に保存する場所として使用できます。これにより同一のオブジェクトを重複して保存することを防止できるので、アプリケーションを編成するのに役立ちます。

steplib は、現在のライブラリ内でオブジェクトが見つからないときに検索されるライブラリです（次のセクションも参照）。標準の steplib は、FUSER および FNAT システムファイルのライブラリ SYSTEM です。

『パラメータリファレンス』ドキュメントの関連トピック：

- [STEPLIB - 追加の Steplib ライブラリ](#)
- [BPSFI - バッファプール内のオブジェクトを最初に検索](#)

このセクションでは、以下のトピックについて説明します。

- [Natural Security を使用した追加の Steplib](#)
- [Natural Security を使用しない追加の Steplib](#)
- [Steplib 設定のチェック](#)

### Natural Security を使用した追加の Steplib

Natural Security がアクティブの場合は、各ライブラリのセキュリティプロファイルに `steplib` を追加定義できます。ライブラリのセキュリティプロファイルにエントリを追加すると、Natural Security の外部で定義されたエントリは上書きされます。標準の `steplib SYSTEM` (FUSER および FNAT) より先にこれらの `steplib` でオブジェクトが検索されます。

関連トピック：

- [Steplibs - 『Natural Security』ドキュメント](#)

### Natural Security を使用しない追加の Steplib

Natural Security がアクティブでない場合は、セッションの開始時に、Natural プロファイルパラメータ `STEPLIB` で追加の `steplib` を1つ指定できます。また、`steplib` を定義するために Natural のシステムライブラリ `SYSEXT` に用意されているアプリケーションプログラミングインターフェイスのいずれか (`USR1025N` や `USR3025N` など) を使用して、`steplib` をさらに定義できます。

標準の `steplib SYSTEM` (FUSER および FNAT) より先にこれらの `steplib` でオブジェクトが検索されます。

### Steplib 設定のチェック

#### ▶手順 5.1. 現在のライブラリの `steplib` 設定をチェックするには

- 1 次のシステムコマンドを入力します。

```
TECH
```

[TECH] ウィンドウが表示されます。


- 2 [Steplib] 列が表示されるまで `ENTER` キーを繰り返し押して、ウィンドウをスクロールダウンします。

関連トピック：

- TECH-『システムコマンド』ドキュメント

## オブジェクト実行の検索順序

このセクションでは、要求されたオブジェクトが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から実行される場合に、そのオブジェクトを探してNatural ライブラリおよびシステムファイルが検索される順序について説明します。

 **注意:** プロファイルパラメータ BPSFI が ON に設定されている場合（デフォルト設定は OFF）、最初にバッファプール内のオブジェクトが検索されます。

ユーザーライブラリからユーザー記述オブジェクトが実行される場合、検索順序は次のようになります。

1. FUSER システムファイル内の、システム変数 \*LIBRARY-ID によって定義された現在のライブラリ。
2. 現在のライブラリを定義する Natural Security プロファイルまたは STEPLIB テーブルで指定された steplib（指定された順に）。
3. システム変数 \*STEPLIB で定義されたデフォルトの steplib。
4. FUSER システムファイル内のライブラリ SYSTEM。
5. FNAT システムファイル内のライブラリ SYSTEM。

システムライブラリから **Natural** オブジェクトが実行される場合、検索順序は次のようになります。

1. FNAT システムファイル内の、システム変数 \*LIBRARY-ID によって定義された現在の "SYS" ライブラリ。
2. 現在のライブラリを定義する Natural Security プロファイルまたは STEPLIB テーブルで指定された steplib（指定された順に）。
3. FNAT システムファイル内のライブラリ SYSLIBS（システムコマンドおよびユーティリティによって共有されているオブジェクトが含まれている）。
4. FNAT システムファイル内のライブラリ SYSTEM。
5. FUSER システムファイル内のライブラリ SYSTEM。

FUSER システムファイルが最後に検索されるため、FUSER および FNAT システムファイルの双方で使用されるオブジェクト（Natural ユーティリティのユーザー出口ルーチンなど）は一方のロケーション（FUSER）にのみ配置してください。

『システム変数』ドキュメントの関連トピック：

- *\*LIBRARY-ID*
- *\*STEPLIB*

## デフォルトのライブラリ割り当て

---

Natural セッションを開始すると、ユーザーは Natural によって割り当てられたライブラリにログオンします。Natural 画面では、通常、ライブラリの名前 (ID) を示すフィールド (例えば、**[Library]** または **[LIB=]**) が表示されます。つまりこのライブラリが、Natural オブジェクトが格納される現在のライブラリ (例えば、LIB=SYSTEM) であり、オブジェクトはこのライブラリから取得されます。

### ▶手順 5.2. 現在のライブラリを確認するには

- 次のシステムコマンドを入力します。

```
TECH
```

**[TECH]** ウィンドウが表示されます。現在ログオンしているライブラリは、**[Library]** フィールドにリストされます。

Natural プロファイルパラメータの **AUTO** が **OFF** に設定されている場合、Natural が割り当てるデフォルトのライブラリ ID は **SYSTEM** になります。 **AUTO** を **ON** に設定すると、セッションの開始時にデフォルトライブラリ ID を変更できます。 **AUTO=ON** は、ユーザー ID に対応するライブラリに自動的にログオンすることを表します。ただし、『*Natural Security*』ドキュメントで説明するように、Natural Security で Natural にログオンする場合は別のルールが適用されます。

「[ライブラリへのログオン](#)」で説明するように、別のアプリケーションでメンテナンス機能を実行するか作業を行う場合、状況によっては別のライブラリにログオンする必要があります。

関連トピック：


- *TECH* - 『システムコマンド』ドキュメント
- *AUTO* - 『パラメータリファレンス』ドキュメント
- ログオン - 『*Natural Security*』ドキュメント

## ライブラリへのログオン

特定のライブラリで、Natural オブジェクトの作成または管理を行うとき、あるいは Natural プログラムを実行するときには、ライブラリを切り替えたうえで、そのオブジェクトを含んでいる（またはこれから含める）ライブラリや `steplib`（「[Steplib ライブラリ](#)」を参照）に最初にログオンする必要があります。

ライブラリにログオンするには、通常、システムコマンド `LOGON` を使用します。ただし、『*Natural Security*』ドキュメントで説明するように、Natural Security で Natural にログオンする場合は別のルールが適用されます。

オブジェクトが何もないライブラリに正常にログオンすると、「このライブラリは空です。」という確認メッセージが表示されます。

 **注意:** システムコマンド `LOGON` は、タイプがエラーメッセージやデバッグ環境になっているオブジェクトを認識しません。したがってこの確認メッセージは、エラーメッセージとデバッグ環境を除いたその他のオブジェクトタイプについてのみ報告します。ライブラリにエラーメッセージやデバッグ環境が含まれているかどうかを確認する場合は、`SYSMAIN` ユーティリティ、またはエラーメッセージには `SYSERR` ユーティリティ、デバッグ環境にはデバッグをそれぞれ使用することができます。

ライブラリが空であるかどうかを確認する場合（エラーメッセージおよび／またはデバッグ環境を除く）、「[ライブラリ内のオブジェクトのリスト](#)」で説明しているシステムコマンド `LIST` を使用することができます。

別のライブラリにログオンしたり、Natural セッションを終了したりするまで、ログオンしたライブラリはアクティブの状態に保たれます。

### ▶手順 5.3. ライブラリにログオンするには

- 1 次のシステムコマンドを入力します。

```
LOGON library-ID
```

*library-ID* はアクセスするライブラリの名前 (ID) です。

または：

Natural のメインメニューまたは下位のメニューでは次の手順に従います。

画面右上隅の [**Library**] フィールドで、表示されているライブラリ ID を別のライブラリ ID に変更します（「[コマンドとメニュー機能の使用](#)」の「[メニュー機能の例](#)」も参照）。

- 2 `ENTER` キーを押します。

入力したライブラリ ID が [ライブラリの命名規則](#) に準拠していた場合、「ライブラリ *library-ID* へのログオンができます。」という確認メッセージが表示されます。

関連トピック：

- LOGON および LOGOFF - 『システムコマンド』ドキュメント
- ログオン - 『Natural Security』ドキュメント

## ライブラリの作成

---

ライブラリを作成する場合、システムコマンド LOGON で空のライブラリにログオンしてオブジェクトを作成するか、オブジェクトの転送時に Natural ユーティリティの移動またはコピー機能を使用します。

### ▶手順 5.4. ライブラリを作成するには

- システムコマンド LOGON を使用する場合、「[ライブラリへのログオン](#)」の手順に従います。オブジェクトが何もないライブラリにログオンするためのライブラリ ID を指定します（「[ライブラリの命名規則](#)」も参照）。このライブラリで、少なくとも1つのカタログ化オブジェクトまたは1つのソースオブジェクトを作成します。

または:

既存のライブラリから新しいライブラリにオブジェクトを移動またはコピーする場合は、「[ライブラリのコピー、移動、および名前変更](#)」の手順に従います。

## ライブラリのリスト

---

例えば、SYSMAIN ユーティリティを使用すると、現在の Natural システム環境で利用可能なすべてのライブラリのリストを表示できます。

このセクションの手順では、SYSMAIN メニュー機能またはその対応するコマンドを使用してライブラリをリストする例を示します。

### ▶手順 5.5. メニュー機能を使用してすべてのライブラリをリストするには

- 1 Natural の [Main Menu] の [Maintenance and Transfer Utilities] を選択します。
- 2 ENTER キーを押します。

[Maintenance and Transfer Utilities] 画面が表示されます。

- 3 [Transfer Objects to Other Libraries] を選択します。

または:

次のシステムコマンドを入力します。

```
SYSMAIN
```

- 4 ENTER キーを押します。

次の例のような SYSMAIN ユーティリティのメインメニューが表示されます。

```
17:33:01          ***** NATURAL SYSMAIN UTILITY *****          2005-05-30
User SAG                - Main Menu -

      Code  Object                                Code  Function
      A    Programming Objects                    C    Copy
      D    Debug Environments                      D    Delete
      E    Error Message Texts                    F    Find
      P    Profiles                                L    List
      R    Rules                                    M    Move
      S    DL/I Subfiles                           R    Rename
      V    DDMs                                     ?    Help
      ?    Help                                    .    Exit
      .    Exit

Object Code .. A          Function Code .. _

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren
```

- 5 エラーメッセージとデバッグ環境以外の Natural オブジェクトを含んでいるライブラリをすべてリストするには、**[Object Code]** フィールドに「A」（デフォルト設定）を入力します。エラーメッセージの場合は「E」、デバッグ環境の場合は「D」を入力します。

**[Function Code]** フィールドに「L」（**[List]** を表す）を入力します。

- 6 ENTER キーを押します。

**[List Programming Objects]** 画面が表示されます。

- 7 すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を含むライブラリを検索するには、**[Code]** フィールドに「A」を入力します。

すべてのライブラリを検索するには **[Source Library]** フィールドにアスタリスク (\*) を入力します

(有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「[名前の範囲の指定](#)」を参照してください)。

その他のすべての入力フィールドは変更しません。

- 8 ENTER キーを押します。

〔**Library Selection**〕画面に、現在のシステムファイルで利用できるすべてのライブラリのリストが表示されます。

リストのスクロールダウンには ENTER キー、右スクロールには PF11 キー、左スクロールには PF12 キーを押します。

〔**Library Selection**〕画面から、特定のライブラリを選択してそのライブラリに含まれるオブジェクトをリストできます。〔**Library Selection**〕画面の詳細については、下記の[関連トピック](#)を参照してください。

### ▶手順 5.6. コマンドを使用してすべてのライブラリをリストするには

- 1 次のシステムコマンドを入力します。

```
SYSMAIN LIST ALL * IN LIBRARY *
```

- 2 ENTER キーを押します。

〔**Library Selection**〕画面に、現在のシステムファイルで利用できるすべてのライブラリのリストが表示されます。


関連トピック：

- ライブラリのリストと選択 - 『SYSMAIN ユーティリティ』ドキュメント

## ライブラリ内のオブジェクトのリスト

---

ライブラリに含まれているオブジェクトのリストを取得するには、システムコマンド LIST を使用するか、Natural ユーティリティ SYSMAIN またはオブジェクトハンドラの対応する機能を使用します。

 **注意:** システムコマンド LIST は、タイプがエラーメッセージやデバッグ環境になっているオブジェクトを処理しません。エラーメッセージやデバッグ環境は、SYSMAIN ユーティリティなどを使用してリストできます。

このセクションでは、LIST を使用してオブジェクトのリストを取得する方法について説明します。



## ▶手順 5.7. LIST を使用してオブジェクトをリストするには

- 1 次のシステムコマンドを入力します。

```
LIST *
```

この場合、アスタリスク (\*) は、すべてのソースオブジェクトとカタログ化オブジェクトをリストすることを表します

(『システムコマンド』ドキュメントの「LIST」の「*object-name-range*」も参照)。

- 2 ENTER キーを押します。

次の例のような「**LIST Objects in a Library**」画面が表示され、現在のシステムファイルにある現在のライブラリ (次の例では TEST) を対象として、そこに含まれているソースオブジェクトとカタログ化オブジェクトがリストで表示されます。

```
12:22:23          ***** NATURAL LIST COMMAND *****          2005-05-09
User SAG          - LIST Objects in a Library -          Library TEST

Cmd  Name      Type      S/C  SM  Version  User ID  Date      Time
---  *          *          *    *  *          *          *          *
___  DBTEST     Program   S/C  S   3.1.05   SAG      2002-02-27 16:46:36
___  DEMOPROG   Program   S/C  S   3.1.04   SAG      2000-12-20 11:40:46
___  DEMOSPRG   Subprogram S/C  S   3.1.04   SAG      2000-12-22 11:32:34
___  DEMOTEXT   Text      S      3.1.04   SAG      2000-12-20 13:59:26
___  EMP-L      Local     S/C      3.1.06   SAG      2002-07-29 12:46:43
___  EMPL       Program   S/C  S   3.1.06   SAG      2002-07-29 12:46:30
___  GDATEST    Global    S/C      3.1.06   SAG      2004-11-22 13:24:32
___  LDATEST    Local     S      4.1.03   SAG      2004-05-21 17:18:13
___  MAPTEST    Map       S/C  S   3.1.04   SAG      2000-12-20 11:40:46
___  PDATEST    Parameter S/C      4.1.02   SAG      2004-05-05 11:46:24
___  PDATEST2   Parameter S/C      4.1.03   SAG      2004-11-23 10:12:07
___  PROCTEST   Processor S/C      4.1.02   SAG      2004-05-05 11:25:22
___  SUBRTEST   Subroutine S/C  S   4.1.00   SAG      2002-07-23 14:37:16
___  TESTPGM    Subprogram S      S   3.1.04   SAG      2000-12-20 11:40:45
                                         14 Objects found

Top of List.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit Sort      --  -  +  ++      >  Canc
```

このリストはオブジェクト名 ([Name] 列) でソートされ、オブジェクトのタイプ (例えば、Program)、使用できるオブジェクトモジュールのタイプ ([S/C] 列: sはソースオブジェクト、cはカタログ化オブジェクト)、オブジェクトを作成または変更したユーザーのIDなど、オブジェクトに関する詳細な情報も表示されます。

「**LIST Objects in a Library**」画面では、オブジェクトを選択してさらに処理することができます。例えば、オブジェクトのソースコードをリストしたり、「ライブラリ内のオブジェクトの削

除」の「**LIST** を使用してオブジェクトを削除するには」で説明するように、ライブラリからオブジェクトを削除することができます。

[**LIST Objects in a Library**] 画面の詳細、および、特定範囲のオブジェクト名のリストやオブジェクトのリストのソートなど、LISTのその他のオプションの詳細については、次の関連トピックを参照してください。

『システムコマンド』ドキュメントの **LIST** 関連トピック：

- *object-name-range*
- 列ヘッダーの説明
- オブジェクトのリストのソート
- オブジェクトへの機能の実行

## オブジェクトのリストの印刷

---

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オブジェクトのリストを印刷できます。

### ▶手順 5.8. オブジェクトのリストを印刷するには

- 1 「ライブラリ内のオブジェクトのリスト」の**手順 1~2**で説明しているように、[**LIST Objects in a Library**] 画面を表示します。
- 2 PF2 キー（印刷）を押します。  
  
[**PRINT**] ウィンドウが表示されます。
- 3 [**Destination**] フィールドに有効なプリンタ名を入力します（現在の環境で利用できるプリンタについては、必要に応じて Natural の管理者にお問い合わせください）。必要に応じて、ページサイズを変更します（デフォルト設定は 60 行です）。
- 4 ENTER キーを押します。  
  
[**Printout Specification**] 画面が表示されます。この画面では、印刷部数などプリンタの設定を行うことができます。
- 5 ENTER キーを押します。  
  
ライブラリに含まれているオブジェクトのリストが、指定したプリンタデバイスで印刷されます。

関連トピック：

- **オブジェクトの印刷**

## ■ LIST - 『システムコマンド』ドキュメント

## ライブラリ内のオブジェクトの検索

SYSMAIN ユーティリティまたはシステムコマンド SCAN を使用すると、ライブラリに含まれているオブジェクトを検索できます。

- SYSMAIN を使用したオブジェクトの検索
- SCAN を使用したオブジェクトの検索

### SYSMAIN を使用したオブジェクトの検索

以下の手順では、SYSMAIN メニュー機能やその対応するコマンドに検索条件を指定し、オブジェクトを検索する例を示します。

#### ▶手順 5.9. メニュー機能を使用してオブジェクトを検索するには

- 1 「メニュー機能を使用してすべてのライブラリをリストするには」の手順 1~4 に従って、SYSMAIN ユーティリティの [Main Menu] を表示します。
- 2 すべてのタイプのオブジェクトを検索するには、[Object Code] フィールドに「A」（デフォルト設定）を入力します。

（例外：エラーメッセージの場合は「E」、デバッグ環境の場合は「D」。）

[Function Code] フィールドに「F」（Find）を入力します。

- 3 ENTER キーを押します。

[Find Programming Objects] 画面が表示されます。

- 4 すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を検索するには、[Code] フィールドに「A」を入力します。

すべてのオブジェクト名を検索するには、[ObjectName] フィールドにアスタリスク（\*）を入力します。アスタリスク（\*）はデフォルト設定です。

（有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」を参照してください。）

[Source Library] フィールドに、検索を実行するライブラリの ID を入力します。

[Criteria] フィールドで、N（いいえ）を Y（はい）に変更します。N はデフォルト設定です。

その他のすべての入力フィールドは変更しません。

5 ENTER キーを押します。

以下の例に示すように、[Additional Criteria] ウィンドウが表示されます。この画面では、追加の検索条件を入力できます。

```

18:23:16          ***** NATURAL SYSMAN UTILITY *****          2005-05-09
User SAG              - Find Programming Objects -

      +-----+
      !      --- Additional Criteria ---      !
      !                                         !
      ! Object Type ..... PM_____          !
      ! Date/Time From .. 2002-05-09 _____ !
      ! Date/Time To .... 2005-05-01 _____ !
      ! User ID ..... SAG_____            !
      ! Terminal ID ..... _____         !
      !                                         !
      !                                         !
      ! Command ===>                          !
      !                                         !
      Code .. !                                !
      Object Name .. +-----+
      Source Library ... TEST_____ Database .... 10___ File .. 32___
      Options                      Criteria .... Y

Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren Fsec Fdic Fnat
    
```

- 6 1つ以上の検索条件を入力して、ENTER キーを押します。

ウィンドウが閉じます。[Criteria] フィールドの前にあるプラス (+) 記号は、オブジェクトタイプ以外の追加の条件がウィンドウで指定されていることを表します。

- 7 もう一度 ENTER キーを押します。

[Find Selection] 画面が表示され、現在のシステムファイル内の指定したライブラリに含まれており、さらに、指定した検索条件に一致したソースオブジェクトとカタログ化オブジェクトがすべて表示されます。上記の例では、タイプが P (プログラム) および M (マップ) であり、2002 年 5 月 9 日~2005 年 5 月 1 日の期間に、ユーザー SAG が作成または変更した全オブジェクトがリストされます。

- 8 「ライブラリが正常にリストされました。」というメッセージが表示されるまで、Enter キーを押してリストをスクロールダウンします。

[Find Selection] 画面では、オブジェクトを選択してさらに処理することができます。例えば、『SYSMAIN ユーティリティ』ドキュメントの「プログラミングオブジェクトの選択リスト」で説明しているように、オブジェクトのソースコードをリストすることができます。

関連トピック：

- メニュー機能とコマンドの使用 - 『SYSMAIN ユーティリティ』ドキュメント

#### ▶手順 5.10. コマンドを使用してオブジェクトを検索するには

- 1 次のシステムコマンドを入力します。

```
SYSMAIN FIND ALL * TYPE PN WITH USER user-ID IN library-ID
```

上記の意味は次に示すとおりです。

ALLは、すべてのソースオブジェクトとカタログ化オブジェクトを検索対象として選択することを表します。

アスタリスク (\*) は、すべてのオブジェクト名を検索対象として選択することを指定しています (『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」も参照)。

PNは、検索するオブジェクトのタイプを指定しています。Pはプログラム、Nはサブプログラムを表しています。

*library-ID* は検索するライブラリの ID です。

*user-ID* は検索するユーザーの ID です。

- 2 ENTER キーを押します。


[Find Selection] 画面が表示され、タイプがプログラムおよびサブプログラムになっているソースオブジェクトとカタログ化オブジェクトのうち、指定したユーザーが作成または変更したオブジェクトがすべてリストされます。

関連トピック：

- [ダイレクトコマンドのキーワードと変数 - 『SYSMAIN ユーティリティ』ドキュメント](#)

### SCAN を使用したオブジェクトの検索

以下の手順では、システムコマンド SCAN を使用してソースをスキャンし、特定の文字列を探すことでオブジェクトを検索する例を示します。

 **注意:** SCAN は、タイプがエラーメッセージやデバッグ環境になっているオブジェクトを処理しません。Natural ユーティリティ SYSERR の **Scan in messages** 機能を使用すると、エラーメッセージをスキャンできます。

#### ▶手順 5.11. メニュー機能を使用してソースをスキャンするには

- 1 次のシステムコマンドを入力します。

```
SCAN
```

- 2 ENTER キーを押します。

次の例のような **[Scan Objects in Libraries]** 画面が表示されます。この画面では、オブジェクトの範囲とスキャン対象値を指定できます。

```
18:24:53          ***** NATURAL SCAN COMMAND *****          2005-05-10
User SAG          - Scan Objects in Libraries -          Library TEST

          Code  Function
          T    Statistics
          L    List of Objects Containing Scan Value
          S    Object Lines with Scan Value
          ?    Help
          .    Exit

Code ..... L
Scan value ..... LOCAL_____
Replace value ... GLOBAL_____
Library ..... TESTLIB_
Object name ..... *_____ Selection list .. N
Object type(s) .. *_____
Absolute scan ... N          Trace ..... N

Command ==>
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit
```

〔**Code**〕 フィールドには「L」（ソースオブジェクトのリスト）を入力してスキャン対象値、ライブラリ、およびオブジェクトの範囲を指定します。アスタリスク（\*）を指定すると、すべてのオブジェクト名とオブジェクトタイプが選択されます。

〔**Replace value**〕 フィールドには、指定されたスキャン対象値（例えば、LOCAL）を置換するための文字列（例えば、GLOBAL）を入力できます。

- 3 ENTER キーを押します。

現在のシステムファイル内の指定したライブラリを対象とした〔**Select Objects for Library**〕画面が表示され、指定した検索条件に一致したすべてのソースオブジェクトがリストで表示されます。上記の例では、ライブラリ TESTLIB 内にあり、文字列 LOCAL を含む全オブジェクトが選択されます。

〔**Select Objects for Library**〕画面では、オブジェクトの選択、スキャン対象値を含むソース行の表示や変更を実行できます。

#### ▶手順 5.12. コマンドを使用してソースをスキャンするには

- 1 次のシステムコマンドを入力します。

```
SCAN FUNC=L,SVAL=scan-value,LIB=library-ID,*
```

上記の意味は次に示すとおりです。

*scan-value* はソースをスキャンするための文字列（例えば、LOCAL）です。

*library-ID* は検索するライブラリの ID（例えば、TESTLIB）です。

アスタリスク（\*）は、すべてのソースオブジェクトを検索対象として選択することを表します。

- 2 ENTER キーを押します。

関連トピック：

- SCAN - 『システムコマンド』ドキュメント

## ライブラリのコピー、移動、および名前変更

ライブラリをコピーまたは移動するには、ソースライブラリのすべての Natural オブジェクトを、既存または新規のターゲットライブラリへコピーまたは移動します。

ライブラリの名前を変更する場合、ライブラリの移動と同じ手順を実行する必要があります。ライブラリのすべてのオブジェクトを新しいターゲットライブラリに移動すると、古いライブラリは自動的に削除されます。

SYSMAIN ユーティリティまたはオブジェクトハンドラを使用して、特定のライブラリから別のライブラリへ1つまたは複数のオブジェクトをコピーまたは移動することができます。以下の手順では、メニュー機能や対応するコマンドを使用して、SYSMAINのすべてのオブジェクトをコピーまたは移動する例を示します。

### ▶手順 5.13. メニュー機能を使用してすべてのオブジェクトをコピーまたは移動するには

- 1 「メニュー機能を使用してすべてのライブラリをリストするには」の手順1~4に従って、SYSMAIN ユーティリティの **[Main Menu]** を表示します。
- 2 すべてのタイプのオブジェクトを選択するには、**[Object Code]** フィールドに「A」（デフォルト設定）を入力します。

（例外：エラーメッセージの場合は「E」、デバッグ環境の場合は「D」。）

**[Function Code]** フィールドに「C」（**[Copy]** を表す）または「M」（**[Move]** を表す）を入力します。

- 3 ENTER キーを押します。

入力したファンクションコードに応じて、**[Copy Programming Objects]** または **[Move Programming Objects]** 画面が表示されます。

- 4 すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を選択するには、**[Code]** フィールドに「A」を入力します。

**[Sel. List]**（リストの選択）フィールドで、Y（はい）をN（いいえ）に変更します。Yはデフォルト設定です。

すべてのオブジェクト名を選択するには **[Object Name]** フィールドにアスタリスク（\*）を入力します。アスタリスク（\*）はデフォルト設定です。

（有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「**名前の範囲の指定**」を参照してください。）

**[Source Library]** フィールドに、コピーまたは移動するオブジェクトを含んでいるライブラリの ID を入力します。



[**Target Library**] フィールドに、オブジェクトのコピーまたは移動先となる既存のライブラリまたは新規のライブラリの ID を入力します。

その他のすべての入力フィールドは変更しません。

- 5 ENTER キーを押します。

現在のシステムファイル内の指定したソースライブラリとターゲットライブラリの間で、すべてのソースオブジェクトとカタログ化オブジェクトがコピーまたは移動され、「機能が正しく実行されました。」というメッセージが表示されます。

#### ▶手順 5.14. コマンドを使用してすべてのオブジェクトをコピーまたは移動するには

- 1 次のシステムコマンドのいずれかを入力します。

```
SYSMAIN COPY ALL * FM old-library TO new-library
```

(オブジェクトのコピー)

または

```
SYSMAIN MOVE ALL * FM old-library TO new-library
```

(オブジェクトの移動)

上記の意味は次に示すとおりです。

ALL は、すべてのソースオブジェクトとカタログ化オブジェクトを選択することを表します。

アスタリスク (\*) は、すべてのオブジェクト名を選択することを指定しています (『SYSMAIN ユーティリティ』ドキュメントの「[名前の範囲の指定](#)」も参照)。

*old-library* は、コピーまたは移動するオブジェクトを含んでいるライブラリの ID です。

*new-library* は既存ライブラリまたは新規ライブラリの ID です (「[ライブラリの命名規則](#)」も参照)。

- 2 ENTER キーを押します。

コピーまたは移動が正常に実行された場合、処理されたコマンドとそれに続いて「機能が正しく実行されました。」というメッセージが画面に表示されます。

オブジェクトの置換や名前変更など、SYSMAIN ユーティリティの機能の詳細については、『ユーティリティ』ドキュメントを参照してください。

## ライブラリ内のオブジェクトの削除

Natural 環境では、ライブラリは最低1つのソースオブジェクトまたはカタログ化オブジェクトがある場合にのみ管理対象となります。ライブラリからすべてのオブジェクトを削除すると、そのライブラリはそれ以降使用できなくなります。

システムコマンド LIST または Natural ユーティリティを使用して、ライブラリから1つ以上のオブジェクトを削除することができます。

以下の手順では、システムコマンド LIST を使用して、現在のライブラリからオブジェクトを削除します。また、SYSMAIN ユーティリティのメニュー機能またはコマンドを使用して、別のライブラリからオブジェクトを削除します。

### ▶手順 5.15. LIST を使用してオブジェクトを削除するには

- 1 次のシステムコマンドを入力します。

```
LIST *
```

この場合、アスタリスク (\*) は、すべてのソースオブジェクトとカタログ化オブジェクトをリストすることを表します

(『システムコマンド』ドキュメントの「LIST」の「*object-name-range*」も参照)。

- 2 ENTER キーを押します。

[**LIST Objects in a Library**] 画面が表示され、現在のシステムファイルにある現在のライブラリを対象として、そこに含まれているソースオブジェクトとカタログ化オブジェクトがリストで表示されます。

- 3 [**Cmd**] 列で目的のオブジェクトの横に、次のように入力します。

```
DE
```

次のサンプル画面のようになります。

- 4 ENTER キーを押します。

次の例のような [DELETE] ウィンドウが表示されます。

```

12:22:23          ***** NATURAL LIST COMMAND *****          2005-05-09
User SAG          - LIST Objects in a Library -          Library TEST

Cmd  Name          Type          S/C  SM Version  User ID  Date          Time
---  *          *          *   * *          *          *          *
DE  DBTEST         Pr +-----DELETED-----+ 002-02-27  16:46:36
DE  DEMOPROG       Pr !          ! 000-12-20  11:40:46
DE  DEMOSPRG       Su !          Please select one item: ! 000-12-22  11:32:34
DE  DEMOTEXT       Te !          ! 000-12-20  13:59:26
___ EMP-L          Lo !          _ Confirm each deletion ! 002-07-29  12:46:43
___ EMPL          Pr !          _ Delete without confirmation ! 002-07-29  12:46:30
___ GDATEST       G1 !          _ Exit (no deletion) ! 004-11-22  13:24:32
___ LDATEST       Lo !          ! 004-05-21  17:18:13
___ MAPTEST       Ma +-----+ 000-12-20  11:40:46
___ PDATEST       Parameter     S/C    4.1.02   SAG      2004-05-05  11:46:24
___ PDATEST2      Parameter     S/C    4.1.03   SAG      2004-11-23  10:12:07
___ PROCTEST      Processor     S/C    4.1.02   SAG      2004-05-05  11:25:22
___ SUBRTEST      Subroutine    S/C    S 4.1.00   SAG      2002-07-23  14:37:16
___ TESTPGM       Subprogram    S      S 3.1.04   SAG      2000-12-20  11:40:45
                                   14 Objects found

Top of List.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit Sort      --      -      +      ++      >      Canc

```

- 5 目的のオプションの横に任意の文字を入力して項目をマークし、ENTER キーを押します。

[**Confirm each deletion**] を選択すると、削除するオブジェクトごとにウィンドウが表示されます。このウィンドウでは、削除を確認するために、オブジェクトの名前を入力します。

[**Delete without confirmation**] を選択すると、すぐに削除が実行されます。

[**Exit**] を選択すると、削除がキャンセルされます。

#### ▶手順 5.16. SYSMAIN メニュー機能を使用してオブジェクトを削除するには

- 「メニュー機能を使用してすべてのライブラリをリストするには」の**手順 1~4**に従って、SYSMAIN ユーティリティの [**Main Menu**] を表示します。
- すべてのタイプのオブジェクトを選択するには、[**Object Code**] フィールドに「A」（デフォルト設定）を入力します。  
  
(例外：エラーメッセージの場合は「E」、デバッグ環境の場合は「D」。)  
  
[**Function Code**] フィールドに「D」（[**Delete**] を表す）を入力します。
- ENTER キーを押します。

[Delete Programming Objects] 画面が表示されます。

- すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を選択するには、[Code] フィールドに「A」を入力します。

[Sel. List]（リストの選択）フィールドで、Y（はい）をN（いいえ）に変更します。Yはデフォルト設定です。

すべてのオブジェクト名を選択するには [Object Name] フィールドにアスタリスク（\*）を入力します。アスタリスク（\*）はデフォルト設定です。

（有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」を参照してください。）

[Source Library] フィールドに、削除するライブラリの ID を入力します。

その他のすべての入力フィールドは変更しません。

- ENTER キーを押します。

現在のシステムファイル内の指定したライブラリから、すべてのソースオブジェクトとカタログ化オブジェクトが削除され、「機能が正しく実行されました。」というメッセージが表示されます。

#### ▶手順 5.17. SYSMAIN コマンドを使用してすべてのオブジェクトを削除するには

- 次のシステムコマンドを入力します。

```
SYSMAIN DELETE ALL * FM library-ID
```

上記の意味は次に示すとおりです。

ALL は、すべてのソースオブジェクトとカタログ化オブジェクトを選択することを表します。

アスタリスク（\*）は、すべてのオブジェクト名を選択することを指定しています（『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」も参照）。

*library-ID* は、削除するオブジェクトを含んでいるライブラリの ID です。

- ENTER キーを押します。

削除が正常に実行された場合、処理されたコマンドとそれに続いて「機能が正しく実行されました。」というメッセージが画面に表示されます。

## 6 Natural オブジェクトの管理と実行

---

▪ Natural オブジェクトについて .....	50
▪ Natural エディタまたはユーティリティの使用 .....	52
▪ オブジェクトの選択と表示 .....	55
▪ オブジェクトの作成と編集 .....	56
▪ オブジェクトのチェックとテスト .....	60
▪ オブジェクトの保存とカタログ .....	62
▪ オブジェクトディレクトリ情報の表示 .....	65
▪ オブジェクトのコピー .....	66
▪ オブジェクトの印刷 .....	68
▪ オブジェクトの名前の変更 .....	69
▪ オブジェクトの移動 .....	71
▪ オブジェクトの削除 .....	72
▪ プログラムの実行 .....	74

オブジェクトは、アプリケーションの一種のコンポーネントです。Natural アプリケーションは、特定のタスクを実行するために相互に連動する一群のオブジェクトで構成されています。

Natural アプリケーションのセットアップと管理に利用できるオブジェクトには、Natural オブジェクトと外部オブジェクトがあります。

外部オブジェクトとは、Naturalの開発機能とは異なるツールで作成され、Natural環境やAdabas環境の外部に保存されているオブジェクトのことです。外部オブジェクトの例としては、ビットマップ、XMLソース、HTMLファイル、DL/Iサブファイル、Predictルールなどがあります。

このセクションでは、Naturalオブジェクトの全般的な情報を提供し、オブジェクトの作成、管理、削除、実行に必要な手順について説明します。

Natural オブジェクトに対するすべての操作は、Natural コマンドおよび／またはメニュー機能で実行します。コマンドとメニュー機能の使い方の詳細については、「[コマンドとメニュー機能の使用](#)」を参照してください。

関連トピック：

- [ライブラリ内のオブジェクトのリスト](#)
- [ライブラリ内のオブジェクトの検索](#)
- [ライブラリ内のオブジェクトの削除](#)

## Natural オブジェクトについて

---

Natural オブジェクトであるかどうかは、次の特性でわかります。

- Natural システムファイルに保存されている。
- カタログ化オブジェクトとソースオブジェクトの両方、またはこのいずれかで構成されている。
- Natural エディタまたはユーティリティで作成された。

このセクションでは、以下のトピックについて説明します。

- [カタログ化オブジェクト](#)
- [ソースオブジェクト](#)

## ■ オブジェクトタイプ

### カタログ化オブジェクト

カタログ化オブジェクトとは、Naturalオブジェクトの実行形式（コンパイルされた形式）のことです。カタログ化オブジェクトは、Naturalコンパイラによって作成され、オブジェクトモジュールとしてNaturalシステムファイルに格納されます。ソースコードのコンパイルおよびカタログ化オブジェクトの作成を、オブジェクトのカタログ化と呼びます。カタログ化オブジェクトは、Naturalシステムコマンド CATALOG または STOW を使用して作成されます。

実行時にカタログ化オブジェクトはNaturalバッファプールにロードされ、Naturalランタイムシステムによって実行されます。Naturalオブジェクトは、カタログ化オブジェクトとしてNaturalシステムファイルに格納されている場合にのみ、実行または相互参照が可能となります。

カタログ化オブジェクトを変更または逆コンパイルすることはできません。

### ソースオブジェクト

ソースオブジェクト（保存オブジェクト）には、人間が理解できる形式のNaturalソースコードが含まれています。ソースコードは、Naturalシステムコマンド SAVE または STOW を使用して、Naturalシステムファイルにソースオブジェクトとして保存されます。

ソースオブジェクトに含まれているソースコードを実行するには、ソースコードをコンパイルして、Naturalランタイムシステムが解釈および実行できる生成済みのオブジェクトコードを作成する必要があります。

関連トピック：

- Naturalシステムファイル-『Naturalシステムアーキテクチャ』ドキュメント
- [オブジェクトの保存とカタログ](#)

### オブジェクトタイプ

Naturalアプリケーションでは、いくつかのタイプのNaturalオブジェクトを使用して、効率的なアプリケーション構造を構築し、特定のプログラミング要件やアプリケーション要件に対応することができます。Naturalオブジェクトのタイプには、プログラム、サブプログラム、ルーチン、データエリアがあります。オブジェクトのすべてのタイプについては、『プログラミングガイド』の「オブジェクトタイプ」を参照してください。

データ定義モジュール（DDM）の詳細については、『プログラミングガイド』の「Naturalデータ定義モジュール」を参照してください。

## Natural エディタまたはユーティリティの使用

Natural オブジェクトの作成、管理、削除を実行する場合、Natural エディタまたは Natural ユーティリティを使用します。

メンテナンス機能の中には、一部のオブジェクトで使用できない機能もあります。例えば、アダプタタイプのオブジェクトを編集することはできません。

システムコマンド EDIT または [Development Functions] 画面で指定できるオブジェクトタイプの場合は、Natural エディタを使用することができます。Natural では、プログラムエディタ、データエリアエディタ、マップエディタなど、指定したオブジェクトタイプに応じて適切なエディタが起動されます。例えば、プログラムタイプのオブジェクトの場合はプログラムエディタが起動されます。

Natural ユーティリティは、追加的な管理サービスが必要なオブジェクトタイプや、DDM など、ライブラリでのメンテナンス対象にならないオブジェクトタイプに使用します。ユーティリティには、独自のエディタがあります。

次の表では、Natural のオブジェクトタイプとそのエディタまたはユーティリティを簡単に示します。

オブジェクトタイプ	エディタまたはユーティリティ
ローカルデータエリア グローバルデータエリア パラメータデータエリア	データエリアエディタ
マップ (画面のレイアウト) マップエディタプロファイル デバイスプロファイル	マップエディタ
プログラム サブプログラム サブルーチン コピーコード ヘルプルーチン クラス テキスト プログラムエディタプロファイル	プログラムエディタ
エラーメッセージ	SYSERR ユーティリティ
データ定義モジュール (DDM)	SYSDDM ユーティリティ
コマンドプロセッサソース	SYSNCP ユーティリティ
パラメータプロファイル	SYS Parm ユーティリティ
デバッグ環境	デバッグ



関連トピック：

- 『エディタ』ドキュメント
- 『ユーティリティ』ドキュメント

このセクションでは、以下のトピックについて説明します。

- [Natural エディタの起動](#)
- [Natural ユーティリティの起動](#)
- [エディタ設定の指定](#)

## Natural エディタの起動

### ▶手順 6.1. Natural エディタを起動するには

- システムコマンド EDIT を使用します。

EDIT の使用例については、「[システムコマンドの例](#)」を参照してください。

または:

Natural のメインメニューで [[Development Functions](#)] メニューを表示し（「Natural のメインメニュー」を参照）、[[Create Object](#)] または [[Edit Object](#)] 機能を選択します。

エディタを起動する例については、「[コマンドとメニュー機能の使用](#)」の「[メニュー機能の例](#)」を参照してください。

関連トピック：

- [EDIT - 『システムコマンド』ドキュメント](#)

## Natural ユーティリティの起動

### ▶手順 6.2. Natural ユーティリティを起動するには

- 次のシステムコマンドのいずれかを入力します。

```
SYSERR
```

(エラーメッセージの場合)

```
SYSDDM
```

(DDM の場合)

```
SYSNCP
```

(コマンドプロセッサソースの場合)

SYSPARM

(パラメータプロファイルの場合)

TEST

(デバッグ環境の場合)

または:

Natural のメインメニューで目的のメニューを表示し、目的のユーティリティを選択します。

- SYSERR、SYSDDM、および SYSNCP の場合、 [\[Maintenance and Transfer Utilities\]](#)
- SYSPARM の場合、 [\[Development Environment Settings\]](#)
- TEST の場合、 [\[Debugging and Monitoring Utilities\]](#)

関連トピック：

- [Natural のメインメニュー](#)

## エディタ設定の指定

Natural プログラムエディタまたはデータエリアエディタで操作を行う場合、エディタプロファイル機能を使用して、エディタの現在の設定を表示し、ソースコードの編集時に有効にする設定を指定することができます。

### ▶手順 6.3. エディタプロファイル設定を表示または変更するには

- 1 プログラムエディタまたはデータエリアエディタのコマンドプロンプトで、次のように入力します。

PROFILE

- 2 ENTER キーを押します。

[\[Editor Profile\]](#) 画面が表示されます。

画面に表示されるフィールドやオプションの詳細については、『エディタ』ドキュメントの「エディタプロファイル」を参照してください。

## オブジェクトの選択と表示

ソースオブジェクトを変更せずに、ソースコードの表示またはコピーを目的としてソースオブジェクトを表示することができます。指定したオブジェクトのソースコードが、対応するエディタの編集エリアに読み取り専用モードで表示されます。

リストからオブジェクトを選択したり、表示するオブジェクトの名前を指定したりすることができます。

このセクションでは、システムコマンド `LIST` を使用してソースコードをリストする方法について説明します。 `LIST` の代替機能として、「Natural のメインメニュー」で説明している [\[Development Functions\]](#) メニューの [\[List Object\(s\)\]](#) 機能を使用することもできます。

### ▶手順 6.4. オブジェクトのリストからオブジェクトを選択するには

- 1 「`LIST` を使用してオブジェクトをリストするには」の [手順 1~2](#) に従って、[\[LIST Objects in a Library\]](#) 画面を表示します。
- 2 [\[Cmd\]](#) 列で目的のオブジェクトの横に、次のように入力します。

```
LI
```

- 3 ENTER キーを押します。

選択したオブジェクトのソースコードが表示されます。

### ▶手順 6.5. 指定したオブジェクトのソースコードを表示するには

- 1 次のシステムコマンドを入力します。

```
LIST object-name
```

この場合、`object-name` は表示するオブジェクトの名前です。

`object-name` を指定しない場合、ソースワークエリアに現在含まれているソースコードが表示されます。

- 2 ENTER キーを押します。

指定したオブジェクトのソースコードが、読み取り専用モードで表示されます。

関連トピック：

- [ライブラリ内のオブジェクトのリスト](#)
- [LIST - 『システムコマンド』ドキュメント](#)

## オブジェクトの作成と編集

---

このセクションでは、Natural エディタを使用して Natural オブジェクトを作成および編集する手順について説明します。前述した Natural ユーティリティの使い方の詳細については、『ユーティリティ』ドキュメントの関連するセクションを参照してください。

- [現在の環境のチェック](#)
- [プログラミングモードの設定](#)
- [Natural プログラミング言語の使用](#)
- [ソースコードの作成](#)
- [ソースオブジェクトの編集](#)
- [オブジェクトタイプの設定](#)

### 現在の環境のチェック

Natural オブジェクトは、現在のシステムファイルにある現在のライブラリ内に作成されます。オブジェクトの作成や編集を開始する前に、オブジェクトの保存や取得を行うライブラリにログオンしていることを確認します。

ライブラリの割り当てとライブラリの切り替えについては、「[デフォルトのライブラリ割り当て](#)」と「[ライブラリへのログオン](#)」を参照してください。

### プログラミングモードの設定

Natural には、レポートモードとストラクチャードモードの2つのプログラミングモードがあります。

これらの2つのモードの説明とレポートからストラクチャード（またはその逆）にモードを切り替える方法の詳細については、「*Natural* のメインメニュー」の「[プログラミングモード](#)」を参照してください。

### Natural プログラミング言語の使用

Natural プログラミング言語は、ステートメント、システム関数、およびシステム変数で構成されています。

Natural ステートメントは、Natural プログラムソースの作成に使用するプログラミング命令です。

Natural システム関数は、例えば、算術関数を実行する場合に使用されます。

Natural システム変数は、Natural で提供され、生成される標準的な変数です。システム変数は、例えば日時の取得に使用されます。

関連トピック：

- 『ステートメント』ドキュメント（概要）
- 『システム関数』ドキュメント
- 『システム変数』ドキュメント

## ソースコードの作成

このセクションでは、システムコマンド `EDIT` とプログラムエディタを使用してソースコードを作成する方法を、例を示して説明します。さらに、エディタコマンドの例とソース内での移動方法についても説明します。

`EDIT` の代替機能として、「*Natural* のメインメニュー」で説明している **[Development Functions]** メニューの **[Create Object]** 機能を使用することもできます。

### ▶手順 6.6. ソースコードを入力するには

- 1 次のシステムコマンドを入力します。

```
EDIT object-type
```

この場合、*object-type* は作成するオブジェクトのタイプです。

例えば、プログラムタイプのオブジェクトを作成するには、次のように入力します。

```
EDIT PROGRAM
```

*object-type* を指定しない場合、プログラムエディタがデフォルトで起動されます。

（「**オブジェクトタイプの設定**」も参照。）

- 2 ENTER キーを押します。

プログラムエディタの編集エリアが表示されます。この例では、オブジェクトタイプ（ここでは Program）が画面の最上部に表示されています。

```
>
> + Program                               Lib SYSTEM
All  .....1.....2.....3.....4.....5.....6.....7..
0010
0020
0030
0040
0050
```

```
0060
0070
0080
0090
0100
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+... S 0   L 1
```

- 3 編集エリアが空ではない場合、エディタコマンドプロンプト (>) で、次のエディタコマンドを入力します。

```
CLEAR
```

さらに ENTER キーを押します。

CLEAR により、ソースワークエリアの内容が削除されます。

- 4 端末エミュレーション（例えば、Entire Connection）のコピー機能と貼り付け機能を使用して、またはソースコードを入力して、空の編集エリアの第 1 行（番号は 0010）からソースコードの挿入を開始します。

小文字から大文字への自動的な変換を無効にする場合は、『エディタ』ドキュメントの「一般的なデフォルト設定」の説明に従い、エディタプロファイルでデフォルト設定を変更します。

- 5 画面での入力時に、空の行を追加する場合は、次のエディタコマンドを入力します。

```
ADD
```

さらに ENTER キーを押します。

エディタコマンド ADD により、空の行が 9 行追加されます。これらの行のうち、ユーザーが入力を行った行のみがプログラムソースに追加されます。次に ENTER キーを押すと、空行が削除されます。『エディタ』ドキュメントの「エディタのデフォルト」で説明しているように、エディタプロファイルでこのデフォルト設定を変更することができます。すべてのプログラムエディタコマンドの詳細については、『プログラムエディタ』ドキュメントを参照してください。

#### ▶手順 6.7. ソースをスクロールするには

- 1 ソースコードの先頭に戻るには、次のエディタコマンドを入力します。

```
TOP
```

- 2 ソースコードの末尾に移動するには、次のエディタコマンドを入力します。

BOT

- 3 ソースコードで1ページスクロールダウンするには、PF8キーまたはENTERキーを押します。
- 4 ソースコードで1ページスクロールアップするには、PF7キーを押します。

すべてのプログラムエディタコマンドの詳細については、『プログラムエディタ』ドキュメントの「[配置のためのエディタコマンド](#)」を参照してください。

## ソースオブジェクトの編集

ソースコードをソースオブジェクトとして保存した後は（「[オブジェクトの保存とカタログ](#)」を参照）、ソースオブジェクトの名前を指定して、ソースオブジェクトにNaturalエディタを開きます。

### ▶手順 6.8. ソースオブジェクトのソースコードを編集するには

- 1 次のシステムコマンドを入力します。

```
EDIT object-name
```

この場合、*object-name*は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既存のソースオブジェクトの名前です。

- 2 Enter キーを押します。

指定したソースオブジェクトのソースコードが、対応するエディタの編集エリアに読み取り専用モードで表示されます。

EDITの代替機能として、「Naturalのメインメニュー」で説明している **[Development Functions]** メニューの **[Edit Object]** 機能を使用することもできます。

EDITの代わりに、システムコマンドREADを使用することもできます（「[オブジェクトのコピー](#)」を参照）。

関連トピック：

- *EDIT* - 『システムコマンド』ドキュメント

## オブジェクトタイプの設定

オブジェクトタイプは、オブジェクトの作成時に指定します（デフォルト設定はプログラム）。またオブジェクトタイプは、既存のソースオブジェクトをソースワークエリアに読み取るときに自動的に設定されます。プログラムエディタまたはデータエリアエディタで作業するときは、エディタコマンド `SET TYPE` を使用して、いつでもオブジェクトタイプを変更できます。

### ▶手順 6.9. オブジェクトタイプを変更するには

- 1 次のエディタコマンドを入力します。

```
SET TYPE object-type
```

この場合、*object-type* は作成するオブジェクトのタイプです。

例えば、次のように入力します。

```
SET TYPE SUBPROGRAM
```

- 2 ENTER キーを押します。

コマンドで指定された新しいオブジェクトタイプが画面に表示されます（この例では、Subprogram）。

関連トピック：

■ *SET TYPE* - 『プログラムエディタ』ドキュメント

*SET TYPE* - 『データエリアエディタ』ドキュメント

## オブジェクトのチェックとテスト

ソースコードのコンパイル（カタログ化）では、構文チェックが行われ、実行可能なオブジェクトコードが生成されます。

ソースワークエリア内のソースコードは、ソースコードを最初に保存しなくてもコンパイルできます（「[オブジェクトの保存とカタログ](#)」を参照）。さらにプログラムの実行と合わせて、プログラムタイプのオブジェクトのソースコードをコンパイルできます。「[プログラムの実行](#)」も参照してください。

### ▶手順 6.10. 構文チェックのためにソースコードをコンパイルするには

- 1 次のシステムコマンドを入力します。

```
CHECK
```



- 2 ENTER キーを押します。

構文エラーが検出されない場合、ソースワークエリアに含まれているソースコードがコンパイルされます。

#### ▶手順 6.11. プログラムの実行のためにソースコードをコンパイルするには

- 1 次のシステムコマンドを入力します。

```
RUN
```

- 2 ENTER キーを押します。

構文エラーが検出されない場合、ソースワークエリアに含まれているソースコードがコンパイルされ、生成されたコードが実行されます。

関連トピック：

- CHECK - 『システムコマンド』ドキュメント
- RUN - 『システムコマンド』ドキュメント

### 構文エラーのオンラインヘルプ

エラーメッセージが表示されない場合、ソースコードのコンパイルは正常に完了しています。

コンパイル中に構文エラーが検出された場合、次の例に示すように、エラーメッセージが画面に表示され、エラーのあるステートメント行が強調表示され、E とマークされます。

```
>
> + Program      PGM01      Lib SYSTEM
....+....1....+....2....+....3....+....4....+....5....+....6....+....7...
0250  RD1. READ EMPLOYEES-VIEW BY NAME
0260      STARTING FROM #NAME-START
0270      THRU #NAME-END
0280 *
0290      IF LEAVE-DUE >= 20
0300          PERFORM MARK-SPECIAL-EMPLOYEES
0310      ELSE
0320          RESET #MARK
0330      END-IF
0340 *
E 0350      DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20 #MARK
0360 *
0370  END-READ
0380 *
0390  IF *COUNTER (RD1.) = 0
0400      REINPUT 'PLEASE TRY ANOTHER NAME'
0410  END-IF
0420 *
```

```
0430 END-REPEAT
0440 *
      ....+....1....+....2....+....3....+....4....+....5....+... S 49   L 25
NAT0305 Text string must begin and end on the same line.
```

エラーを修正するまで、オブジェクトはコンパイルできません。構文エラーがある場合、ソースコードはソースオブジェクトとしてのみ保存できます（次のセクションを参照）。エラーの説明と問題解決のアドバイスについては、オンラインヘルプ機能を参照してください。

### ▶手順 6.12. エラーメッセージのヘルプを表示するには

- 1 次のシステムコマンドのいずれかを入力します。

```
HELP nnnn
```

または

```
? nnnn
```

*nnnn* は 4 桁のエラー番号です。

例えば、次のように入力します。

```
HELP NAT0305
```

- 2 ENTER キーを押します。

[Natural System Message] 画面に、指定したエラーの説明が表示されます。

オンラインヘルプの詳細については、「[エラーメッセージの詳細情報](#)」を参照してください。

## オブジェクトの保存とカタログ

---

ソースワークエリアに現在含まれているソースコードを、システムコマンド SAVE を使用してソースオブジェクトとして保存できます。SAVE では、ソースコードのカタログ化（コンパイル）は実行されないため、構文チェックは実行されません。

ソースワークエリアに現在含まれているソースコードを、ソースオブジェクトおよびカタログ化オブジェクト（コンパイル済み）として保存するには、システムコマンド STOW を使用します。

ソースワークエリアに現在含まれているソースコードをカタログし、カタログ化オブジェクトとしてのみ保存するには、システムコマンド CATALOG を使用します。CATALOG を使用すると、ソースコードはソースオブジェクトとして保存されないため、編集が可能です。「[複数のオブジェクトのカタログ](#)」も参照してください。

**▶手順 6.13. 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**

- 1 エディタコマンドプロンプトに、次のように入力します。

```
SAVE object-name
```

この場合、*object-name* は作成するソースオブジェクトの名前です。オブジェクトの名前は一意であり、**オブジェクトの命名規則**に従っている必要があります（関連セクションを参照）。

SAVE に適用されるすべての構文規則については、『システムコマンド』ドキュメントを参照してください。

- 2 ENTER キーを押します。

ソースコードが指定した名前で、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ライブラリに保存されます。

**▶手順 6.14. ソースコードを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保存するには**

- 1 エディタコマンドプロンプトに、次のいずれかを入力します。

```
STOW object-name
```

または

```
CATALOG object-name
```

この場合、*object-name* は作成するソースオブジェクトおよび／またはカタログされるオブジェクトの名前です。オブジェクトの名前は一意であり、**オブジェクトの命名規則**に従っている必要があります。

STOW および CATALOG に適用されるすべての構文規則については、『システムコマンド』ドキュメントを参照してください。

- 2 ENTER キーを押します。

STOW を使用する場合、ソースコードは、指定した名前で、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ライブラリに保存されます。さらに、生成されたオブジェクトコードは、同じライブラリとシステムファイルに、カタログ化オブジェクトとして保存されます。

CATALOG を使用する場合、ソースコードは、指定した名前で、カタログ化オブジェクトとしてのみ、現在のシステムファイル内の現在のライブラリに保存されます。ソースコードは、ソースオブジェクトとしては、システムファイル内に保存されません（また、既存のソースオブジェクトに対してこのコマンドを実行する場合、更新も実行されません）。ソースコードの保存と更新は、SAVE または STOW でのみ実行できます。

オブジェクトがソースオブジェクトまたはカタログ化オブジェクトのどちらで保存されているか確認する場合は、「[オブジェクトディレクトリ情報を表示するには](#)」を参照してください。

## 複数のオブジェクトのカタログ

現在のライブラリに含まれている複数のソースオブジェクトのカタログと再カタログを実行するには、システムコマンド CATALL を使用します。

### ▶手順 6.15. 複数のオブジェクトをカタログするには

- 1 次のシステムコマンドを入力します。

```
CATALL
```

- 2 ENTER キーを押します。

次の例のような [Catalog Objects in Library] 画面が表示され、処理するオブジェクト、実行するコマンド、エラーレポートの作成などの追加オプションを指定できます。

```
16:10:20          ***** NATURAL CATALL COMMAND *****          2006-12-21
User SAG          - Catalog Objects in Library -          Library SAGTEST

Catalog Objects from .. *_____ (start value, range, input list)
                    to .... _____ (end value)

                                X Recatalog only existing modules
                                _ Catalog all sources
Select object types:
X   Global data areas          Select function:
X   Parameter data areas      Save
X   Local data areas          X Catalog
X   Copycodes                 Stow
X   Texts                     Check
X   External Subroutines      Select options:
X   Subprograms               Condition code in batch
X   Help routines             X Renumbr source-codes lines
X   Maps                       Keep result list
X   Adapter                   X Processing information
X   Programs                  X Error report
X   Classes                    Extended error report

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit AddOp Sel.                               Canc
```

画面に表示されるオプションの詳細については、『システムコマンド』ドキュメントの「CATALL」を参照してください。

関連トピック：

- コンパイルの例- 『Natural システムアーキテクチャ』 ドキュメント

## オブジェクトディレクトリ情報の表示

Naturalオブジェクトのディレクトリには、オブジェクト名、オブジェクトを含むライブラリの名前、ソースオブジェクトの作成日または変更日など、オブジェクトの一般的な情報が含まれます。

### ▶手順 6.16. オブジェクトディレクトリ情報を表示するには

- 1 次のシステムコマンドを入力します。

```
LIST DIR object-name
```

この場合、**object-name** は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既存のオブジェクトの名前です。

例えば、次のように入力します。

```
LIST DIR PGMTEST
```

- 2 ENTER キーを押します。

次に示すプログラム PGMTEST の例のような、**[List Directory]** 画面が表示されます。

```
15:24:19          ***** NATURAL LIST COMMAND *****          2005-05-27
User SAG          - List Directory -          Library SAGTEST

Directory of Program PGMTEST          Saved on ... 2005-05-27 15:24:15
-----
Library .... SAGTEST    User-ID ..... SAG      Mode ..... Structured
TP-System .. CICS      Terminal-ID .. TCIJ
Op-System .. MVS/ESA   Transaction .. QA41
NAT-Ver .... 4.1.3
Source size .....          1028 Bytes

Directory of Program PGMTEST          Cataloged on 2005-05-27 15:24:15
-----
Library .... SAGTEST    User-ID ..... SAG      Mode ..... Structured
TP-System .. CICS      Terminal-ID .. TCIJ
Op-System .. MVS/ESA   Transaction .. QA41
NAT-Ver .... 4.1.3    Used GDA .....
Size of global data .....          0 Bytes
Size in DATSIZE .....          720 Bytes
Size in buffer pool .....          3348 Bytes
```

```
Size of OPT-Code ..... 0 Bytes
Initial OPT string .....
```

「**List Directory**」画面の詳細については、『システムコマンド』ドキュメントの「LIST」にある「オブジェクトディレクトリ情報の表示」を参照してください。

## オブジェクトのコピー

ソースワークエリアに含まれているソースコードをコピーするか、SYSMAIN などの Natural ユーティリティのコピー機能を使用して、新しいオブジェクトを作成できます。

### ▶手順 6.17. ソースワークエリアからソースコードをコピーするには

- 1 次のシステムコマンドを入力して、コピーするソースコードを読み取ります。

```
READ object-name
```

この場合、*object-name* はコピーするソースコードを含むオブジェクトの名前です。

- 2 ENTER キーを押します。

指定したソースオブジェクトのソースコードが、ソースワークエリアに読み取られます。

- 3 次のシステムコマンドのいずれかを入力します。

```
SAVE object-name
```

または

```
STOW object-name
```

この場合、*object-name* は作成するオブジェクトの名前です。

- 4 ENTER キーを押します。

新しいオブジェクトは、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オブジェクトとして保存されるか (SAVE を使用する場合)、カタログ化オブジェクトとして保存されます (STOW を使用する場合)。

### ▶手順 6.18. SYSMAIN を使用して 1 つ以上のオブジェクトをコピーするには

- 1 「メニュー機能を使用してすべてのライブラリをリストするには」の**手順 1~4**に従って、SYSMAIN ユーティリティの「**Main Menu**」を表示します。

- 2 すべてのタイプのオブジェクトを選択するには、**[Object Code]** フィールドに「A」（デフォルト設定）を入力します。メニュー画面で個別にリストされているオブジェクトタイプに対して、エラーメッセージの「E」など、別のコードを入力します。

**[Function Code]** フィールドに「C」（**[Copy]** を表す）を入力します。

- 3 ENTER キーを押します。

**[Copy Programming Objects]** 画面が表示されます。

- 4 すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を選択するには、**[Code]** フィールドに「A」を入力します。

**[Sel. List]**（リストの選択）フィールドで、Y（はい）をN（いいえ）に変更します。Yはデフォルト設定です。

**[ObjectName]** フィールドに、コピーするオブジェクトの名前を入力するか、名前の範囲を指定します。アスタリスク（\*）を入力すると、すべてのオブジェクト名が選択されます。アスタリスク（\*）はデフォルト設定です。

（有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」を参照してください。）

**[Source Library]** フィールドに、コピーするオブジェクトを含んでいるライブラリの ID を入力します。

**[Target Library]** フィールドに、オブジェクトのコピー先となる既存のライブラリまたは新規のライブラリの ID を入力します。

その他のすべての入力フィールドは変更しません。

- 5 ENTER キーを押します。

現在のシステムファイル内の指定したソースライブラリとターゲットライブラリの間で、すべてのソースオブジェクトとカタログ化オブジェクトがコピーまたは移動され、「機能が正しく実行されました。」というメッセージが表示されます。

関連トピック：

- READ - 『システムコマンド』ドキュメント

## オブジェクトの印刷

ソースオブジェクトのソースコードを印刷するには、システムコマンド LIST を使用します。

また「[オブジェクトのリストの印刷](#)」で説明しているように、ライブラリに含まれるオブジェクトのリストを印刷することもできます。

### ▶手順 6.19. ソースオブジェクトを印刷するには

1 次のいずれかの方法を選択します。

- 「LIST を使用してオブジェクトをリストするには」の[手順 1~2](#)に従って「**LIST Objects in a Library**」画面を表示し、リストからオブジェクトを選択します。

[Cmd] 列で目的のオブジェクトの横に、次のように入力します。

```
PR
```

Enter キーを押します。

- または：

次のシステムコマンドを入力します。

```
LIST object-name
```

この場合、*object-name* は印刷するオブジェクトの名前です。

Enter キーを押します。

指定したオブジェクトのソースコードが、読み取り専用モードで表示されます。

PF2 キーを押します。

[PRINT] ウィンドウが表示されます。

2 [Destination] フィールドに有効なプリンタ名を入力します（現在の環境で利用できるプリンタについては、必要に応じて Natural の管理者にお問い合わせください）。必要に応じて、ページサイズを変更します（デフォルト設定は 60 行です）。

3 Enter キーを押します。

[Printout Specification] 画面が表示されます。この画面では、印刷部数などプリンタの設定を行うことができます。

4 Enter キーを押します。

指定したソースオブジェクトが、指定したプリンタデバイスで印刷されます。



関連トピック：

- [ライブラリ内のオブジェクトの印刷](#)
- [LIST - 『システムコマンド』ドキュメント](#)

## オブジェクトの名前の変更

1つのオブジェクトの名前を変更する場合はシステムコマンド `RENAME` を使用し、複数のオブジェクトの名前を変更する場合は Natural ユーティリティ `SYSMAIN` を使用します。

`RENAME` の代替機能として、「Natural のメインメニュー」で説明している [\[Development Functions\]](#) メニューの [\[Rename Object\]](#) 機能を使用することもできます。

### ▶手順 6.20. `RENAME` を使用してオブジェクトの名前を変更するには

- 1 次のシステムコマンドを入力します。

```
RENAME object-name
```

この場合、`object-name` は名前を変更するオブジェクトの名前です。

- 2 `ENTER` キーを押します。
- 3 [\[Rename Object\]](#) ウィンドウが開いて、指定されたオブジェクトの名前が [\[Name\]](#) フィールドに表示されます。
- 4 [\[New Name\]](#) フィールドに、新しいオブジェクト名を入力します。  
必要に応じて、[\[New Type\]](#) フィールドに新しいオブジェクトタイプを入力します。
- 5 `ENTER` キーを押します。

「オブジェクトの名前を変更しました。」というメッセージが表示されます。

### ▶手順 6.21. `SYSMAIN` を使用して1つ以上のオブジェクトの名前を変更するには

- 1 「メニュー機能を使用してすべてのライブラリをリストするには」の[手順 1~4](#)に従って、`SYSMAIN` ユーティリティの [\[Main Menu\]](#) を表示します。
- 2 すべてのタイプのオブジェクトを選択するには、[\[Object Code\]](#) フィールドに「A」（デフォルト設定）を入力します。メニュー画面で個別にリストされているオブジェクトタイプに対して、エラーメッセージの「E」など、別のコードを入力します。

[\[Function Code\]](#) フィールドに「R」（[\[Rename\]](#) を表す）を入力します。

- 3 `ENTER` キーを押します。

[\[Rename Programming Objects\]](#) 画面が表示されます。

- 4 すべてのタイプのオブジェクトモジュール（カタログ化オブジェクトとソースオブジェクト）を選択するには、[Code] フィールドに「A」を入力します。

[Name] フィールドに、名前を変更するオブジェクトの名前、または名前の範囲を指定します（例えば、下記のサンプル画面の TEST\*）。アスタリスク（\*）を入力すると、すべてのオブジェクト名が選択されます。アスタリスク（\*）はデフォルト設定です。

（有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「名前の範囲の指定」を参照してください。）

1つのオブジェクトの名前のみを変更する場合、[New Name] フィールドに新しい名前を入力して、[Sel. List] フィールドで Y（はい）を N（いいえ）に変更します。

[Source Library] フィールドに、名前を変更するオブジェクトを含んでいるライブラリの ID を入力します。

必要に応じて [Target Library] フィールドに、名前を変更したオブジェクトの保存先となる既存のライブラリまたは新規のライブラリの ID を入力します。

その他のすべての入力フィールドは変更しません。

- 5 ENTER キーを押します。

ウィンドウが表示されます。名前を変更したオブジェクトのコピーを保存する場合は、ここで「Y」（はい）を入力できます。

- 6 ENTER キーを押します。

オブジェクトの範囲を指定した場合、以下の例のような [Rename Selection] 画面に、指定した選択条件（以下のサンプル画面では、TEST\*）に一致したすべてのオブジェクトがリストで表示されます。

ソースオブジェクトとカタログ化オブジェクトの両方の名前を変更する場合、[C] 列で目的のオブジェクトの横に「A」を入力します。次に示すように [New Name] 列に新しい名前を入力します。

```

16:39:39          ***** NATURAL SYSMAIN UTILITY *****          2005-05-30
User SAG          -          Rename Selection          -

RENAME ALL TEST* WITH XREF N IN SAGTEST WHERE DBID 10 FNR 32

  C  Name      Type      S/C      New NameC  Name      Type      S/C      New Name
  -  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -
A  TEST+     Progrm   S        PGMT+___  _  TEST+2   Progrm   S        _____
A  TESTCHAR  Progrm   S/C      CHARTEST  A  TESTDIR  Progrm   S        PGMDIR__
_  TESTDISP  Progrm   S/C      _____  _  TESTDIS2  Progrm   S/C      _____
_  TESTMMO   Proc     S/C      _____  A  TESTPGM_  Progrm   S/C      PGMTTEST_
    
```

```

_ TESTTEST  Progrm  S      _____  _ TESTXXX2  Progrm  S      _____
A TEST1    Subpgm  S/C    SUBTEST1    A TEST10   Subpgm  S/C    SUB10____
A TEST2    Subpgm  S/C    SUBTEST2    _ TEST666  Progrm  S/C    _____

Enter New Name and options, or '?' (Help) or '.' (Exit): _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren          Canc

```

- 7 ENTER キーを押します。

【Message Text】列が表示され、名前を変更した各オブジェクトの横に確認メッセージが表示されます。元のオブジェクトのコピーを保持するオプションをオンにしたかどうかに応じて、メッセージには、【Renamed as】または【Copied as】が表示されます。

## オブジェクトの移動

特定のライブラリから別のライブラリにオブジェクトを移動するには、SYSMAINなどのNaturalユーティリティを使用します。

### ▶手順 6.22. SYSMAIN メニュー機能を使用してオブジェクトを移動するには

- 「メニュー機能を使用してすべてのライブラリをリストするには」の手順1~4に従って、SYSMAINユーティリティの【Main Menu】を表示します。
- すべてのタイプのオブジェクトを選択するには、【Object Code】フィールドに「A」（デフォルト設定）を入力します。メニュー画面で個別にリストされているオブジェクトタイプに対して、エラーメッセージの「E」など、別のコードを入力します。

【Function Code】フィールドに「M」（【Move】を表す）を入力します。

- ENTER キーを押します。

【Move Programming Objects】画面が表示されます。

- すべてのタイプのオブジェクトモジュール（ソースオブジェクトとカタログ化オブジェクト）を選択するには、【Code】フィールドに「A」を入力します。

【Sel. List】（リストの選択）フィールドで、Y（はい）をN（いいえ）に変更します。Yはデフォルト設定です。

[**ObjectName**] フィールドに、移動するオブジェクトの名前を入力するか、名前の範囲を指定します。アスタリスク (\*) を入力すると、すべてのオブジェクト名が選択されます。アスタリスク (\*) はデフォルト設定です。

(有効な名前の範囲については、『SYSMAIN ユーティリティ』ドキュメントの「[名前の範囲の指定](#)」を参照してください。)

[**Source Library**] フィールドに、移動するオブジェクトを含んでいるライブラリの ID を入力します。

[**Target Library**] フィールドに、オブジェクトの移動先となる既存のライブラリまたは新規のライブラリの ID を入力します。

その他のすべての入力フィールドは変更しません。

- 5 ENTER キーを押します。

確認ウィンドウが表示されます。

- 6 移動を実行するには ENTER キーを押し、移動をキャンセルするにはピリオド (.) を入力します。

移動が正常に完了すると、すべてのソースオブジェクトとカタログ化オブジェクトが、現在のシステムファイル内の指定したソースライブラリから指定したターゲットライブラリへ移動され、「機能が正しく実行されました。」というメッセージが表示されます。

## オブジェクトの削除

---

オブジェクトを削除するには、システムコマンド DELETE、システムコマンド LIST、または SYSMAIN などの Natural ユーティリティを使用します。LIST または SYSMAIN を使用してオブジェクトを削除する方法の詳細については、「[ライブラリ内のオブジェクトの削除](#)」を参照してください。

DELETE の代替機能として、「Natural のメインメニュー」で説明している [**Development Functions**] メニューの [**Delete Object**] 機能を使用することもできます。

### ▶手順 6.23. DELETE を使用して1つ以上のオブジェクトを削除するには

- 1 次のシステムコマンドのいずれかを入力します。

```
DELETE object-name
```

または

```
DELETE object-name*
```

または

## DELETE \*

上記の意味は次に示すとおりです。

*object-name* は削除するオブジェクトの名前です。

*object-name\**は選択する特定範囲のオブジェクトです（例えば、TEST\*では、TESTで始まるすべてのオブジェクトが選択されます）。

アスタリスク (\*) では、現在のシステムファイル内の現在のライブラリにあるすべてのオブジェクトが選択されます。

2 ENTER キーを押します。

- オブジェクトを個別に指定した場合、[DELETE] ウィンドウが表示されます。

削除を確認するには、オブジェクトの名前を入力します。

- オブジェクトの範囲を指定した場合、[Delete Sources and Objects] 画面が表示されま

す。ソースオブジェクトとカタログ化オブジェクトの両方を削除する場合、[M] 列で目的のオブジェクトの横に「B」を入力します。

ENTER キーを押します。

[DELETE] ウィンドウが表示されます。

目的のオプションの横に任意の文字を入力して項目をマークします。

[Confirm each deletion] では、削除する最初のオブジェクトに対して [DELETE] ウィンドウが表示されます。オブジェクトの名前を入力した後、削除を確認するために ENTER キーを押すと、削除する次のオブジェクトに対して [DELETE] ウィンドウが表示されま

す。

[Delete without confirmation] を選択すると、すぐに削除が実行されます。

[Exit (no deletion)] を選択すると、削除がキャンセルされます。

3 ENTER キーを押します。

[Delete Sources and Objects] 画面が表示され、削除対象として選択したオブジェクトの横にメッセージが表示されます。このメッセージでは、オブジェクトが削除されたこと、または削除がキャンセルされたこと（削除されていないこと）が示されます。

関連トピック：

- DELETE - 『システムコマンド』ドキュメント

## プログラムの実行

プログラムタイプのオブジェクトは、システムコマンドを使用して実行できます。その他のすべてのタイプのオブジェクトは、このプログラムまたは下位のオブジェクトで参照している場合にのみ、実行または呼び出すことができます。『プログラミングガイド』の「呼び出されるオブジェクトの複数レベル」も参照してください。

プログラムを実行するには、システムコマンド `RUN` または `EXECUTE` を使用します。

`EXECUTE` の代替機能として、「*Natural* のメインメニュー」で説明している **[Development Functions]** メニューの **[Execute Program]** 機能を使用することもできます。

`RUN` では、ソースワークエリアに現在含まれているソースコードが実行されるか、システムファイルに保存されているカタログ化オブジェクトが実行されます。

`EXECUTE` では、カタログ化オブジェクトのみが実行されます。`RUN` とは異なり、`EXECUTE` では、ソースワークエリア内の対応するソースコードに対して加えられた最新の変更は実行されません。これらの変更が実行対象になるのは、ソースオブジェクトを更新して、再コンパイルを実行した後です。

カタログ化オブジェクトの実行は、ソースワークエリアに現在含まれているソースコードには影響を及ぼしません。

### ▶手順 6.24. `RUN` を使用してプログラムを実行するには

- 1 次のシステムコマンドのいずれかを入力します。

```
RUN
```

または

```
RUN program-name
```

この場合、*program-name* はソースワークエリアに読み取られるプログラムタイプのソースオブジェクトの名前です。

- 2 `ENTER` キーを押します。

構文エラーが検出されない場合、ソースワークエリアに含まれているソースコードがコンパイルされ、実行されます。

### ▶手順 6.25. `EXECUTE` を使用してプログラムを実行するには

- 1 次のシステムコマンドを入力します。

```
EXECUTE program-name
```

この場合、*program-name* はプログラムタイプのカタログ化オブジェクトの名前です。

キーワード EXECUTE の指定は任意です。*program-name* の指定で十分です。

2 ENTER キーを押します。

プログラムが実行されます。

関連トピック：

- RUN - 『システムコマンド』ドキュメント
- EXECUTE - 『システムコマンド』ドキュメント
- オブジェクトの実行 - 『Natural システムアーキテクチャ』ドキュメント
- [オブジェクト実行の検索順序](#)
- オブジェクトのロードと実行の例 - 『Natural システムアーキテクチャ』ドキュメント





# 7 Natural のメインメニュー

---

▪ Natural のメインメニューの表示またはクローズ .....	78
▪ Development Functions .....	80
▪ Development Environment Settings .....	85
▪ Maintenance and Transfer Utilities .....	86
▪ Debugging and Monitoring Utilities .....	87
▪ Example Libraries .....	87
▪ Other Products .....	88

Natural のメインメニューでは、Natural の開発機能、環境設定、ユーティリティ、およびサンプルライブラリにアクセスできます。

このセクションでは、Natural のメインメニューと下位メニューで提供される機能や、入力オプションについて説明します。

## Natural のメインメニューの表示またはクローズ

---

Natural のメインメニューを表示またはクローズするには、2つの方法があります。

- メニューモードのオンとオフを切り替えることで、デフォルト設定を定義できます。メニューモードでは、Natural のメインメニューは次に開始されるセッションのために自動的に表示されます。
- Natural のセッション中でも、必要に応じて、Natural のメインメニューを表示またはクローズすることができます。

### ▶手順 7.1. セッションを開始する前にメニューモードのオンとオフを切り替えるには

- Natural の起動時に、次のいずれかのプロファイルパラメータを指定します。

```
MENU=ON
```

(メニューモードをアクティブにします)

または

```
MENU=OFF
```

(メニューモードを非アクティブにします)

『パラメータリファレンス』ドキュメントの「MENU」も参照してください。

### ▶手順 7.2. セッション中に Natural のメインメニューを表示またはクローズするには

- 1 次のシステムコマンドのいずれかを入力します。

```
MAINMENU
```

(メニューを表示します)

または

```
MAINMENU OFF
```

(メニューをクローズします)

2 ENTER キーを押します。

Natural のメインメニューは、次の例のようになります。

```

15:00:52          ***** NATURAL *****          2002-12-18
User SAG          - Main Menu -          Library TEST

                Function

                _ Development Functions
                _ Development Environment Settings
                _ Maintenance and Transfer Utilities
                _ Debugging and Monitoring Utilities
                _ Example Libraries
                _ Other Products
                _ Help
                _ Exit Natural Session

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                                  Canc

```


セッションの開始時に、ユーザーにはデフォルトのライブラリが割り当てられ、これが画面に表示されます。上記のサンプル画面では、ライブラリの ID（名前）が画面の右上隅にある **[Library]** フィールドに表示されます。「[デフォルトのライブラリ割り当て](#)」も参照してください。

[メニュー機能の実行](#)の詳細については、「[コマンドとメニュー機能の使用](#)」の関連セクションを参照してください。

Natural のメインメニューにリストされる各機能では、同じ名前のメニューが表示され、さらに機能を選択することができます。

Natural のメインメニューの機能	対応するメニューにある機能の説明
<a href="#">Development Functions</a>	Natural アプリケーションを構成するプログラム、マップ、データエリア、およびその他のコンポーネントを作成し、管理します。
<a href="#">Development Environment Settings</a>	Natural セッションに影響するさまざまな設定を表示したり変更したりします。
<a href="#">Maintenance and Transfer Utilities</a>	特定のオブジェクトの作成、管理、または他の環境へのオブジェクトの転送を行うための Natural ユーティリティを起動します。

Natural のメインメニューの機能	対応するメニューにある機能の説明
<b>Debugging and Monitoring Utilities</b>	Natural アプリケーションのモニタリングと処理フローでのエラー検出を行うための Natural ユーティリティを起動します。
<b>Example Libraries</b>	サンプルプログラムとアプリケーションプログラミングインターフェイス (API) を含むライブラリを選択します。
<b>Other Products</b>	その他の Software AG 製品を起動します。

 **注意:** Natural のメインメニューと下位メニューでのメッセージ行/PF キー行の位置と色は、『オペレーション』ドキュメントで説明しているユーザー出口ルーチン USR2003P によって変えられる場合があります。

## Development Functions

〔Development Functions〕メニューにリストされる機能は、Natural でアプリケーションを開発するとき最も必要になる機能です。これらの機能は、現在ログオンしているライブラリで使用できるすべての Natural オブジェクトに適用されます。

次の表では、〔Development Functions〕メニューのフィールドについて説明します。

フィールド	説明
<b>User</b>	現在のセッションにログオンしている Natural ユーザーの ID。
<b>Library</b>	現在アクティブなライブラリ。 「 <a href="#">Natural ライブラリの使用</a> 」も参照してください。
<b>Mode</b>	プログラミングモード。レポートモードまたはストラクチャードモードです。「 <a href="#">プログラミングモード</a> 」を参照してください。
<b>Work area empty</b>	ソースワークエリアにソースがロードされていないことを示します。 ソースがすでにソースワークエリアにロードされている場合、例えば Program PROGX など、オブジェクトの <a href="#">タイプ</a> と <a href="#">名前</a> が表示されます。
<b>Code</b>	<b>Create Object</b> の C など、使用する機能に対応するコード。 「 <a href="#">メニュー機能の実行</a> 」も参照してください。
<b>Type</b>	program の P など、オブジェクトのタイプ。詳細については、『 <a href="#">プログラミングガイド</a> 』の「 <a href="#">オブジェクトタイプ</a> 」を参照してください。 既存の Natural オブジェクトの名前を指定する場合、〔Type〕フィールドは空白のままかまいません。 オブジェクトタイプを変更する場合、「 <a href="#">オブジェクトタイプの設定</a> 」も参照してください。
<b>Name</b>	オブジェクトの名前です。

フィールド	説明
	有効なオブジェクト名の詳細については、「 <a href="#">オブジェクトの命名規則</a> 」を参照してください。
Command ==>	<p>コマンド行。この行は、Natural コマンドを入力できる入力フィールドです。</p> <p>例：</p> <p>PROGX という既存のプログラムを編集するには、次のシステムコマンドを入力します。</p> <p>EDIT PROGX</p> <p>「<a href="#">コマンドとメニュー機能の使用</a>」も参照してください。</p>
PF (ファンクションキー)	<p>PF キー（ファンクションキー）は、コマンドまたはメニュー機能の代わりに使用することができます。画面の下部にある PF キー行は、どの機能がどのキーに割り当てられているかを示します。</p> <p>「<a href="#">標準的な PF キー</a>」も参照してください。</p>

次の表では、[Development Functions] メニューの機能について説明します。ほとんどのメニュー機能に、同等の Natural システムコマンドがあります。『システムコマンド』ドキュメントでは、これらの代替的なシステムコマンドを表にして、さらに関連セクションで詳細に説明しています。

機能	コード	説明
Create Object	C	<p>プログラム、マップ、データエリアなど、新しいオブジェクトを作成できる <b>Natural エディタ</b> を起動します。</p> <p>作成するオブジェクトのタイプと名前を指定します。[Type] フィールドに疑問符 (?) を入力すると、その機能で使用できる全タイプのリストが表示されます。このリストからオブジェクトタイプを選択できます。</p> <p>「<a href="#">オブジェクトの作成と編集</a>」も参照してください。</p>
Edit Object	E	<p><b>Natural エディタ</b> を起動して、指定したオブジェクトのソースを編集モードで表示します。</p> <p>編集する既存のオブジェクトの名前を指定します。オブジェクトの選択リストも表示できます。「<a href="#">オブジェクトの範囲の指定</a>」も参照してください。</p> <p>[Type] フィールドに疑問符 (?) を入力すると、その機能で使用できる全タイプのリストが表示されます。このリストからオブジェクトタイプを選択できます。</p> <p>「<a href="#">オブジェクトの作成と編集</a>」も参照してください。</p> <p>同等のシステムコマンド：EDIT</p>
Rename Object	R	<p>指定したオブジェクトの名前および/またはオブジェクトタイプを変更する [Rename Object] ウィンドウを表示します。</p>

機能	コード	説明
		<p>「<a href="#">オブジェクトの名前の変更</a>」も参照してください。</p> <p>同等のシステムコマンド：RENAME</p>
Delete Object	D	<p>指定したオブジェクトに対して <b>[Delete]</b> ウィンドウを表示します。 <b>[Delete]</b> ウィンドウでは、関連する入力フィールドに再びオブジェクトの名前を入力して削除を確認します。</p> <p>「<a href="#">オブジェクトの範囲の指定</a>」で説明しているように、オブジェクトの選択リストも表示できます。このリストでは、1つ以上のオブジェクトを削除対象としてマークできます。</p> <p>「<a href="#">オブジェクトの削除</a>」も参照してください。</p> <p>同等のシステムコマンド：DELETE</p>
Execute Program	X	<p>プログラムタイプオブジェクトを実行します。</p> <p>実行するオブジェクトの名前を指定します。</p> <p>その他のオブジェクトタイプは、それ自身で実行することができず、別のオブジェクトから呼び出す必要があります。</p> <p>「<a href="#">プログラムの実行</a>」も参照してください。</p> <p>同等のシステムコマンド：EXECUTE</p>
List Object(s)	L	<p>指定したオブジェクトのソースコードを表示します。</p> <p>表示するオブジェクトの名前を指定します。オブジェクトの選択リストも表示できます。「<a href="#">オブジェクトの範囲の指定</a>」も参照してください。</p> <p><b>[Type]</b> フィールドに疑問符 (?) を入力すると、その機能で使用できる全タイプのリストが表示されます。このリストからオブジェクトタイプを選択できます。</p> <p>「<a href="#">ライブラリ内のオブジェクトのリスト</a>」も参照してください。</p> <p>同等のシステムコマンド：LIST</p>
List Subroutines Used	S	<p>どのオブジェクトがどの外部サブルーチンやクラスを使用するかを確認します。</p> <p>同等のシステムコマンド：ROUTINES</p>

このセクションでは、以下のトピックについて説明します。

- [プログラミングモード](#)
- [Natural エディタ](#)

## ■ オブジェクトの範囲の指定

### プログラミングモード

Natural には、レポートモードとストラクチャードモードの2つのプログラミングモードがあります。アプリケーションの構造がより明確になるため、ストラクチャードモードを排他的に使用することをお勧めします。したがって、Natural のチュートリアル『ファーストステップ』と『エディタ』ドキュメントに記載されているすべての説明と例は、ストラクチャードモードを対象としています。レポートモードの特性は考慮されていません。

プログラミングモードの詳細については、『プログラミングガイド』の「Natural プログラミングモード」を参照してください。

【Development Functions】メニューの右上隅にある【Mode】フィールドは、その時点で有効になっているプログラミングモード、つまり、ストラクチャードまたはレポートを示しています。

#### ▶手順 7.3. プログラミングモードを切り替えるには

- 1 【Development Functions】画面の右上隅にある【Mode】フィールドでは、ストラクチャードモードをオンにする場合は S、レポートモードをオンにする場合は R で、最初の位置にある指定を上書きします。

または:

次のシステムコマンドのいずれかを入力します。

```
GLOBALS SM=ON
```

(ストラクチャードモードをオンにします)

または

```
GLOBALS SM=OFF
```

(レポートモードをオンにします)

- 2 ENTER キーを押します。

【Mode】フィールドの表示が、Reporting から Structured に、またはその逆に変更されます。

関連トピック：

- Natural プログラミングモード - 『プログラミングガイド』
- GLOBALS - 『システムコマンド』ドキュメント

## Natural エディタ

[Development Functions] メニューで指定したオブジェクトタイプに応じて、Natural では、プログラムエディタ、マップエディタ、データエリアエディタなど、適切なエディタが起動されます。これらのエディタの詳細については、『エディタ』ドキュメントの関連セクションを参照してください。

### オブジェクトの範囲の指定

[Edit Object]、[List Object(s)]、[Delete Object] 機能のオプションでは、個別にオブジェクトの名前を指定するか、名前の範囲を指定します。名前の範囲を指定すると、オブジェクトのリストが表示されます。このリストでは、オブジェクトの編集やリスト表示、または削除したいオブジェクトをマークするなどの操作の対象とするオブジェクトを1つ以上選択できません。

#### ▶手順 7.4. すべてのオブジェクトをリストするには

- 1 [Name] フィールドにアスタリスクを入力します。

\*

- 2 ENTER キーを押します。

現在のライブラリ内にあるすべてのオブジェクトのリストが表示されます。

#### ▶手順 7.5. 開始値を使用してオブジェクトをリストするには

- 1 [Name] フィールドに、開始値とそれに続けてアスタリスク (\*) を入力します。

値の末尾にアスタリスクを指定して入力するこのオプションをアスタリスク表記と呼びます。

例えば、次のように入力します。

AB\*

- 2 ENTER キーを押します。

現在のライブラリを対象として、名前が AB で始まるすべてのオブジェクト（例えば、AB、AB1、ABC、ABEZ）のリストが表示されます。このリストには、AA1 や ACB で始まるオブジェクト名は含まれません。



**注意:** 同等のシステムコマンド LIST で説明しているように、[List Object(s)] 機能には、オブジェクト名の範囲を指定するオプションがあります。



## Development Environment Settings

次の表では、[Development Environment Settings] メニューの機能について簡単に説明し、これらの機能に対応する Natural システムコマンドを示します。システムコマンドの詳細については、『システムコマンド』ドキュメントの関連するセクションを参照してください。

機能	説明	対応するコマンド
<b>Function-Key Settings</b>	Natural セッションで使用する PF キーに機能を割り当てます。	KEY
<b>Compilation Settings</b>	Natural オブジェクトのコンパイル方法を制御するオプションを設定します。	COMPOPT
<b>Session Parameter Settings</b>	Natural セッションパラメータの設定を変更します。  『パラメータリファレンス』ドキュメントの「 <a href="#">Natural 環境の構成</a> 」と「セッションパラメータ」も参照してください。	GLOBALS
<b>Profile Parameter Settings</b>	Natural プロファイルパラメータの設定を変更します。  プロファイルパラメータについては、『パラメータリファレンス』ドキュメント、『オペレーション』ドキュメントの「プロファイルパラメータの使用法」で説明しています。  システムコマンド SYSPARM では、『ユーティリティ』ドキュメントで説明している同じ名前のユーティリティが起動されます。	SYSPARM
<b>Technical Session Information</b>	現在のユーザー ID、ライブラリ、オペレーティングシステムなど、Natural セッションの技術情報を表示します。	TECH
<b>System File Information</b>	Natural システムファイルの現在の定義を表示します。  『Natural システムアーキテクチャ』ドキュメントの「 <a href="#">Natural システムファイル</a> 」も参照してください。	SYSPROF
<b>Product Installation Information</b>	サイトにインストールされた製品のリストとこれらの製品の情報を表示します。	SYSPROD
<b>Security Profile Information</b>	使用できるのは、Natural Security がインストールされている場合に限られます。  現在有効になっているセキュリティプロファイルを表示します。	PROFILE

## Maintenance and Transfer Utilities

次の表では、[Maintenance and Transfer Utilities] メニューの機能について簡単に説明し、これらの機能に対応する Natural システムコマンドを示します。これらの各コマンドでは、『ユーティリティ』ドキュメントで説明している Natural ユーティリティが起動されます。

機能	ユーティリティの説明	対応するコマンド
<b>Maintain Error Messages</b>	Natural アプリケーションで発行するメッセージの作成とメンテナンスを実行します。	SYSERR
<b>Maintain DDMs</b>	データ定義モジュール (DDM) の作成とメンテナンスを実行します。	SYSDDM
<b>Maintain Command Processors</b>	Natural アプリケーションで使用するコマンドプロセッサの作成とメンテナンスを実行します。	SYSNCP
<b>Maintain Remote Procedure Calls</b>	リモートプロシージャコールの確立と維持を実行し、リモートサーバーにある Natural サブプログラムの実行に必要な設定を提供します。	SYSRPC
<b>Transfer Objects to Other Libraries</b>	異なるライブラリ間で Natural オブジェクトを転送します。	SYSMAIN
<b>Transfer Objects to Other System Files</b>	<p>Natural オブジェクトのアンロードまたはロードを実行します。</p> <p>システムコマンド SYSUNLD を使用すると、オブジェクトのアンロードまたはロードを行う最初のユーティリティメニューを表示できます。また、システムコマンド NATUNLD または NATLOAD を使用すると、下位のロードユーティリティまたはアンロードユーティリティを直接起動することができます。</p> <p>NATUNLD ユーティリティ：Natural システムファイルからワークファイルに Natural オブジェクトをアンロードします。</p> <p>NATLOAD ユーティリティ：ワークファイルから Natural システムファイルに Natural オブジェクトをロードします。</p> <p>NATUNLD と NATLOAD の機能は、Natural オブジェクトハンドラと同様に実行できます。代わりにオブジェクトハンドラを使用することをお勧めします。</p>	SYSUNLD
<b>Transfer Objects to Other Platforms</b>	<p>Natural オブジェクトと Adabas FDT を特定のハードウェアプラットフォームから別のプラットフォームに転送します。</p> <p>SYSTRANS と同じ機能を、オブジェクトハンドラで実行できます。代わりにオブジェクトハンドラを使用することをお勧めします。</p>	SYSTRANS
<b>Transfer Objects to Other Systems</b>	オブジェクトハンドラを起動して、Natural 環境に配布するために Natural オブジェクトと外部オブジェクトを処理します。	SYSOBJH

## Debugging and Monitoring Utilities

次の表では、[**Debugging and Monitoring Utilities**] メニューの機能について簡単に説明し、これらの機能に対応する Natural システムコマンドを示します。これらの各コマンドでは、『ユーティリティ』ドキュメントで説明している Natural ユーティリティが起動されます。

機能	ユーティリティの説明	対応するコマンド
<b>Debugging</b>	プログラムの処理フローで発生したエラーを検索します。	TEST
<b>Logging of Database Calls</b>	データベースコマンドのログを記録します。	TEST DBLOG
<b>Issuing Adabas Calls</b>	Adabas コマンドをデータベースに直接渡します。	SYSADA
<b>Buffer Pool Maintenance</b>	Natural バッファプールをモニタリングし、要件に合わせてプールを調整します。	SYSBPM
<b>Editor Buffer Pool Maintenance</b>	Software AG Editor のバッファプールをモニタリングし、要件に合わせてプールを調整します。	SYSEDT
<b>TP-Specific Monitoring</b>	Natural の TP モニタ固有の特性をモニタリングして管理します。	SYSTP
<b>Data Collection and Tracing</b>	Natural アプリケーションの処理フローについて、モニタリングデータとアカウントングデータを収集します。	SYSRDC
<b>Error Information on Abnormal Termination</b>	Software AG の担当技術者によるエラー診断のために必要な情報を提供します。	DUMP

## Example Libraries

Natural のメインメニューで [**Example Libraries**] を選択すると、ライブラリのリストが表示されます。これらのライブラリには、デモンストレーションを目的とするサンプルプログラムや、Software AG が提供するアプリケーションプログラミングインターフェイス (API) が含まれています。

ライブラリ	内容
<b>SYSEXP</b>	『プログラミングガイド』で例示し、説明しているサンプルプログラム。
<b>SYSEXR</b>	『ステートメント』ドキュメントと『システム変数』ドキュメントで例示し、説明しているサンプルプログラム。
<b>SYSEXV</b>	Natural の新機能を説明するためのサンプルプログラム。
<b>SYSEXT</b>	API と、API を使用するためのサンプルプログラム。  『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 SYSEXT も参照してください。
<b>SYSEXTP</b>	特定の TP モニタのみで使用する特別な機能向けのサンプルプログラムと API。

## Other Products

---

Natural のメインメニューで [**Other Products**] を選択すると、Software AG のアドオン製品のリストが表示されます。これらの製品はサイトにインストールされ、このメニューからアクセスできます。

## 8 出力ファイルとワークファイル

---

出力ファイルとワークファイルは、Naturalオンライン環境またはバッチ環境向けに論理的に定義されており、基盤となるオペレーティングシステムまたはTPモニタのNaturalパラメータや制御ステートメントを使用して、ファイルまたはプリンタに物理的に割り当てることができます。割り当ては、ランタイムに現在のセッションについて変更できます。

ワークファイルでのデータの書き込み、読み取り、出力ファイルでの書き込みには、適切なNaturalステートメントを使用します。

物理的な出力ファイルまたはワークファイルの割り当ては、対応するNatural環境で管理されているオブジェクトとは依存関係にありません。したがって、割り当てが変更された場合でも、オブジェクト内で出力ファイルまたはワークファイルを参照するステートメントを変更する必要はありません。

出力ファイルとワークファイルの全般情報については、『Naturalシステムアーキテクチャ』ドキュメントの「出力ファイル- ワークファイル」を参照してください。

出力ファイルとワークファイルの使い方の詳細については、次に挙げるドキュメントを参照してください。

『オペレーション』：

- z/OS バッチモードのNaturalで 사용되는データセット
- BS2000/OSD バッチモードセッションで使用されるNaturalデータセット
- z/VSE ライブラリメンバに対するNATVSE出力およびワークファイルサポート
- 出力ファイルおよびワークファイルのサポート (VM/CMS環境)
- サーバー環境における外部データセットでの出力ファイルおよびワークファイルの処理-z/OS環境のサーバーとしてのNatural
- 出力レポートおよびワークファイル

『ステートメント』：

- ワークファイル/PC ファイルの制御
- *CLOSE WORK FILE*
- *DEFINE WORK FILE*
- *READ WORK FILE*
- *WRITE WORK FILE*

『パラメータリファレンス』：

- *PRINT* - 出力ファイルの割り当て
- *WORK* - ワークファイルの割り当て

## 9 Natural 環境の構成

---

■ プロファイルパラメータの使用 .....	92
■ セッションパラメータの使用 .....	95
■ パラメータ化のレベル .....	97

Natural パラメータは、Natural 環境のコンフィグレーションを管理します。

Natural パラメータを使用すると、開発／生成プロセスの標準化や自動化、または個々のユーザーのニーズに応じた標準設定の調整を行うことができます。例えば、レポート作成時のデフォルト値の設定、レポートサイズの定義、またはエディタのソースエリアなど必要とされるストレージエリアサイズの定義を行う場合に、Natural パラメータを使用します。

Natural 環境の特質の大半はすでに Software AG によって定義されていますが、Natural 管理者は、すべての Natural ユーザーに対して有効となる別のデフォルト環境設定を構成することができます。また、個々のユーザーは、デフォルト環境設定にダイナミックプロファイルパラメータまたはセッションパラメータを上書きするという方法で、自身のニーズに応じた設定を適用することができます。

## プロファイルパラメータの使用

---

プロファイルパラメータは、スタティックまたはダイナミックに指定されます。

スタティックパラメータは、Natural のインストール中に Natural パラメータモジュール NATPARAM で指定されます。各 Natural セッションのデフォルトとして使用されます。

ダイナミックパラメータは、Natural の起動時に指定されます。Natural SYSPARM ユーティリティを使用すると、ダイナミックパラメータセットを事前定義することができます。

ダイナミックパラメータの定義済みセットは、パラメータプロファイルタイプの Natural オブジェクトです。

「[セッション開始時にプロファイルパラメータを設定するには](#)」で説明するように、セッションの開始時には、1つ以上のパラメータプロファイルと、1つ以上のシングルプロファイルパラメータを指定できます。

セッションの開始時に指定するパラメータ（パラメータプロファイルの使用に関係なく）は、セッションに割り当てられた標準的なパラメータプロファイル内で指定されたパラメータ値、および／または Natural パラメータモジュール NATPARAM 内で指定されたパラメータ値よりも優先されます。「[パラメータ化のレベル](#)」も参照してください。

### ▶手順 9.1. NATPARAM でプロファイルパラメータを設定するには

- 『オペレーション』ドキュメントの「[Natural パラメータモジュールの使用](#)」で説明する手順に従います。

### ▶手順 9.2. SYSPARM を使用してパラメータプロファイルを作成するには

- 1 次のシステムコマンドを入力して、SYSPARM ユーティリティを起動します。



## SYSPARM

- 2 ENTER キーを押します。

Natural SYSPARM ユーティリティのメニューが表示されます。

- 3 次の例に示すように、[Code] フィールドに「A」 (**Add New Profile**) を入力し、**[Profile]** フィールドに作成するパラメータプロファイルの名前を入力します。

```

11:36:19          ***** NATURAL SYSPARM UTILITY *****          2005-06-16
User SAG                      - Menu -

          Code  Function

          L    List Profiles
          D    Display Profile
          A    Add New Profile
          M    Modify Profile
          C    Copy Profile
          X    Delete Profile
          ?    Help
          .    Exit

          Code ..... A
          Profile .. TESTPROF   DBID ..   10  FNR .....   1640
          Copy to .. _____ Password ..
                                   Cipher ....

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                                           Canc

```

- 4 ENTER キーを押します。

SYSPARM ユーティリティの編集エリアが表示されます。

- 5 以下のパラメータプロファイルの例に示すように、必要なプロファイルパラメータとパラメータ値を入力します。指定できるパラメータ設定の詳細については、『パラメータリファレンス』ドキュメントの「プロファイルパラメータ」にあるプロファイルパラメータの説明を参照してください。

```

16:57:37          ***** NATURAL SYSPARM UTILITY *****          2005-06-16
> FUSER=(10,32) <
> LS=250,PS=60 <
> WORK=((6-8),AM=PC) <
> DB=(ADAV7,*) <
> PC=ON <
> AUTO=ON <
> MENU=OFF <
> <
> <
> <
Help with parameters .. _____ (Profile name: TESTPROF)

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  Check Save          Insrt Del   Copy  Canc
    
```

- 6 PF4 (Check) キーを押してチェックし、必要に応じて構文を修正します。  
 適用される構文については、『オペレーション』ドキュメントの「パラメータ値のダイナミックな割り当て」を参照してください。
- 7 PF5 (Save) キーを押して、パラメータプロファイルをソースオブジェクトとして、指定した名前での現在のシステムファイル（デフォルト設定は FNAT）に保存します。

▶手順 9.3. セッション開始時にプロファイルパラメータを設定するには

- 1つ以上のパラメータプロファイル（可能な場合）、および／または1つ以上のプロファイルパラメータを入力します。パラメータプロファイルとプロファイルパラメータは、任意の順序で入力できます。ただし、パラメータが複数回指定されている場合（次の例では、FUSER）、最後の指定でこのパラメータに割り当てられた値が使用されます（次の例では、10,245）。

```
PROFILE=TESTPROF FUSER=(10,123) DSIZE=40 LS=250,PS=50 FUSER=(10,245)
```

関連トピック：

- プロファイルパラメータの使用方法 - 『オペレーション』ドキュメント
- プロファイルパラメータの概要 - 『パラメータリファレンス』ドキュメント
- プロファイルパラメータの概要 - 『オペレーション』ドキュメント
- パラメータ値の割り当て - 『オペレーション』ドキュメント
- 『SYSPARM ユーティリティ』ドキュメント

## セッションパラメータの使用

セッションパラメータは、アクティブな Natural セッション内や Natural オブジェクト内で指定されます。セッションパラメータの主な目的は、Natural プログラムの実行を制御することです。

### ▶手順 9.4. セッションパラメータを設定するには

- 『パラメータリファレンス』ドキュメントの「セッションパラメータの設定方法」で説明する手順に従います。

### ▶手順 9.5. 現在のセッションのパラメータ設定をチェックまたは変更するには

- 次のいずれかの方法を選択します。
  1. 次のシステムコマンドを入力します。

```
GLOBALS
```

ENTER キーを押します。

次の例のような [Session Parameters] 画面に、セッションパラメータの現在の設定が表示されます。

```
09:51:55          ***** NATURAL GLOBALS COMMAND *****          2005-06-16
                   - Session Parameters -

(CC) Cond.Prog.Execution ..... OFF  (MT) Max.CPU Time ..... _____0
(CF) Term.Control Character .... %  (NC) Nat.Sys.Commands ..... OFF
```

```

(DC) Dec. Character ..... . (OPF) Overw.Prot.Fields ..... ON
(DFOUT) Date Format Output ..... S (PD) Page Dataset ..... _50
(DFSTACK) Date Format Stack Cmd .. S (PM) Print Mode ..... RP
(DFTITLE) Date Format Title ..... S (PS) Page Size ..... _23
(DU) Dump Generation ..... OFF (REINP) Reinput on Error ..... ON
(EJ) Page Eject ..... ON (SA) Sound Alarm ..... OFF
(FS) Default Format ..... OFF (SF) Spacing Factor ..... _1
(FCDP) FC on Dyn.Prot.Fields ... ON (SL) Source Line Length ..... _72
(IA) Input Assign ..... = (SM) Structured Mode ..... OFF
(ID) Input Delimiter ..... , (TS) Translate Sys.Prog ..... OFF
(IM) Input Mode ..... F (WH) Wait on Hold ..... OFF
(LE) Limit Error ..... OFF (ZD) Zero Division ..... ON
(LS) Line Size ..... _80 (ZP) Zero Printing ..... ON
(LT) Limit ..... __99999999

Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit          Updat                               Canc

```

カッコで囲んだ文字は、セッションパラメータを表します。『システムコマンド』ドキュメントの「*GLOBALS*」の「パラメータのリスト」も参照してください。

目的のパラメータの横に入力されたデフォルト値を上書きすることで、パラメータの設定を変更できます。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』ドキュメントの「セッションパラメータ」にあるセッションパラメータの説明を参照してください。

例：

日付フォーマットのパラメータ *DFOUT* の設定を変更するには、**[(DFOUT) Date Format Output]** 入力フィールドの横にある値（上の例では、*S*）を *I* に変更します。

変更を保存するには、PF5 (Updat) キーを押します。

日付変数のフォーマットが *yy-mm-dd* (例えば、2005-06-16) から *yyymmdd* (例えば、20050616) に変更されます。

## 2. 次のシステムコマンドを入力します。

```
GLOBALS parameter=value
```

上記の意味は次に示すとおりです。

*parameter* はセッションパラメータです (次の例では、DFOUT)。

*value* はこのセッションパラメータの有効値です (次の例では、I)。

複数のパラメータと値を指定できます。

例えば、次のように入力します。

```
GLOBALS DFOUT=I PS=60
```

ENTER キーを押します。

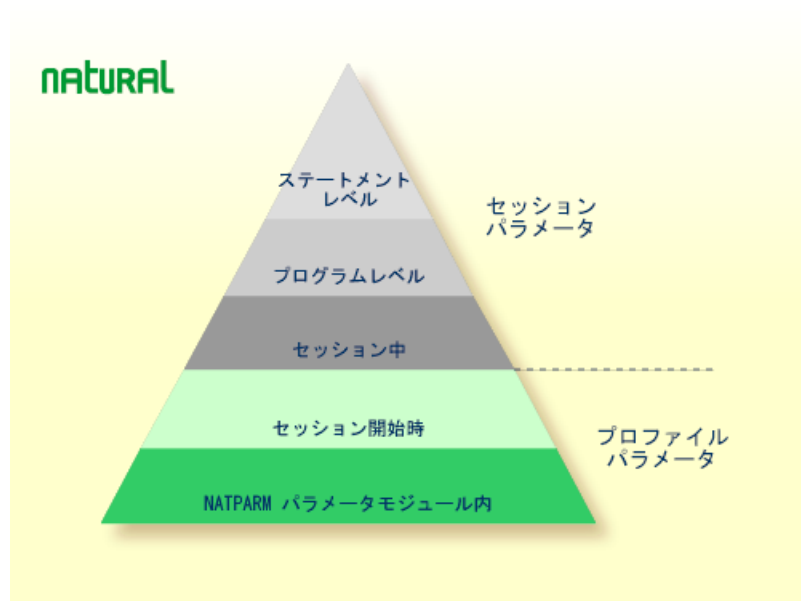
関連トピック：

- セッションパラメータについて - 『パラメータリファレンス』ドキュメント
- セッションパラメータの概要 - 『パラメータリファレンス』ドキュメント
- GLOBALS - 『システムコマンド』ドキュメント

## パラメータ化のレベル

Naturalパラメータの設定レベルは階層構造になっています。上のレベルで設定されたパラメータ値は、下のレベルで定義された値より優先されます。例えば、パラメータをダイナミックに指定すると、Naturalパラメータモジュール NATPARM で対応するパラメータに対して設定されたスタティックな指定より、新しいパラメータ値が優先されます。

下の図は、Naturalパラメータの設定可能なタイミングとその階層構造を（底辺が最下レベル、頂点が最上レベルのピラミッド方式で）示したものです。



関連トピック：

- *Natural* パラメータ階層 - 『オペレーション』ドキュメント

# 10      ルールと命名規則

---

▪ 標準の文字設定 .....	100
▪ オブジェクトの命名規則 .....	100
▪ ライブラリの命名規則 .....	101
▪ ユーザー定義変数の命名規則 .....	102

このセクションでは、Natural 固有のルールおよび命名規則について説明します。

## 標準の文字設定

---

このセクションでは、Natural に適用される標準の文字設定のリストが複数の表にまとめられています。この設定は、図形文字に対するコードポイントの割り当てを指定する IBM コードページ 1140（アメリカ英語）に準拠しています。コードポイントは 16 進値で表されます。IBM コードページの詳細については、該当する IBM のドキュメントを参照してください。

ディスプレイ制御装置またはエミュレータによって、16 進値に対して表示する文字が決まります。1140 以外のコードページを使用すると、次のような結果になる可能性があります。

- 文字が Natural で受け入れられない 16 進値になります。
- 標準の文字設定と異なる文字が、標準文字用に定義された 16 進値と同じになります。

例：

ドイツ語コードページ 1141 で使用される文字 @（アットマーク）には、16 進値 7C が存在しません。

イギリス英語コードページ 1146 で使用される文字 £（ポンド記号）の 16 進値は、アメリカ英語コードページ 1140 で使用される \$（ドル記号）と同じになります。

ドイツ語コードページ 1141 で使用される文字 §（セクション記号）の 16 進値は、アメリカ英語コードページ 1140 で使用される文字 @ と同じになります。

## オブジェクトの命名規則

---

このセクションでは、Natural オブジェクトを Natural システムファイルに保存したりカタログ化したりする場合に適用される命名規則について説明します。

Natural オブジェクト名には 1～8 文字（下の表に示されている文字）を使用でき、最初の文字は次のいずれかである必要があります。

- 大文字の英字
- 番号記号 (#)
- プラス記号 (+)

最初の文字が番号記号 (#) またはプラス記号 (+) の場合、少なくとも 1 文字を名前に追加する必要があります。



例外：

Natural DDM 名には 1～32 文字（下の表に示されている文字）を使用でき、最初の文字は大文字の英字である必要があります。

Natural オブジェクト名には、次の文字を使用できます。

文字	EBCDIC 16 進数値	ISO 図形文字名	注釈
A～Z	C1～C9 D1～D9 E2～E9	大文字のラテン文字 (A～Z)	大文字の英字
0 - 9	F0～F9	数字の 0～9	数値
-	60	ハイフン - マイナス	ハイフン
_	6D	下線	下線
/	61	ソリダス	スラッシュ
@	7C	アットマーク	
\$	5B	ドル記号	
&	50	アンパサンド	言語コードでのみ使用できます。  『プログラミングガイド』の「Natural オブジェクトの言語の定義」も参照してください。
#	7B	番号記号	ハッシュ記号
+	4E	プラス記号	最初の文字にのみ使用できます。

## ライブラリの命名規則

このセクションでは、Natural ライブラリに適用される命名規則について説明します。

ライブラリの名前 (ID) には 1～8 文字を使用でき、"SYS" で始めることはできません。接頭辞 "SYS" は、Natural システムライブラリ用に予約されています。

ライブラリ名は大文字の英字から始める必要があります。他の英字も大文字である必要があります。ライブラリ名には空白文字を含めることはできません。

ライブラリ名には、次の文字を使用できます。

文字	EBCDIC 16 進数値	ISO 図形文字名	注釈
A～Z	C1～C9 D1～D9 E2～E9	大文字のラテン文字 (A～Z)	大文字の英字
0 - 9	F0～F9	数字の 0～9	数値
-	60	ハイフン - マイナス	ハイフン
_	6D	下線	下線 最後の文字として使用できません。

## ユーザー定義変数の命名規則

---

このセクションでは、ユーザー定義変数に適用される命名規則について説明します。

- 変数名の長さ
- 変数名の制限
- 変数名に使用できる文字
- 変数名の最初の文字
- 変数名の大文字と小文字

ユーザー定義変数の詳細については、『プログラミングガイド』の「ユーザー定義変数」セクションを参照してください。

### 変数名の長さ

ユーザー定義変数名には、1～32 文字を使用できます。

32 文字を超える変数名を使用できますが（複雑なアプリケーションで、意味のある長い変数名によりプログラムの読みやすさが向上する場合など）、最初の 32 文字は区別できるように一意である必要があります。残りの文字は Natural によって無視されます。

### 変数名の制限

ユーザー定義変数の名前には、Natural の予約キーワードは使用できません。

1つの Natural プログラム内では、ユーザー定義変数とデータベースフィールドに同じ名前を使用できません。参照エラーを引き起こすことがあります。『プログラミングガイド』の「データ構造の条件指定」を参照してください。

## 変数名に使用できる文字

ユーザー定義変数名には、次の文字を使用できます。

文字	EBCDIC 16 進数値	ISO 図形文字名	注釈
A~Z	C1~C9 D1~D9 E2~E9 81~89 91~99 A2~A9	大文字/小文字のラテン文字 (A~Z)	大文字/小文字の英字  小文字を最初の文字として使用することはできません。
0-9	F0~F9	数字の 0~9	数値
-	60	ハイフン-マイナス	ハイフン
_	6D	下線	下線
/	61	ソリダス	スラッシュ
@	7C	アットマーク	
\$	5B	ドル記号	
&	50	アンパサンド	
#	7B	番号記号	ハッシュ記号
+	4E	プラス記号	最初の文字にのみ使用できます。

## 変数名の最初の文字

名前の最初の文字は、次のいずれかである必要があります。

文字	EBCDIC 16 進数値	ISO 図形文字名	注釈
A~Z	C1~C9 D1~D9 E2~E9	大文字のラテン文字 (A~Z)	大文字の英字
&	50	アンパサンド	
#	7B	番号記号	ハッシュ記号
+	4E	プラス記号	

最初の文字が番号記号 (#)、プラス記号 (+)、またはアンパサンド (&) である場合は、少なくとも 1 文字を名前に追加する必要があります。

最初の文字としてプラス記号 (+) を使用しているグローバルデータエリア (GDA) の変数は、レベル1で定義する必要があります。『プログラミングガイド』の「グローバルデータエリア」を参照してください。他のレベルは再定義だけで使用します。

アプリケーションに依存しない変数 (AIV) およびグローバルデータエリア (GDA) の変数についてのみ、名前の最初の文字としてプラス記号 (+) を使用できます。

AIV名はプラス記号 (+) で始まる必要があります。『ステートメント』ドキュメントの「DEFINE DATA」セクションにある「アプリケーションに依存しない変数の定義」も参照してください。

名前の先頭文字にアンパーサンド (&) を使用すると、ソースプログラムのダイナミック変更（『ステートメント』ドキュメントの RUN ステートメントを参照）や、ダイナミックに置換可能な文字として処理ルールの定義（『マップエディタ』ドキュメントの該当する説明を参照）を行うことができます。

### 変数名の大文字と小文字


小文字は、変数名の 2 番目以降の文字としてのみ入力できます。

COMPOPT システムコマンド（『システムコマンド』ドキュメントも参照）の LOWSRCE オプションが ON に設定されている場合は、変数名の一部として入力された小文字は大文字に内部変換されます。

小文字は大文字に変換されないため、次の両方の条件が満たされている場合には小文字と大文字が異なるものとして解釈されます。

- COMPOPT システムコマンドの LOWSRCE オプションが OFF（デフォルト値）に設定されている場合。
- エディタでの入力が大文字に変換されない場合。エディタでの大文字への変換は、エディタプロファイルオプションおよびオペレーティングシステムに応じたオプションによって制御されます。

例えば、これにより名前 #FIELD と #field は 2 つの異なるフィールド名として解釈されます。

 **注意:** 互換性の理由から、メインフレームコンピュータで開発した Natural アプリケーションを Windows、UNIX、または OpenVMS に移植する場合には、この機能を使用しないでください。変数名の一部として小文字を使用する場合は、変数名を大文字と小文字に関係なく一意にすることをお勧めします。

## 索引

---

---