

Natural for UNIX

システムコマンド

バージョン 8.4.1

2017 年 10 月

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

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

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

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

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

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

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

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

目次

前書き	vii
1	1
表記規則	2
オンライン情報	2
データ保護	3
2 システムコマンドの発行	5
コマンド入力	6
コマンド行	6
NEXT プロンプト	6
MORE プロンプト	6
3 システムコマンド構文	7
構文要素	8
コマンド構文の例	9
4 カテゴリ別システムコマンド	11
Natural での操作	12
Natural 開発環境	12
Natural オブジェクトでのアプリケーションの管理	13
モニタリング、デバッグ、トレース	14
その他の Software AG 製品との Natural	15
5 CATALL	17
対話式モードでの CATALL	18
バッチモードでの CATALL	19
6 CATALOG	21
7 CHECK	23
8 CLEAR	25
9 COMPOPT	27
構文説明	28
コンパイラオプション	28
コンパイラパラメータの指定	28
10 DEBUG	31
11 EDIT	33
構文 1	34
構文 2	36
構文 3	36
12 EXECUTE	37
構文説明	38
EXECUTE コマンドの例	39
13 FIN	41
14 GLOBALS	43
構文説明	44
パラメータのリスト	44
SET GLOBALS とその他のステートメントとの相互作用	45
15 HELP	47

16 INPL	49
17 KEY	51
コマンドの割り当て	52
すべてのキーのアクティブ化／非アクティブ化 - KEY ON/OFF	53
個別キーのアクティブ化／非アクティブ化 - KEY key=ON/OFF	53
18 LAST	55
19 LASTMSG	57
20 LIST	59
構文の概要	60
個々のソースの表示	61
オブジェクトのリストの表示	62
オブジェクトの数とサイズの表示	63
ディレクトリ情報の表示	64
ビューの表示	64
リソースオブジェクトのファイル情報の表示	65
エラーメッセージコンテナのファイル情報の表示	65
21 LIST XREF	67
22 LOGOFF	69
23 LOGON	71
24 MAIL	73
25 PROFILE	75
26 PROFILER	77
27 PURGE	79
28 READ	81
29 RENAME	83
30 RENUMBER	85
31 RETURN	87
32 RPCERR	89
33 RUN	91
34 SAVE	93
35 SCAN	95
メニューオプション	96
サブコマンド	97
Natural Security 環境での SCAN	98
36 SCRATCH	99
37 SETUP	101
構文説明	102
例：SETUP/RETURN	103
38 STOW	105
39 STRUCT	107
ソースコード行のインデント	108
40 SYSDDM	111
41 SYSERR	113
42 SYSEXT	115
43 SYSEXV	117

44	SYSFILE	119
45	SYSMAIN	121
46	SYSNCP	123
47	SYSOBJH	125
48	SYSPROD	127
49	SYSPROF	129
50	SYSRPC	131
51	SYSWIZDB	133
52	SYSWIZDW	135
53	TECH	137
54	UNCATALOG	139
55	UPDATE	141
56	XREF	143

前書き

このドキュメントでは Natural システムコマンドについて説明します。

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。

このドキュメントは次の項目で構成されています。

システムコマンドの発行	Natural システムコマンドの入力時に適用される一般的なルールについて説明します。
システムコマンド構文	Natural システムコマンドの構文記述に使用される記号について説明します。
カテゴリ別システムコマンド	カテゴリ別に分類された Natural システムコマンドの概要について説明します。

システムコマンド一覧（アルファベット システムコマンド（アルファベット順）の説明です。
順）

vrs または *vr* の表記

このドキュメントで使用されている場合、表記 *vrs* または *vr* は、関連する製品バージョン（『用語集』の「バージョン」も参照）を表します。

1

■ 表記規則	2
■ オンライン情報	2
■ データ保護	3

表記規則

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

オンライン情報

Software AG マニュアルの Web サイト

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

Software AG Empower 製品のサポート Web サイト

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

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

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

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

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

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

Software AG TECHcommunity

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

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

データ保護

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

2 システムコマンドの発行

■ コマンド入力	6
■ コマンド行	6
■ NEXT プロンプト	6
■ MORE プロンプト	6

コマンド入力

システムコマンドを実行するには、次のいずれかの方法でコマンドを入力します。

- **コマンド行**で入力します。
- Natural **NEXT** または **MORE** プロンプトで入力します。

以下のルールが適用されます。

- コマンド入力では、大文字と小文字は区別されません。
- コマンドはコンテキスト依存です。
- 一部の Natural コマンドは、現在アクティブなオブジェクト以外のオブジェクトにも影響を与えます。

構文の記述で使用する記号の詳細については、「[システムコマンド構文](#)」を参照してください。

コマンド行

メインメニューから「**Direct**」メニューを選択すると、システムコマンドを入力する「**Direct Command**」ウィンドウが表示されます。

NEXT プロンプト

NEXT プロンプトは、保留になっている出力が残っていない場合に Natural アプリケーションやプログラムで表示されます。

MORE プロンプト

MORE プロンプトは、保留になっている出力がまだあることを示すために出力画面の下段に表示されます。MORE プロンプトに応答してシステムコマンドを入力すると、プログラムの実行は中断されて、システムコマンドが実行されます。

3 システムコマンド構文

■ 構文要素	8
■ コマンド構文の例	9

構文要素

システムコマンドの構文の記述には、次の記号を使用します。

要素	説明
ABCDEF	斜体ではない大文字の用語は Natural キーワードまたは Natural 予約語のいずれかで、示されたとおりに入力する必要があります。
<u>ABCDEF</u>	大文字のオプション用語全体に下線が付いている（ハイパーリンクではない）場合は、その用語がデフォルト値であることを示します。用語を省略すると、下線の付いた値が適用されます。
<u>ABCDEF</u>	大文字の用語の一部に下線が付いている（ハイパーリンクではない）場合は、下線部分はその用語の入力可能な省略形です。
<i>abcdef</i>	斜体の文字は、変数情報を表します。この用語を指定する場合は、有効な値を入力する必要があります。
[]	角カッコ内の要素はオプションです。 角カッコに複数の行が入れ子になって含まれている場合は、各行がオプションの選択肢です。選択肢の中から 1 つ選択できます。
{ }	中カッコに複数の行が入れ子になって含まれている場合は、各行が選択肢です。選択肢の中から 1 つのみ選択する必要があります。
	選択肢は縦棒で区切られます。
...	省略記号の前の用語は任意に繰り返すことができます。省略記号の後の数値は用語の繰り返し回数を示します。 省略記号の前の用語が角カッコまたは中カッコに囲まれている場合、省略記号はカッコ全体に適用します。
,...	コンマと省略記号の前の用語は任意に繰り返すことができます。繰り返す場合は、繰り返しをコンマで区切る必要があります。コンマと省略記号の後の数値は用語の繰り返し回数を示します。 コンマと省略記号の前の用語が角カッコまたは中カッコに囲まれている場合、コンマと省略記号はカッコ全体に適用します。
:...	コロンと省略記号の前の用語は任意に繰り返すことができます。繰り返す場合は、繰り返しをコロンの区切る必要があります。コロンと省略記号の後の数値は用語の繰り返し回数を示します。 コロンと省略記号の前の用語が角カッコまたは中カッコに囲まれている場合、コロンと省略記号はカッコ全体に適用します。
その他の記号 ([] { } , : ; ... を除く)	他のすべての記号は、この表で定義しているものを除いて、示されたとおりに入力する必要があります。 例外：

要素	説明
	SQL スカラ連結演算子は 2 つの縦線で表現され、構文定義に記述されているとおりに完全に入力する必要があります。

コマンド構文の例

`CATALOG [object-name [library-id]]`

- CATALOG は Natural キーワードであり、指定する場合は正しく入力する必要があります。下線部分 (CAT) は省略形として入力できることを示しています。
- *object-name* および *library-id* はユーザーが設定するオペランドであり、処理するプログラムの名前とそのプログラムが存在するライブラリ ID を指定します。
- 大カッコは、*object-name* および *library-id* がオプションであることを示します。必要に応じて指定してください。カッコによるグループ化は、CATALOG だけを指定することも、あるいは、CATALOG の後にプログラム名だけ、またはプログラム名とライブラリ ID を指定することも可能であることを示しています。ただし、プログラム名を指定せずにライブラリ ID を指定することはできません。

4 カテゴリ別システムコマンド

■ Natural での操作	12
■ Natural 開発環境	12
■ Natural オブジェクトでのアプリケーションの管理	13
■ モニタリング、デバッグ、トレース	14
■ その他の Software AG 製品との Natural	15

この章は、カテゴリ別の Natural システムコマンドの概要です。

Natural での操作

Command	簡単な説明
FIN	Natural セッションを終了します。
LAST	最後に実行されたシステムコマンドを表示し、再度実行することができます。
LOGOFF	ライブラリ ID に SYSTEM を設定し、Adabas パスワードに空白を設定します。ソースプログラムワークエリアの内容は、このコマンドの影響を受けません。
LOGON	ユーザーにライブラリ ID を設定します。セッション中に保存した全ソースまたはオブジェクトプログラムが、指定したライブラリに保存されます (SAVE、CATALOG、または STOW コマンドに別のライブラリ ID を明示的に指定する場合を除く)。
RETURN	SETUP コマンドで設定した戻り位置に戻ります。
SETUP	RETURN コマンドを使用して制御を戻すことができる戻り位置を設定します。これにより、Natural セッション中にアプリケーション間を容易に移動できます。

Natural 開発環境

Command	簡単な説明
COMPOPT	Natural オブジェクトのコンパイル方法を制御するコンパイルオプションを設定します。
GLOBALS	Natural セッションパラメータの設定を変更します。
HELP	Natural ヘルプシステムを呼び出します。
KEY	Natural セッションで使用するキーに、機能を割り当てます。
LAST	最後に実行されたシステムコマンドを表示し、再度実行することができます。
LIST XREF	現在ログオンしているライブラリのアクティブクロスリファレンスをすべて表示します。 (適用されるのは、Predict がインストールされている場合に限られます)
SYSEXT	Natural アプリケーションプログラミングインターフェイスを含む SYSEXT ユーティリティを呼び出します。
SYSEXV	現在の Natural バージョンおよびデバッグヒントで採用された新機能の例が含まれている SYSEXV ユーティリティを呼び出します。
SYSFILE	ワークファイルおよび出力ファイルの情報を表示します。
SYSPROD	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。
SYSPROF	Natural システムファイルの現在の定義を表示します。
UPDATE	プログラムによるデータベース更新の実行を阻止します。

Command	簡単な説明
XREF	Predict の「アクティブクロスリファレンス」機能の使用を制御します。プログラムまたはデータエリアが参照しているオブジェクトについてのドキュメントを、Predict に自動で作成します。 (適用されるのは、Predict がインストールされている場合にに限られます)

Natural オブジェクトでのアプリケーションの管理

Command	簡単な説明
CATALL	現在のライブラリのすべてのオブジェクトまたは選択したオブジェクトをカタログ（コンパイル）します。
CATALOG	現在、エディタのワークエリアにあるソースコードをカタログ化（コンパイル）し、構文が正しければ、作成されたカタログ化オブジェクトを Natural システムファイルに保存します。
CHECK	エディタのワークエリアに現在表示されているソースコードに構文エラーが含まれていないことを確認します。 構文チェックは、RUN、CATALL、CATALOG、および STOW システムコマンドの一部としても実行されます。
CLEAR	エディタのワークエリアの内容がクリアされます。
EDIT	ソースコードを作成または変更するためのエディタを開きます。
EXECUTE	カタログ化された（コンパイルされた）プログラムを実行します。このプログラムは、カタログ化オブジェクトとして Natural システムファイルに保存されています。
INPL	INPL ユーティリティを呼び出します。Software AG インストールデータセットをシステムファイルにロードするためにのみ使用します。
LIST	現在のライブラリに含まれている 1 つまたは複数のオブジェクト、またはエディタのワークエリアの内容をリストします。
PURGE	現在のライブラリから 1 つまたは複数のソースオブジェクトを削除します。
READ	Natural システムファイルからエディタのワークエリアに、ソースオブジェクトを読み込みます。
RENAME	オブジェクトの名前、またはオブジェクトの名前とタイプを変更します。
RENUMBER	現在エディタのワークエリアにあるソースコードの番号を変更します。
RUN	現在エディタのワークエリアにあるソースプログラムを、コンパイルして実行します。
SAVE	現在エディタのワークエリアにあるソースコードをソースオブジェクトとして Natural システムファイルに保存します。
SCAN	ソース内の文字列を検索し、その文字列を置換するオプションを指定します。
SCRATCH	ソースオブジェクトや対応するカタログ化オブジェクトを現在のライブラリから削除します。

カテゴリ別システムコマンド

Command	簡単な説明
STOW	ソースコードをカタログ化（コンパイル）し、ソースオブジェクトとカタログ化オブジェクトの両方として現在の Natural システムファイルに保存します。
STRUCT	プログラムソースの構造的なインデントを実行します。これは、構造的な不整合の発見に役立ちます。
SYSDDM	Natural データ定義モジュール（DDM）を作成およびメンテナンスします。
SYSERR	Natural アプリケーションでユーザーに対して表示するメッセージを作成およびメンテナンスします。
SYSMAIN	Natural システム内の Natural オブジェクトを、ライブラリ間で転送します。
SYSNCP	Natural アプリケーションで使用するコマンドプロセッサを作成およびメンテナンスします。
SYSOBJH	Natural 環境での分散のために、Natural と Natural 以外のオブジェクトを処理します。
SYSRPC	SYSRPC ユーティリティを呼び出して、リモートプロシージャコールを作成およびメンテナンスします。つまり、リモートサーバーにあるサブプログラムを実行するために必要な設定を提供します。
SYSWIZDB	Natural データブラウザを呼び出します。このブラウザは、Natural スタジオ内の開発ツールウィザードです。これにより、ファイル構造を表示、出力、または保存することができます。
SYSWIZDW	Natural Dialog Wizard を呼び出します。これは、特定用途のためのダイアログを作成するツールです。定義済みダイアログボックスに、目的に適合する複数のレイアウトを設定できます。
UNCATALOG	現在のライブラリで1つまたは複数のカタログ化オブジェクトを削除します。

モニタリング、デバッグ、トレース

Command	簡単な説明
DEBUG	このコマンドは、Natural デバッガを呼び出すために使用します。
HELP	Natural ヘルプシステムを呼び出します。
LASTMSG	最後に発生したエラー状況に関する追加情報を表示します。
RPCERR	Natural RPC（リモートプロシージャコール）に関連している場合は、最後の Natural エラー番号とメッセージを表示します。また、最後の EntireX Broker 理由コードと関連メッセージを表示します。
TECH	Natural セッションの技術的な情報およびその他の情報を表示します。

その他の Software AG 製品との Natural

次のシステムコマンドは、サイトにインストールされている他の Software AG 製品についてのみ使用できます。

製品	Command	簡単な説明
Natural Security	MAIL	メールボックスを呼び出し、その内容と満了日のいずれかまたは両方を変更します。メールボックスは、Natural ユーザーにメッセージを送る掲示板です。
	PROFILE	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。このプロファイルは、現在の Natural 環境の実際の使用条件を知らせます。
Predict	LIST XREF	現在ログオンしているライブラリのアクティブクロスリファレンスをすべて表示します。
	XREF	Predict の「アクティブクロスリファレンス」機能の使用を制御します。プログラムまたはデータエリアが参照しているオブジェクトについてのドキュメントを、Predict に自動で作成します。

5 CATALL

■ 対話式モードでの CATALL	18
■ バッチモードでの CATALL	19

このコマンドは、現在のライブラリ内のすべてのオブジェクトまたは選択したオブジェクトをカタログ、チェック、保存、または格納するために使用します。

対話式モードでの CATALL

CATALL

このコマンドを発行すると、**「Catalog Objects in Library」** ウィンドウが表示されます。このウィンドウで、処理するオブジェクトのタイプを指定します。オブジェクトは、オブジェクトタイプがウィンドウにリストされている順に処理されます。「**バッチモードでの CATALL**」セクションの TYPES に関する情報も参照してください。さらに、実行するアクションと処理対象のオブジェクトを選択できます。

このウィンドウで指定できる項目は、次のとおりです。

開始する名前	現在のライブラリで選択されているタイプのすべてのオブジェクトを処理する場合は、アスタリスク (*) を入力します。 オブジェクトの番号を特定の範囲に制限する場合は、名前にアスタリスク表記を使用できます。
存在するモジュールのみに適応	このオプションをマークすると、現在のライブラリにカタログ化オブジェクトがすでに存在するオブジェクトのみが再カタログされます。カタログ化オブジェクトが存在しないソースオブジェクトは処理されません。
全てのソースに適応	このオプションをマークすると、選択したすべてのオブジェクトが処理されます。
Action	PF2 キーを押すと、選択したオブジェクトに適用するアクションを次の中から1つ選択できます。 ■ Catalog ■ Check ■ Save ■ Stow これらのアクションは、同名のシステムコマンドに対応します。 注意: Natural Security では、特定のアクションが許可されない場合があります。
行番号再設定	デフォルトでは、保存または格納されたソースのソースコード行の番号も再設定されます。 行番号の自動再設定を望まない場合は、このフィールドの X を空白で上書きします。
オブジェクトタイプ	デフォルトでは、CATALL は現在のライブラリのあらゆるタイプのオブジェクトに適用されます。つまり、すべてのオブジェクトタイプが X でマークされていま

	す。特定のタイプのオブジェクトが CATALL で処理されないようにするには、該当する X を空白で上書きします。
マップソース変換	Natural の以前のバージョンで作成されたマップは、Natural バージョン 3.1 以降と必ずしも互換性がありません。このオプションをマークすると、カタログ処理中にマップが確実に変換されます。このオプションによって、マップがソースオブジェクトとカタログ化オブジェクトの両方として変換され、保存されます。

CATALL の処理中は統計ウィンドウが表示され、現在カタログされているオブジェクトがリストされます。



注意: CATALL の処理中にいずれかのキーを押すと、CATALL は停止します。

処理が正常に完了すると、情報メッセージが表示されます。

オブジェクトが正常にカタログされなかった場合は、そのオブジェクト名、エラー番号、およびエラーの行を示したウィンドウが表示されます。

CATALL コマンドが、Natural プログラム内からパラメータを上下に並べて呼び出された場合はバッチモードであるとみなされ、コンパイルエラーが発生してもエラーダイアログは表示されません。ダイアログの代わりに、Natural の一時ディレクトリ (TMP_PATH) に *library.CTL* という名前のファイルが作成されます。エラーが検出されなかった場合、このファイルは空ですが、エラーが検出された場合は、失敗したオブジェクトが行番号とエラーコードとともにファイルに含まれます。

バッチモードでの CATALL

<code>CATALL object-name [<u>RECAT</u>] [TYPEStypes] [<u>CATALOG</u> <u>CHECK</u> <u>SAVE</u> <u>STOW</u>] [options ...]</code>	

「Catalog Objects in Library」ウィンドウで実行可能なさまざまな指定に対応し、システムコマンド CATALL で直接指定できる次のようなオプションもあります。

<i>object-name</i>	カタログされるオブジェクトの名前。 現在のライブラリで指定されているタイプのすべてのオブジェクトをカタログする場合は、アスタリスク (*) を入力します。 オブジェクトの番号を特定の範囲に制限する場合は、名前にアスタリスク表記を使用できます。
RECAT / ALL	「Catalog Objects in Library」画面の「Apply action only to existing modules」または「Apply action to all sources」オプションに対応します。RECAT はデフォルトです。

TYPES types	〔Catalog Objects in Library〕画面でマークされるオブジェクトタイプに対応します。可能な <i>types</i> (オブジェクトタイプがリストされた順にオブジェクトが処理される) : D DDM G グローバルデータエリア L ローカルデータエリア A パラメータデータエリア C コピーコード T テキスト 7 機能 N サブプログラム S サブルーチン H ヘルプルーチン M マップ 8 アダプタ P プログラム 4 クラス 3 ダイアログ * すべてのタイプ (デフォルト) <i>types</i>は、1つの文字列 (グローバルに対する GLA など)、ローカル、およびパラメータデータエリアで指定する必要があります。デフォルトでは、CATALL は現在のライブラリ内にあるすべてのタイプのオブジェクトに適用されます。	
CATALOG / CHECK / SAVE / STOW	〔 Catalog Objects in Library 〕画面の同名のアクションに対応します。CATALOGはデフォルトです。	
options	NOREN	ソースオブジェクトのソースコード行の自動番号再設定は行われません。

 **注意:** 個々のコマンドコンポーネントは、空白または入力デリミタ文字 (セッションパラメータ ID で定義) で区切る必要があります。

6 CATALOG

CATALOG [*object-name* [*library-id*]]

関連するコマンド：SAVE | STOW.

このコマンドは、現在、Naturalエディタのワークエリアにあるソースコードをカタログ化（コンパイル）し、（構文が正しければ）作成されたカタログ化オブジェクトを現在のNaturalシステムファイルに保存するために使用します。

 **重要:** CATALOG コマンドは、プロファイルパラメータ RECAT が ON に設定されている場合には使用できません。この場合は、STOW コマンドを使用してオブジェクトをコンパイルおよび保存します。

CATALOG	<i>object-name</i> を指定しない場合、オブジェクトは、EDIT または READ などを使用してソースワークエリアに最後に読み込まれたオブジェクトの名前で現在のライブラリにカタログされます。既存のオブジェクトコードは置き換えられます。
CATALOG <i>object-name</i>	新しいオブジェクトが作成されます。 <i>object-name</i> には、新しいオブジェクトがカタログされるときの名前を指定します。オブジェクトは現在のライブラリに保存されます。オブジェクトが存在する場合、コマンドは拒否されます。
CATALOG <i>object-name</i> <i>library-id</i>	新しいオブジェクトを別のライブラリにカタログするには、そのライブラリの <i>library-id</i> を指定する必要があります。オブジェクトが存在する場合、コマンドは拒否されます。

 **注意:** 有効ではないパラメータファイルに FDIC システムファイルが指定された場合は、CATALOG コマンドが発行されたときに、Natural によって適切なエラーメッセージが表示されます。

7 CHECK

CHECK

このコマンドは、現在エディタワークエリアにあるソースコードの構文にエラーが含まれていないかどうかをチェックするために使用します。

構文エラーが検出されると、構文チェックは中断され、エラーが含まれる行が表示されます。ここでは、行を修正する（その後整合性チェックが継続します）か、または表示されている行を修正しないでENTERキーを押すことができます。この場合は整合性チェック手順が停止し、エディタが開きます。



注意: 構文チェックは、[RUN](#)、[STOW](#)、[CATALOG](#)、および [CATALL](#) コマンドの一部としても実行されます。

8 CLEAR

CLEAR

このコマンドは、エディタのソースワークエリアをクリアするために使用します。新しいプログラムを作成するときに、ソースワークエリアに別のオブジェクトがある場合に使用することができます。

9 COMPOPT

■ 構文説明	28
■ コンパイラオプション	28
■ コンパイラパラメータの指定	28

COMPOPT [*option=value* ...]

このシステムコマンドは、各種のコンパイラオプションを設定するために使用します。設定したオプションは、Natural オブジェクトのコンパイル時に評価されます。

オプションなしで COMPOPT コマンドを入力すると、画面が表示されます。この画面では、以降で説明するオプションを有効または無効にできます。

個々のオプションのデフォルト設定は、Natural パラメータファイル内の対応するプロファイルパラメータを使用して設定されます。

構文説明

COMPOPT	オプションなしで COMPOPT システムコマンドを発行すると、[Compilations Options] 画面が表示されます。ここで使用可能なキーワードについては、次で説明します。
COMPOPT <i>option=value</i>	画面でオプションを変更する代わりに、COMPOPT コマンドを使用して直接指定することもできます。 例： COMPOPT DBSHORT=ON

コンパイラオプション

次のコンパイラオプションを使用できます。各オプションの目的および可能な設定の詳細については、対応する Natural プロファイルパラメータの解説を参照してください。

DBSHORT | GFID | KCHECK | MASKCME | MAXPREC | PCHECK | PSIGNF | THSEP | TQMARK

コンパイラパラメータの指定

コンパイラパラメータは、次のような異なるレベルで指定できます。

1. デフォルト設定として指定

個々のコンパイラパラメータのデフォルト設定は、コンフィグレーションユーティリティの [Compiler Options] カテゴリを使用して指定し、Natural パラメータファイル NATPARM に保存されます。

2. セッション開始時に指定

セッション開始時に、対応するプロファイルパラメータを指定することによって、コンパイラオプション設定を上書きできます。

3. アクティブな **Natural** セッション中に指定

アクティブな **Natural** セッション中に COMPOPT システムコマンドを使用してコンパイラのパラメータ値を変更するには2つの方法があります。コマンド割り当て (COMPOPT *option=value*) を使用して直接行う方法、およびオプションなしで COMPOPT コマンドを発行して [**Compilation Options**] 画面を表示する方法です。コンパイラオプションに割り当てられた設定は、次の LOGON コマンドを別のライブラリに発行するまで有効です。別のライブラリへの LOGON 時には、デフォルト設定（上記の項目 1 を参照）に戻ります。例：

```
OPTIONS KCHECK=ON
DEFINE DATA LOCAL
1 #A (A25) INIT <'Hello World'>
END-DEFINE
WRITE #A
END
```

4. **Natural** オブジェクトで

プログラムやサブプログラムなどの **Natural** オブジェクトに、OPTIONS ステートメントを使用してコンパイラパラメータを設定できます。例：

```
OPTIONS KCHECK=ON
WRITE 'Hello World'
END
```

OPTIONS ステートメントで定義したコンパイラオプションは、このオブジェクトのコンパイルにのみ影響し、COMPOPT コマンドで設定した一連の設定を更新することはありません。

10

DEBUG

`DEBUGobject-name`

このコマンドは、Naturalデバッガを呼び出すために使用します。コマンドには、デバッグするオブジェクトの名前を指定します。

デバッガは、リモート開発環境（SPoD）で Natural スタジオを使用している場合にのみ使用できます。



注意: このコマンドは、バッチモードでは実行できません。

11

EDIT

■ 構文 1	34
■ 構文 2	36
■ 構文 3	36

このコマンドは、Natural ソースを作成および編集する目的で Natural エディタを呼び出すために使用します。



注意: このコマンドは、バッチモードでは実行できません。

コマンド構文には3つの形式があります。これについては、以降のドキュメントで説明されています。

関連するコマンド：[READ](#)。

『Naturalの使用』ドキュメントの「オブジェクトの命名規則」も参照してください。

構文 1

```
EDIT [object-type] [object-name [library-id]]
```

object-type

次のオブジェクトタイプを編集できます。

{	<u>C</u> CLASS	}
	4	
<u>C</u> OPYCODE		
<u>G</u> LOBAL		
<u>H</u> ELPROUTINE		
<u>L</u> OCAL		
<u>M</u> AP		
{	<u>P</u> ARAMETER	}
	<u>P</u> ROGRAM	
	{	
		SUBPROGRAM
		N
		}
	<u>S</u> UBROUTINE	
	<u>I</u> EXT	
	<u>V</u> IEW	
	7 (Function の場合)	

呼び出されるエディタは、編集するオブジェクトタイプに応じて、次のように異なります。

- ローカルデータエリア、グローバルデータエリア、またはパラメータデータエリアは、データエリアエディタで編集します。

- マップはマップエディタで編集します。
- クラスはプログラムエディタで編集します。
- EDIT VIEW は、現在のライブラリ内で *object-name* が指定されている場合にのみ機能します。表示するオブジェクトが DDM の場合は、DDM サービスが呼び出されます。
- その他すべてのオブジェクトタイプ（プログラム、サブプログラム、サブルーチン、7（機能用）、ヘルプルーチン、コピーコード、テキスト、説明）は、プログラムエディタで編集します。



注意: テキストオブジェクト「説明」は、メインフレームのみで使用可能です。説明とは、Predict データディクショナリに保存され、メンテナンスされているプログラム説明です。このタイプのオブジェクトは、Predict がインストールされている場合にのみ編集できます。

オブジェクトタイプについては、『プログラミングガイド』の「*Natural* アプリケーション管理のオブジェクト」を参照してください。エディタについては、『エディタ』ドキュメントを参照してください。



注意: プログラムエディタから EDIT *object-name* を入力すると、*Natural* オブジェクトのタイプがコピーコード、クラス、7（機能用）、ヘルプルーチン、プログラム、サブプログラム、サブルーチン、テキストのいずれの場合でも、並行する編集セッションが開きます。プログラムエディタコマンド NEXT の説明も参照してください。

編集するオブジェクトの名前を指定する場合、そのオブジェクトタイプを指定する必要はありません。

object-name

EDIT コマンドを使用して、編集するオブジェクトの名前を指定します。オブジェクト名の最大長は 8 文字です。



注意: DDM の場合、最大長は 32 文字です。

この後は、*Natural*によってオブジェクトが適切なエディタの編集ワークエリアにロードされ、後続の **SAVE**、**CATALOG**、または **STOW** コマンドに対してオブジェクト名が設定されます。

object-name を指定せず、ソースワークエリアにオブジェクトが存在しない場合は、空のプログラムエディタ画面が呼び出され、そこでプログラムを作成できます。ソースワークエリアが空ではない場合、オブジェクトは適切なエディタにロードされます。

library-id

編集するオブジェクトが現在ログオンしているライブラリに存在しない場合は、編集するオブジェクトが存在するライブラリの *library-id* を指定する必要があります。

NaturalSecurity がアクティブな場合は、*library-id* を指定できません。これは、現在のライブラリ内にあるオブジェクトしか編集できないことを意味します。

構文 2

EDIT [^{*}
object-type] { ^{*}
object-name }

編集するオブジェクトの名前を覚えていない場合は、この形式の EDIT コマンドを使用してオブジェクトのリストを表示し、そのリストから希望するオブジェクトを選択できます。

EDIT *	現在のライブラリに存在するすべてのオブジェクトのリストを表示します。
EDIT <i>object-type</i> *	現在のライブラリに存在する指定タイプのすべてのオブジェクトのリストを表示します。

特定範囲のオブジェクト群からオブジェクトを1つ選択するには、システムコマンド の記述と同じ方法で、*object-name* にアスタリスク表記とワイルドカード表記を使用できます。

構文 3

EDIT FUNCTION*subroutine-name*

EDIT FUNCTION コマンドは、最大 32 文字のサブルーチン名（オブジェクト名ではない）を使用してサブルーチンを編集するために使用できます。

 **注意:** この構文で使用するキーワード FUNCTION が、上記のリストにある Natural **オブジェクトタイプ**7（機能用）と同じではないことに注意してください。『プログラミングガイド』でファンクションオブジェクトタイプの説明を参照してください。

例：

```
DEFINE SUBROUTINE CHECK-PARAMETERS
...
END-SUBROUTINE
END
```

上記のサブルーチンがオブジェクト名 CHCKSUB で保存されている場合は、サブルーチン CHECK-PARAMETERS を次のコマンドで編集できます。

```
EDIT S CHKSUB
```

または、次のコマンドも使用できます。

```
EDIT F CHECK-PARAMETERS
```

12 EXECUTE

■ 構文説明	38
■ EXECUTE コマンドの例	39

```
{ EXECUTE [REPEAT]    program-name    [library-id] }
{ program-name [parameter ...]
```

このコマンドは、プログラムタイプのNaturalオブジェクトモジュールを実行するために使用します。オブジェクトモジュールは、あらかじめNaturalシステムファイルにカタログしておく（つまり、オブジェクト形式で保存しておく）か、またはNaturalニュークリアスにリンクしておく必要があります。オブジェクトモジュールを実行しても、現在エディタワークエリアにあるソースには影響しません。

構文説明

EXECUTE	キーワード EXECUTE はオプションであり、 <i>program-name</i> （実行するプログラムの名前）だけで指定できます。
REPEAT	実行されるプログラムで複数の画面が出力される場合に、これらの画面をプロンプトの介入なしに順番に出力するには、キーワード REPEAT をキーワード EXECUTE とともに指定します。
<i>program-name</i>	実行するプログラムの名前。 <i>library-id</i> を指定しない場合、Natural では、指定されたプログラムが現在のライブラリまたは現在の <i>steplib</i> ライブラリのいずれかにある場合でしか実行できません。デフォルトの <i>steplib</i> は SYSTEM です。
<i>library-id</i>	プログラムが別のライブラリにある場合は、そのライブラリの <i>library-id</i> を指定します。この場合は、指定したライブラリにプログラムが実際に保存されているときに限り実行できます。 SYSTEMを除き、SYS で始まる <i>library-id</i> は指定できません。
<i>parameter</i>	キーワード EXECUTE なしでプログラム名を指定してプログラムを実行するときは、プログラムにパラメータを渡すことができます。これらのパラメータは、実行されるプログラムの最初の INPUT ステートメントによって読み込まれます。 各パラメータは、位置指定パラメータまたはキーワードパラメータとして、セッションパラメータ ID で指定する場合と同様に、個別指定を空白または入力デリミタ文字でそれぞれ区切って指定できます。 注意: 渡されるパラメータの1つに空白が含まれているか、またはパラメータが空白を含む文字列である場合は、プログラム名の直後に入力デリミタが設定されているときにのみ転送が実行されます。

EXECUTE コマンドの例

```
EXECUTE PROG1
```

```
EXECUTE PROG1 ULIB1
```

```
PROG1
```

```
PROG1 VALUE1 VALUE2 VALUE3
```

```
PROG1 VALUE1, VALUE2, VALUE3
```

```
PROG1 PARM1=VALUE1, PARM2=VALUE2, PARM3=VALUE3
```

```
PROG1 PARM3=VALUE3 PARM1=VALUE1 VALUE2
```

```
PROG1,ab cd ef,gh,de fg,ab
```


13

FIN

EIN

このコマンドは、Naturalセッションを終了するために使用します。バッチモードだけでなく、オンラインセッションにも適用されます。

バッチモードセッションは、コマンド入力データセットでエンドオブファイル状態が検出された場合も終了します。

14 GLOBALS

■ 構文説明	44
■ パラメータのリスト	44
■ SET GLOBALS とその他のステートメントとの相互作用	45

GLOBALS [*parameter*=*value* ...]

このコマンドは、Natural セッションパラメータを設定するために使用します。

 **注意:** バッチモードでは、このコマンドはパラメータが指定された場合のみ実行可能です。例えば、GLOBALS SM=ON はバッチモードで実行できます。

構文説明

GLOBALS	GLOBALS コマンドをパラメータなしで入力すると、画面が表示され、ここでパラメータ設定を変更できます。
<i>parameter</i>	パラメータは任意の順序で指定できますが、空白で区切る必要があります。 指定する複数のパラメータが1つのコマンド行に入りきらない場合は、GLOBALS コマンドを複数に分けて発行する必要があります。 例： <pre>GLOBALS DC=, ID=.</pre> 注意: 一部のセッションパラメータは、上述の画面を使用してのみ変更でき、コマンド行を使用して変更することはできません。

パラメータのリスト

次の表に、システムコマンド GLOBALS で指定できるセッションパラメータのリストを示します。

パラメータ	関数
CC	バッチモードでのエラー処理
CF	端末コマンドの文字
CO	コンパイラ出力
CPCVERR	コードページ変換エラー
DBSHORT	データベースショートネームの解釈
DC	小数点表記の文字
DO	出力データの順序の表示
DU	ダンプ生成
EJ	ページ換え
ENDIAN	コンパイルしたオブジェクトのエンディアンモード
FS	ユーザー定義変数のデフォルトのフォーマット／長さ設定

パラメータ	関数
GFID	グローバルフォーマット ID
IA	INPUT 割り当て文字
ID	INPUT 区切り文字
IM	入力モード
LE	処理ループの制限超過時の処理
LS	Line Size
LT	処理ループの制限
ML	メッセージ行の位置
PD	ページデータセットのサイズ
PM	Print Mode
PS	Natural レポートのページサイズ
SA	サウンド端末アラーム
SF	フィールド間の空白
SM	ストラクチャードモードでのプログラミング
THSEPCH	千桁単位セパレータ文字
ZD	ゼロ除算チェック
ZP	ゼロ出力

SET GLOBALS とその他のステートメントとの相互作用

SET GLOBALS ステートメント

システムコマンド GLOBALS と SET GLOBALS ステートメントでは、変更と同じパラメータが提供されます。両方とも同じ Natural セッション内で使用できます。GLOBALS コマンドで指定したパラメータ値は、新しい GLOBALS コマンドまたは SET GLOBALS ステートメントで上書きされるか、セッションが終了するか、またはユーザーが別のライブラリにログオンするまで有効です。

セッションパラメータ設定に影響を与えるその他のステートメント

一部のパラメータ値は、プログラムの実行中に LIMIT、EJECT、および FORMAT の各ステートメントを使用するか、または INPUT、DISPLAY、PRINT、および WRITE の各ステートメントに指定したフォーマットエントリで上書きできます。

15

HELP

$\left\{ \begin{array}{l} \text{HELP} \\ ? \end{array} \right\} \left[\begin{array}{l} [\text{NAT}]nnnn \\ \text{USER}[nnnn] \end{array} \right]$
--

このコマンドは、エラーメッセージのオンラインヘルプを呼び出すために使用します。



注意: このコマンドは、バッチモードでは実行できません。

HELP	ヘルプメニューを表示します。
HELP [NAT]nnnn	HELP と番号（最大 4 桁、接頭辞 "NAT" は任意）を入力すると、この番号に関する Natural のエラー条件の詳細メッセージテキスト、つまり、Natural システムのエラーメッセージ NATnnnn の長文テキストが表示されます。
HELP USERnnnn	現在のライブラリのライブラリ固有のエラーメッセージ番号 nnnn の長文テキストを表示します。

16 INPL

INPL [R]

このコマンドは、Natural の INPL ユーティリティを呼び出すために使用します。このユーティリティは、Software AG インストールデータセットをシステムファイルにロードするためにのみ使用します。詳細については、INPL オンラインヘルプおよびプラットフォーム固有のインストールドキュメントを参照してください。

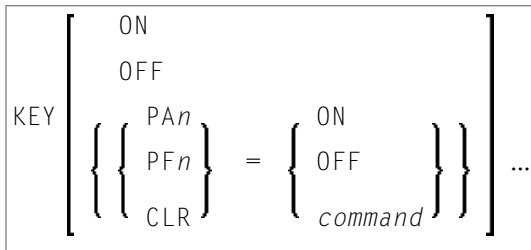
上記以外の場合は、オブジェクトハンドラを使用してオブジェクトをシステムファイルにロードします。

INPL	パラメータなしで INPL コマンドを入力すると、INPL ユーティリティが呼び出されます。
INPL R	INPL ユーティリティ機能の Natural Security リカバーが呼び出されます。これは、Natural Security がインストールされている場合にのみ使用できる機能です。 Natural Security ライブラリ SYSSEC へのアクセスをリセットするために使用できます。ユーザー DBA、ライブラリ SYSSEC、およびこの2つの間のリンクが初回インストール後と同じに再定義され、SYSSEC へのその他すべてのリンクが取り消されます。『Natural Security』ドキュメントの「Countersignatures」セクションの「Inaccessible Security Profiles」も参照してください。

詳細については、『ツールおよびユーティリティ』ドキュメントの「INPL ユーティリティ」を参照してください。

17 KEY

■ コマンドの割り当て	52
■ すべてのキーのアクティブ化／非アクティブ化 - KEY ON/OFF	53
■ 個別キーのアクティブ化／非アクティブ化 - KEY key=ON/OFF	53



このコマンドは、ビデオ端末のキーボードのキーに機能を割り当てるために使用します。割り当てた機能を変更、アクティブ化、非アクティブ化することもできます。

対象となるキーは次のとおりです。

- PA1 ～ PA3、
- PF1 ～ PF24
- CLEAR

上記の各キーに対し、次の機能のいずれかを割り当てることができます。

- Natural のシステムコマンド
- Natural の端末コマンド
- ユーザー定義コマンド

コマンドモード（「**Direct Command**」ウィンドウ）で対応するキーを押すたびに、割り当てられたコマンドが実行されます。



注意:

1. システムコマンド KEY を使用した割り当ては、ほとんどの場合、プログラムの SET KEY ステートメントによる割り当てとは独立しています。
2. ファンクションキー割り当ては Natural 管理者がプロファイルパラメータ KEY を使用して行うこともできます。
3. このコマンドは、バッチモードでは実行できません。

コマンドの割り当て

パラメータなしで KEY コマンドのみを入力すると、「**Function-Key Assignments**」画面が表示されます。この画面では、入力フィールドにコマンド名を入力することによって、個別キーにコマンドを割り当てることができます。

キーに別のコマンドを割り当てるには、入力フィールドの既存エントリを上書きします。

コマンドの割り当てを削除するには、入力フィールドのエントリを削除するか、または空白で上書きします。

KEY コマンドでキーを直接指定することによって、個別キーにコマンドを割り当てることもできます。例えば、次のようになります。

```
KEY PF1=CLEAR
```

割り当てるコマンドに空白が含まれる場合は、コマンドをアポストロフィで囲む必要があります。例えば、次のようになります。

```
PF13='UPDATE OFF'
```

すべてのキーのアクティブ化／非アクティブ化 - KEY ON/OFF

コマンド KEY OFF/ON を使用すると、ファンクションキーのすべての割り当てを非アクティブにしたり、再びアクティブにしたりすることができます。

KEY OFF	ファンクションキーを押すと、そのキーがアクティブではないことを示すメッセージが返されます。
KEY ON	KEY OFF によって以前に非アクティブ化されたすべてのファンクションキーの割り当てを再びアクティブにします。

キーのアクティブ化／非アクティブ化は、**[Function-Key Assignments]** 画面の右上隅にある **[Activate Keys]** フィールドの ON/OFF のエントリを上書きすることによっても可能です。

個別キーのアクティブ化／非アクティブ化 - KEY key=ON/OFF

コマンド KEY key=OFF/ON を使用すると、特定の key に割り当てられたコマンドを非アクティブにしたり、再びアクティブにしたりすることができます。

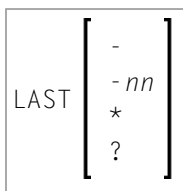
KEY key=OFF	特定の key に割り当てられたコマンドを非アクティブにします。例えば、次のようになります。
	KEY PF24=OFF

KEY key=ON	以前に非アクティブ化されたコマンド割り当てを再びアクティブにします。例えば、次のようになります。
	KEY PF24=ON



注意: 個々のキーを非アクティブ（PF24=OFF など）にした後で、すべてのキーを非アクティブ（KEY=OFF）にしてから、再びすべてのキーをアクティブ（KEY=ON）にすると、個別に非アクティブ化されたキーもアクティブ化されます。

18 LAST



LAST コマンドは、最後に実行されたコマンドを表示するために使用します。さらに、表示されたコマンドを再実行できます。また、これらのコマンドが実行される前に、コマンドを上書きすることもできます。

LAST コマンドでは、実際に入力したシステムコマンドのみを表示できます。入力したコマンドの結果として Natural が内部的に発行したコマンドは LAST コマンドでは処理できません。

LAST	最後に発行されたコマンドが、[Direct Command] ウィンドウに配置され、実行可能になります。
LAST -	最後に発行されたコマンドが、[Direct Command] ウィンドウに配置され、実行可能になります。 再び LAST - を入力すると、最後から 2 番目のコマンドが [Direct Command] ウィンドウに配置されます。 繰り返し LAST - を入力すると、コマンドごとに「ページ」を戻すことができます。 注意: 手動で繰り返し入力する代わりに、システムコマンド KEY によって LAST - を PF キーに割り当てすることもできます。
LAST -nn	Natural では、発行されたコマンドの最後の 20 個が「記憶」されています。したがって、nn には 20 以内を指定する必要があります。 最後から nn+1 番目のコマンドが、[Direct Command] ウィンドウに配置され、実行可能になります。
LAST *	ウィンドウが表示され、最後に発行された 20 個のコマンドが示されます。10 個以上のコマンドが表示される場合には、PF8 キーと PF7 キーを使用して上下にスクロールします。

	■ 単一のコマンドを再実行するには、カーソルでコマンドをマークして F5 キーを押すか、または任意の文字でコマンドをマークして ENTER キーを押します。 ■ 複数のコマンドを再実行するには、実行する順に番号でコマンドをマークして ENTER キーを押します。番号の昇順でコマンドが実行されます。
LAST ?	LAST コマンドのヘルプ機能呼び出します。

19 LASTMSG

LASTMSG

LASTMSG コマンドを使用すると、最後に発生したエラー状況に関する付加的な情報を表示できます。

Natural でエラーメッセージが表示された場合、このエラーは、実際のエラーではなく別のエラーによって生成されたエラーである可能性があります（連鎖的に別のエラーを引き起こすことがあります）。このようなケースでは、LASTMSG コマンドを使用すると、発行されたエラーを、最初にエラー状況を引き起こしたエラーまでさかのぼってトレースすることができます。

LASTMSG コマンドを入力すると、最後に発生したエラー状況に対して表示されたエラーメッセージ、およびこのエラーを引き起こしたすべての先行する（表示されなかった）エラーメッセージを取得します。

➤手順 19.1. 対応するエラーに関する情報を表示するには

- これらのメッセージの 1 つをカーソルでマークし、ENTER キーを押します。

次の情報が表示されます。

- エラー番号
- エラーが発生した行番号
- エラーが発生したオブジェクトの名前、タイプ、およびレベル
- オブジェクトを含むライブラリの名前、データベース ID、およびファイル番号
- エラークラス（system = Natural が発行したエラー、user = ユーザーアプリケーションが発行したエラー）
- エラータイプ（runtime（ランタイム）、syntax（構文）、command execution（コマンド実行）、session termination（セッション終了）、program termination（プログラム終了）、remote procedure call（リモートプロシージャコール））

- エラーが発生した日付および時刻



注意: ライブラリ SYSEXT には、アプリケーションプログラミングインターフェイス USR2006 があります。このユーザー API を使用すると、LASTMSG によって提供されたエラー情報を Natural アプリケーションに表示できます。

Natural RPC（リモートプロシージャコール）：

サーバーでのエラーの場合、データベース ID、ファイル番号、日付、および時刻は表示されません。

20

LIST

■ 構文の概要	60
■ 個々のソースの表示	61
■ オブジェクトのリストの表示	62
■ オブジェクトの数とサイズの表示	63
■ ディレクトリ情報の表示	64
■ ビューの表示	64
■ リソースオブジェクトのファイル情報の表示	65
■ エラーメッセージコンテナのファイル情報の表示	65

このシステムコマンドは、1つのオブジェクトのソースコードを表示したり、現在のライブラリに含まれている1つまたは複数のオブジェクトをリストしたりするために使用します。



注意: このコマンドは、バッチモードでは実行できません。

この章では、次のトピックについて説明します。

コマンド [LIST](#) [XREF](#) の説明も参照してください。

構文の概要

```
LIST [ [object-type] object-name
      COUNT [ *
              object-name-range ]
      DIRECTORY [object-name]
      VIEW [view-name]
      RESOURCE [name]
      ERROR [name] ]
```

object-type

```
*
{
  { CLASS }
  4
COPYCODE
DATA-AREAS
  GLOBAL
  LOCAL
  PARAMETER
  { DIALOG }
  3
  7 (機能用)
  8 (アダプタ用)
MAP
  { PROCESSOR }
  CP
  5
PROGRAM
ROUTINES
  HELPROUTINE
  { SUBPROGRAM }
  N
  SUBROUTINE
TEXT
```

object-name

*object-name*の代わりに、オブジェクトの名前を指定することもできます（長さは最大8文字）。また、アスタリスク（*）表記を指定することもできます。下記の「例」を参照してください。

個々のソースの表示

LIST	パラメータなしで LIST コマンドのみを入力すると、ソースワークエリアの内容がリストされます。
LIST <i>object-name</i>	LIST コマンドで1つのオブジェクト名を入力する場合は、 <i>object-type</i> を指定する必要はありません。
LIST <i>object-type</i> <i>object-name</i>	<i>object-type</i> を指定する場合は、 <i>object-name</i> も指定する必要があります。

	どちらの場合も、オブジェクトのソースコードがリストされます。
--	--------------------------------

オブジェクトのリストの表示

LIST <i>object-name</i>	現在のライブラリにあるすべてのオブジェクトをリストするには（DDMを除く）、 <i>object-type</i> ではなく、 <i>object-name</i> にアスタリスク（*）を指定します。
LIST <i>object-type</i> <i>object-name</i>	
	特定のタイプのオブジェクトをすべてリストするには、 <i>object-name</i> に特定の <i>object-type</i> とアスタリスク（*）を指定します。
	特定の範囲にあるオブジェクトをリストするには、 <i>object-name</i> にアスタリスク表記（*）とワイルドカード表記（?）のいずれかまたは両方を使用します。

例

- 現在のライブラリにあるすべてのオブジェクトを、オブジェクトのタイプに関係なくリストします（ビュー、リソース、エラーを除く）。

```
LIST *
```

- 現在のライブラリにあるすべてのサブルーチンをリストします。

```
LIST S *
```

- 名前が SYS で始まるすべてのオブジェクト（任意のタイプ）をリスト表示します。

```
LIST SYS*
```

- 名前が SYS で始まるすべてのマップをリスト表示します。

```
LIST M SYS*
```

- 現在のライブラリにあるオブジェクト PRG01 のディレクトリ情報をリストします。

```
LIST DIR PRG01
```

- NATAL、NATURAL、NAT_{vr}AL などのオブジェクトをすべてリスト表示します（_{vr} は関連する製品バージョンを表しています）。

```
LIST N?T*AL
```

オブジェクトへの機能の実行

選択リストからのオブジェクトに対して機能を実行するには、左側の列で適切なファンクションコードを指定してオブジェクトをマークします。

ファンクションコードは次のとおりです。

Code	関数
C	オブジェクトのソースコードをチェックします。
D	オブジェクトのソースコードを読み取ります。
E	オブジェクトのソースを編集します（システムコマンド EDIT と等価）。
L	オブジェクトのソースコードをリストします。
I	オブジェクトのソースコードのディレクトリ情報をリストします（ LIST DIRECTORY object-name と等価）。
H	オブジェクトのソースのハードコピーを印刷します。
R	オブジェクトのソースを実行（つまり、コンパイルして実行）します（システムコマンド RUN と等価）。
X	オブジェクトを実行します（システムコマンド EXECUTE と等価）。
S	オブジェクトをソース形式およびオブジェクト形式で格納します（システムコマンド STOW と等価）。
U	オブジェクトのソースフォームおよびオブジェクトフォームを削除します。
.	終了。

疑問符 (?) を入力するか、または F2 キーを使用して、選択したオブジェクトに使用可能なファンクションコードのリストを表示します。

オブジェクトの数とサイズの表示

LIST COUNT	現在のライブラリに保存されているすべてのオブジェクトの数とサイズ（バイトまたは 1MB を超える場合は KB 単位）を含むサマリレポートを表示します。 リスト表示されている数とサイズは、ソース (Saved) オブジェクトとして保存されたすべてのオブジェクトまたはカタログ化 (Cataloged) オブジェクトとして保存されたすべてのオブジェクトのいずれか、および保存されたオブジェクトとカタログ化されたオブジェクト (Stowed) の両方が存在するすべてのオブジェクトを対象とします。
LIST COUNT *	保存／カタログ化されたオブジェクトの数とサイズが、検出されたオブジェクトタイプごとにリスト表示されている、すべてのオブジェクトのサマリレポートを表示します。
LIST COUNT <i>object-name-range</i>	指定したオブジェクト範囲のサマリレポート（上記と同じ）を表示します。 有効な範囲表記：

	<i>value</i> >	<i>value</i> 以降の名前を持つすべてのオブジェクト。
	<i>value</i> <	<i>value</i> 以前の名前を持つすべてのオブジェクト。
	<i>value</i> *	<i>value</i> で始まる名前を持つすべてのオブジェクト。

ディレクトリ情報の表示

LIST DIRECTORY	現在ソースワークエリアで最後にアクティブなオブジェクトのディレクトリ情報を表示します。 ■ ソースコード： 「保存された」日付と時刻、ライブラリ名、ユーザーID、プログラミングモード（レポートिंगまたはストラクチャード）、Naturalバージョン、コードページ情報（使用可能な場合）、オペレーティングシステム、サイズ、エンコード ■ オブジェクトコード： 「カタログされた」日付と時刻、ライブラリ名、ユーザーID、プログラミングモード、Naturalバージョン、コードページ情報（使用可能な場合）、オペレーティングシステム、サイズ、エンディアンモード Natural以外（Naturalの制御下でない）のエディタでソースコードを編集できるため、保存されたソースコードのディレクトリ情報は常に正確であるとは限りません。
LIST DIRECTORY <i>object-name</i>	指定されたオブジェクトに関するディレクトリ情報を表示します。<i>object-name</i> にアスタリスク表記（*）を使用した場合は、既存のオブジェクトのディレクトリ情報が連続して表示されます。



注意: 表示されるコードページ情報には、コードページの先頭の 32 文字のみが示されます。

ビューの表示

LIST VIEW	すべてのビューのリストを表示します（DDM）。
LIST VIEW <i>view-name</i>	1つのビュー名を指定すると、指定したビューが表示されます。 <i>view-name</i> でアスタリスク表記を使用すると、すべてのビュー（*）または特定の範囲にあるビューのリストを表示できます（例：A*）。

リソースオブジェクトのファイル情報の表示

LIST RESOURCE <i>name</i>	指定されたリソースオブジェクトに関するファイル情報を表示します。 <i>name</i> には、アスタリスク（*）表記のみを使用できます。
---------------------------	---

例 - 名前が W で始まるすべてのリソースオブジェクトのファイル情報を表示します。

```
LIST RESOURCE W*
```

エラーメッセージコンテナのファイル情報の表示

LIST ERROR <i>name</i>	指定されたエラーメッセージコンテナ <i>Nnn</i> APMSL.MSG に関するファイル情報を表示します。 <i>nn</i> には言語コードが入ります。 <i>name</i> には、アスタリスク（*）表記のみを使用できます。
------------------------	---

21

LIST XREF

LIST XREF

このコマンドは、Predict をインストールしている場合にのみ使用できます。

現在ログオンしているライブラリのアクティブクロスリファレンスをすべて表示します。

詳細については、『Predict』ドキュメントの「*Natural* の XREF」を参照してください。

22 LOGOFF

LOGOFF

関連するコマンド：[LOGON](#)。

このコマンドを使用すると、ライブラリ ID を SYSTEM に設定し、Adabas パスワードを空白に設定できます。ソースプログラムワークエリアの内容は、このコマンドの影響を受けません。

LOGOFF によって、Natural グローバルパラメータの設定が影響を受けることはありません。

Natural Security 環境での LOGOFF の処理については、『*Natural Security*』ドキュメントの「*Logging On*」セクションの「*How to End a Natural Session*」を参照してください。



注意: LOGOFF では、Natural セッションが終了することはありません。

➤手順 22.1. セッションを終了するには

- システムコマンド [FIN](#) を使用するか、または TERMINATE ステートメントを含むプログラムを実行します。

23 LOGON

`LOGON library-id[password]`

関連するコマンド：LOGOFF。

LOGON コマンドは、ライブラリへのログオン、または新規ライブラリの作成に使用します。ライブラリ ID を指定すると、セッション中に保存するすべてのソースプログラムやオブジェクトプログラムは、指定ライブラリに保存されます（ただし、[SAVE](#)、[CATALOG](#)、または [STOW](#) コマンドで別のライブラリ ID を指定した場合を除く）。

LOGON コマンドによって、現在アクティブなウィンドウ内にあるソースプログラムが直接影響を受けることはありません。

LOGON では、すべての Natural グローバルデータエリア、アプリケーション独立変数（AIV）、SET KEY ステートメントで行われたすべての割り当て、保持されている ISN リストが解放されます。DDM バッファエリア内の DDM も解放されます。

LOGON <i>library-id</i>	ライブラリ ID は 1～8 文字で指定し、間に空白を入れないようにします。ライブラリ ID には、次の文字を使用できます。	
	A～Z	英大文字
	0～9	数字
	-	ハイフン
	_	下線
	ライブラリ ID の先頭文字は、英大文字にする必要があります。	
LOGON <i>library-id</i> <i>password</i>	Adabas パスワードを指定します。『Natural Security』ドキュメントの「Library Maintenance」セクションで「Session Parameters」を参照してください。	

Natural Security 環境下での LOGON の処理については、『Natural Security』ドキュメントの「Logging On」を参照してください。

24 MAIL

MAIL $\left[\left\{ \begin{array}{c} * \\ ? \\ mailbox-id \end{array} \right\} \right]$
--

MAIL コマンドを使用すると、Natural Security 環境下でメッセージをブロードキャストするための一種の「掲示板」として使用されるメールボックスを起動できます。メールボックスの内容と有効期限のいずれかまたは両方を変更できます。

MAIL	MAIL コマンドをパラメータなしで入力すると、メールボックス ID の入力を要求するウィンドウが表示されます。
MAIL *	使用可能なすべてのメールボックスを含むリストが表示されるため、メールボックスをこのリストから選択します。
MAIL ?	
MAIL <i>mailbox-id</i>	<i>mailbox-id</i> (最大8文字) を指定すると、対応するメールボックスが直接起動します。 <i>mailbox-id</i> は、Natural Security で事前に定義しておく必要があります。

詳細については、『Natural Security』ドキュメントの「Mailboxes」を参照してください。

25 PROFILE

PROFILE コマンドは、Natural Security がインストールされている場合にのみ使用可能です。

PROFILE

PROFILE コマンドを使用すると、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できます。このプロファイルは、現在の Natural 環境の実際の使用条件を知らせます。

詳細については、『*Natural Security*』ドキュメントの「*PROFILE Command*」を参照してください。

26 PROFILER

PROFILER

このコマンドはプロファイラユーティリティを呼び出します。プロファイラユーティリティは、Naturalアプリケーションの内部プロセスフローを監視し、アプリケーションのパフォーマンスを分析します。

プロファイラユーティリティはバッチモードでのみ実行できます。

詳細については、『ツールおよびユーティリティ』ドキュメントの「プロファイラユーティリティ」を参照してください。

27 PURGE

PURGE [*object-name* ...]

PURGE コマンドを使用すると、1 つまたは複数のソースオブジェクトを削除できます。

-  **注意:** プロファイルパラメータ RECAT が ON に設定されている場合、対応するカタログ化オブジェクトが存在すると、PURGE コマンドではそのソースが拒否されます。
-  **注意:** FUSER システムファイルのライブラリに DDM が存在している場合は、PURGE コマンドでそれらの DDM を削除できます。

PURGE	<i>object-name</i> を指定せずに PURGE コマンドを入力すると、現在のライブラリにあるすべてのオブジェクトのリストが表示されます。このリストでは、削除するオブジェクトをマークできます。
PURGE <i>object-name</i>	<i>object-name</i> として、削除するオブジェクトの名前を指定します。現在ログオンしているライブラリに保存されているオブジェクトだけを削除できます。 特定の文字列で始まるすべてのオブジェクトを削除する場合は、 <i>object-name</i> にアスタリスク (*) を使用します。

28 READ

`READ object-name [library-id]`

関連するコマンド：[EDIT](#)。

READ コマンドを使用すると、ソース形式で保存されているオブジェクトをソースワークエリアに転送できます。現在のソースワークエリア内にあるオブジェクトは、新たに読み込まれたオブジェクトによって上書きされます。

『*Natural*の使用』ドキュメントの「オブジェクトの命名規則」も参照してください。

READ <i>object-name</i>	読み込むオブジェクトの名前です。 ライブラリ ID を指定せずに <i>object-name</i> のみを指定すると、そのオブジェクトが現在ユーザーがログオンしているライブラリに保存されている場合にのみ読み込まれます。
READ <i>object-name</i> <i>library-id</i>	読み込むオブジェクトが存在するライブラリです。 <i>object-name</i> と <i>library-id</i> の両方を指定した場合、 <i>Natural</i> では、指定されたライブラリ ID にそのオブジェクトが保存されているときにのみ読み込みが行われます。

29 RENAME

```
RENAME [old-name [new-name [new-type]]
```

このコマンドを使用すると、Naturalオブジェクトに別の名前を指定できます。また、オブジェクトタイプを変更することもできます。

名前を変更できるオブジェクトは、一度に1つのみです。名前を変更するオブジェクトは、現在ログオンしているライブラリに保存されている必要があります。一貫性を確保するために、Naturalでは、ソースコードとオブジェクトコードのいずれかまたは両方の名前が変更されます。

『Naturalの使用』ドキュメントの「オブジェクトの命名規則」も参照してください。

RENAME	パラメータを指定せずに RENAME コマンドを発行すると、 [Rename Object] ウィンドウが表示され、コマンド行と同じパラメータを指定できます。	
<i>old-name</i>	<i>old-name</i> には、名前を変更するオブジェクトの現在の名前を指定します。	
<i>new-name</i>	今後オブジェクトは <i>new-name</i> に指定した名前で保存されます。	
<i>new-type</i>	ソース形式のオブジェクトの名前を変更する場合は、対応する文字を <i>new-type</i> に指定することによって、そのオブジェクトのオブジェクトタイプも変更できます。	
	<i>new-type</i> に指定可能な値は、次のとおりです。	
	3	ダイアログ
	4	クラス
	5	プロセッサ
	7	関数
	8	アダプタ
	9	リソース
	A	パラメータデータエリア
	C	コピーコード
	G	グローバルデータエリア

	H	Helproutine
	L	ローカルデータエリア
	M	マップ
	N	Subprogram
	O	マクロ
	P	プログラム
	S	サブルーチン
	T	テキスト
	○	ルール
	Z	レコーディング

30 RENUMBER

RENUMBER [(n)]

このコマンドを使用すると、現在Naturalエディタのワークエリア内にあるソースコードの番号を再設定できます。



注意: 英数字定数またはUnicode定数の数値を変更する場合は、プロファイルパラメータ RNCONST が ON に設定されていることを確認してください。

RENUMBER	パラメータを使用せずにコマンドを入力した場合、ワークエリア内のソース行の総数に応じて、次のデフォルトの増分値が番号の再設定に使用されます。	
	0001～0999 行：	10 単位で増加
	1000～1999 行：	5 単位で増加
	2000～4999 行：	2 単位で増加
	5000 行以上：	1 単位で増加
RENUMBER (n)	n には、番号再設定の増分として 1～9999 の値を指定できます。指定した増分値によって番号再設定で 9999 行の上限を超える場合は、代わりにデフォルトの増分値が使用されます。	

『プログラミングガイド』の「参照するソースコード行番号の変更」を参照してください。

31 RETURN

RETURN $\left[\left\{ \begin{array}{c} I \\ nn \\ * \end{array} \right\} \right]$

RETURN コマンドを使用すると、前の（または最初の）Natural アプリケーションに戻ることができます。

RETURN	パラメータを指定せずに RETURN を発行した場合は、制御は前のアプリケーションに返されます（SETUP システムコマンドの定義）。前のアプリケーションに関する情報はすべて削除されます。前のアプリケーションがない場合は、制御は初期のアプリケーションに戻されます。 RETURN を発行しても戻り位置が設定されていない場合、RETURN コマンドは無視されます。 Natural Security の場合： RETURN を発行して戻り位置が設定されていなかった場合には、LOGOFF コマンドが実行されます。
RETURN I	制御を初期のアプリケーションに直接戻します。このオプションを使用すると、以前のアプリケーションに関するすべての定義が削除されます（初期のアプリケーションを除く）。
RETURN nn	このコマンドは、制御を nn つ前のアプリケーションに戻します。このオプションを使用すると、nn つ前以降のアプリケーションの情報はすべて削除されます。
RETURN *	現在設定されているすべての戻り位置のリストを表示します。このリストでは、戻り位置を選択できます。

詳細な説明および例については、[SETUP](#) コマンドを参照してください。

32 RPCERR

RPCERR

RPCERR コマンドを使用すると、RPC に関連している場合には、最後の Natural エラー番号とメッセージを表示できます。また、最後の Broker 理由コードと関連メッセージも表示できます。さらに、最後の Broker コールからのノードおよびサーバー名も取得可能です。

詳細については、『*Natural RPC (リモートプロシージャコール)*』ドキュメントの「*Natural RPC 環境の運用*」セクションで「*RPC セッションのステータスのモニタ*」を参照してください。

33 RUN

`RUN [REPEAT] [program-name [library-id]]`

RUN コマンドを使用すると、ソースプログラムをコンパイルして実行できます。ソースワークエリア内または Natural システムファイル内にあるプログラムを実行できます。

RUN	<i>program name</i> を指定しない場合、Natural では、現在ワークエリア内にあるプログラムがコンパイルおよび実行されます。
REPEAT	REPEAT を指定すると、プログラムが複数の出力画面を生成する場合に、間にプロンプトメッセージを出さずに連続で画面を出力します。プログラムが終了すると、Natural はコマンドモードに入ります。
<i>program-name</i>	実行するプログラムの名前です。 <i>program-name</i> を指定してライブラリ ID を指定しない場合は、指定のプログラムが現在ログオンしているライブラリ ID のもとに保存されているときにのみ、そのソースプログラムがソースワークエリアに読み込まれ、コンパイルして実行されます。現在のライブラリ ID で保存されていない場合は、エラーメッセージが表示されます。
<i>library-id</i>	実行するプログラムが存在するライブラリです。 <i>program-name</i> と <i>library-id</i> の両方を指定する場合は、指定のライブラリ ID のもとにこのプログラムが保存されているときにのみ、検索が行われ、コンパイルして実行されます。現在のライブラリ ID で保存されていない場合は、エラーメッセージが表示されます。 <i>library-id</i> の設定値を SYS (SYSTEM を除く) で始めることはできません。

34

SAVE

`SAVE [object-name [library-id]]`

関連するコマンド：[STOW](#) | [CATALOG](#).

このコマンドは、現在 Natural エディタのワークエリアに含まれるソースコードをソースオブジェクトとして現在の Natural システムファイルに保存するために使用します。



注意: SAVE コマンドは、プロファイルパラメータ `RECAT` が ON に設定されている場合には使用できません。この場合は、[STOW](#) コマンドを使用してオブジェクトをコンパイルおよび保存します。

SAVE	<i>object-name</i> を指定しないでコマンドを使用する場合は、ソースワークエリア内の現在のソースオブジェクトが現在のライブラリに保存されます。既存のソースコードは置換されます。
SAVE <i>object-name</i>	新しいソースオブジェクトが作成されます。 <i>object-name</i> として、保存するソースオブジェクトの名前を指定します。新しいソースオブジェクトは現在のライブラリに保存されます。ソースオブジェクトが存在する場合、コマンドは拒否されます。
SAVE <i>object-name</i> <i>library-id</i>	ソースオブジェクトを別の名前で作成するか、または新しく作成したオブジェクトを保存する場合、ソースオブジェクトはデフォルトで現在のライブラリに保存されます。別のライブラリに保存する場合は、 <i>object-name</i> の後に必要な <i>library-id</i> を指定する必要があります。新しいソースオブジェクトが作成されます。ソースオブジェクトが存在する場合はコマンドが拒否されます。

35

SCAN

■ メニューオプション	96
■ サブコマンド	97
■ Natural Security 環境での SCAN	98

SCAN

このコマンドは、オブジェクト内の文字列を検索するために使用します。このとき、オプションとして文字列を別の文字列で置換することもできます。

対象になるのは、1つのオブジェクト、指定した設定で始まるすべてのオブジェクト、またはライブラリ内のすべてのオブジェクトです。スキャンも、対象を特定のオブジェクトタイプに制限できます。

 **重要:** ソースワークエリアは SCAN コマンドによって使用されます。そのため、SCAN コマンドを使用する前に、**SAVE** または **STOW** コマンドを発行する必要があります。

 **注意:** このコマンドは、バッチモードでは実行できません。

この章では、次のトピックについて説明します。

メニューオプション

マップ、DDM、およびデータエリアを除くすべてのオブジェクトは、スキャンユーティリティを使用して変更できます。フルスクリーンエディタを使用して、スキャンされた行以外の行を変更することもできます。オブジェクトが変更されたら、それを保存してエディタを終了します。その後、スキャン処理を継続できます。

SCAN コマンドを発行すると、次の項目を指定できるウィンドウが表示されます。

フィールド	説明	
スキャン対象値	検索する値。	
置換値	スキャン対象値を置き換える値。	
削除文字列	スキャン対象値が削除されます (Y/N)。	
Object name	スキャン対象のオブジェクト：	
	空白	すべてのオブジェクト。
	*	
	name*	名前が name で始まるすべてのオブジェクト。
オブジェクトタイプ	スキャンするオブジェクトのタイプ。	
	P	プログラム
	C	コピーコード
	N	サブプログラム
	S	サブルーチン
	H	ヘルプルーチン
	M	マップ

フィールド	説明	
	G	グローバルデータエリア
	L	ローカルデータエリア
	A	パラメータデータエリア
	T	テキスト
	4	クラス
	*	すべてのタイプ
完全スキャン	○	完全スキャンを行います。
	N	完全スキャンが"N"に設定されている場合は、スキャン対象値を空白または特定の文字で区切る必要があります。
ライブラリ	スキャン対象のライブラリの ID。デフォルトは現在のライブラリです。	
Case sensitive	N	大文字、小文字またはその混在のいずれであるかに関係なく、スキャン対象値を検索します。
	○	検索は、指定されたとおりのスキャン対象値に対して行われます。

適切な指定を行い、ENTER キーを押します。検索されたスキャン対象値を含む行を表示するウィンドウが開きます。

```

+----- Search value MOVE in Program READVAR-----+
| Commands:                                           |
| 0440  MOVE EMPLOYEES-VIEW.NAME          TO #NAME (J) |
| 0450  MOVE EMPLOYEES-VIEW.FIRST-NAME TO #FIRST (J)  |
| 0960      MOVE #NAME(6) TO #START-NAME              |
| 0980      MOVE 'D' TO #DIR                        /* new direction DESCENDING |
| 1020      MOVE #NAME(1) TO #START-NAME              |
| 1050      MOVE 'A' TO #DIR                        /* new direction ASCENDING |
| 1090      MOVE #NAME(6) TO #START-NAME              |
| 1180      MOVE #NAME(6) TO #START-NAME              |
+-----+

```

サブコマンド

必要な任意のサブコマンドを上記のウィンドウに入力できます。コマンドモードに入るには、**[Commands]** 行にカーソルを置き、ENTER キーを押します。

続いて、次のコマンドのいずれかを選択できます。

Command	関数
編集	オブジェクトを編集します。
List	現在ソースワークエリアに表示されているオブジェクトをリストします。
Ignore	現在スキャン中のオブジェクトを無視し、いずれの変更も保存せず、次のオブジェクトに移ってスキャンを続行します。
Quit	スキャン処理を終了します。

サブコマンドは、最初の文字を入力することでも呼び出すことができます。

コマンドモードを非アクティブにするには、ESC キーを押します。

編集ルール

- [Replace] オプションが使用されている場合、オブジェクトがスキャンユーティリティで更新された場合、またはその両方の場合は、[Ignore] または [Quit] が指定されていない限り、次のオブジェクトがスキャンされる前にオブジェクトが常に保存されます。
- PASSWORD=、PASSW=、CIPHER=、または CIPH= を含む行は、SCAN コマンドによって無視されます。
- スキャンユーティリティのソースオブジェクトの行の長さは、72 に制限されます。
- 置換値が原因で行が 80 文字を超えた場合は、行が自動的に分割されます。

Natural Security 環境での SCAN

Natural Security 環境で SCAN を使用するには、システムコマンド **LIST**、**EDIT** および **READ** を現在のライブラリのセキュリティプロファイルで許可する必要があります。[Replace] オプションを使用する場合は、システムコマンド **SAVE** も許可する必要があります。

Natural Security 環境では、SCAN コマンドが一部のライブラリで使用できない場合があります。

36 SCRATCH

```
SCRATCH [ { *  
            object-name... } ]
```

このコマンドは、1つ以上のオブジェクトを（ソースとオブジェクトの両方の形式で）削除するときに使用します。ソースワークエリアの内容には影響しません。

SCRATCH	<i>object-name</i> または <i>object-name</i> を指定せずに、アスタリスク (*) を指定して SCRATCH コマンドを入力すると、現在のライブラリのすべてのオブジェクトまたは選択したすべてのオブジェクトのリストが表示されます。そのリストで、削除 (SCRATCH) する各オブジェクトをマークできます。
SCRATCH *	
SCRATCH <i>object-name</i>	<i>object-name</i> として、削除するオブジェクトの名前を指定します。現在ログオンしているライブラリ内に存在しているオブジェクトのみを削除できます。 特定の文字列で始まるすべてのオブジェクトを削除する場合は、 <i>object-name</i> にアスタリスク (*) を使用します。



注意: 有効ではないパラメータファイルに FDIC システムファイルが指定されていると、SCRATCH コマンドが発行されるときに Natural は該当するエラーメッセージを表示します。

37 SETUP

■ 構文説明	102
■ 例：SETUP/RETURN	103

SETUP [*application-name*] [*command-name*] [I]

このコマンドは、RETURN コマンドで制御を返す先のアプリケーションを定義するために使用します。これにより、Natural セッション中にアプリケーション間を容易に移動できます。

この章では、次のトピックについて説明します。

構文説明

SETUP システムコマンドで発行できるコマンド構文とパラメータについて説明します。パラメータを省略する場合は、INPUT 区切り文字を使用して、次のパラメータの先頭をマークできます。

SETUP	パラメータを何も指定せずに SETUP を発行すると、コマンド情報の入力のためのメニューが表示されます。
<i>application-name</i>	制御が返されるアプリケーションの名前です。最大8文字（A8）の英数字を使用できます。 <i>application-name</i> が空白の場合、LOGON コマンドは発行されません。これにより、同じアプリケーション内で複数の戻り位置が持てるようになります。 <i>application-name</i> が "*" の場合は、システム変数 *LIBRARY-ID の現在の設定（SETUP が発行された時点の設定）が、RETURN 発行時に LOGON コマンドを作成するために使用されます。
<i>command-name</i>	アプリケーションに制御が返されたときに実行するコマンドの名前です。最大60文字（A60）の英数字を使用できます。 <i>command-name</i> が空白の場合、LOGON の後にコマンドは発行されません。これは、すでにスタートアッププログラムが定義されているような Natural Security のアプリケーションに便利です。 <i>command-name</i> が "*" の場合は、システム変数 *STARTUP の現在の設定（SETUP が発行された時点の設定）が、RETURN 発行時のスタートアップコマンドとして使用されます。
I	I オプションを指定すると、以前の SETUP コマンドで定義されていた戻り位置はすべて削除され、SETUP I で指定されたアプリケーションが新しい開始アプリケーションとして定義されます。 非セキュリティ環境では、ライブラリ SYSTEM から別のライブラリにログオンし、戻り位置が設定されていないと、ログオンしたライブラリが開始戻り位置となります。

例：SETUP/RETURN

1. Natural セッションを開始します（デフォルトのアプリケーション = APPL1）。

戻り位置 APPL1 がレベル 1 に定義されます。

2. コマンド LOGON APPL2 を発行します。
3. 2つのコマンドをスタックするプログラムを実行します（戻り位置を設定し、別のアプリケーションに移動します）。

```
SETUP *,MENU
LOGON APPL3
```

戻り位置 APPL2, STARTUP MENU がレベル 2 に定義されます。

4. LOGON APPL4 コマンドを発行します（他のアプリケーションを選択します）。
5. RETURN という設定を持つ PF キーを押します。Natural がユーザーに対して次のコマンドを発行します。

```
LOGON APPL2
MENU
```

APPL2 に戻り、レベル 2 を削除します。

6. 次のコマンドが含まれたプログラムを実行します。

```
SETUP *,MENU
LOGON APPL5
```

戻り位置 APPL2, STARTUP MENU がレベル 2 に定義されます。

7. 次のコマンドが含まれたプログラムを実行します。

```
SETUP *,MENU
LOGON APPL6
```

戻り位置 APPL5, STARTUP MENU がレベル 3 に定義されます。

8. 次のコマンドが含まれたプログラムを実行します。

```
SETUP *,MENU
LOGON APPL7
```

戻り位置 APPL6, STARTUP MENU がレベル 4 に定義されます。

9. 次のコマンドが含まれたプログラムを実行します。

```
SETUP *,MENU
LOGON APPL8
```

戻り位置 APPL7, STARTUP MENU がレベル 5 に定義されます。

10. 次のコマンドが含まれたプログラムを実行します。

```
SETUP *,MENU
LOGON APPL9
```

戻り位置 APPL8, STARTUP MENU がレベル 6 に定義されます。

11. RETURN 2 コマンドを発行します (2 レベル戻ります)。

Natural は 2 つ前のセッションである APPL7 にユーザーを戻します (ここで APPL8 の情報はすべて消去されます)。レベル 6 (APPL8) は削除され、レベル 5 (APPL7) は制御を受けてレベルが削除されます。

- 12 コマンド RETURN を発行します。

レベル 4 (APPL6) がアクティブ化され、レベルが削除されます。Natural は APPL7 の 1 つ前セッションである APPL6 にユーザーを戻します。

13. コマンド RETURN を発行します。

レベル 3 (APPL5) がアクティブ化され、レベルが削除されます。Natural は APPL6 の 1 つ前セッションである APPL5 にユーザーを戻します。

14. コマンド RETURN I を発行します。

レベル 2 (APPL2) が削除され、レベル 1 (APPL1) が制御を受けます。

38 STOW

STOW [*object-name* [*library-id*]]

関連するコマンド：SAVE | CATALOG.

このコマンドは、現在の Natural システムファイルに Natural オブジェクトをカタログ化（コンパイル）して（ソース形式とオブジェクト形式で）保存するために使用します。このコマンドは、CATALOG とそれに続く SAVE とみなすことができます。

STOW	<i>object-name</i> を指定しないでコマンドを使用する場合は、ソースエリアに保持されるソースコードと生成されたコードが、同じ名前で現在のライブラリに保存されます。既存のソースおよびオブジェクトコードは置換されます。
STOW <i>object-name</i>	このコマンドは、 <i>object-name</i> という名前の新しいオブジェクト（ソースと生成されたコード）を現在のライブラリに保存するために使用します。オブジェクトがソースまたはカタログのいずれかの形式で存在する場合、コマンドは拒否されます。
STOW <i>object-name</i> <i>library-id</i>	<i>object-name</i> と <i>library-id</i> の両方を指定すると、新しいオブジェクトが作成され、指定のライブラリ ID にその名前で保存されます。オブジェクトがソースまたはカタログのいずれかの形式で存在する場合、コマンドは拒否されます。



注意: 有効ではないパラメータファイルに FDIC システムファイルが指定された場合は、STOW コマンドが発行されたときに、Natural によって適切なエラーメッセージが表示されます。

39 STRUCT

■ ソースコード行のインデント	108
-----------------------	-----

STRUCT[(n)]

このコマンドを使用して、エディタのワークエリアに現在ある Natural オブジェクトのソースコードのインデントを実行します。

STRUCT	デフォルトでは（つまり、(n) が指定されていない場合）、インデントは 2 単位の位置ずつ行われます。
STRUCT (n)	パラメータ (n) を提供して、インデントに使用するスペースの数を指定できます。 設定可能値：1～9。 例： STRUCT (5)

次のステートメントタイプが STRUCT コマンドの影響を受けます。

- 処理ループ（READ、FIND、FOR など）
- 条件ステートメントブロック（AT BREAK、IF、DECIDE FOR など）
- DO/DOEND ステートメントブロック、
- DEFINE DATA ブロック、
- インラインサブルーチン

この章では、次のトピックについて説明します。

ソースコード行のインデント

ソースコードの行に、プログラムの階層構造を反映したインデントを行うことができます。

 **注意:** レポーティングモードのプログラムに行われるインデントはストラクチャードモードのプログラムとは異なります。

部分的なインデント

特殊ステートメント /*STRUCT OFF および /*STRUCT ON を使用して構造的なインデントからプログラムソースを部分的に除外できます。このステートメントはソースコード行の先頭から入力する必要があります。この 2 つのステートメント間のソースコード行は STRUCT コマンドを発行しても元のままです。

例：

インデントする前のプログラムは次のとおりです。

```
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
2 PERSONNEL-ID
2 FULL-NAME
3 FIRST-NAME
3 NAME
1 VEHI VIEW OF VEHICLES
2 PERSONNEL-ID
2 MAKE
END-DEFINE
FIND EMPL WITH NAME = 'ADKINSON'
IF NO RECORDS FOUND
WRITE 'NO RECORD FOUND'
END-NOREC
FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
END-FIND
END-FIND
END
```

同じプログラムをインデントすると、次のようになります。

```
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 FULL-NAME
    3 FIRST-NAME
    3 NAME
1 VEHI VIEW OF VEHICLES
  2 PERSONNEL-ID
  2 MAKE
END-DEFINE
FIND EMPL WITH NAME = 'ADKINSON'
  IF NO RECORDS FOUND
    WRITE 'NO RECORD FOUND'
  END-NOREC
  FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
    DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
  END-FIND
END-FIND
END
```


40

SYSDDM

SYSDDM

このコマンドは、Natural データ定義モジュール（DDM）の作成およびメンテナンスに必要な機能を提供する DDM サービスを呼び出します。

詳細については、『エディタ』ドキュメントの「DDM サービス」を参照してください。



注意: このコマンドは、バッチモードでは実行できません。

Natural Single Point of Development の注意事項：

このコマンドは、リモート開発環境の Natural スタジオコマンド行経由では使用できません。DDM はノード DDM の下のツリービューにリストされ、SYSDDM ユーティリティの全機能はコンテキストメニューまたはメニューバー経由で有効になるためです。

41

SYSERR

SYSERR

このコマンドは SYSERR ユーティリティを呼び出します。

SYSERR ユーティリティを使用して、独自のアプリケーション固有メッセージを作成できます。

- SYSERR ユーティリティを使用してエラーや情報メッセージを Natural コードから分離し、個別に管理できます。
- 各種メッセージに対するメッセージの統一およびメッセージの定義と同様に、メッセージを別の言語に翻訳したり、メッセージに拡張メッセージを連結したりすることができます。
- SYSERR ユーティリティを使用して、既存の Natural システムメッセージのテキストを修正することもできますが、新規 Natural リリースではこの修正は失われるので推奨できません。

詳細については、『ツールおよびユーティリティ』ドキュメントの「SYSERR ユーティリティ」を参照してください。

42 SYSEXT

SYSEXT

このコマンドは SYSEXT ユーティリティを呼び出します。

このユーティリティは、ライブラリ SYSEXT に含まれるさまざまな Natural アプリケーションプログラミングインターフェイスを表示するために使用します。

詳細については、『ツールおよびユーティリティ』ドキュメントの「SYSEXT - *Natural* アプリケーションプログラミングインターフェイス」を参照してください。

43 SYSEXV

SYSEXV

このコマンドは SYSEXV ユーティリティを呼び出します。

SYSEXV ユーティリティを使用すると、現行および以前の一部バージョンの Natural で使用可能な新機能の例を参照できます。

詳細については、『ツールおよびユーティリティ』ドキュメントの「SYSEXV ユーティリティ」を参照してください。

44

SYSFILE

SYSFILE

このコマンドは、ワークファイルと出力ファイルの情報を表示します。次の情報を取得できます。

- レポート
- 論理デバイス
- 物理デバイスの定義
- プリンタプロファイルの定義
- ワークファイルの定義

ワークファイルと出力ファイルの詳細については、次を参照してください。

- 『コンフィグレーションユーティリティ』ドキュメントの「プリンタプロファイル」
- 『コンフィグレーションユーティリティ』ドキュメントの「デバイス／レポート割り当て」
- 『オペレーション』ドキュメントの「ワークファイル」

45

SYSMAIN

SYSMAIN

このコマンドは SYSMAIN ユーティリティを呼び出します。このユーティリティは、コピー、移動、削除など、Natural オブジェクトに対する操作を実行するために使用します。SYSMAIN ユーティリティは、ある環境の Natural システムにあるオブジェクトを、インポート機能を使用して別の環境に転送するために使用することもできます。

詳細については、『ツールおよびユーティリティ』ドキュメントの「SYSMAIN ユーティリティ」を参照してください。



注意: このコマンドは、バッチモードでは実行できません。

46

SYSNCP

SYSNCP

このコマンドは SYSNCP ユーティリティを呼び出します。

詳細については、『ツールおよびユーティリティ』ドキュメントの「*SYSNCP* ユーティリティ」を参照してください。

47

SYSOBJH

SYSOBJH

このコマンドはオブジェクトハンドラを呼び出します。オブジェクトハンドラを使用して、Natural 環境での配布のために Natural オブジェクトと Natural 以外のオブジェクトを処理します。

詳細については、『ツールおよびユーティリティ』ドキュメントの「オブジェクトハンドラ」を参照してください。

48 SYSPROD

SYSPROD

このコマンドで、Naturalサイトにインストールされている製品を確認できます。現在のNaturalバージョン、Natural 選択ユニット、および Natural 環境で実行している製品に関する情報を取得します。

コマンドを入力すると、ダイアログが表示され、インストールされている製品ごとに次のような情報が表示されます。

- 製品名
- 製品バージョン（『用語集』の「バージョン」も参照）
- インストール日

リストにある製品の一部については、ダイアログの [Cmd] 列に行コマンドでマークすることにより、追加情報を取得できます。

行コマンド	説明
EX	詳細な製品情報を表示します。
HI	製品情報の履歴を表示します。
SC	製品のサブコンポーネントを表示します。



注意: 一部の製品については、行コマンドを使用できません。

49

SYSPROF

SYSPROF

このコマンドは、Natural システムファイルの現在の定義を表示します。

各システムファイルについて、次の情報がに表示されます。これは、次に挙げるページのいずれかが現在表示されている場合に PF4 キーを押すことに相当します。

- ファイル名
- データベース ID
- ファイル番号
- データベースタイプ

また、データベース ID とファイル番号の定義済み組み合わせごとに、次の情報を表示できます。

- ファイルシステム内のパス（PF5 キーを押した場合）
- 割り当てられている論理ファイル番号（PF6 キーを押した場合）

50

SYSRPC

SYSRPC	CSMASS	{ <i>individual-syntax</i> }
	PING	
	SGMASS	
	SM REPLACE	
	SRVLIST	

このシステムコマンドは、リモートプロシージャコールをメンテナンスするために使用される SYSRPC ユーティリティを呼び出します。

RPC 固有のタスクを実行するために、SYSRPC システムコマンドでダイレクトコマンドを指定できます。これらのタスクと、ダイレクトコマンドに適用される *individual-syntax* について詳細は、『ツールおよびユーティリティ』ドキュメントの「SYSRPC ユーティリティ」を参照してください。

ダイレクトコマンド	目的と関連トピック
CSMASS	RPC コールに必要なバッファサイズを計算します。 「サイズ要件の計算」および「SYSRPC CSMASS コマンドの使用」を参照してください。
PING	サーバーに対して ping を実行します。 「Pinging an RPC Server」および「Using the SYSRPC PING Direct Command」を参照してください。
SGMASS	複数のインターフェイスオブジェクトを生成します。 「Generating Multiple Interface Objects」および「Using the SYSRPC SGMASS Command」を参照してください。
SM REPLACE	サービスディレクトリ内の 1 つまたは複数の項目を置き換えます。 「Replacing Items in the Service Directory」を参照してください。

ダイレクトコマンド	目的と関連トピック
SRVLIST	EntireX Broker に登録された Natural RPC サーバーに関する情報を提供します。 「 <i>Listing Servers Registered on EntireX Broker</i> 」を参照してください。

SYSRPC ユーティリティ機能を適用してサーバーとクライアントシステム間の通信フレームワークを確立する方法の詳細については、『*Natural RPC（リモートプロシージャコール）*』ドキュメントを参照してください。

51

SYSWIZDB

SYSWIZDB

このコマンドは、Naturalスタジオ内の開発ツールウィザードであるデータブラウザを呼び出します。これにより、ファイル構造を表示、出力、または保存することができます。

52 SYSWIZDW

SYSWIZDW

このコマンドは、特定の用途のためのダイアログを作成するツールであるダイアログウィザードを呼び出します。定義済みダイアログボックスに、目的に適合する複数のレイアウトを設定できます。

53 TECH

TECH

このコマンドは、Naturalセッションに関する次の技術情報およびその他の情報を表示します。

- ユーザー ID
- ライブラリ ID
- Natural バージョン（『用語集』の「バージョン」も参照）
- スタートアップ（開始）トランザクション
- Natural Security インジケータ
- オペレーティングシステム名およびバージョン
- マシンクラス
- ハードウェア
- TP モニタ（メインフレームおよびリモート構成の Windows（*TPSYS）のみ）
- デバイスタイプ
- 端末 ID（メインフレームおよびリモート構成の Windows のみ）
- コードページ
- ロケール
- 最後に発行されたコマンド
- 最後に発生したエラーに関する情報
- 現在アクティブなすべての STEPLIB の名前、データベース ID およびファイル番号
- 現在アクティブな Natural オブジェクトおよび上位レベルの全オブジェクトの名前、タイプ、レベル、および下位レベルのオブジェクトを呼び出すステートメントの行番号（メインフレーム、UNIX および OpenVMS のみ）

**注意:**

1. キャラクタユーザーインターフェイスアプリケーションの場合のみ：アプリケーションの任意のポイントからこの情報を表示するには、端末コマンド %<TECH を使用できます。
2. このコマンドは、リモートセッションでも使用できます。バッチモードですべての情報を読み取ることができます。

54 UNCATALOG

UNCATALOG [*object-name* ...]

このコマンドは、1つ以上のオブジェクトモジュールを削除します。

現在ログオンしているライブラリに保存されているオブジェクトだけを削除できます。UNCATALOG コマンドはソースワークエリアの内容には影響しません。

UNCAT	UNCATALOG コマンドを入力するときに <i>object-name</i> を指定しないか、またはアスタリスク (*) を1つ指定すると、現在のライブラリにカタログされているすべてのオブジェクトのリストが表示されます。リスト上で、削除するオブジェクトをマークできます。
UNCAT *	
UNCAT <i>object name</i>	<i>object-name</i> として、削除するオブジェクトの名前を指定します。 複数のオブジェクトを削除する場合、 <i>object-names</i> は1つ以上の空白（または現在定義されているデリミタ文字）で区切る必要があります。 特定の文字列で始まるすべてのオブジェクトを削除する場合は、 <i>object-name</i> にアスタリスク (*) を使用します。選択されたすべてのオブジェクトを含むリストが表示されます。そのリストで、削除する各オブジェクトをマークできます。

 **注意:** 有効ではないパラメータファイルに FDIC システムファイルが指定されていると、UNCATALOG コマンドが発行されるときに Natural は該当するエラーメッセージを表示します。

55 UPDATE

UPDATE { ON OFF }

このコマンドは、Naturalプログラムによって実行されるデータベースの更新を制御するために使用します。

UPDATE ON	データベースの更新が許可されます。
UPDATE OFF	データベースの更新が許可されません。 UPDATE、STORE、またはDELETEステートメントを指定したプログラムが実行される場合、データベースの更新は行われません。代わりに、次の画面 I/O で、NAT1010 警告メッセージが表示されます。 さらに、UPDATE または DELETE ステートメントが含まれるデータベースループでは、レコードがホールド状態になりません（read with hold なし）。

56 XREF

XREF	[	ON	]
		OFF	
		FORCE	
		DOC	
		?	

このコマンドは、Predict をインストールしている場合にのみ使用できます。Predict の "active cross-references" 機能の使用を制御します。

アクティブクロスリファレンス機能は、あるプログラム／データエリアを参照するオブジェクトに関して、データディクショナリ内に、自動的にドキュメントを作成します。これらのオブジェクトとして、プログラム、サブプログラム、サブルーチン、ヘルプルーチン、マップ、データエリア、データベースビュー、データベースフィールド、ユーザー定義変数、処理ルール、エラー番号、ワークファイル、プリンタ、クラスおよび 保持 (RETAIN) ISN 集合があります。

アクティブクロスリファレンスは、プログラム／データエリアがカタログされたときに生成されます。

クロスリファレンスデータは、システムコマンド [LIST](#) の XREF オプションを使用して参照できます。

アクティブクロスリファレンスの詳細については、Predict のドキュメントを参照してください。

次のコマンドオプションを使用できます。

XREF	パラメータなしで XREF コマンドを入力すると、メニュー／ダイアログが表示されてオプションを指定できます。
XREF ON	アクティブクロスリファレンス機能を有効にします。Naturalプログラム／データエリアをカタログするたびに、クロスリファレンスデータが適切な Predict エントリに保存されます。
XREF OFF	アクティブクロスリファレンス機能を無効にします。クロスリファレンスデータは保存されません。カタログされるオブジェクトの既存のクロスリファレンスデータは削除されます。
XREF FORCE	オブジェクトをカタログ化できるのは、オブジェクトに Predict エントリが存在する場合のみです。オブジェクトがカタログされるとき、そのクロスリファレンスデータは Predict に保存されます。Predict エントリが存在しない場合は、オブジェクトをカタログ化することはできません。
XREF DOC	オブジェクトをカタログ化できるのは、オブジェクトに Predict エントリが存在する場合のみです。ただし、オブジェクトがカタログされるとき、そのクロスリファレンスデータは Predict に保存されず、オブジェクトの既存のクロスリファレンスデータは削除されます。Predict エントリが存在しない場合は、オブジェクトをカタログ化することはできません。
XREF ?	XREF ? を使用すると、XREF コマンドのヘルプ機能呼び出すことができます。

Natural Security の考慮事項

Natural Security がインストールされている場合、XREF 設定はライブラリセキュリティプロファイルでライブラリごとに設定されます。セキュリティプロファイルによっては、XREF コマンドのいくつかのオプションを利用できないことがあります。