

## Natural for Windows

### Glossary

バージョン 6.3.3

October 2008

This document applies to Natural バージョン 6.3.3 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 1992-2008. 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

Glossary .....	1
A .....	1
B .....	3
C .....	3
D .....	4
E .....	6
F .....	8
G .....	9
H .....	9
I .....	9
J .....	11
K .....	11
L .....	11
M .....	12
N .....	12
O .....	16
P .....	17
R .....	18
S .....	19
T .....	22
U .....	23
V .....	23
W .....	23
X .....	24
Z .....	25
あ行 .....	25
か行 .....	31
さ行 .....	36
た行 .....	42
な行 .....	44
は行 .....	46
ま行 .....	53
や行 .....	54
ら行 .....	55
わ行 .....	57



## Glossary

---

### 承認、商標、免責事項

この用語集には、Natural 固有の用語と一部の汎用的なコンピュータ用語に加えて、International Business Machines Corporation (IBM) の用語と説明も含まれています。IBM の用語は、IBM が著作権を有する下記のマニュアルから IBM の許可を得て抜粋しており、この用語集ではアスタリスク (\*) を付記してあります。

*Terminology*, Copyright © International Business Machines Corporation.

さらに、この用語集には Unicode Consortium の用語と説明も含まれています。Unicode Consortium の用語は、Unicode, Inc. が著作権を有する下記のマニュアルから Unicode, Inc. の許可を得て抜粋しています。

*Glossary of Unicode Terms*, Copyright © Unicode, Inc.

Software AG およびすべての Software AG 製品名は、Software AG 社の商標または登録商標です。この用語集に記載されているその他の企業名、製品名、オペレーティングシステムおよび関連する用語は、各所有者の商標です。

この用語集の説明は、ユーザーがわかりやすいように概略レベルで記述されています。Software AG は、説明の正確性または完全性について責任を負いません。

## A

---

ActiveX コントロール	<i>Windows</i> : ラジオボタンやリストボックスなどのユーザーインターフェイスを備えた、標準化されたプログラミングコンポーネント。Natural ダイアログで使用することができます。Natural では、レジストリから使用できる ActiveX コントロールの選択リストを呼び出すことができます。コンポーネントブラウザを使用して、Property ページにデフォルト設定を表示できます。
ActiveX コンポーネント	<i>Windows</i> : ビジネスロジックや計算など、外部プログラミングコンポーネントの標準化された Microsoft インターフェイス。ActiveX コンポーネントにユーザーインターフェイスは含まれていません。

Adabas	Software AG のマルチスレッド化された高性能なデータベース管理システム。複数のプラットフォームに適応できます。
AIV = アプリケーション独立変数	<ol style="list-style-type: none"><li>1. メインフレーム：すべてのアクティブな AIV をリストする Natural システムコマンド。</li><li>2. プログラム全域で値を割り当てののに使用される Natural 変数。「ユーザー定義変数」セクション（『プログラミングガイド』）および DEFINE DATA ステートメント（『ステートメント』ドキュメント）を参照してください。</li></ol>
Ajax = 非同期 JavaScript + XML	ユーザー要求を即座に処理する対話型のブラウザベースアプリケーションを構築する方法の 1 つ。  「 <a href="#">Natural for Ajax</a> 」も参照してください。
ALF = アクセス層フォーマット	<i>Predict</i> ：データベースシステムから独立した、事前定義されたフォーマットを指します。このフォーマットを使用して、Predict Case と Natural エンジニアリングワークベンチとの間でデータを交換することができます。Predict Coordinator で送信するデータはすべて、まず送信媒体に配置する必要があります。
AMODE = アドレッシングモード	プログラム実行中に使用される仮想アドレスのビット数（24 または 31）。
API	下記の「 <a href="#">アプリケーションプログラミングインターフェイス</a> 」を参照してください。
AppID	Windows システムレジストリでは、各アプリケーションは AppID で示されます。AppID はグローバルユニーク ID であり、レジストリキー HKEY_CLASSES_ROOT\AppID に保存されています。 <b>DCOM</b> では、AppID を使用してアプリケーションのクラスをグループ化します。また、セキュリティ設定なども AppID ベースで定義されます。Natural では、NaturalX サーバー ID ごとに 1 つの AppID がレジストリに作成されます。  詳細については、「 <a href="#">サーバー ID</a> 」を参照してください。
ASM = 認可サービスマネージャ	以下の認可されたオペレーティングシステム機能を、Natural 内で使用できるようにします。 <ul style="list-style-type: none"><li>■ Natural バッファプールオブジェクトの伝搬。</li><li>■ システム管理機能（<b>SMF</b>）レコードの書き込み。</li><li>■ セッション情報プール（<b>SIP</b>）への Natural セッション情報の保存。</li></ul>

ASMでは、PCルーチン経由で機能が提供され、ASM自身のアドレススペースで実行されます。

## B

BLOB	BLOB（「バイナリラージオブジェクト」）は、非常に大きな変数のことです。
BMP = 基本多言語面	
BMP = バッチメッセージ処理	IMS/TM:IMS/TMの制御下でバッチモードで非同期に実行されるプログラム。
BOM = バイトオーダマーク	
BT = BACKOUT TRANSACTION	データベーストランザクションを取り消して、トランザクション開始前のデータ状態を復元するために発行されるNaturalステートメント。「トランザクション」およびEND TRANSACTION (ET)も参照してください。

## C

CATALOG	Naturalでは、「カタログ」という語はNaturalシステムコマンドCATALOGを指します。このコマンドは、ソースコードをコンパイル（カタログ）し、生成されたオブジェクトコードを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します。
CF = カップリング機能	z/OS：Sysplex環境に参加しているオペレーティングシステムによって共有されているデータへのアクセスを制御および調整する際に使用されるハードウェアコンポーネント。CFは、Sysplex環境で通信するために認可サービスマネージャ (ASM) およびロールサーバーによって使用されます。
CGI = Common Gateway Interface	Naturalでインターネットサービスを提供するために、Natural Web インターフェイスによって使用されます。
CICS = 顧客情報管理システム	z/OS および z/VSE：IBM のオンライントランザクション処理システム。
CICSplex = CICS Complex	リソースマネージャとして動作する、相互接続されたCICSリージョンのセット。顧客のビジネスニーズに対して首尾一貫したサービス群を提供するために結合されたものです。*
CLOB	CLOBは、「文字ラージオブジェクト」のことです。

CMS = Conversational Monitor System	VM：対話型の一般的なタイムシェアリング機能、問題解決機能、およびプログラム開発機能を提供する仮想マシンオペレーティングシステム。 <b>VM</b> 制御プログラムの制御下でのみ作動します。*
COM=コンポーネントオブジェクトモデル	プログラムコンポーネントオブジェクトを開発およびサポートするための Microsoft のフレームワーク。COM は、同一コンピュータ内でのクライアントとサーバーの通信を可能にするインターフェイス群を提供します。異なるマシンのコンポーネントは、 <b>DCOM</b> を使用して結合できます。
Com-plete	Software AG の汎用的で完全に会話形式のオンライントランザクション処理 (OLTP) ソフトウェア。
Con-form	<b>Con-nect</b> のメインフレームバージョンで自動的にインストールされるテキストフォーマッタ。
Con-nect	Software AG のオフィス通信システム。電子メール、テキスト処理、ドキュメント/時間管理など、多様なオフィスサポート機能を利用できます。
CSA = 共通システム記憶域 / 共通記憶領域	<ol style="list-style-type: none"><li>1. CICS：共通システム記憶域は、CICS の操作に必要な記憶域とデータを含む主要な CICS ストレージコントロールブロックです。*「<b>ECSA</b>」も参照してください。</li><li>2. z/OS：共通記憶領域は、z/OS のすべてのアドレススペースに使用できる領域です。</li></ol>
CSECT = 制御セクション	プログラマによって再配置可能な単位であると指定されたプログラム部分。この部分のすべての要素は、隣接するメインストレージにロードされます。*
CSV = Comma-Separated Values	統計表を含むファイルはカラムが区切られた CSV 形式で保存されるので、スプレッドシートで操作するのに便利です。
CUI = キャラクターユーザーインターフェイス	

## D

---

DB	「データベース」の略です。
DB2 = DATABASE 2	データをテーブル形式でユーザーに示すリレーショナルデータベース管理システム。SQL 要求を発行する CICS アプリケーションプログラムによってアクセスすることができます。*
DBA	「データベース管理者」の略です。



DBCS = ダブルバイト文字 セット	各文字が2バイトで表される文字セット。この文字セットは通常、シングルバイト文字セット ( <b>SBCS</b> ) で扱えるより多くのコードを必要とする言語 (中国語や日本語など) を表すのに使用されます。
DBD = データベース記述	<i>DL/1</i> : <b>DL/I</b> データベースの物理的特質の記述。データベースの構造、セグメントキー、物理編成、名前、アクセスメソッド、デバイス、およびその他の詳細を定義します。*
DBID = データベース ID	データベース番号。
DCL = データ制御言語/ DIGITAL Command Language	SQL : データ制御言語では、権限の許可/取り消しを行うステートメントを使用して、データのセキュリティ面を操作できます。  <i>OpenVMS</i> : DIGITAL Command Language (Hewlett-Packard 社の商標) は、OpenVMS環境でオペレーティングシステム命令を処理します。
DCOM = 分散コンポーネン トオブジェクトモデル	Microsoft のコンポーネントテクノロジー。異なるコンピュータ上のオブジェクトが、インターネット用プロトコルや Web ベースのプロトコルなど共通のプロトコルを介して通信できるようになります。DCOM は、分散環境でのソフトウェアコンポーネントの対話方法を指定する分散コンポーネントソフトウェアモデルに <b>COM</b> を拡張します。
DCSS = 非連続共有セグメン ト	<i>CMS</i> : 仮想マシンのアドレス範囲外の仮想ストレージ。読み取り専用データまたは再入可能コードを含めることができます。DCSS は非隣接セグメントを仮想マシンのアドレススペースに連結して、プログラムを取り出すことができるようにします。*
DDL = データ定義言語	SQL データ構造を作成、修正、および削除するために使用します。
DDM = データ定義モジュー ル	Natural プログラミングオブジェクトによって参照される物理データベースファイルの論理定義。DDM には、プログラミングオブジェクト内でのフィールドの使用に関連した、ファイルの各フィールドに関する情報が含まれています。メインフレームでは、DDM はシステムファイル FDIC に保存されます。Windows、UNIX、および OpenVMS では、DDM は関連ライブラリのシステムファイル FUSER に保存されます。DDM 構造の詳細については、『プログラミングガイド』の「Adabas データベース内のデータへのアクセス」の「データ定義モジュール - DDM」を参照してください。  『エディタ』ドキュメントの「DDM エディタ」を参照してください。
DIGITAL command language (DCL)	「 <b>DCL</b> 」を参照してください。

DL/I = Data Language/One	階層型データベースを操作するためのアクセスメソッド。
DLL	「ダイナミックリンクライブラリ」の略です。
DML = データ操作言語	<b>SQL</b> データ構造を操作するために使用される言語。
DSECT	「ダミー制御セクション」の略です。
DTD = 文書型定義	SGML および XML ドキュメントのスキーマ指定方式。DTD は、ドキュメントに含まれるか外部サブセットに属し、URL ごとにそのドキュメントのドキュメントタイプ宣言の内部から参照されます。よく知られている DTD に、DocBook、CML、IBTWSH、HTML などがあります。dtd2html は、SGML DTD に対して HTML ドキュメントを作成します。XML の場合は、DTD が新しい XML スキーマ指定方式によって置き換えられます。

---

## E

ECSA = 拡張共通システム記憶域	z/OS : 16 MB を超える z/OS 仮想ストレージの主な要素。ここには、すべてのアクティブな仮想ストレージアクセススペースによってアドレス指定できる、ページ可能なシステムデータ領域が含まれます。16 MB 以内の共通システム記憶域 ( <b>CSA</b> ) が複製されます。*
ELPA = 拡張リンクパック記憶域	z/OS : 16 MB を超える z/OS 仮想ストレージの主な要素。リンクパック記憶域 ( <b>LPA</b> ) が複製されます。*
Entire Connection	ホスト - PC 間の通信を処理するための Software AG 製品。Entire Connection では、メインフレーム端末エミュレーション、高度なアプリケーションプログラムインターフェイス、無人のワークステーション操作、データ変換機能および関連する通信タスクを使用できます。
Entire DB Engine	<b>Adabas</b> に基づいた、Software AG のエンティティリレーションシップデータベース。
Entire Net-Work	データベースアプリケーション、照会ツール、連携処理アプリケーション、アプリケーション開発ツールおよびその他のソフトウェアに共通の通信サービスを提供する、Software AG のミドルウェア。
Entire Operations	コンピュータセンタでのバッチ処理の準備、操作、モニタリング、ロギングを自動化する、Software AG のオンライン制御およびスケジューリングシステム。

Entire Output Management	異機種クライアント／サーバー環境で、データを作成したアプリケーションまたはプログラムを変更せずにあらゆるタイプの出力データを処理できる Software AG 製品。
Entire Screen Builder	Web からホストへの通信を処理する Software AG 製品。 Entire Screen Builder を使用すると、クライアント／サーバー環境であらゆるタイプのホスト端末通信が可能になります。 表示クライアントでは、従来どおりの緑色の画面端末エミュレーションから「GUIfied」 Windows ダイアログまで、さまざまなスタイルでの表示が可能です。
Entire System Server	Natural 環境でオペレーティングシステムサービスを提供する、Software AG の独立したソフトウェアパッケージ。 Entire System Server は、Entire Net-Work でリンクされているコンピュータにインストールされた場合、異機種のオペレーティングシステムが含まれている可能性のある分散コンピューティング環境をサポートします。
Entire Transaction Propagator	データベースファイルを複製するときに使用する Software AG 製品。 この製品を使用すると、Adabas ユーザーは単一データベース内または分散ネットワーク内のデータベースファイルを複写または複製することができます。
EntireX	オープンエンタープライズコンピューティング向けの Software AG のミドルウェアテクノロジー。 広範囲な運用環境に移植可能なコンポーネントを備えた分散アプリケーションを開発するための、強力なインフラストラクチャが提供されます。 EntireX では、メインフレーム上などにあるレガシーアプリケーションと GUI フロントエンドとを統合することができます。
EntireX Broker	EntireX Broker は、IT アーキテクチャ内での分散アプリケーションコンポーネント間の通信を制御します。 Broker がサポートする通信モデルには、同期／非同期、クライアント／サーバー、ピアツーピア、パブリッシュ／サブスクライブなど、さまざまなタイプがあります。 関連テクノロジーとして、EntireX があります。
EntireX Broker スタブ	<b>Natural RPC</b> ランタイムと、クライアントおよびサーバー間で整列されたデータを交換する EntireX Broker トランスポート層の間のインターフェイス。
EntireX DCOM	異機種ネットワーク環境で <b>DCOM</b> コンポーネントを開発するときに使用する Software AG 製品。
EntireX Manager	これは、 <b>System Management Hub</b> に変わりました。
ESDS = エントリ順データセット	VSAM：レコードがデータベースに入力された順に並んでいるデータセット。 このデータセットにインデックスは含まれておらず、

処理するにはアドレス指定による直接アクセスまたはシーケンシャルアクセスを行います。新規レコードはデータセットの最後に追加されます。\*

ET = END TRANSACTION データベーストランザクションの完了を確認するために発行される Natural ステートメント。「トランザクション」および「BACKOUT TRANSACTION (BT)」も参照してください。

Extract *Predict* : *Predict* の事前定義されたオブジェクトタイプの 1 つ。  
Extract は、主に次の 2 つの機能を実行します。

- オブジェクトの論理的グループ化
- *Predict Coordinator* で送信されるオブジェクトのセットの作成

## F

FDIC クロスリファレンス情報および DDM を含むシステムファイル。ユーザーマシンに *Predict* がインストールされている場合、FDIC が *Predict* ディクショナリファイルです。*Predict* がインストールされていない場合、FDIC は FNAT または FUSER と同一で、メインフレームでは DDM のみを含みます。FDIC は、FDIC プロファイルパラメータで指定します。詳細については、『パラメタリファレンス』の記載を参照してください。

FDICX VSAM : VSAM ベースクラスタ **FDIC** に対する代替インデックスパス。

FDT = フィールド定義テーブル *Adabas* : *Adabas* ファイルのフィールド定義テーブル。*Adabas* によって格納され、使用されます。

FNAT Natural システムオブジェクトおよびパラメータプロファイルが保存される Natural システムファイル。指定するには、FNAT プロファイルパラメータを使用します。詳細については、『パラメタリファレンス』の記載を参照してください。

FNR = ファイル番号 通常は、Natural システムファイルのファイル番号を指します。

FSEC Natural Security システムファイル。

FSEQ 「ファイル順序」の略です。

FSPOOL Natural Advanced Facilities スプールシステムファイル。「システムファイル」も参照してください。

FUSER すべてのユーザー作成オブジェクトが保存される Natural システムファイル。指定するには、FUSER プロファイルパラメータを使用

します。詳細については、『パラメータリファレンス』の記載を参照してください。「システムファイル」も参照してください。

## G

**GDA=グローバルデータエリア** 複数のプログラミングオブジェクトによる共有が可能なデータ要素が含まれた Natural プログラミングオブジェクト。

詳細については、『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。

**GP** 「生成プログラム」の略です。

**GUI** 「グラフィカルユーザーインターフェイス」の略です。

**GUID** GUID (グローバルユニーク ID) は、COM/DCOM モデル全体を通して確実に一意である定数です。ネットワーク上でクラスとそのインターフェイスを明確に識別するために使用します。Natural クラスをDCOMクラスとして登録する場合、クラスおよびその各インターフェイスにGUIDを割り当てる必要があります。Natural では、GUIDは英数字定数で表され、これはデータエリアエディタで生成できます。Natural スタジオクラスビルダは、自動的に新しいGUIDを個々の新しいクラスに割り当てます。

詳細については、『プログラミングガイド』の「グローバルユニークID - GUID」を参照してください。

## H

**HF=階層ファイルシステム** z/OS UNIX サービスで使用できる UNIX ファイルシステム。

**HTML** 「ハイパーテキストマークアップ言語」の略です。

**HTTP** 「ハイパーテキスト転送プロトコル」の略です。

**HTTP Cookie** サーバーにおいて、クライアントアプリケーションのシステム状態に関する情報を格納または検索するメカニズムの役を果たします。Web ベースアプリケーションが選択された項目に関する情報、ユーザー設定、登録情報、および後で検索可能な他の情報を格納することができるのも、このメカニズムがあるおかげです。REQUEST DOCUMENT ステートメントで使用されます。詳細については、『ステートメント』ドキュメントの「REQUEST DOCUMENT」を参照してください。

## I

I/O バッファ = 入力／出力 バッファ	最も頻繁に使用されるデータおよびデータ関係を含む記憶域。これにより、物理的な入力／出力動作を最小限に抑えてコンピュータ時間を節約することができます。
ICU = International Components for Unicode	<a href="http://www-306.ibm.com/software/globalization/icu/index.jsp">http://www-306.ibm.com/software/globalization/icu/index.jsp</a> も参照してください。
ID = 識別子	<p><i>Predict</i> : 1つの例外を除いて、Predict オブジェクトは2つの属性、すなわちオブジェクトタイプと ID で一意に識別されます。したがって、タイプの異なるオブジェクトが同じ ID を持つこともあります。</p> <p>フィールドオブジェクトの場合、3つの属性、すなわちオブジェクトタイプ、所属先ファイルの ID、およびフィールド自身の ID によって一意に識別されます。したがって、異なるファイル内のフィールドオブジェクトが同じ ID を持つこともあります。</p>
IIS	「Internet Information Services」 (Microsoft) の略です。
IMS = Information Management System	<b>DL/I</b> データベースのデータにアクセスできるようになるために、 <b>CICS</b> によって使用されるデータベースマネージャ。IMS は、階層構造のデータ配列、および IMS データベースを操作するアプリケーションプログラム内の共通アクセス方式を規定します。*
IMS/DB = Information Management System/Database	<i>z/OS</i> : データを階層構造に編成し、IMS/DB データベースを修正するプログラムにアクセスするために、 <b>DL/I</b> データベースへのアクセスを提供する管理システム。
IMS/DC = Information Management System / Data Communication	現在は使用されていません。 <b>IMS/TM</b> に変わりました。
IMS/TM = Information Management System / Transaction Manager	<i>z/OS</i> : IBM が提供するオンライントランザクション処理システムのコンポーネント。
IMS/VS = Information Management System / Virtual Storage	複合データベースおよびネットワークを管理できる、データベース／データ通信 (DB/DC) システム。*
INPL = Initial Natural Program Load	Natural モジュールまたは Software AG データセットの DDM (Natural INPL テープなど) をワークファイル 1 からロードしたりスキャンする場合に使用する Natural ユーティリティです。さらに、Natural Security 環境を強制的に初期化できる Natural Security リカバリー機能もあります。
IOCB	「入力／出力コントロールブロック」 の略です。



IPL=初期プログラムロード	オペレーティングシステムのロードを初期化するプロシージャ。
ISAPI = Internet Server Application Programming Interface	Natural でインターネットサービスを提供するために Natural Web インターフェイスによって使用されます。
ISN = 内部シーケンス番号	<b>Adabas</b> データベースファイルに保存された可変長レコードの一意の識別子。各 ISN には、レコードが物理的に格納されている Adabas ブロックの相対 Adabas ブロック番号 (RABN) が割り当てられます。
IUPD = INPL 更新	<b>INPL</b> の修正を含みます。

## J

JCL = ジョブ制御言語	z/OS および VSE/ESA : <b>ジョブ</b> および特定のオペレーティングシステムでのその要件を記述するために使用される制御言語。*
	ジョブは、Natural のインストール/メンテナンス時、およびバッチモードでの Natural の実行時に必要です。
JCS = ジョブ制御文	VSE/ESA : <b>ジョブ</b> 制御言語 ( <b>JCL</b> ) の命令文。
JES=ジョブエントリサブシステム	z/OS : IBM がライセンスを持つプログラム。システムへのジョブの投入およびジョブによって作成されたすべての出力データの処理を行います。*

## K

KSDS=キー順データセット	レコードがキー順にロードされ、インデックスによって制御されている VSAM ファイル。*
----------------	--

## L

LDA = ローカルデータエリア	複数のプログラミングオブジェクトによる使用が可能なデータ要素定義が含まれた Natural プログラミングオブジェクト。詳細については、『プログラミングガイド』の「ローカルデータエリア」を参照してください。
LPA = リンクパック記憶域	z/OS : 16 MB 以内の z/OS 仮想ストレージの主要要素。LPA を構成するストレージには、システムによって共有されるすべての共通再入可能モジュールが含まれます。LPA では、モジュールのコピーが共有される形になるので実際のストレージを節約できます。また、キー 0 のプログラムでさえも LPA コードを書き換えることができないため保護上の効果を上げることができ、モジュールの分岐が可能であるためパスの長さを短縮できます。LPA は 16 MB

を超えると、拡張リンクパック記憶域 (**ELPA**) として複製されま  
す。\*

LU = 論理単位

ユーザーにシステムネットワークアーキテクチャ (**SNA**) へのア  
クセスを提供する、基本となるネットワークエンティティ。

## M

---

MPP = メッセージ処理プロ  
グラム

1. 端末から受け取ったメッセージを処理する、またはメッセージ  
に応答するプログラム。
2. **IMS/VS** では、トランザクションによって起動し、オンライン  
IMS/VS データベースおよびメッセージキューにアクセスできる  
アプリケーションプログラム。\*

## N

---

NATPARM

Natural のすべてのプロファイルパラメータ設定が含まれたパラ  
メータファイル (Windows/UNIX/OpenVMS) またはパラメータ  
モジュール (メインフレーム)。このファイルがないと、Natural  
は実行できません。初期状態の NATPARM には、Software AG に  
よって提供されるシステムデフォルトが含まれます。Natural にシ  
ステムデフォルト以外のパラメータ値を使用する場合は、  
NATPARM を修正するか、ユーザー独自のパラメータファイル/  
モジュールを作成します。

メインフレーム : NATPARM は、ソース形式で提供されるロード  
モジュールです。アセンブルして、ニュークリアスおよび/または  
フロントエンドモジュールにリンクする必要があります。

Windows/UNIX/OpenVMS : NATPARM は、**Natural** コンフィグ  
レーションユーティリティを使用して編集できるバイナリファイ  
ルです。

Natural Advanced Facilities

Natural Advanced Facilities は NATSPOOL から構成されていま  
す。NATSPOOL は、Natural プログラムの出力をスプールして、  
後で実際のプリンタに送信することができるスプール/レポート  
管理システムです。

Natural Construct

柔軟性の高い多様なアプリケーション構造テンプレートを提供す  
るアプリケーションジェネレータ。Natural ソフトウェア生成の多  
くの面を自動化します。IT エキスパートもそれ以外のユーザーも  
同様に使うことができます。



- Natural Construct Spectrum** Windows からメインフレーム Natural へのアクセスを提供します。アプリケーション開発者は、Natural Construct Spectrum およびソフトウェア開発キット (SDK) を使用して、クライアント / サーバーおよび Web アプリケーションのすべてのコンポーネントを作成することができます。これには例えば、メンテナンス機能やブラウズ機能を実行する Natural オブジェクトサブプログラムや、そうした Natural オブジェクトサブプログラムと通信する GUI ダイアログまたは Web ページなどがあります。アプリケーションのサーバーコンポーネントとクライアントコンポーネントとの間の通信は、Construct Spectrum のミドルウェアコンポーネントに加えて、Entire Broker と Entire Net-Work の組み合わせ (または、TCP/IP を使用するよう設定された Entire Broker) によって実行されます。
- Natural Engineer** メインフレーム / Windows : Natural コードのチェック、メンテナンス、および定義要件への準拠を行うためのツール。将来、標準の適用、柔軟性の維持、新しいテクノロジーへの対応を行うために、Natural アプリケーションを再構成およびメンテナンスすることができます。
- Natural for Ajax** Natural for Ajax では、Ajax (非同期 JavaScript + XML) テクノロジーを使用する機能豊富なインターネットアプリケーションを作成できます。これにより、UNIX およびメインフレームプラットフォーム上の Natural ユーザーは、GUI デスクトップアプリケーションに似た、ブラウザベースのユーザーインターフェイスを備えた Natural アプリケーションを開発して使用できます。
- 詳細については、『*Natural for Ajax*』ドキュメントを参照してください。
- Natural ISPF (Integrated Structured Programming Facility)** アプリケーションのライフサイクルを通してアプリケーションの構築、テストおよびメンテナンスを行うために使用される、Software AG のアプリケーション開発ツールです。
- Natural Optimizer Compiler** メインフレーム : 可能な場合は常にマシンコードを生成する (可能でなければ中間コードを生成する) Natural コンパイラ。
- Natural RPC (リモートプロシージャコール)** Natural 環境で RPC 手法を実装します。Natural RPC を使用すると、Natural はネットワーク上のリモートコンピュータにある Natural サブプログラム、アプリケーション、プロシージャ、およびオブジェクトメソッドを、EntireX Broker を介して呼び出すことができます。クライアントは、サービスを提供するサーバーに要求を送信します。要求の伝送は、クライアントとサーバーの **スタブ** を介して実行されます。

	<p>クライアント、サーバー、およびサービス間の接続は、Natural ユーティリティ <b>SYSRPC</b> を使用して Natural クライアントごとに設定されます。</p> <p>詳細については、『<i>Natural</i> リモートプロシージャコール (RPC) 』ドキュメントを参照してください。</p>
Natural Security	<p>Natural 環境へのアクセスの制御、チェック、防止を行える総合的なセキュリティシステム。</p> <p>詳細については、『<i>Natural Security</i>』ドキュメントを参照してください。</p>
Natural Web アダプターサーバーエクステンション	<p>Natural Web インターフェイスの一部。CGI、ISAPI、NSAPI など、さまざまなサーバーインターフェイスの実装。</p>
Natural Web インターフェイス	<p>Natural Web インターフェイスを使用すると、既存の Natural アプリケーションを快適に Web 化できます。Natural Web インターフェイスは、Web サーバー (HTTP サーバー) とユーザーの Natural 環境とをリンクするものです。これは、別のサーバーマシン (メインフレームなど) または HTTP サーバーと同じマシン (Netscape の通信サーバーや Microsoft の IIS など) でも使用可能です。</p>
Natural オブジェクトハンドラ	<p>アプリケーション配布用のオブジェクトを処理します。これは、ソース環境のオブジェクトをワークファイルにアンロードし、それらをワークファイルからターゲット環境にロードすることによって実行されます。Natural オブジェクトハンドラは、ライブラリ SYSOBJH にあるユーティリティ SYSOBJH、およびダイレクトコマンドインターフェイスで構成されています。詳細については、『<i>ツールおよびユーティリティ</i>』ドキュメントの「オブジェクトハンドラ」セクションにある「<i>全般的な情報</i>」を参照してください。</p>
Natural 開発サーバー	<p>Natural スタジオ開発環境をリモートの Natural 環境にマップすることができます。その後、Natural スタジオの共通ワークエリアから、リモート環境で Natural アプリケーションを開発およびテストすることができます。</p> <p>詳細については、Natural 開発サーバーに関するドキュメントを参照してください。</p>
Natural コマンド	<p>Natural コマンドは、Natural セッション中に各種機能を実行するために使用します。</p>
Natural コマンドプロセッサ (NCP)	<p>2つのコンポーネント、すなわちメンテナンスとランタイムで構成されます。SYSNCP ユーティリティは、アプリケーション内のナビゲーションを定義および制御するのに使用されるすべての機能</p>

が含まれたメンテナンス部分です。詳細については、『ツールおよびユーティリティ』ドキュメントを参照してください。PROCESS COMMAND ステートメント（『ステートメント』ドキュメントを参照）は、Natural プログラムを呼び出すために使用されるランタイム部分です。

**Natural コンフィグレーションユーティリティ** *Windows/UNIX/OpenVMS* : Natural コンフィグレーションユーティリティは、次のファイルのメンテナンスを行うために使用します。

- グローバルコンフィグレーションファイルおよびローカルコンフィグレーションファイル。
- Natural パラメータファイル。

詳細については、『コンフィグレーションユーティリティ』ドキュメントを参照してください。

**Natural スタジオ** *Windows* : Natural の GUI 開発環境。Natural スタジオでは、元は別々であるエディタとツールを、1つの使いやすいワークエリアに統合できます。

『*Natural* スタジオの使用』および『ファーストステップ』を参照してください。

**Natural セッション** Natural セッションは、ユーザー依存の Natural ランタイムコンテキストです。Natural ランタイムシステムで Natural プログラムを実行する際に必要です。

**Natural デバッガ** *Mainframe/Windows/UNIX* : デバッガを使用すると、以下のことを行うことができます。

- 変数を修正して、Natural アプリケーションのプログラムフローを一時的に制御する。
- コール履歴でプログラムフローをチェックする。
- ブレイクポイントまたはプログラム割り込みの条件を使用して変数の内容をチェックし、Natural プログラム内の論理アプリケーションエラーを検出する。
- 変数を恒久的に監視する。

デバッガを使うと、他人の作成したプログラムを簡単に理解できるようになり、アプリケーション開発にかかる時間を短縮できるようになり、初心者であれば Natural ロジックの理解をさらに深めることができます。

詳細については、『デバッガ』ドキュメントを参照してください。

*OpenVMS* : デバッガは、SPoD 環境で Natural for Windows を使用している場合にのみ利用可能です。リモート Natural 環境にマップして、Natural スタジオに統合されたデバッガを使用できます。

Natural ランタイム	<ol style="list-style-type: none"> <li>1. Natural アプリケーションを実行するために必要な環境を提供する製品。</li> <li>2. Natural コードを解釈実行する仮想マシン。</li> </ol>
NaturalX	オブジェクトベースの Natural アプリケーションの作成に使用します。これにより、Windows プラットフォームで分散オブジェクトテクノロジー (現在は DCOM) を使用して、分散 Natural アプリケーションを作成できます。
NaturalX クライアント	NaturalX クライアントは、NaturalX オブジェクトを作成する、または NaturalX オブジェクトにアクセスするプロセスです。
NaturalX クラス	<p>NaturalX クラスは、オブジェクトベースのプログラミングアプローチに従って、データ構造 (オブジェクト) を対応する機能 (メソッド) とともにカプセル化します。</p> <p>クラスオブジェクトのオブジェクトの内部構造はデータエリア (オブジェクトデータエリア) で定義されます。クラスのメソッドはサブプログラムとして実装されます。</p> <p>NaturalX クラスを <b>DCOM</b> に認識させるには、Natural コマンド REGISTER を使用します。その後、このクラスはネットワーク上でアクセス可能になります。</p> <p>クラスのタイプは、内部、外部、またはローカルです。</p> <p>クラスの詳細については、「プログラミング手法」、「クラスの定義」、「クラスとオブジェクトの使用」および「内部、外部、およびローカルクラス」を参照してください。</p>
NaturalX サーバー	NaturalX サーバーは、1 つの Natural セッションを管理するプロセスです。NaturalX サーバーによって管理される Natural セッションは、COM オブジェクトをホストするために使用されます。
NSAPI = Netscape アプリケーションプログラミングインターフェイス	Natural でインターネットサービスを提供するために Natural Web インターフェイスによって使用されます。

## O

ODA = オブジェクトデータエリア	オブジェクト指向プログラミング: オブジェクトのすべてのプロパティの現在値が格納される場所。クライアントからプロパティ
--------------------	---

としてアクセスできない他の変数も、オブジェクトデータエリアで定義できます。これらの変数は、オブジェクトの内部状態をメンテナンスするために、オブジェクトのメソッドによって使用されます。1つのクラスのすべてのオブジェクトのオブジェクトデータエリアの構造は、DEFINE CLASS ステートメントの OBJECT USING 節で指定されます。オブジェクトデータは、データエリアエディタでローカルデータエリアとして作成されます。

*openUTM* = オープン汎用トランザクションモニター  
*BS2000/OSD* オペレーティングシステム：オンライン環境用の TP アクセスメソッド。

## P

*PCB* = プログラム通信ブロック  
**DL/I** または **IMS** データベースに対するアプリケーションプログラムのインターフェイスを記述する DL/I または IMS コントロールブロック。記述するインターフェイスはその他に、メッセージ処理 / バッチメッセージ処理プログラム用のものや、メッセージの入力元および宛先に対するものである場合もあります。「**PSB = プログラム指定ブロック**」も参照してください。\*

*PDA* = パラメータデータエリア  
 サブプログラム、外部サブルーチン、またはヘルプルーチンにパラメータとして渡されるデータ要素を定義するために使用される Natural プログラミングオブジェクト。詳細については、『プログラミングガイド』の「パラメータデータエリア」を参照してください。

*PLOG* = プロテクションログ  
**Adabas** : Adabas データベースに変更が加えられたときに、レコードおよびその他の要素の変更前と変更後のイメージを記録するファイル。再スタート後に、最後に完了したトランザクションまたは **ET** までデータベースを復元するのに使用されます。

*Predict*  
 Natural で 4 GL の開発を行うための、Software AG のオープンかつ操作可能なデータディクショナリ。アプリケーションメタデータの中央リポジトリであり、ドキュメント機能とクロスリファレンス機能を利用できます。Predict では定義からコードを自動生成できるので、開発およびメンテナンスの生産性を向上できます。

*Predict Coordinator*  
*Predict* : 異なる FDIC ファイル間および Predict と Natural Engineering Workbench との間でのデータ交換を可能にするユーティリティ。一時的なストレージの働きをする、コーディネータ FDIC と呼ばれる独自の FDIC ファイルを使用します。Coordinator 機能を含むアプリケーションが、Natural ライブラリ SYSDICBE 内にあります。

*ProgID*  
**DCOM** クラスの ProgID (プログラム ID) は、クライアントプログラム内でクラスを識別するための意味のある名前になっています。

す。Natural クラスの場合は、REGISTER コマンドでクラスをDCOM クラスとして登録するときに、DEFINE CLASS ステートメントの *class-name* オペランドで定義された名前が ProgID としてレジストリに書き込まれます。

PSB = プログラム指定ブロック

アプリケーションプログラムに使用できるデータベースおよびメッセージの論理的宛先を記述する **DL/I** または **IMS** コントロールブロック。PSB は、1つ以上のプログラム通信ブロック (**PCB**) で構成されています。\*

PU = 物理単位

**SNA** : システムネットワークアーキテクチャ (SNA) では、物理単位は、論理単位 (**LU**) 間の通信セッションをサポートするネットワークノードを指します。

## R

RABN

「相対 Adabas ブロック番号」の略です。

RAD

「高速アプリケーション開発モデル」の略です。

RBA = 相対バイトアドレス

VSAM : 格納されたレコードまたはコントロールインターバルの、その所属先であるデータセットに割り当てられたストレージスペースの先頭からの変位バイト数。\*

Report listing

*Predict*: Report listing タイプの Predict オブジェクトは、Predict Coordinator による送信操作および変換機能のロギングのために使用されます。

RPC = リモートプロシージャコール

リモートコンピュータ上のアプリケーション、プロシージャ、およびオブジェクトメソッドをネットワーク経由で呼び出すために適用されるクライアント/サーバー通信手法。クライアントは、サービスを提供するサーバーに要求を送信します。要求の伝送は、クライアントとサーバーの **スタブ** を介して実行されます。

『*Natural* リモートプロシージャコール (RPC)』も参照してください。

RPC スタブ

*Natural* RPC : サーバープログラムの呼び出しを行うクライアントサブプログラム。RPC スタブは、対応するサーバーサブプログラムと同じ名前で、同じパラメータを含みます。

RPC スタブはクライアント側で CALLNAT 要求を受け付け、*Natural* CALLNAT パラメータを整列させて RPC クライアントランタイムに渡します。RPC クライアントランタイムは、次にそのパラメータをクライアントの EntireX Broker スタブに渡します。詳細については、『*Natural* リモートプロシージャコール (RPC)』ドキュメント



ントの「スタブサブプログラムの作成」および「Natural RPC 自動実行の操作」を参照してください。

RRDS = 相対レコードデータセット	<b>VSAM</b> : データセット編成の一種。レコードは固定長で、相対レコード番号 ( <b>RRN</b> ) によってアクセスされます。*
RRN = 相対レコード番号	<b>VSAM</b> : 相対レコードデータセット ( <b>RRDS</b> ) に格納された各レコードの一意の識別子。
RSL = レコード共有レベル	<b>VSAM</b> : 複数システムにわたる複数のアドレススペースから VSAM データセットをレコードレベルで共有できるオプション。

## S

SAA = システムアプリケーションアーキテクチャ	IBMシステムとデータを処理するための共通の標準およびプロシージャ群。SAAにより、異なるソフトウェア、ハードウェア、およびネットワーク環境が共存できるようになります。SAAは、異なるシステムで一貫したアプリケーションプログラムを設計および開発するための基盤を提供します。*
SBCS = シングルバイト文字セット	ダブルバイト文字セット ( <b>DBCS</b> ) と異なり、各文字が1バイトで表される文字セット。
SCP = システム制御プログラム	メインフレーム : Natural SCP 環境は Natural CICS インターフェイスの一部です。SCP 障害はすべて、ダンプで知ることができます。
SDI	「単一ドキュメントインターフェイス」の略です。
Single Point of Development	Windows : Natural 5 for Windows で実現された、プラットフォームに依存しない開発の概念。Natural 5 で重点が置かれているのは、次のとおりです。 <ul style="list-style-type: none"> <li>■ Natural スタジオを使用した z/OS アプリケーションのリモート開発</li> <li>■ XML ツールキットや、Natural からインターネット上の任意のリソースに直接アクセスできる機能に代表される、高度な Web および XML 機能</li> </ul>
SIP = セッション情報プール	Natural セッション情報レコードを保存します。認可サービスマネージャ ( <b>ASM</b> ) との関連で使用します。
SIR = セッション情報レコード (Natural) / セッション情報検索 (IBM)	Natural : Natural CICS インターフェイスでは、すべてのアクティブな Natural セッションに関する情報が永久に保持されます。セッションごとにセッション情報レコード (SIR) が維持されます。

IBM：特定のゲートウェイまたはすべてのゲートウェイセッションを対象にしたセッション情報検索の有効化／無効化をオペレータが実行できるようになる機能。ゲートウェイセッションの終了時に、ゲートウェイを通過する最新のシーケンス番号またはFID0番号に関するトレース情報が、当該セッションまたは全セッションのSIRを有効にしたすべてのシステムサービス制御ポイント（SSCP）に渡されます。この情報は、要求元ホストに渡される場合もあります。\*

SIT=システム初期化テーブル	システム関数を初期化および制御するための情報、ユーザー指定バージョンのCICSモジュールおよびテーブルを選択するためのモジュール接尾辞、初期化プロセスの制御に使用する情報などが含まれたCICSテーブル。リソース定義マクロDFHSITを使用して複数のSITを生成し、次にSITシステム初期化パラメータを使用して、現在の要件に最も合ったSITを初期化時に選択することができます。*
SMA = System Maintenance Aid	「 <a href="#">System Maintenance Aid</a> 」を参照してください。
SMF = システム管理機能	さまざまなシステムおよびジョブに関する情報を収集、記録するz/OSの標準機能。*
SNA = システムネットワークアーキテクチャ	ネットワークでの情報単位の送信やネットワークの構成および操作の制御を行うための論理的構造、フォーマット、プロトコル、操作手順を定義するIBMのアーキテクチャ。SNAの構造は階層化されているため、情報の発信源および最終宛先（ユーザー）は、情報交換に使用される特定のSNAネットワークサービスおよび機能から独立を保ち、その影響を受けないようにすることができます。*
SOAP = シンプルオブジェクトアクセスプロトコル	EntireX XML Wrapper との組み合わせで、任意のNaturalアプリケーションをHTTPで送信されるXMLドキュメントによって起動できるようにする、XMLベースのメッセージ通信規約。
Software AG Editor	メインフレーム：Software AG Editor は、Natural プログラムエディタの代わりとして使用することができます。Natural ISPF をインストールし、エディタプロファイルを適切に設定する必要があります（エディタプロファイルの詳細については、『エディタ』ドキュメントを参照してください）。
SPA = スクラッチパッドエリア	<b>IMS/TM</b> ：ユーザートランザクション中に処理されるデータを一時的に保存するための端末固有の記憶域。この記憶域の消去はプログラムによって制御されます。
SPoD	「 <a href="#">Single Point of Development</a> 」を参照してください。



SQL = 構造化照会言語	リレーショナルデータベース内のデータを定義および操作する場合に使用されるプログラミング言語。*
SSA = セグメント検索指数	<b>IMS</b> または <b>DL/I</b> : 処理するセグメントまたはセグメントグループを識別する DL/I コールの一部。SSA は単純なセグメント名の場合もあれば、フィールド値の制約が指定セグメントタイプに含まれるように修飾されている場合もあります。*
steplib	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によって共有される複数のオブジェクトの単一の保存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。さらに、Natural Security がインストールされている場合は、特定のオブジェクトへのアクセスを制限するために steplib を使用できます。
STOW	Natural では、"stow" という語は Natural システムコマンド STOW を指します。このコマンドは、ソースコードをソースオブジェクトおよびカタログ化オブジェクトとして Natural システムファイルに保存すると同時にカタログ（コンパイル）します。オブジェクトは両方とも同じタイムスタンプになります。
Super Natural	メインフレームまたは PC ファイルからデータを抽出して処理するための、Software AG のエンドユーザーツール。メニュー方式の照会指定、対話型のレポートレイアウト指定、エンドユーザープロファイルの柔軟なカスタマイズといった機能を利用できます。
SVC = スーパーバイザコール	ストレージの割り当てなど、オペレーティングシステム機能のインターフェイスの働きをする要求。SVC により、オペレーティングシステムは不適切なユーザー入力から保護されます。オペレーティングシステム要求はすべて、SVC によって処理される必要があります。*
Syncpoint = 同期地点	<b>CICS</b> および <b>IMS/TM</b> : アプリケーションプログラムが実行される論理的な地点。この地点では、プログラムによってデータベースに加えられた変更は整合性の取れた完全なものであり、データベースにコミットすることができます。同期地点では、それまで保留されていた出力が宛先に送られ、入力がメッセージキューから削除され、他のアプリケーションでのデータベース更新が可能になります。*
SYSPARM ユーティリティ	メインフレーム : パラメータプロファイルを作成およびメンテナンスするために使用されます。プロファイルパラメータの文字列を一度指定し、その文字列にプロファイル名を付けて保存すると、

	ダイナミックパラメータ <code>PROFILE=profile-name</code> を使用して <b>Natural</b> を起動することができます。そのプロファイル名で格納されているパラメータの文字列は、ダイナミックパラメータとして <b>Natural</b> に渡されます。
Sysplex (System Complex の略)	z/OS : パフォーマンスの改善、作業負荷の均衡化、システム可用性の保証を実現する目的で、z/OS オペレーティングシステムの複数のイメージを結合したもの。
SYSRPC	リモートプロシージャコールを使用する各 <b>Natural</b> クライアントに対して、サービスおよびサーバー接続を構成するためのユーティリティ。『ツールおよびユーティリティ』ドキュメントの「 <b>SYSRPC</b> ユーティリティ」を参照してください。
System Maintenance Aid (SMA)	Software AG メインフレーム製品をインストールおよびメンテナンスするとき使用される Software AG 製品。
System Management Hub	Software AG の製品およびクロスプラットフォーム製品管理フレームワーク。このアーキテクチャにより、既存製品のプラグイン統合、実装固有の管理テクノロジーの分離、製品に依存しないクロスプラットフォームフレームワークとしての再利用が可能になります。
SYSTEM ライブラリ	<b>Natural</b> オブジェクトが現在のライブラリにも <code>steplib</code> にも見つからない場合に検索されるライブラリ。FNAT と FUSER の両方に SYSTEM ライブラリがあります。「 <a href="#">steplib</a> 」を参照してください。

## T

TAC	「トランザクションコード」の略です。
TCB = タスクコントロールブロック	z/OS および VSE/ESA : オペレーティングシステムによって割り当てられるコントロールブロック。システムに関連するタスクのすべての情報を含みます。
TD = 一時データ	<b>CICS</b> : 一時的に保存されるデータ。一時データは、使用されなくなったときに自動的に削除されます。
TIAM = Terminal Interactive Access Method	BS2000/OSD オペレーティングシステム : オンライン環境用の TP アクセスメソッド。
TP モニタ (テレプロセシングモニタ / トランザクション処理モニタ)	z/OS および VSE/ESA : オンライントランザクション処理 (OLTP) アプリケーションの監視および管理を行うための制御プログラム。TP モニタテクノロジーはオペレーティングシステムを補完するものであり、リソースへのアクセスをカプセル化する中間層の働きをします。したがって、ユーザーまたはアプリケーションプログラ

マは、データベースアクセスなどのオペレーティングシステムレベルの技術的詳細に關与する必要がなくなります。

## U

UCB	「装置コントロールブロックユーザー」の略です。
Unicode	Natural ドキュメントで使用されている Unicode 用語の詳細については、Unicode のホームページで公開されている現在のバージョンの Unicode の用語集を参照してください。  <ul style="list-style-type: none"> <li>■ <a href="http://www.unicode.org/">http://www.unicode.org/</a></li> <li>■ <a href="http://www.unicode.org/versions/Unicode4.0.0/b1.pdf">http://www.unicode.org/versions/Unicode4.0.0/b1.pdf</a></li> </ul>
UTM=汎用トランザクションモニタ	現在は使用されていません。 <b>openUTM</b> に変わりました。

## V

VM = 仮想マシン	<ol style="list-style-type: none"> <li>1. 特定のユーザーが排他的に使用しているように見える仮想データ処理システム。ただし、その機能は現実のデータ処理システムのリソースを共有することによって達成される。*</li> <li>2. <b>z/VM</b> では、単一ユーザーに割り当てられる仮想プロセッサ、仮想ストレージ、仮想デバイス、および仮想チャネルサブシステムを指します。専用の拡張記憶域がある場合は、それも含まれます。*</li> </ol>
VM/CMS = 仮想マシン/会話型モニタシステム	多数の対話ユーザーに適応するため、 <b>VM</b> 環境で <b>CMS</b> を統合した IBM のシステム。
VM/SP = 仮想マシン/システムプロダクト	VM/SP は IBM のオペレーティングシステムで、各ログオンユーザーに仮想マシンを与えます。*
VSAM = Virtual Storage Access Method	異なるデータセット編成（キー順データセット ( <b>KSDS</b> )、エン트리順データセット ( <b>ESDS</b> )、または相対レコードデータセット ( <b>RRDS</b> )）のレコードをメンテナンスするための IBM のアクセスメソッド。
VTAM = Virtual Telecommunications Access Method	システムネットワークアーキテクチャ ( <b>SNA</b> ) の遠隔処理装置およびそのユーザーと通信するためのアプリケーションプログラミングインターフェイス (API)。

## W

WTO = Write-to-Operator オプションのユーザーコードサービス。修正の必要があると考えられるエラーやシステムの異常な状況を知らせるメッセージをシステムコンソールオペレータに書いて送ることができます。\*

## X

X-array X-array (拡張可能な配列) は、ランタイムにオカレンス数を変更できる配列です。これにより、より効率的にメモリを管理することができます。例えば、大量の配列オカレンスを短期間だけ使用してから、アプリケーションが配列を使用しなくなった時点でメモリを削減することができます。詳細については、『プログラミングガイド』の「X-array」を参照してください。

XCF = システム間カップリング機能 IBM の z/OS オペレーティングシステムの機能の 1 つです。単一システムイメージを維持する一方で、多重システム環境をサポートするのに必要ないくつかの初期 MVS サービスを提供します。XCF を使用して結合されたシステムは、**Sysplex** と呼ばれています。\*

XML 「拡張マークアップ言語」の略です。

XML ツールキット *Windows* : 開発者が Natural で XML ドキュメントを処理できるようにします。ツールキットには、Natural ソースコードを生成し、次の機能を提供するウィザードが用意されています。

- Natural データ定義の DTD へのマッピング
- シリアライズ
- DTD を Natural データ定義にマッピング
- XML ファイルの解析およびその内容の Natural データ構造への割り当て

XRef Evaluation *Windows* : XRef Evaluation プラグインは、開発サーバーファイルで CAT または STOW コマンドを実行中に作成されたクロスリファレンス情報をナビゲートするときを使用します。この情報は Natural スタジオにツリー表示されます。参照する側 (アクティブ) と参照される側 (パッシブ) の両方のタイプのリファレンスが表示されます。ツリー表示内で、どちらのリファレンスにも移動することができます。Single Point of Development ドキュメントを参照してください。

XRef データ (アクティブクロスリファレンス) *Predict* : 実装されたオブジェクトの XRef データは、Predict に、ドキュメンテーションオブジェクトから独立して格納されます。Predict のアクティブ検索機能では、Xref データや Predict ドキュメンテーションオブジェクトから情報を検索して、以下の点が決定されます。

- ディクショナリ内のドキュメントされたオブジェクトがまだ実装されていないかどうか。
- 実装されたメンバがまだドキュメントされていないかどうか。
- ドキュメンテーションデータと実装が異なるかどうか。

Natural オブジェクトがカタログされる時、Natural オブジェクトの Xref データレコードが Natural によって作成されます。プログラミングオブジェクトがカタログされ、XREF パラメータがこれに応じて指定されている場合にも、XRef データが生成されます。

## Z

z/VM = 仮想マシン/エンタープライズシステムアーキテクチャ

IBM がライセンスを持つプログラム。複数のコンピューティングシステムが存在するような形で、単一のコンピュータのリソースを管理します。各仮想マシンは実際のマシンと同等に機能します。  
\*

Zap

実行形式モジュールの指定アドレスの内容を変更するための、プログラムエラー修正。IBM の用語集では、パッチと呼ばれています。

## あ行

アクティブ検索

*Predict* : アクティブ検索は、XRef データおよび Predict ドキュメンテーションオブジェクトから情報を検索する機能です。ドキュメンテーションとシステム実装との比較を行うことができます。

アクティブヘルプ

*Predict* : 機能を実行するのに必要なパラメータ値が設定されていない場合、アクティブヘルプとして、すべての設定可能値が含まれた選択ウィンドウが表示されます。この機能は、アスタリスクが付いたすべての入力フィールドに適用されます。

アクティブリファレンスデータ

*Predict* : 「**XRef データ**」を参照してください。

アクティベーションポリシー

DCOM クラスとして登録される NaturalX クラスの属性。Natural セッションを他のクラスと共有するかどうかを定義します。

NaturalX では、**DCOM** でサポートされているさまざまなオプションを、次の 3 つのアクティベーションポリシーの形式で組み合わせます。

- ExternalMultiple
- ExternalSingle

## ■ InternalMultiple

クラスのアクティベーションポリシーを設定するには、REGISTER コマンドの一部として指定するか、DEFINE CLASS ステートメントで指定するか、プロファイルパラメータ ACTPOLICY を使用します。

### アスタリスク表記

*Predict* : ワイルドカード文字 "\*" (アスタリスク) を使用して、いくつかのグループの値を扱う入力パラメータを指定することができます。例えば、ID として「datab\*」と入力すると、「datab」で始まるすべての ID が対象となります。

### アソシエーション

*Predict* : Predict ドキュメンテーションオブジェクトは、アソシエーションを使用してリンクすることができます。タイプの異なるオブジェクトをリンクするには、異なるタイプのアソシエーションを使用します。アソシエーションは単方向（親子アソシエーション）です。

*Predict* のメタデータ管理機能を使用して、データディクショナリ管理者は追加のアソシエーションタイプを定義できます。

### アドレススペース

Natural が稼動している領域。

### アブストラクト

*Predict* : Predict の中の各オブジェクトは、オブジェクトに関する短いコメントを提供するアブストラクトを持つことができます。アブストラクトには、30文字までの行を最大16行含めることができます。Predictの以前のバージョンでは「コメント」または「短いコメント」と呼ばれていましたが、他の Software AG 製品との互換性の理由で "アブストラクト" に変更されました。

### アプリケーション

相互接続されるプログラミング要素を論理的観点で集めたもの。これらの要素がまとまって、特定のビジネス問題に対するビジネスロジックをカバーする機能単位を形成します。アプリケーションは、一連のライブラリ、Natural オブジェクト、および（または）サブアプリケーション（ビジネスオブジェクト）で構成されます。ライブラリの内容（Natural オブジェクトやリソースなど）は、複数の異なるアプリケーションに属している場合があります。アプリケーションに関する情報（アプリケーション記述）は、すべてのプラットフォームからアクセスできる開発サーバーファイルに保管されます。

アプリケーションには、次の2つの種類があります。

#### ■ 基本アプリケーション

基本アプリケーションは、1つのプラットフォーム上の1つのアプリケーションで構成されます。例えば、劇場チケットの予約システムなどが挙げられます。



	<ul style="list-style-type: none"> <li>■ 複合アプリケーション 複合アプリケーションは、複数の基本アプリケーションで構成されます。例えば、チケット予約、お客様への請求、上演情報の郵送などのアプリケーションが組み合わさった劇場管理アプリケーションが挙げられます。基本アプリケーションは複数のプラットフォームに置くことができます。</li> </ul>
アプリケーションウィザード	<i>Windows</i> : 特定のアプリケーションに属する定義を行うためのツール。ユーザーは、表示されるダイアログボックスに従うことで、既存のアプリケーションに対するアプリケーション記述を作成できます。
アプリケーション記述	1つのアプリケーションに属するすべての要素のリスト。開発サーバーファイルに格納されています。アプリケーション情報が物理的に格納されているのは、データベースファイル（メインフレームの現在のFDIC）です。「 <a href="#">開発サーバーファイル</a> 」を参照してください。
アプリケーション記述層	アプリケーション記述へのアクセスを操作するコード。
アプリケーションプログラミングインターフェイス (API)	ユーザーのデータへのアクセスやデータ修正、あるいは、Natural、サブコンポーネントまたはサブプロダクトに固有のサービスの使用を可能にするプログラミングオブジェクト。アプリケーションプログラミングインターフェイスの目的は、Naturalステートメントではアクセス不可能な情報の検索や修正、またはサービスを使用することにあります。アプリケーションプログラミングインターフェイスは、CALLNATステートメントまたはCALLステートメントを使用してNaturalプログラミングオブジェクトから呼び出すことができます（各アプリケーションインターフェイスによって異なります）。
アプリケーションワークスペース	<i>Windows</i> : アプリケーションをツリー形式でビジュアル表示するのに使用します。アプリケーションワークスペースには、現在のアプリケーションに属するすべての要素が表示されます。
アンロード	<p><i>Predict</i> : Predict オブジェクトをアンロードするには、次の2つの方法があります。</p> <ul style="list-style-type: none"> <li>■ 抽出を使用します。メニュー機能で、1つの抽出の中に含まれたオブジェクトだけをアンロードすることができます。1回のアンロード操作で最大10個の抽出を指定できます。</li> <li>■ アンロードのコマンドを使用します。コマンド行に個々のオブジェクトを指定するか、オブジェクトの範囲を指定します。この機能の範囲はパラメータによって制限できます。</li> </ul>

- イベントドリブンプログラミング** *Windows* : イベントドリブンプログラミングでは、グラフィカルユーザーインターフェイスから受け取った入力データによってアプリケーションを起動できます。したがって、コードの実行順序は発生するイベントに依存し、次にユーザーの操作内容に依存します。『プログラミングガイド』の「イベントドリブンプログラミングについて」を参照してください。
- インスタンス** オブジェクト指向プログラミングモデルでは、データ構造と関数（「メソッド」）がオブジェクト内に一緒に納められます。各オブジェクトは、オブジェクトの内部構造およびそのインターフェイス、プロパティ、メソッドを記述するクラスに所属します。特定のクラスに属するオブジェクトは、そのクラスのインスタンスとも呼ばれます。
- インターフェイス** インターフェイスは、クライアントにサービスを提供するために、クラスによって使用されます。インターフェイスは、メソッドとプロパティのコレクションです。クライアントは、クラスのオブジェクトを作成し、そのインターフェイスのメソッドとプロパティを使用することによって、これらのサービスにアクセスします。
- インターフェイスは次のように定義します。
- INTERFACE 節を定義して、インターフェイス名を指定します。
  - インターフェイスのプロパティを PROPERTY 定義で定義します。
  - インターフェイスのメソッドを METHOD 定義で定義します。
- インターフェイス GUID** NaturalX クラスを **DCOM** クラスとして登録する場合、ネットワーク上でインターフェイスを明確に識別できるグローバルユニーク ID (GUID) を各インターフェイスに対して定義する必要があります。GUID をインターフェイスに割り当てるには、INTERFACE ステートメントの ID 節で指定します。Natural では、GUID は英数字定数で表され、これはデータエリアエディタで生成できます。Natural スタジオクラスビルダは、自動的に新しい GUID を個々の新しいインターフェイスに割り当てます。
- インターフェイスの継承** インターフェイスの継承とは、異なるクラスに同じインターフェイスを異なる方法で実装することです。これにより、このインターフェイスのみに依存するクライアントプログラムの作成が可能になります。このようなプログラムでは、このインターフェイスを持つクラスをどれでも操作できます。
- インターフェイスモジュール** インターフェイスを定義する Natural コピーコードモジュール。インターフェイスモジュールは、含まれるインターフェイスを定義するためにクラス内で使用できます。クラスはメソッドとプロパティ実装を書き換えることができますが、インターフェイスの



他のすべての設定はインターフェイスモジュールの定義どおりに使用されます。

- インポート *Predict* : コーディネータの Import 機能は、データを送信媒体から *Predict* 環境にインポートする場合に使用します。テキストをさまざまな外部ソース (Natural、PC ASCII ファイル、Connect ドキュメントなど) から *Predict* オブジェクトのテキスト属性 (任意の *Predict* オブジェクトの拡張記述、ファイルのサブクエリなど) にインポートすることもできます。
- エクスポート *Predict* : コーディネータの Export 機能は、*Predict* データを他のシステムとデータ交換できる形式でエクスポートする場合に使用します。テキストを *Predict* オブジェクトの属性 (拡張記述、ファイルのサブクエリなど) または表示指向の *Predict* 機能 (検索、アクティブ検索など) の出力から外部出力先 (Natural、Connect ドキュメント、PC ASCII ファイルなど) にエクスポートすることもできます。
- エレメンタリフィールド *Adabas* : 複数の値ではなく単一の値 (スカラ) を持つフィールド。「マルチプルバリューフィールド」または「ピリオディックグループ」を参照してください。  
*Predict* : 「フィールド」を参照してください。 *Predict* の以前のバージョンでは、フィールドタイプの *Predict* オブジェクトが「エレメンタリフィールド」と呼ばれていました。
- エンディアンモード *Windows/UNIX/OpenVMS* : コンパイラが GP を生成するときのアーキテクチャのタイプ。次の 2 つのモードがあります。  
ビッグエンディアン  
数値の上位バイトがメモリの最下位アドレスに格納され、下位バイトが最上位アドレスに格納されます (大きな桁が最初に来ます)。  
リトルエンディアン  
数値の下位バイトがメモリの最下位アドレスに格納され、上位バイトが最上位アドレスに格納されます (小さな桁が最初に来ます)。  
ENDIAN セッション/プロファイルパラメータによって決まります。ENDIAN パラメータは、ポータブル GP の実行パフォーマンスを向上させる場合に使用します。
- エントリ順データセット (ESDS) 「ESDS」を参照してください。

- オートメーションオブジェクト *Windows* : オートメーションインターフェイス経由で使用できるプログラム。詳細については、Microsoftのドキュメントを参照してください。
- オーナー *Predict* : オーナーとは、基本的に1人または複数のユーザーのグループを指し、例えば組織上の単位を表す場合もあります。オーナーをオーナーユーザーリストとオブジェクトオーナーリストに割り当てることによって、*Predict* で責任者をドキュメントすることができます。
- オブジェクト *オブジェクト指向プログラミング* : オブジェクト指向プログラミングモデルでは、データ構造と関数（「メソッド」）がオブジェクトと一緒に納められます。各オブジェクトは、オブジェクトの内部構造およびそのインターフェイス、プロパティ、メソッドを記述するクラスに所属します。
- Predict* : *Predict* の情報は、データディクショナリオブジェクト、ドキュメンテーションオブジェクト、*Predict* オブジェクトの形式で格納されます。データディクショナリオブジェクトは、*Predict* オブジェクトまたはドキュメンテーションオブジェクトとも呼ばれます。「[オブジェクトタイプ](#)」も参照してください。
- Natural* : 「[カタログ化オブジェクト](#)」、「[ソースオブジェクト](#)」、および「[オブジェクトタイプ](#)」も参照してください。
- オブジェクトタイプ *Natural* : *Natural* オブジェクトタイプの例として次のものが挙げられます。プログラム、マップ、コピーコード、テキスト、サブプログラム、ヘルプルーチン、サブルーチン、クラス、データエリア（グローバル、ローカル、パラメータ）、ダイアログ、ファンクション。詳細については、『[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。
- Predict* : *Predict* の情報は、データディクショナリオブジェクト、ドキュメンテーションオブジェクト、*Predict* オブジェクトの形式で格納されます。データディクショナリオブジェクトは、*Predict* オブジェクトまたはドキュメンテーションオブジェクトとも呼ばれます。*Predict* ドキュメンテーションオブジェクトのタイプとしては、データベースやプログラムなどがあります。*Predict* のメタデータ管理機能を使用して、その他のオブジェクトタイプを定義することができます。オブジェクトタイプには、それぞれのタイプ固有の属性があります。
- オブジェクトタイプコード *Predict* : *Predict* メタ構造内のオブジェクトのタイプは、一意のオブジェクトタイプコードによって識別されます。オブジェクトタイプコードは2文字で構成されています（例えば、ファイルの場合はFI）。

オブジェクトデータエリア (ODA)	オブジェクトデータエリアには、オブジェクトの内部状態のメンテナンス用にオブジェクトの全プロパティの現在値が保存されます。1つのクラスのすべてのオブジェクトのオブジェクトデータエリアの構造は、DEFINE CLASS ステートメントの OBJECT USING 節で指定されます。オブジェクトデータエリアは、データエリアエディタでローカルデータエリアとして作成されます。
オブジェクトデータ変数	オブジェクト指向プログラミング：クラスのオブジェクトデータエリアの各プロパティには、プロパティ値を格納するために変数が必要になります。この変数をオブジェクトデータ変数と言います。オブジェクトデータエリアでは他の変数も定義できますが、それらはクライアントからプロパティとしてアクセスすることはできず、それらのメソッドでのみ使用されます。
オブジェクトのロック	<i>Windows</i> ：リモート開発環境でのプログラムの同時更新を防ぎます。詳細については、『SPoDを使用したリモート開発』ドキュメントの「オブジェクトのロック」を参照してください。
オペランド	次のいずれかがオペランドとなります。 <ul style="list-style-type: none"><li>■ 定数</li><li>■ ユーザー定義変数またはユーザー定義関数</li><li>■ ビューフィールド</li><li>■ システム変数またはシステム関数</li></ul> オペランドは Natural ステートメントと連携して使用します。詳細については、『ステートメント』ドキュメントの「オペランド定義テーブル」を参照してください。
親	<i>Predict</i> ：アソシエーションは、Predict オブジェクトと他の Predict オブジェクトとの関係をドキュメントするために使用されます。アソシエーションは、子または親としてオブジェクトをオブジェクトにリンクすることにより確立されます。例えば、データベースはファイルの親であり、ファイルはフィールドの親です。

## か行

概念ファイル	<i>Predict</i> ：アプリケーション開発の初期の段階で、設計目的のために Conceptual タイプのファイルオブジェクトが使用されます。
開発サーバーファイル	アプリケーション記述が物理的に格納されているデータベースファイル（メインフレーム上の現在の FDIC にある）。「 <a href="#">アプリケーション記述</a> 」を参照してください。

外部インターフェイス	<i>Windows</i> : 外部インターフェイスとは、インターフェイスモジュールで定義されるインターフェイスのことで、クラス別に含まれています。
外部オブジェクト	<i>Predict</i> : <i>Predict</i> でドキュメンテーションオブジェクトを使用してドキュメントされたアプリケーション (データ処理システム) のオブジェクト。 <i>Generation</i> 機能を使用すると、ドキュメンテーションオブジェクトから外部オブジェクトを作成できます。また、 <i>Incorporate</i> 機能を使用すると、外部オブジェクトからドキュメンテーションオブジェクトを作成できます。 <i>Comparison</i> 機能を使用すると、外部オブジェクトとドキュメンテーションオブジェクトを比較できます。
外部クラス	<i>NaturalX</i> クラスのタイプは、ローカル、内部、または外部です。これは、クラスが登録された方法によって決まります。外部クラスは、 <i>REGISTER</i> コマンドオプション <i>ES</i> ( <i>ExternalSingle</i> ) または <i>EM</i> ( <i>ExternalMultiple</i> ) を使用して、 <b>DCOM</b> クラスとして登録されたクラスです。外部クラスのオブジェクトは、他のプロセスによって作成およびアクセスされる場合があります。  詳細については、「 <a href="#">ローカルクラス</a> 」および「 <a href="#">内部クラス</a> 」を参照してください。
仮想マシン	<i>Predict</i> : 仮想マシンタイプの <i>Predict</i> オブジェクトは、ネットワークタイプのオブジェクトとともに、データ処理システムのハードウェアおよびオペレーティングシステム環境をドキュメントします。
カタログ化オブジェクト	<i>Natural</i> では、カタログ化オブジェクトとは <i>Natural</i> 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、オブジェクトモジュールとして <i>Natural</i> システムファイルに格納されます。  「 <a href="#">ソースオブジェクト</a> 」、「 <a href="#">オブジェクト</a> 」、および「 <a href="#">オブジェクトタイプ</a> 」も参照してください。
カレントオブジェクト	<i>Predict</i> : ある種の検索 (例: 子を持つファイル) では、指定された選択条件に合うオブジェクトの属性と、これらのオブジェクトに関係付けられたオブジェクトの属性の両方がレポートされます。出力オプションを指定している場合、指定された選択条件に合うオブジェクトがカレントオブジェクトと呼ばれます。
カレントな仮想マシン / カレントなネットワーク	<i>Predict</i> : ネットワークオブジェクトや仮想マシンオブジェクトは、情報処理システムのハードウェアおよびオペレーティングシステム環境をドキュメントします。 <i>Predict</i> は、カレントな仮想マシンおよびカレントなネットワークを保存して、次のような形で使用します。

- データベースオブジェクトおよび仮想マシンオブジェクトのデフォルトの親として使用します。
- Generate/Incorporate/Compare 機能、および Predict の特殊機能メニューから呼び出される一部の AOS 機能のターゲット環境として使用します。

カレントな仮想マシンおよびカレントなネットワークは、Predict のデフォルトで指定されます。

## 関係

*Predict* : 「[ファイル関係](#)」を参照してください。

## 関連オブジェクト

*Predict* : 親子オブジェクトに対する総称的な用語。

## キーワード

*Natural* : 『プログラミングガイド』の「*Natural* 予約キーワード」を参照してください。

*Predict* : データディクショナリオブジェクトを参照する追加の手段として、キーワードタイプのオブジェクトが使用されます。どのオブジェクトにも、最大 32 個のキーワードを割り当てることができます。

## 基本アプリケーション

「[アプリケーション](#)」を参照してください。

## キャッシュ

メインメモリから最後に検索されたデータのコピーが一時的に保管されるのに使用されるバッファ領域。これによって、システムファイルおよび外部データベース/ファイルシステムへの不必要なアクセスを避け、CPU タイムを削減してレスポンスタイムを向上させることができます。

## 共有ニュークリアス

z/OS および VSE/ESA : *Natural* ニュークリアスの環境に依存しない部分。これがインストールされることで、ニュークリアスの 1 つのコピーを複数の TP 依存 *Natural* ドライバが使用できるようになり、バッチモードでの使用も可能になります。

## 組み込み

*Predict* : *Predict* データディクショナリオブジェクトは、外部オブジェクト（既存の *Natural* データ定義モジュール、Adabas フィールド定義テーブルなど）を組み込むことによって作成できます。

## クライアントスタブ

「[RPC スタブ](#)」を参照してください。

## クラス

*Natural* クラスは、データ構造（オブジェクト）を対応する機能（メソッド）とともにカプセル化します。「[NaturalX クラス](#)」も参照してください。

クラスオブジェクトのオブジェクトの内部構造はデータエリア（オブジェクトデータエリア）で定義されます。クラスのメソッドはサブプログラムとして実装されます。

クラス GUID	NaturalX クラスを <b>DCOM</b> クラスとして登録する場合は、ネットワーク上でクラスを明確に識別できるグローバルユニーク ID (GUID) をクラスに定義する必要があります。Natural で GUID をクラスに割り当てるには、DEFINE CLASS ステートメントの ID 節で指定します。GUID は、データエリアエディタで生成できる英数字定数で表されます。Natural スタジオクラスビルダは、自動的に新しい GUID を個々の新しいクラスに割り当てます。
クラスタ	VSAM : VSAM に定義されるデータセット。クラスタは、キー順データセット ( <b>KSDS</b> )、エントリ順データセット ( <b>ESDS</b> )、または相対レコードデータセット ( <b>RRDS</b> ) のいずれかです。*
クラスビルダ	Windows : NaturalX および DCOM コンポーネントクラスの定義および実装を、Natural クラスとしてサポートします。Natural クラスは、Natural オブジェクト (例えばオブジェクトデータ領域) やクラスソースだけに存在するオブジェクト (例えばインターフェイスコンポーネント) などのさまざまなコンポーネントで構成することができます。クラスビルダは、クラスの各コンポーネントをノードの形で表します。ノードを選択することによって、クラスおよびそのコンポーネントを文脈に依存した方法で管理することができます。『エディタ』ドキュメントの「クラスビルダ」を参照してください。
クラス名	DEFINE CLASS ステートメントの <i>class-name</i> オペランドで定義される名前。クラスのオブジェクトを作成する CREATE OBJECT ステートメントで、この名前が使用されます。『ステートメント』ドキュメントの「DEFINE CLASS」も参照してください。
クラスモジュール名	Natural クラスが定義される Natural モジュールの名前。
グローバルパラメータ	「 <a href="#">セッションパラメータ</a> 」を参照してください。
クロスリファレンスリスト	<i>Predict</i> : 検索メニューまたはアクティブ検索メニューの Predict 関数 X によって作成されるリスト。特定のアソシエーションによって所定のオブジェクトのリンク先となっているすべてのオブジェクトが表示されます。
結合フィールド	<i>Predict</i> : 標準ファイルまたはマスタファイルのフィールドからコピーされたフィールドを結合フィールドと呼びます。
検索	<i>Predict</i> : 情報 (オブジェクト、その属性およびアソシエーション) の選択された出力を端末に、バッチモードの場合はスプールに送るサブシステム。
検索処理	<i>Predict</i> : デクシヨナリから情報を検索するときには、検索タイプと出力モードを指定する必要があります。これらの設定の組み



---

	合わせによって、レポートヘッダーに示される検索操作のタイプが決定します。
検索タイプ	<i>Predict</i> : 検索タイプにより、ディクショナリから検索する <i>Predict</i> 情報のタイプが決定されます。検索タイプには、タイプに依存しないもの（子を持ったオブジェクトなど）と、タイプ固有のもの（ファイル間相違など）があります。
検索モデル	<i>Predict</i> : 検索モデルは、検索構造と、作成するレポートの内容およびレイアウトの定義という 2 つの部分で構成されます。
子	<i>Predict</i> : アソシエーションは、 <i>Predict</i> オブジェクトと他の <i>Predict</i> オブジェクトとの関係をドキュメントするために使用されます。アソシエーションは、子または親としてオブジェクトをオブジェクトにリンクすることにより確立されます。例えば、フィールドはファイルの子であり、ファイルはデータベースの子です。
コーディネータ FDIC	<i>Predict</i> : コーディネータのロード機能またはインポート機能を使用するときに、データを一時的に格納するために使用される <i>Predict</i> ファイル。このコーディネータ FDIC でコンフリクトが解決されていないと、ロード機能またはインポート機能を続行できません。機能が正常に実行されると、コーディネータ FDIC のデータは削除されます。
コーディネータチェックサイクル	<i>Predict</i> : オブジェクトは、 <i>Predict Coordinator</i> を使用してロードまたはインポートされる場合、コンフリクト管理、セキュリティチェック、整合性チェックという 3 つの段階から成るチェックサイクルを通過する必要があります。いずれかの段階のコンフリクトがメイン FDIC またはコーディネータ FDIC で解決されていない場合、ロード機能またはインポート機能を続行できません。
コード単位	処理または交換用にエンコードされたテキストの単位を表す最小のビット組み合わせ。
コードページ	「 <a href="#">Unicode</a> 」も参照してください。
コードポイント	Unicode コードスペースの値 (U+000000~U+10FFFF)。
コンパイラ	Natural コンパイラは、ソースプログラムを中間コードに翻訳します。「 <a href="#">Natural Optimizer Compiler</a> 」も参照してください。
コンフィグレーションファイル	<i>Windows/UNIX/OpenVMS</i> : 以下のコンフィグレーションファイルを調べれば、FNAT/FUSER の基本ディレクトリがわかります。

- ローカルコンフィグレーションファイル - **NATURAL.INI**  
ローカルコンフィグレーションファイルには、バッファプール割り当てとインストール指定が含まれています。このファイルは、バージョンごとの **Natural** インストールディレクトリの *etc* ディレクトリ内にあります。
- グローバルコンフィグレーションファイル - **NATCONF.CFG**  
グローバルコンフィグレーションファイルには、DBMS 割り当て、ディクショナリサーバー割り当て、プリンタプロファイルの割り当て、およびシステムファイル割り当てが含まれています。

詳細については、『コンフィグレーションユーティリティ』ドキュメントの「コンフィグレーションファイルのパラメータの概要」を参照してください。

## コンフリクト管理

*Predict* : コーディネータチェックサイクルの第1段階です。Predict バージョン 3.3 以上では、すべてのオブジェクトが内部 ID を持っています。コンフリクト管理とは、こうした内部 ID から生じるコンフリクトを解決することを指します。例えば、オブジェクト ID が同一の 2 つのオブジェクトで内部 ID が異なるといったコンフリクトが考えられます。

内部 ID から生じるコンフリクトがコーディネータ FDIC で解決されていなければ、データをメイン FDIC にロードまたはインポートすることはできません。

## コンポーネントブラウザ

*Windows* : **NaturalX** アプリケーション開発に使用できる ActiveX コンポーネントを表示するのに使用します。以下の機能から構成されています。

- 使用可能な ActiveX コンポーネントとそれらのディスパッチおよびデュアルインターフェイスがリストされます。
- データタイプが **Natural** フォーマットにマップされます。
- 外部コンポーネントのヘルプファイルに直接アクセスできます。
- **Natural** プログラミング例が自動的に生成されます。

## さ行

### サーバー

*Predict* : サーバータイプの Predict オブジェクトは、ノードタイプのオブジェクトとともに、リモートプロシージャコールをドキュメントするために使用されます。

### サーバー ID

サーバー ID は、**NaturalX** サーバーを識別する文字列です。**Windows** システムレジストリ内の **Natural** が所有するキーであ



	り、所定の NaturalX サーバーに属するすべてのクラスがまとめられています。これは 32 文字の任意の英数字文字列で、空白を含まず、大文字と小文字は区別されません。サーバー ID は Natural パラメータ COMSERVERID= <i>serverid</i> で定義されます。
サーバータスク	<i>Natural RPC</i> : サーバータスクとは、サービス (サブプログラム) を提供する Natural タスクのことです。これは、通常、バッチタスクまたは非同期タスクです。これはサーバー名で識別されます。
サーバーフロントエンド	<i>z/OS</i> : Natural DB2 ストアドプロシージャサーバーと Natural 開発サーバーによって使用される Natural サーバー環境のコンポーネント。サーバーフロントエンドは、サーバー環境の初期化、セッションのロールイン/ロールアウト、Natural オブジェクトの実行などの追加機能をフロントエンドサーバーに付与する、機能拡張されたバッチドライバ (NATOS、NATTSO) です。
サーバー名	<i>Natural RPC</i> : CALLNAT が実行されるサーバーの名前。  EntireX Broker では、サーバー名は EntireX Broker 属性ファイルの SERVER フィールドに定義された名前です。
サービスディレクトリ	<i>Natural RPC</i> : サーバーが提供するサービス (サブプログラム) に関する情報を含みます。サービスディレクトリは、各クライアントノードからローカルでアクセスできるか、またはリモートディレクトリサーバーに配置されている場合は、RDS セッションパラメータで参照できます。サービスディレクトリを実装する SYSRPC ユーティリティで生成されます。  サービスディレクトリは、SYSRPC ユーティリティによって生成される Natural サブプログラム NATCLTGS によって実装されます。
サブタイプ	<i>Predict</i> : オブジェクトタイプはサブタイプを持つことができます。オブジェクトタイプファイルのサブタイプとして、例えば Adabas ファイル、Adabas ユーザービュー、DB2 テーブル、DB2 ビューなどがあります。オブジェクトタイプファイルのサブタイプは、ファイルタイプとも呼ばれます。
システム	<i>Predict</i> : システムタイプの Predict オブジェクトは、アプリケーションまたはその一部を形成するプログラムをドキュメントします。
システム関数	Natural によって提供されるプログラム済みの関数。『システム関数』ドキュメントを参照してください。
システムコマンド	Natural システムコマンドは、Natural プログラミング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。
システムファイル	Natural システムファイル (FNAT、FUSER) には、Natural システムが機能するために必要な情報、データ、プログラム、モジュールなどが含まれます。Predict、Natural Security、Natural Advanced Facilities、Natural for VSAM などの製品では、それぞれ固有のシステムファイル (FDIC、FSEC、FSPOOL、FDICX) が必要とされます。
システムプログラム	<i>Predict</i> : ソースコードとして利用できないプログラムは、subtye E (外部オブジェクト) および language Z (システムプログラム) のプログラムタイプの Predict オブジェクトによってドキュメントされます。Predict でシステムプログラムを作成すると必ず、そのシステムプログラムの Xref データが作成されます。
システム変数	システム変数は、システム情報を表示する場合に使用します。これらは、Natural プラグラム内のどこからでも参照できます。『システム変数』ドキュメントを参照してください。
システムライブラリ	インストール時に FNAT 内に作成される Natural の内部ライブラリ。ユーザーが修正することはできません。Natural システムライブラリの名前はすべて、"SYS" で始まります (メインフレームではいくつかの例外があります)。
自動更新	<i>Predict</i> : 上位階層レベルにあるフィールドが修正されたときに、そのフィールドから派生する下位階層レベルのすべてのフィールドが自動更新されること。  階層レベルは次のようになっています。 <ul style="list-style-type: none"><li>■ 標準ファイル</li><li>■ マスタファイル</li><li>■ ユーザービュー</li></ul>
出力モード	<i>Predict</i> : 出力モードは、検索タイプとともに、Predict 検索機能によるディクショナリデータの評価方法を決定します。あらゆる検索タイプに対してすべての出力モードを使えるわけではないことに注意してください。
ジョブ	ユーザーによって定義された、コンピュータが達成すべき作業単位。ジョブという用語は、曖昧にジョブの代表的なものを指して使われることがあります。こうした代表的なものに、コンピュータプログラム、ファイル、オペレーティングシステムへの制御文などがあります。*

処理ルール	マップフィールドに定義された処理プロシージャのことで、フィールドの内容をチェックし、内容に応じて対応します。複数のフィールドをチェック対象にすることもできます。処理ルールは、Predictにまとめて保存するか（フリールール）、 <b>DDM</b> フィールドに永久に割り当てる（自動ルール）ことができます。
スタティックパラメータ	<p>メインフレーム：Naturalパラメータモジュール（NATPARM）のNTPRM マクロおよび他のマクロで指定されたプロファイルパラメータによって割り当てられたもの。このパラメータモジュールは、その後NaturalニュークリアスでアSEMBルされてリンクされます。指定されないパラメータには、すべてデフォルト値が割り当てられます。</p> <p>Windows/UNIX/OpenVMS：NATPARM.SAGで指定されたパラメータ設定。</p>
スタブ	ローカルアプリケーションプログラムとブローカの間にリンクを設定するソケットプログラム。「 <b>RPCスタブ</b> 」も参照してください。
ステートメント	『ステートメント』ドキュメントを参照してください。
ストレージスペース	<i>Predict</i> ：ストレージスペースタイプのPredictオブジェクトは、DB2ストレージグループをドキュメントします。
スレッド	<p>メインフレーム：特定のNaturalセッションのすべてのデータが含まれるストレージスペース。</p> <p>Windows/UNIX/OpenVMS：オペレーティングシステムによるCPU時間の割り当て対象となる、アプリケーション内の基本（メモリ）エンティティ。スレッドが存在することで、シングルおよびマルチプロセッサでの並行コンピューティングが可能になります。スレッドはそれぞれ独立して実行されます。</p>
制限	<i>Predict</i> ：オブジェクトに関する情報を検索するときに選択条件として使用する、キーワード、オーナー、テキスト文字列の組み合わせ。制限は、プロファイルに保存することも、一時的な使用の目的で指定することもできます。
整合性チェック	<i>Predict</i> ：コーディネータチェックサイクルの第3段階です。ファイル番号がデータベース内でユニークであるかなど、送信されるオブジェクトの論理的整合性がチェックされます。メインフレーム機能と同様のチェックが実行されます。
整合性チェック	<i>Predict</i> ：整合性チェックタイプのPredictオブジェクトは、フィールド値の確認を行うためのルールをドキュメントします。整合性

チェックタイプの Predict オブジェクトからは、Natural 処理ルールを生成することができます。

**生成**

*Predict* : 生成機能を使用すると、ファイルタイプの Predict オブジェクトから外部データ定義オブジェクトを生成することができます。

**生成タスク**

*Predict* : 実装計画にいくつかの生成タスクを含めると、データ定義オブジェクトの生成を快適に実行することができます。生成タスクとは、Predict 生成機能呼び出すタスクのことです。

**生成プログラム (GP)**

コンパイラによって生成される実行形式のプログラムオブジェクト。

メインフレーム : メインフレームプラットフォーム上でのみ実行可能なプログラム。メインフレームでは、**Natural Optimizer Compiler** を使用して生成プログラムを最適化することもできます。

Windows/UNIX/OpenVMS : Natural 5.1 for Windows で導入された **ポータブル GP** は、Natural によってサポートされているすべての Windows、UNIX、および OpenVMS プラットフォーム上で実行することができます (メインフレームでは実行できません)。必要な場合は、オペレーティングシステム依存 GP を生成することもできます。

*同義語* : GP の同義語として、「オブジェクト」という用語がしばしば使われます。Predict Application Control では、「ロード形式」という用語が使われます。

**セッション**

*Natural* : Natural セッションは、ユーザー依存の Natural ランタイムコンテキストです。Natural ランタイムシステムで Natural プログラムを実行する際に必要です。

**セッションパラメータ**

現在の Natural セッション内で、システムコマンド GLOBALS (または SET GLOBALS ステートメント) を使用して割り当てられ、スタティック/ダイナミック割り当てを書き換えます。

Natural セッションパラメータは、レポートサイズなどの係数やフィールド表示方法を制御するために、特定の Natural ステートメントで使用されます。

これらのパラメータは、Natural のインストール時に Natural 管理者によって Natural の全ユーザーに有効なデフォルト値に設定されます。

---

	<p>詳細については、『パラメータリファレンス』を参照してください。</p> <p>セッションに適用されるパラメータ値を確認するには、システムコマンド GLOBALS を入力してください。</p>
セッション変数	<p><i>Predict</i>：多くの <i>Predict</i> 機能ではパラメータ値を指定する必要があります。 <i>Predict</i> では、パラメータ値は一時的にセッション変数内に格納されます。パラメータ値を省略しても、可能であればこのようなセッション変数からパラメータ値を取得できます。セッション変数から取得できる具体的なパラメータ値については、機能やパラメータタイプによって異なります。</p>
接続	<p><i>Predict</i>:外部オブジェクトとドキュメンテーションオブジェクトとの接続は、外部オブジェクトがドキュメンテーションオブジェクトから作成されたか、またはドキュメンテーションオブジェクトが外部オブジェクトから取り込まれたときに確立されます。接続された外部オブジェクトとドキュメンテーションオブジェクトのほとんどは、明示的に切断できます。</p> <p>接続は、ドキュメンテーションオブジェクトの属性である実装ポイントを使用して、<i>Predict</i> 側で実現されます。</p>
千桁単位セパレータ	<p>大きい整数値の出力を構築する場合、一般的には整数の3桁ごとにセパレータが挿入されて、千桁ごとに値が分割されます。このセパレータは、「千桁単位セパレータ」と呼ばれます。</p>
千桁単位セパレータ文字	<p>「千桁単位セパレータ文字」は、ランタイムに千桁単位セパレータとして使用される文字です。</p>
選択ウィンドウ	<p><i>Predict</i>：「<a href="#">アクティブヘルプ</a>」を参照してください。</p>
選択条件	<p><i>Predict</i>：<i>Predict</i> 検索機能は、個々のオブジェクトまたはオブジェクトグループに適用することが可能です。検索機能を適用するオブジェクトを選択するには、検索条件を指定します。</p>
送信媒体	<p><i>Predict</i>：<i>Predict Coordinator</i> を使用してデータを送信する場合、送信媒体として、PC ワークファイル、Natural ワークファイル、またはデータベース内のファイルを使用できます。コーディネータのアンロード機能やエクスポート機能を使用する場合、ターゲット環境が送信媒体です。ロード機能やインポート機能を使用する場合は、ソース環境が送信媒体です。</p>
ソースオブジェクト	<p>Naturalでは、ソースオブジェクト（保存オブジェクト）にNaturalソースコードが格納されます。ソースオブジェクト自体は、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。</p>

「カタログ化オブジェクト」、「オブジェクト」、および「オブジェクトタイプ」も参照してください。

ソースフィールド	<i>Predict</i> : 派生フィールドの定義で使用されるフィールド。派生フィールドとは、スーパー、ハイパー、サブフィールド、ディスクリプタ、およびフォネティックディスクリプタを指します。
属性	<i>Predict</i> : <i>Predict</i> ドキュメンテーションオブジェクトには、オブジェクトを記述する属性が含まれています。

## た行

ダイナミック千桁単位セパレータ	<i>Natural</i> では、「ダイナミック千桁単位セパレータ」は編集マスクの特殊文字です。これは、ランタイムで挿入される千桁単位セパレータの位置を示します。
ダイナミック定義	DEFINE DATA 節に含まれていない <i>Natural</i> 変数の定義。
ダイナミックパラメータ	<i>Natural</i> の開始時に個々のパラメータおよび/または代替パラメータファイルを指定することによって割り当てられます。現在の <i>Natural</i> セッションに対して有効です。
ダイナミック変数	DYNAMIC 属性を持つ変数を使用すると、開発時にスペース制限を定義せずに、 <i>Natural</i> で大きなバイナリデータ構造および英数字データ構造を処理できます。ダイナミック変数は長さを指定せずに定義されます。ダイナミック変数をターゲットオペランドとして明示的に使用するか、EXPAND または RESIZE ステートメントとともに黙示的に使用した場合、メモリが実行時に割り当てられます。ダイナミック変数は、DEFINE DATA ステートメント内でしか定義することはできません。『プログラミングガイド』の「ダイナミック変数およびフィールドについて」を参照してください。
タイプ情報	<i>NaturalX</i> クラスが <b>DCOM</b> クラスとして登録されている場合は、Windows システムレジストリエントリによって、クラスに対してタイプライブラリが生成され、クラスに接続されます。クライアントは、タイプライブラリに含まれているタイプ情報を使用して、コンパイル時またはランタイムに、クラスのインターフェイス、メソッド、およびプロパティの説明をチェックできます。
タイプライブラリ	<i>NaturalX</i> クラスが <b>DCOM</b> クラスとして登録されている場合は、Windows システムレジストリエントリによって、クラスに対してタイプライブラリが生成され、クラスに接続されます。クライアントは、タイプライブラリに含まれているタイプ情報を使用して、コンパイル時またはランタイムに、クラスのインターフェイス、メソッド、およびプロパティの説明をチェックできます。

タスク	メインフレーム：多重プログラミングまたは多重処理環境では、制御プログラムによってコンピュータが達成すべき1つの作業単位として扱われる、1つまたは複数の命令シーケンスを指します。 *  Windows/UNIX/OpenVMS：「プロセス」を参照してください。
単一マスタビュー	<i>Predict</i> ：Adabas ユーザービューなど、常に1つのマスタファイルだけから派生するビュー。
端末コマンド	セッションパラメータCFを使用すると、他の特殊文字を制御文字として定義することができます。『端末コマンド』ドキュメントを参照してください。
ツリービュー	要素の階層を視覚的に表示したもの。
定数	Natural では、次のタイプの定数がサポートされています。 <ul style="list-style-type: none"><li>■ 数値定数</li><li>■ 英数字定数</li><li>■ 日付／時刻の定数</li><li>■ 16進の定数</li><li>■ 論理定数</li><li>■ 浮動小数点定数</li><li>■ 属性定数</li><li>■ ハンドル定数</li></ul> 『プログラミングガイド』の「ユーザー定義定数」を参照してください。
データスペース	<i>Predict</i> ：データスペースタイプの <i>Predict</i> ドキュメンテーションオブジェクトは、DB2 テーブルスペースと SQL/DS DBspace をドキュメントします。
データ制御言語 (DCL)	「 <a href="#">DCL</a> 」を参照してください。
データ定義オブジェクト	<i>Predict</i> ：生成機能で作成された外部オブジェクト。Adabas 圧縮ユーティリティ、COBOL コピーコード、データ定義モジュール (DDM)、DB2 データベース、整合性ルールなどの定義。
データ定義言語	「 <a href="#">DDL</a> 」を参照してください。
データ定義モジュール (DDM)	「 <a href="#">DDM</a> 」を参照してください。



- データディクショナリオブジェクト *Predict* : 「[オブジェクト](#)」を参照してください。
- データビュー データビューは、[データ定義モジュール \(DDM\)](#) で定義されたデータベースフィールドを参照します。データビューには、DDMのフィールド定義のすべてまたは一部を表示できます。データビューは、DEFINE DATAステートメントまたはデータエリアで定義されます。
- データベース *Predict* : データベースタイプの *Predict* ドキュメンテーションオブジェクトは、一連の物理ファイルと論理ファイルをドキュメントします。データベースタイプのオブジェクトを形成するためのファイルの収集方法は、ドキュメントされるデータベースのタイプによって異なります。
- デバッグ 「[Natural デバッグ](#)」を参照してください。
- 独立データベース *Predict* : Adabas Star を使用してアクセスできないデータベース。
- トランザクション
1. 単一ユーザーによって行われるアプリケーションデータ入力。ここから一連の情報交換および操作手順の実行が開始されます (入力 - データ処理 - 出力)。プロセスを開始するにはトランザクションコードが必要です。
  2. Natural では、データベースの更新処理はトランザクション (データベーストランザクション) に基づいて実行されます。つまり、すべてのデータベース更新要求は論理トランザクション単位で処理されます。論理トランザクションとは、データベースに含まれる情報の論理的整合性を確実なものにするためには完全に実行する必要のある最小作業単位のことです。ユーザーによって定義されます。ステートメント END TRANSACTION (**ET**) と BACKOUT TRANSACTION (**BT**) を使用すると、トランザクションを開始してから以降のデータベースの全レコードがロック解除されます。『プログラミングガイド』のデータベースアクセスに関するセクションも参照してください。

## な行

---

- 内部 ID *Predict* : *Predict* バージョン 3.3 から、すべてのオブジェクトは内部 ID を持っています。この ID は、オブジェクトが追加されたときに自動的に割り当てられます。内部 ID は全体を通して一意であり、そのオブジェクトが追加されて以降ずっと変わることはありません。

内部インターフェイス	クラスで直接定義されるインターフェイス、またはインターフェイスモジュールで定義されるインターフェイスモジュールのインターフェイス。
内部クラス	<p>NaturalX クラスのタイプは、ローカル、内部、または外部です。これは、クラスが登録された方法によって決まります。内部クラスは、REGISTER コマンドオプション IM (InternalMultiple) を使用して、<b>DCOM</b> クラスとして登録されたクラスです。内部クラスのオブジェクトを他のプロセスから作成することはできませんが、アクセスすることはできます。そのためには、例えばメソッドの戻り値などの形で、オブジェクトがクライアントプロセスに渡されている必要があります。</p> <p>詳細については、「<a href="#">ローカルクラス</a>」および「<a href="#">外部クラス</a>」を参照してください。</p>
内部コード	<p><i>Predict</i> : 各 <i>Predict</i> オブジェクトタイプは、内部コードによって内部的に識別されます。内部コードは <i>Predict</i> によって割り当てられ、ユーザーが変更することはできません。</p>
ニュークリアス	<p>Natural カーネルを構成するサービスプログラム（メモリ管理、文字列操作、オペレーティングシステムインターフェイス、<a href="#">コンパイラ</a>、<a href="#">ランタイム環境</a>など）の集合体。</p> <p><i>メインフレーム</i> : Natural ニュークリアスは2つの不可欠の部分から構成されています。1つは、オペレーティングシステムや TP モニタシステムから独立して実行される部分、もう1つは、これらに関連して実行される部分です。どちらの部分も、1つのニュークリアスを形成するために一緒にリンクすることができます。「<a href="#">共有ニュークリアス</a>」も参照してください。</p> <p><i>Windows/UNIX/OpenVMS</i> : ニュークリアスは、オペレーティングシステムに依存する Natural の一部です。</p>
ネットワーク	<p><i>Predict</i> : ネットワークは、仮想マシンタイプの <i>Predict</i> オブジェクトと連携して、データ処理システムのハードウェアおよびオペレーティングシステム環境をドキュメントします。</p>
ノード	<p><i>Predict</i> : ノードタイプの <i>Predict</i> オブジェクトは、サーバータイプのオブジェクトとともに、リモートプロシージャコールをドキュメントするのに使用されます。</p>
ノード名	<p><i>Natural RPC</i> : リモート CALLNAT の送信先ノードの名前。EntireX Broker では、ノード名は、EntireX Broker 属性ファイルの BROKER-ID フィールドに定義された EntireX Broker 名です。「<a href="#">Natural RPC</a>」も参照してください。</p>

## は行

### バージョン

製品のバージョンは、バージョン番号の最初の2桁で示されます。Software AG では、製品に追加した機能やテクノロジーの量に応じて、メジャーバージョンとマイナーバージョンを区別しています。その他の数字は修正レベルを示します。

製品ドキュメントでは、*vrs*、*vr*、または単に *v* という表記が、例えばデータセットやモジュール名などの中で、現在の製品バージョンのプレースホルダとしてよく使用されています。

プレースホルダ	意味	定義
v	バージョン	メジャーバージョン  製品バージョン番号の先頭の数字は、製品の価値を高める主要なアーキテクチャと機能が実装されている、または拡張されていることを示します。
r	リリース	マイナーバージョン  製品バージョン番号の2番目の数字は、製品の価値を高める機能が追加または拡張されていることを示します。
s	システムメンテナンスレベル	修正レベル  修正レベルの場合、エラー修正のみが含まれ、新機能は含まれません。すべての変更および修復に関するドキュメントが付属しています。  修正レベルに機能変更を含める必要がある場合は、例外処理プロセスによって、対応する品質保証アクティビティが確実に実行されます。このような機能変更はドキュメントに記載されます。当該の修正レベルをインストールしたときの影響を回避することが主な目的です。  Naturalバージョンの3番目の数字は、システムメンテナンスレベルを示します。  Naturalがサポートする特定のプラットフォームで、更新パッケージ、パッチレベル、サービスパック、修正プログラムなど、追加レベルが存在する場合があります。

パッケージリスト	<p><i>Predict</i> : パッケージリストタイプの <i>Predict</i> オブジェクトは、DB2 パッケージをドキュメントします。</p>
パッシブクロスリファレンス	<p><i>Windows - XRef Evaluation</i> : パッシブクロスリファレンス機能は、カレントオブジェクトを使用しているオブジェクトを表示します。例えば、手元にコピーコードがあって、それがアプリケーションのどの部分に含まれているかを知りたい場合などに、この機能を使用します。結果はツリービューに表示され、参照された（「使用された」）オブジェクトが最上位に表示されます。コピーコード、DDM、メソッドなど、一部のオブジェクトにはデフォルトでパッシブクロスリファレンスのみが含まれています。</p> <p>詳細については、Single Point of Development ドキュメントを参照してください。</p>
パッシブヘルプ	<p><i>Predict</i> : パッシブヘルプでは、各種機能に関する記述情報が提供されます。メニューの Retrieval Type/Function フィールドに疑問符を入力すると、またはヘルプのメインメニューから、コンテキスト依存のオンラインヘルプ情報を表示できます。</p>
バッファプール	<p>Natural では、Natural ニュークリアスが Natural オブジェクトを実行するのに使用するストレージを指します（「<a href="#">オブジェクトタイプ</a>」も参照）。Natural オブジェクトは、実行が要求されるとシステムファイルから読み取られ、バッファプールに置かれます。バッファプール内の Natural オブジェクトは、複数のユーザーが同時に使用できます。</p>
パラメータ	<p>以下の項目を参照してください。</p> <ul style="list-style-type: none"><li>■ <a href="#">ダイナミックパラメータ</a></li><li>■ <a href="#">NATPARM</a></li><li>■ <a href="#">Natural</a> コンフィグレーションユーティリティ</li><li>■ <a href="#">パラメータファイル</a></li><li>■ <a href="#">パラメータモジュール</a></li><li>■ <a href="#">プリンタプロファイル</a></li><li>■ <a href="#">プロファイル</a></li><li>■ <a href="#">プロファイルパラメータ</a></li><li>■ <a href="#">プロファイルパラメータ階層</a></li><li>■ <a href="#">SYSPARM</a> ユーティリティ</li><li>■ <a href="#">セッションパラメータ</a></li><li>■ <a href="#">スタティックパラメータ</a></li></ul>

*Natural* リモートプロシージャコール：各 **Natural RPC** パラメータの詳細については、使用環境の『オペレーション』ドキュメントを参照してください。メインフレーム *Natural* の場合、これらのパラメータは NTRPC マクロに含まれるか（スタティック定義）、プロファイルパラメータ RPC で定義されます（ダイナミック定義）。

#### パラメータファイル

メインフレーム：「**パラメータモジュール**」を参照してください。

*Windows/UNIX/OpenVMS*：デフォルトでは、ユーザーの *Natural* 環境の特質を決めるのに、パラメータファイル NATPARM.SAG 内のパラメータ指定が使用されます。パラメータファイルがないと、*Natural* は実行できません。初期状態の NATPARM パラメータファイルには、Software AG によって提供されるデフォルト値が含まれます。*Natural* にシステムデフォルト以外のパラメータ値を使用する場合は、**Natural コンフィグレーションユーティリティ**を使用して、デフォルトパラメータファイルである NATPARM.SAG を修正するか、ユーザー独自のパラメータファイルを作成します。どのパラメータファイルも、名前は 8 文字で拡張子が .SAG である必要があります。

#### パラメータモジュール

*Windows/UNIX/OpenVMS*：「**パラメータファイル**」を参照してください。

メインフレーム：ユーザーの *Natural* 環境の特質を決めるために使用される、*Natural* のすべてのプロファイルパラメータ設定を含みます。パラメータモジュールがないと、*Natural* は実行できません。Software AG によって提供されるデフォルトパラメータモジュールが、ソース形式の NATPARM です。*Natural* パラメータモジュールはロードモジュールであるため、アセンブルしてリンクする必要があります。*Natural* にシステムデフォルト以外のパラメータ値を使用する場合は、NATPARM を修正するか、ユーザー独自のパラメータファイルを作成します。

*Natural* パラメータは、*Natural* ニュークリアスにリンクする標準パラメータモジュールで定義されます。このモジュールは、*Natural* パラメータ階層の最下位レベルを構成します。*Natural* 管理者は、*Natural* 標準パラメータモジュールに加えて、任意の個数の追加パラメータモジュールを定義できます。追加パラメータモジュールは TP またはオペレーティングシステムライブラリに保存され、*Natural* の開始時に、パラメータ PARM によって代替パラメータモジュールとして使用できます。

#### 非同期 *Natural* セッション

非同期 *Natural* セッションは、どの端末にも関係しない「オンライン」セッションです（つまり、TP モニタ配下で実行中）。したがって、端末ユーザーと対話することはできません。

ビュー	<i>Predict</i> : 「 <a href="#">ユーザービュー</a> 」を参照してください。
標準ファイル	<i>Predict</i> : 標準ファイルは、1つの編成内で処理されるすべてのデータ（データ定義／会社標準）のレイアウト全体をドキュメントします。標準ファイル内のフィールドは、実装されたデータ構造を直接参照しません。標準ファイル内のフィールド定義は異なるタイプのマスタファイルに自動更新されます。  「 <a href="#">自動更新</a> 」を参照してください。
標準フィールド	<i>Predict</i> : 標準ファイル内のフィールド。標準ファイルおよび標準フィールドでは、別のファイル内のフィールドも標準的な使用を強制されます。
ピリオディックグループ	<i>Adabas</i> : ピリオディックグループフィールドでは、 <b>FDT</b> 内の連続フィールド（一連の <a href="#">エレメンタリフィールド</a> や <a href="#">マルチプルバリューフィールド</a> ）が定義されます。これらのフィールドはレコード内でまとめて繰り返されます。ピリオディックグループによって定義されたフィールドは、Adabas のバージョンと FDT の定義に応じて、レコードごとに最大 65,534 回繰り返すことができます。Natural で Adabas のピリオディックグループにアクセスするには、すべてのフィールドを別々の配列として定義するか、1グループを1つの配列として定義します。
ファイル	<i>Predict</i> : ファイルタイプの <i>Predict</i> オブジェクトには、フィールドの集合体の定義が含まれます。
ファイルカップリング	<i>Adabas</i> : ファイルカップリングは、特定のファイルのレコードが別のファイルのレコードに関連付けられている（カップリングされている）場合に、それらのレコードを選択するために使用します。ファイルは Adabas ディスクリプタを使用して連結（カップリング）されます。ファイルの連結を物理的に行う場合は（ハードカップリング）、ISN リストを使用します。論理的に行う場合は（ソフトカップリング）、検索結果（FIND ステートメント）をランタイムにダイナミックに評価します。
ファイル関係	<i>Predict</i> : ファイル関係タイプの <i>Predict</i> オブジェクトは、ファイル内のフィールドを使用して設定される2つのファイル間の論理的または物理的关系をドキュメントします。 <i>Predict</i> の以前のバージョンでは、「ファイル関係」は「関係」と呼ばれていましたが、他の Software AG 製品との互換性の理由で名前が変更されました。
フィールド	<i>Predict</i> : フィールドタイプの <i>Predict</i> オブジェクトは、 <a href="#">ファイル</a> 内部にある参照の最小論理単位をドキュメントします。
フィールド定義テーブル (FDT)	「 <b>FDT</b> 」を参照してください。



ブートストラップモジュール	プログラムを呼び出すために使用されるモジュール。ブートストラップモジュールは、プログラムを実行するのに必要な環境を提供します。
複合アプリケーション	「 <a href="#">アプリケーション</a> 」を参照してください。
複数マスタビュー	<i>Predict</i> : 1つ以上のマスタファイルから起動できるビュー（例：ORACLE ビュー）。
物理ファイル	<i>Predict</i> : 「 <a href="#">マスタファイル</a> 」を参照してください。
プラグインマネージャ	<i>Windows</i> : Natural スタジオのユーザーインターフェイスは、プラグインによって拡張できます。プラグインは、プラグインマネージャから起動および停止することができます。Natural スタジオ機能自体の一部がプラグイン形式で提供されます。プラグインのサンプルが SYSEXPLG ライブラリのソースコード内で提供されています。詳細については、『 <i>Natural スタジオの使用</i> 』ドキュメントの「プラグインマネージャ」、および『 <i>プラグインによる Natural スタジオの拡張</i> 』ドキュメントを参照してください。
プリンタプロファイル	<i>Windows/UNIX/OpenVMS</i> : プリンタのプロファイル情報は、グローバルコンフィグレーションファイル内に指定されています。このファイルは、Software AG によって提供される印刷デフォルト値が初期設定されています。グローバルコンフィグレーションファイルの設定は、 <a href="#">Natural コンフィグレーションユーティリティ</a> を使用して変更できます。
プレースホルダ	<i>Predict</i> : 他のオブジェクトにリンクされているオブジェクトをロード／インポートしたときに、参照先オブジェクトがロード／インポートされておらず、ターゲット環境にも存在しない場合、ターゲット環境に参照先オブジェクトのプレースホルダが追加されます。このプレースホルダの目的は、ターゲット環境に参照先オブジェクトのオブジェクト ID を確保し、後で旧環境内のリンクを新環境内で再作成できるようにすることです。  プレースホルダには次の情報が含まれます。 <ul style="list-style-type: none"><li>■ オブジェクト ID</li><li>■ 内部 ID</li><li>■ サブタイプ（存在する場合）</li><li>■ 送信ステータスプレースホルダ</li></ul>
プログラム	<i>Predict</i> : プログラムタイプの <i>Predict</i> オブジェクトは、異なるタイプおよび言語のオブジェクトを処理するデータをドキュメントします。



プログラムドリブンアプリケーション	イベントではなくプログラムによって実行コード部分が制御されるアプリケーション。実行可能なコードの最初の行から実行が開始され、定義されたパスウェイに従ってアプリケーションを通過し、あらかじめ設定された順序で指示どおりに追加のプログラムを呼び出します。
プロセス	オペレーティングシステムのプロセスは、自身のメモリスペース、コード、データや他のオペレーティングシステムリソースを持つ作業要素のことで、1つまたは複数のスレッドで構成されます。
プロトコルアクセス層	通信プロトコル (TCP/IP) とクライアント/サーバーの間でのアクセス呼び出しのコーディング。
プロパティ	クライアントからアクセスできるオブジェクトの属性。Natural クラスでは、オブジェクトのプロパティ値はオブジェクトデータエリアに格納されます。したがって、プロパティごとにオブジェクトデータ変数を割り当てる必要があります。詳細については、「 <a href="#">オブジェクトデータ変数</a> 」を参照してください。
プロパティ実装	プロパティに割り当てられるオブジェクトデータ変数。詳細については、「 <a href="#">オブジェクトデータ変数</a> 」を参照してください。
プロファイル	メインフレーム：SYSPARMユーティリティを使用して、プロファイルパラメータの文字列を指定し、プロファイル名を付けて保存することができます。プロファイルを使用するには、ダイナミックパラメータ <code>PROFILE=profile-name</code> を指定して Natural を開始します。そのプロファイル名で格納されているパラメータの文字列は、ダイナミックパラメータとして Natural に渡されます。ここで言うプロファイルとは、プロファイル名を付けて保存したプロファイルパラメータの文字列を指します。
プロファイルパラメータ	<p>NATPARM に定義されているすべての Natural パラメータ（ドライバパラメータやフロントエンドパラメータなどは含まれません）。Natural プロファイルパラメータは、Natural 環境のさまざまな特質を定義します。Natural 管理者のみにすべてのプロファイルパラメータを設定する権限があります。</p> <p>プロファイルパラメータの値は、次の3つのソースから取得されます。</p> <ul style="list-style-type: none"><li>■ Natural SET GLOBALS ステートメントまたは GLOBALS システムコマンドで、ランタイムに割り当てられるセッションパラメータ（最も優先度が高い）。</li><li>■ 現在の Natural セッションに有効なダイナミック割り当て。こうした割り当てを行うには、Natural の起動時に個々のパラメータおよび/または代替パラメータファイルを指定します。</li></ul>

- Natural モジュール/パラメータファイル NATPARM で指定されているスタティック割り当て（最も優先度が低い）。メインフレームでは、Natural ニュークリアスにリンクしているパラメータモジュールがこれに当たります。Windows、UNIX、および OpenVMS では、パラメータファイル NATPARM.SAG がこれに当たります。

詳細については、Natural を実行しているプラットフォームの『オペレーション』ドキュメントを参照してください。

#### プロファイルパラメータ階層

Natural プロファイルパラメータは、下の表に示されているように、異なる階層構造レベルで設定されます（優先度は上の方が高くなります）。

セッション中（最も優先度が高い）	開発環境設定。 プログラム/ステートメントレベル設定。 セッションパラメータ設定。 Natural Security 定義。
セッション開始時にダイナミック	ダイナミックパラメータ入力。 事前定義されたユーザーパラメータプロファイル。 事前定義されたダイナミックパラメータ設定。
スタティック	代替パラメータファイル/モジュール。 Natural 標準パラメータファイル/モジュール。

『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方法」も参照してください。

#### フロントエンド

Windows/UNIX/OpenVMS：画面およびプリンタ出力を用意します。

メインフレーム：「[フロントエンドスタブ](#)」を参照してください。

#### フロントエンドスタブ

z/OS：Natural DB2 ストアドプロシージャサーバーと Natural 開発サーバーによって使用される Natural サーバー環境のコンポーネント。フロントエンドスタブは、クライアント/サーバープロトコルと Natural サーバーフロントエンドとの間で通信を行います。

#### 分散コンポーネントオブジェクトモデル (DCOM)

「[DCOM](#)」を参照してください。

#### 変換データベース

Predict：Adabas Star の使用時に論理ファイル番号から物理的なファイルを見つけるために使用する ADASTAR 変換テーブルが含まれたデータベース。

編集行	<i>Predict</i> : Add、Copy、Modify の各機能の入力画面の最後にある行。この行により、特定の属性を作成または修正するかどうかが決まります。編集行の入力フィールドに Y を入力すると、 <i>Predict</i> エディタが呼び出されます。
変数	「 <a href="#">システム変数</a> 」および「 <a href="#">ユーザー定義変数</a> 」を参照してください。
ポータブル GP	<i>Windows/UNIX/OpenVMS</i> : Natural バージョン 5 でカタログされた GP は、Natural がサポートする UNIX、OpenVMS、および Windows プラットフォームに移植することができます。つまり、Natural バージョン 5 でカタログされた GP は、これらのプラットフォームで再コンパイルせずに実行できます。この機能を使用すると、UNIX、OpenVMS、Windows プラットフォーム全体でのアプリケーションの展開が容易になります。コマンドプロセッサ GP は以前と同様に動作します。「 <a href="#">生成プログラム</a> 」も参照してください。

## ま行

マスタファイル	<i>Predict</i> : ビューの作成元となったファイル。 <i>Predict</i> では、マスタファイルはデータ階層の中間レベルにあります。マスタフィールドは標準フィールドから属性を継承することができ、マスタフィールド内の属性値は下位レベルのビューおよびユーザービューに自動更新されます。
マスタフィールド	<i>Predict</i> : マスタファイルの中のフィールド。
マッピング	<i>Windows Single Point of Development</i> : リモート開発の場合、開発サーバーを Windows セッションにマップし、次にアプリケーションにマップすることができます。詳細については、『 <a href="#">SPoDを使用したリモート開発</a> 』ドキュメントの「 <a href="#">アプリケーションワークスペース</a> 」、および『 <a href="#">システムコマンド</a> 』ドキュメントの MAP コマンドの説明を参照してください。
マップ	プログラム内で参照される画面レイアウト。プログラム内で参照される画面レイアウトのフォーマット命令を提供します。マップは、入力および出力フィールドを定義して、これらにプログラム変数を割り当てます。マップタイプの Natural オブジェクトは、マップエディタで作成されます。詳細については、Natural の『 <a href="#">エディタ</a> 』ドキュメントを参照してください。
マルチプルバリューフィールド	<i>Adabas</i> : 複数の値（またはオカレンス）を持つことのできるフィールド。1つのファイル内の各マルチプルバリューフィールドに可能な値（オカレンス）の数は、単一レコードで最大 65,534 個です。オカレンスの最大数は、Adabas のバージョンと <a href="#">FDT</a> の定義によっ

て異なります。NaturalでAdabasのマルチプルバリューフィールドにアクセスするには、フィールドを配列として定義します。

メイン FDIC *Predict* : メイン FDIC は、Predict Coordinator を使用した Load または Import 操作のターゲット環境です。

メインフレーム オペレーティングシステム z/OS、VSE/ESA、VM/CMS、および BS2000/OSD を指します。また、これらのオペレーティングシステム上の Natural でサポートされるすべての TP モニタも含まれます。

メソッド オブジェクト指向プログラミング: メソッドは、クライアントによって要求されたときにクラスのオブジェクト/インスタンスが実行できる関数です。

メタデータ構造 *Predict* : Predict データディクショナリ構造は、オブジェクトタイプ、その属性およびアソシエーションタイプで構成されます。この構造は、Predictのメタデータ管理機能を使用して新しいオブジェクトタイプおよびアソシエーションタイプを定義すれば拡張できます。Predictの事前定義されたオブジェクトタイプは修正できません。

メンバ *Predict* : XRef データの対象となる Natural オブジェクトまたは第 3 世代言語オブジェクト。

## や行

ユーザー *Predict* : ユーザータイプの Predict オブジェクトは、システムを誰が使用しているかをドキュメントします。ユーザーとドキュメンテーションオブジェクトとの間の論理的なつながりは、オーナーによって設定されます。「[オブジェクトタイプ](#)」を参照してください。

ユーザー定義変数 ユーザー自身がプログラムで定義するフィールド。プログラム処理中の特定のポイントで得られた値または中間結果を保存して、追加処理または表示できるようにするために使用されます。

ユーザー定義変数を定義するには、DEFINE DATA ステートメントで名前とフォーマット/長さを指定します。『[プログラミングガイド](#)』の「[ユーザー定義変数](#)」を参照してください。

ユーザー出口 [ユーザー出口](#) とは、Natural、サブコンポーネント、またはサブプロダクト内の、制御がユーザー出口ルーチンに渡されるポイントです。

ユーザー出口ルーチン ユーザー出口ルーチンは、定義された [ユーザー出口](#) で制御の役割を担い、データの操作または決定を行います。通常は、サンプル

のユーザー出口がソース形式で提供されています。ユーザー出口に含まれる命令は、ユーザーが記述または調整する必要があります。ほとんどのユーザー出口ルーチンは Natural 言語を利用しています。1つの小さなサブセットについては、アセンブラ言語（メインフレームシステム）または C（メインフレーム以外のシステム）で記述する必要があります。

#### ユーザービュー

*Predict* : ユーザービューは、プログラムのデータ宣言セクションで使用するために定義された、マスタファイルの論理ビューです。ユーザービューのフィールドの数と順序はマスタファイルと異なることがあります。また、特定の互換性ルールのために、ユーザービューのフィールドの特定の属性が物理/論理レベルで対応する値と異なることもあります。

#### 予約語

Natural プログラミング言語の一部。ユーザーには使用できない文字の組み合わせを指します。

『プログラミングガイド』の「Natural 予約キーワード」を参照してください。

## ら行

#### ラージ変数

*Windows/UNIX/OpenVMS* : 文字データおよびバイナリデータのラージ変数は、通常の Natural フォーマット A および B に基づいています。フォーマット A およびフォーマット B の制限は、それぞれ 253 バイトと 126 バイトですが、こうした制限はラージ変数には無効です。ラージ変数の新しい制限は 1 ギガバイトです。このようなスタティックなラージ変数およびフィールドは、定義、再定義、値領域割り当て、変換、ステートメント内での参照などに関して、従来の文字/バイナリ変数およびフィールドと同じ方法で処理されます。文字/バイナリフォーマットに関するすべての規則がラージフォーマットに適用されます。

『ステートメント』ドキュメントの「ダイナミック変数およびフィールドについて」を参照してください。

#### ライブラリ構造

*Predict* : ライブラリ構造タイプの *Predict* オブジェクトには、Natural *steplib* 構造をドキュメントするシステムオブジェクトが含まれます。

#### ラベル

Natural ステートメントは、先頭にラベルを付けてマークすることができます。ラベルは任意の名前です。ラベルでマークされたステートメントは、プログラムの別の場所でラベルを指定することによって参照できます。

- ランタイム
1. コンパイラとは対照的に、Naturalプログラムを実行するNaturalニュークリアスの一部。
  2. Naturalプログラムが実行される時間。
- リンク
- Predict* : Predictドキュメンテーションオブジェクトは、アソシエーションを使用してリンクすることができます（「[アソシエーション](#)」を参照）。
- ルーチン
- ヘルプルーチン、サブプログラム、サブルーチンのように、それだけでは実行できないNaturalオブジェクトタイプを指す総称。
- レジストリ
- Windowsプラットフォームでの、コンフィグレーションデータの中央リポジトリ、またはコンフィグレーションファイルのデータベース。物理的には、レジストリは *System.dat* と *User.dat* という2つのファイルから構成されています。論理的には、レジストリエディタで表示されるコンフィグレーションデータがレジストリです。レジストリには、ハードウェア、ソフトウェア双方の全種類のコンフィグレーション情報が含まれています。例えば、**DCOM**クラスとそのサーバーへの割り当てに関する情報が含まれます。詳細については、Microsoftのドキュメントを参照してください。
- レジストリキー
- レジストリキーは、クラスの登録時にサーバーのWindowsシステムレジストリに作成されるエン트리です。また、レジストリキーは、クライアント登録ファイルの実行時にクライアントシステムレジストリに追加されます。
- レジストリキーおよびその管理についての詳細な予備知識については、該当するプラットフォームのレジストリに関するドキュメントを参照してください。
- ローカルクラス
- NaturalXクラスのタイプは、ローカル、内部、または外部です。これは、クラスが登録された方法によって決まります。ローカルクラスとは、**DCOM**クラスとして登録されていないクラスのことです。したがって、ローカルクラスのオブジェクトは、他のプロセスからは作成することもアクセスすることもできません。現在のNaturalセッションのプログラムからのみ、作成/アクセスすることができます。
- 詳細については、「[内部クラス](#)」および「[外部クラス](#)」を参照してください。
- ロード
- Predict* : **コーディネータ FDIC** から移行データをロードします。ロードされるデータは、**コーディネータチェックサイクル**の3つの段階のすべてを通過する必要があります。



---

ロールサーバー	z/OS：サーバーとしての Natural は、独立したリージョン内で動作するか、DB2 ストアドプロシージャのようなサーバーサブシステムリージョン内で動作します。Natural をサーバーとして実行するには、サービス固有のサーバースタブが必要です。このサーバースタブは、サーバー製品の一部として提供され、すべてのサービス要求を制御するとともに、Natural サーバーフロントエンドの唯一のインターフェイスとして機能します。
論理ファイル	<i>Predict</i> ：論理ファイル定義に、ファイルの物理的実装に関する情報が含まれているとは限りません。基本的に、論理ファイル定義はファイル構造の定義のことです。

## わ行

---

ワークファイルタイプ	特定のワークファイルフォーマットで作成された、Natural で使用可能なファイル。例えば、Entire Connection フォーマットでデータを保存するワークファイルタイプは 2 種類あります。ENTIRE CONNECTION と TRANSFER です。  詳細については、プロファイルパラメータ WORK を参照してください。
ワークファイルフォーマット	ターゲットファイルに保存されるデータの構造を示します。例えば、ワークファイルフォーマット Entire Connection では、2 つのファイル .NCD および .NCF が作成されます。  詳細については、『オペレーション』ドキュメントの「ワークファイルフォーマット」を参照してください。



