

Natural für z/OS

Glossar

Version 9.2.4

Oktober 2025

Dieses Dokument gilt für Natural für z/OS ab Version 9.2.4.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuausgaben bekanntgegeben werden.

Copyright © 1979-2025 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein.

Nähere Informationen zu den Patenten und Marken der Software AG und ihrer Tochtergesellschaften befinden sich unter <http://documentation.softwareag.com/legal/>.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt "License Texts, Copyright Notices and Disclaimers of Third Party Products" entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices, abrufbar unter dem Abschnitt "License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products". Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://softwareag.com/licenses> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Dokument-ID: NATMF-NGLOSSARY-924-20251031DE

Inhaltsverzeichnis

| | |
|---------------|----|
| Glossar | 1 |
| A | 1 |
| B | 5 |
| C | 6 |
| D | 12 |
| E | 15 |
| F | 18 |
| G | 19 |
| H | 20 |
| I | 22 |
| J | 25 |
| K | 25 |
| L | 26 |
| M | 27 |
| N | 28 |
| O | 33 |
| P | 36 |
| Q | 42 |
| R | 42 |
| S | 45 |
| T | 52 |
| U | 54 |
| V | 56 |
| W | 57 |
| X | 58 |
| Y | 59 |
| Z | 59 |

Glossar

Danksagungen, Markenzeichen, Haftungsausschluss

Ergänzend zur spezifischen Terminologie unserer Produkte und bestimmten generellen Fachbegriffen aus dem Bereich Computer enthält dieses Glossar auch englische Begriffe und Beschreibungen der International Business Machines Corporation (IBM). Auszüge aus der folgenden seitens der IBM urheberrechtlich geschützten, englischsprachigen Veröffentlichung werden mit freundlicher Genehmigung der IBM abgedruckt und sind mit einem Stern (*) gekennzeichnet.

IBM Terminology, Copyright © International Business Machines Corporation.

Darüber hinaus enthält dieses Glossar auch Begriffe und Beschreibungen vom Unicode Consortium. Auszüge aus der folgenden, urheberrechtlich geschützten Veröffentlichung werden mit freundlicher Genehmigung der Unicode, Inc. nachgedruckt:

Glossary of Unicode Terms, Copyright © Unicode, Inc.

Da nur Teile der Produkt-Dokumentationen für Natural und Natural-Anwendungen in deutscher Sprache vorliegen, kann dieses Glossar auch als Nachschlagewerk für englische Fachbegriffe dienen, die in gemischtsprachigen Produkt-Dokumentationen verwendet werden.

Alle anderen in diesem Glossar erwähnten Firmennamen, Produktnamen und in Beziehung dazu stehenden Begriffe können Markennamen ihrer betreffenden Eigentümer sein.

Die in diesem Glossar enthaltenen Beschreibungen werden generell zur Erleichterung der Orientierung unserer Benutzer verfügbar gemacht. Wir übernehmen keine Verantwortung für die Exaktheit oder Vollständigkeit der hierin zur Verfügung gestellten Informationen.

A

Abstract (Kurzbeschreibung)

Predict: In Predict kann jedes Objekt einen *Abstract* haben, der kurze Kommentare zu dem Objekt enthält und maximal 16 Zeilen mit jeweils maximal 30 Zeichen umfasst. In früheren Predict-Versionen auch als *Comments* oder *Short Comments* bezeichnet. Der Name wurde aus Gründen der Kompatibilität mit anderen Produkten geändert.

| | |
|--|--|
| Activation Policies (Aktivierungsverfahren) | <p><i>Activation Policy</i> ist ein Attribut einer <i>NaturalX Class</i>, die als <i>DCOM Class</i> registriert werden soll. Sie definiert, ob die <i>Class</i> gemeinsam mit anderen <i>Classes</i> eine <i>Natural Session</i> nutzen kann.</p> <p>NaturalX kombiniert die verschiedenen, durch DCOM unterstützten Optionen, in Form von drei Aktivierungsverfahren:</p> <ul style="list-style-type: none">■ <i>ExternalMultiple</i>■ <i>ExternalSingle</i>■ <i>InternalMultiple</i> <p>Die <i>Activation Policy</i> einer <i>Class</i> kann gesetzt werden mit dem Natural-Systemkommando <code>REGISTER</code>, im Natural-Statement <code>DEFINE CLASS</code> oder mit dem Natural-Profilparameter <code>ACTPOLICY</code>.</p> |
| Active Help (Aktive Hilfe) | <p><i>Predict</i>: Falls zur Ausführung einer Funktion benötigte Parameterwerte fehlen, wird eine aktive Hilfe in Form eines Auswahlfensters angeboten, das alle möglichen Werte enthält. Dies gilt bei allen Eingabefeldern, die mit einem Stern (*) gekennzeichnet sind.</p> |
| Active Reference Data (Aktive Referenzdaten) | <p><i>Predict</i>: Siehe XRef Data.</p> |
| Active Retrieval (Aktive Recherche) | <p><i>Predict</i>: <i>Active Retrieval</i>-Funktionen dienen zur Abfrage von Informationen aus <i>Xref</i>-Daten und aus <i>Predict</i>-Objekten des Typs <i>Documentation</i>. Ein Vergleich der Dokumentation und der Implementierung eines Systems ist möglich.</p> |
| ActiveX Component (ActiveX-Komponente) | <p><i>Windows</i>: Standardisierte Microsoft-Schnittstelle zu externen Programmierungskomponenten, zum Beispiel, fachliche Funktionen oder Berechnungen. <i>ActiveX Components</i> haben keine Benutzerschnittstelle.</p> |
| ActiveX Control (ActiveX-Steuerelement) | <p><i>Windows</i>: Standardisiertes Programmierungssteuerelement mit Benutzerschnittstelle, zum Beispiel: Optionsfelder oder Optionsschaltflächen oder Listfelder, die in Natural-Dialogen verwendet werden können. Natural kann eine Auswahlliste der in der Registry verfügbaren ActiveX-Steuerelemente abrufen. Mit Hilfe des <i>Component Browser</i> können die Standardeinstellungen in den Eigenschaften (<i>Property Pages</i>) eingesehen werden.</p> |
| Adabas | <p>Universelles, multithreading-fähiges und skalierbares Datenbank-Verwaltungssystem. Zeichnet sich aus durch hohe Zuverlässigkeit und Leistungsfähigkeit. Verfügbar für zahlreiche Plattformen.</p> |

| | |
|---|---|
| Address Space (Adressraum) | Bereich, in dem Natural läuft. |
| AIV = Application-Independent Variable (Anwendungsunabhängige Variable) | <ol style="list-style-type: none"> 1. Natural-Variable, die verwendet wird, um Werte programmübergreifend zuzuweisen. Siehe Abschnitte <i>User-Defined Variables</i> (im <i>Leitfaden zur Programmierung</i>) und das <code>DEFINE DATA</code>-Statement (in der <i>Statements-Dokumentation</i>). 2. <i>Großrechner</i>: Natural-Systemkommando <code>AIV</code>, mit dem alle aktiven anwendungsunabhängigen Variablen aufgelistet werden können. Siehe Systemkommando <code>AIV</code>. |
| Ajax = Asynchronous JavaScript and XML | Verfahren zum Erstellen von interaktiven Anwendungen auf Browser-Basis, die Benutzerabfragen unmittelbar verarbeiten. |
| ALF = Access Layer Format (Zugriffsschichtformat) | <i>Predict</i> : Vordefiniertes, datenbankunabhängiges Format, das den Austausch von Daten zwischen <i>Predict Case</i> und <i>Natural Engineering Workbench</i> ermöglicht. Alle mit dem <i>Predict Coordinator</i> zu übertragenden Daten müssen zunächst in ein Übertragungsmedium gestellt werden. |
| AMODE = Addressing Mode (Adressierungsmodus) | Anzahl der Bits (24 oder 31) einer virtuellen Adresse, die während der Programmausführung benutzt wird. |
| API | Siehe Application Programming Interface . |
| AppID (Anwendungskennung) | <p>In der <i>Windows System Registry</i> wird jede Anwendung durch eine Anwendungskennung (<i>AppID</i>) repräsentiert. Dies <i>AppID</i> ist eine global eindeutige Kennung, die unter dem Registrierungsschlüssel <code>HKEY_CLASSES_ROOT\AppID</code> zu finden ist. DCOM benutzt die <i>AppID</i>, um <i>Classes</i> nach Anwendungen zu gruppieren. Darüber hinaus werden Security-Einstellungen auf der Basis der <i>AppID</i> definiert. Natural legt in der Registry für jede <i>NaturalX Server ID</i> eine <i>AppID</i> an.</p> <p>Weitere Informationen siehe Server ID.</p> |
| Application (Anwendung) | Eine logische Sicht auf eine Sammlung miteinander in Verbindung stehender Programmierelemente. Zusammen bilden sie eine funktionale Einheit, die die fachliche Funktionslogik (<i>Business Logic</i>) für ein bestimmtes fachliches Problem abdeckt. Eine Anwendung besteht aus einem Satz Bibliotheken (<i>Libraries</i>) und ihren Natural-Objekten und/oder Unteranwendungen (fachliche Objekte). Die Inhalte einer Bibliothek (Natural-Objekte, Ressourcen usw.) können zu verschiedenen Anwendungen gehören. Informationen bezüglich einer Anwendung (<i>Application Description</i>) werden im Development Server |

File vorgehalten. Auf diese Datei kann von allen Plattformen aus zugegriffen werden.

Es gibt zwei Arten von Anwendungen:

■ **Base Applications (Basis-Anwendungen)**

Basis-Anwendungen bestehen aus einer Anwendung auf einer Plattform, z.B. ein Theatereintrittskarten-Reservierungssystem.

■ **Compound Applications (Verbund-Anwendungen)**

Verbund-Anwendungen bestehen aus mehreren Basis-Anwendungen, z.B. eine Theater-Verwaltungsanwendung, in der Anwendungen für die Reservierung von Eintrittskarten, zum Ausstellen von Kundenrechnungen und zum Versenden von Aufführungsinformationen an Kunden. Die Basis-Anwendungen können sich auf mehreren Plattformen befinden.

| | |
|---|---|
| Application Description (Anwendungsbeschreibung) | Liste aller zu einer Anwendung gehörenden Bestandteile. Diese wird im <i>Development Server File</i> gespeichert. Datenbankdatei (zurzeit FDIC), in der Anwendungsinformationen physisch gespeichert werden. Siehe Development Server File . |
| Application Description Layer (Anwendungsbe- schreibungsschicht) | Code, der den Zugriff auf die Anwendungsbeschreibung regelt. |
| Application Programming Interface – API (Anwen- dungsprogrammierungs- schnittstelle) | Programmierobjekt, das es dem Benutzer ermöglicht, auf Daten lesend oder verändernd zuzugreifen und/oder Dienste zu benutzen, die spezifisch für Natural, einer Unterkomponente oder einem Unterprodukt sind. Zweck eines API ist es, Informationen abzurufen oder zu ändern oder Dienste zu benutzen, auf die nicht über Natural-Statements zugegriffen werden kann. APIs können je nach Anwendungsschnittstelle unter Verwendung eines CALLNAT- oder CALL-Statements aufgerufen werden. |
| Application Wizard (Assis- tent zum Erstellen von Anwendungsbeschreibun- gen) | <i>Windows:</i> Werkzeug, mit dem festgelegt werden kann, was zu einer bestimmten Anwendung gehört. Der Benutzer wird durch Dialogboxen beim Erstellen einer Anwendungsbeschreibung für eine vorhandene Anwendung angeleitet. |
| Application Workspace (Anwendungsarbeitsbe- reich) | <i>Windows:</i> Dient der visuellen Darstellung einer Anwendung in einer Baumstrukturansicht. Der Anwendungsarbeitsbereich zeigt alle, zur aktuellen Anwendung gehörenden Elemente. |
| ASM = Natural Authorized Services Manager | Stellt folgende zugelassenen Betriebssystemfunktionen zur Benutzung innerhalb von Natural zur Verfügung: |

- Weiterleitung von *Natural Buffer Pool*-Objekten,
- Schreiben von Datensätzen in der *System Management Facility (SMF)*,
- Bereithalten von *Natural Session*-Informationen im *Session Information Pool (SIP)*.

Der ASM stellt Funktionen über PC-Routinen zur Verfügung und läuft in einem eigens dafür vorbehaltenen Adressraum.

Association (Verknüpfung) *Predict*: Predict-Objekte des Typs *Documentation* können durch *Associations* miteinander verknüpft werden. Es werden unterschiedliche Typen von *Associations* benutzt, um Objekte verschiedenen Typs miteinander zu verknüpfen. *Associations* sind einseitig wirkend (*Child and Parent Association*).

Der Verwalter des Datendiktions kann mit Hilfe der **Metadata Administration**-Funktionen von Predict zusätzliche Typen von *Associations* definieren.

Asynchronous Natural Session (Asynchrone Natural-Sitzung) Eine asynchrone Natural-Sitzung ist eine Online-Sitzung (die z.B. unter einem TP-Monitor läuft), die mit keinem Terminal verbunden ist und deshalb nicht mit einem Terminal-Benutzer interagieren kann.

Asterisk Notation (Stern-Notation) *Predict*: Ein Stern (*) kann als Platzhalterzeichen benutzt werden, um Eingabeparameter anzugeben, mit denen Gruppen von Werten adressiert werden sollen. Beispiel: Bei Eingabe von `atab*` als Kennung werden alle Kennungen adressiert, die mit den Zeichen `atab` beginnen.

Attribute *Predict*: Predict-Objekte des Typs *Documentation* haben Attribute, die das Objekt beschreiben. .

Automation Object (Automatisierungsobjekt) *Windows*: Programm, das über eine Automatisierungsschnittstelle benutzt werden kann. Weitere Informationen siehe *Windows-Dokumentation*.

B

Base Application (Basis-Anwendung) Siehe *Application*.

Betriebssysteme Alle von Natural und von Natural Add-on-Produkten unterstützten Betriebssysteme sind aufgeführt im Abschnitt *Product Version Availability* unter <https://empower.softwareag.com/>.

| | |
|--------------------------------|---|
| BMP = Basic Multilingual Plane | <i>Unicode</i> : Mehrsprachige Basis-Ebene (andere Bezeichnung: <i>Plane 0</i>), die hauptsächlich die zurzeit gebräuchlichen Schriftsysteme umfasst. |
| BMP = Batch Message Processing | <i>IMS TM</i> : Programme, die asynchron im Batch-Modus unter der Kontrolle von <i>IMS TM</i> laufen. |
| BOM = Byte Order Mark | Die Byte-Reihenfolge-Markierung ist eine charakteristische Bytefolge, die am Anfang eines Datenstroms das Unicode-Zeichen U+FEFF codiert und als Kennung zur Definition der Byte-Reihenfolge und Kodierungsform in UCS/Unicode-Zeichenketten, speziell in Textdateien, dient. |
| Bootstrap Module (Start-Modul) | Modul, das benutzt wird, um ein Programm aufzurufen. Stellt die Umgebung zur Verfügung, die benötigt wird, um das Programm laufen zu lassen. |
| BLOB | Abkürzung für <i>Binary Large Object</i> . Es handelt sich um eine sehr große Variable. |
| BT=BACKOUT TRANSACTION | Natural-Statement, das abgesetzt werden kann, um eine Datenbanktransaktion abubrechen und den vor Beginn der Transaktion existierenden Zustand wiederherzustellen. Siehe auch <i>Transaction</i> und <i>END TRANSACTION (ET)</i> . |
| Buffer Pool | Der <i>Natural Buffer Pool</i> mit gemeinsam genutztem Speicher dient als Zwischenspeicher, in den Natural katalogisierte Objekte (z.B. Programme, siehe auch <i>Object Types</i>) aus einer Natural-Systemdatei zur anschließenden Ausführung durch das Natural-Laufzeitsystem und/oder zur Objektkompilierung durch den Natural-Compiler lädt. Wenn ein Natural-Objekt zur Ausführung angefordert wird, wird es aus der Systemdatei gelesen und in den Buffer Pool gestellt. Dort kann es dann von mehreren Benutzern gleichzeitig benutzt werden. |

C

| | |
|------------------------|---|
| Cache (Pufferspeicher) | Pufferspeicherbereich, der benutzt wird, um Kopien von Daten zwischenzuspeichern, die zuletzt vom Hauptspeicher abgerufen wurden. Dadurch werden unnötige Zugriffe auf Systemdateien und externe Datenbanksysteme oder Dateisysteme vermieden und folglich die CPU-Belastung verringert und die Reaktionszeit verkürzt. |
| CATALOG | In Natural bezieht sich der Begriff <i>Catalog</i> auf das Natural-Systemkommando <i>CATALOG</i> . Dieses Kommando bewirkt, dass der Quellcode kompiliert (katalogisiert) und der generierte Objektcode als katalogisiertes Objekt in einer Natural-Systemdatei gespeichert wird. |

| | |
|--|---|
| Cataloged Object (Katalogisiertes Objekt) | <p>Ein katalogisiertes Objekt in Natural ist die ausführbare (kompilierte) Form eines Natural-Objekts. Ein katalogisiertes Objekt wird als Objektmodul in einer Natural-Systemdatei gespeichert.</p> <p><i>Großrechner:</i> Ein auf dem Großrechner katalogisiertes Objekt ist nur auf Großrechnern ausführbar. Auf dem Großrechner kann ein katalogisiertes Objekt mit Hilfe des <i>Natural Optimizer Compiler</i> optimiert werden.</p> <p><i>Windows und UNIX:</i> Ab Natural Version 5.1 sind katalogisierte Objekte portierbar über alle von Natural unterstützten Windows- und UNIX-Plattformen. Falls erforderlich, kann bei einem katalogisierten Objekt außerdem der <i>Endian Mode</i> angegeben werden.</p> <p><i>Synonyme:</i> Gleichbedeutend werden auch die Bezeichnungen <i>GP</i> oder generiertes Programm (<i>Generated Program</i>) für ein katalogisiertes Objekt verwendet. Im Produkt <i>Predict Application Control</i> wird der Begriff <i>loadable</i> (ladbar) verwendet.</p> <p>Siehe auch <i>Source Object</i>, <i>Object</i> und <i>Object Types</i>.</p> |
| CF = Coupling Facility | <p>z/OS: Hardware-Komponente, die zum Steuern und Koordinieren des Zugangs zu Daten dient, die gemeinsam von Betriebssystemen genutzt werden, die an einer <i>Parallel Sysplex</i>-Umgebung beteiligt sind. Die <i>Coupling Facility</i> wird vom <i>Natural Authorized Services Manager (ASM)</i> und dem <i>Natural Roll Server</i> benutzt, um in einer <i>Parallel Sysplex</i>-Umgebung zu kommunizieren.</p> |
| CGI = Common Gateway Interface | <p>Wird vom <i>Natural Web Interface</i> benutzt, um Internet-Dienste in Natural verfügbar zu machen.</p> |
| CHECK | <p>In Natural bezieht sich der Begriff <i>Check</i> auf das Natural-Systemkommando <i>CHECK</i>, mit dem geprüft werden kann, ob die Syntax des Quellcodes Fehler enthält.</p> |
| Child | <p><i>Predict: Associations</i> werden verwendet, um zu dokumentieren, wie <i>Predict</i>-Objekte mit anderen <i>Predict</i>-Objekten in Beziehung stehen. Eine <i>Association</i> wird hergestellt, indem Objekte entweder als <i>Child</i> oder als <i>Parent</i> mit einem Objekt verknüpft werden. Beispiel: Ein Feld ist das <i>Child</i> einer Datei, eine Datei ist <i>Child</i> einer Datenbank.</p> |
| CICS = Customer Information Control System | <p>z/OS: Online-Transaktionsverarbeitungssystem der IBM.</p> |
| CICSplex = CICS Complex | <p>„A set of interconnected CICS regions acting as resource managers, and combining to provide a set of coherent services for a customer's business needs.“*</p> |

| | |
|---|---|
| Class | <p><i>Natural Classes</i> kapseln Datenstrukturen (Objekte) mit entsprechender Funktionalität (<i>Method</i>). Siehe auch <i>NaturalX Class</i>.</p> <p>Die interne Struktur eines Objekts des Typs <i>Class</i> wird mit einer <i>Data Area (Object Data Area)</i> definiert. Die Methoden einer <i>Class</i> werden als Unterprogramme (Objekttyp <i>Subprogram</i>) implementiert.</p> |
| Class Builder | <p><i>Windows</i>: Unterstützt die Definition und Implementierung von <i>NaturalX</i> und <i>DCOM Component Classes</i> als <i>Natural Classes</i>. Eine <i>Natural Class</i> kann aus verschiedenen Komponenten bestehen: Natural-Objekte (z.B. <i>Object Data Area</i>) oder Objekte, die nur im Quellcode der <i>Class</i> existieren (z.B. Schnittstellenkomponenten). Der <i>Class Builder</i> stellt jede Komponente der <i>Class</i> in Form eines Knotens dar. Durch Markieren der Knoten ist es möglich, die <i>Classes</i> und ihre Komponenten kontextabhängig zu verwalten.</p> |
| Class GUID (Globally Unique Identification) | <p>Soll eine <i>NaturalX Class</i> als DCOM <i>Class</i> registriert werden, muss für die <i>Class</i> eine <i>Globally Unique Identification (GUID)</i>, d.h. eine global eindeutige Kennung, definiert werden, um sicherzustellen, dass die <i>Class</i> in einem Netzwerk eindeutig identifiziert werden kann. In <i>Natural</i> wird die ID-Klausel des <code>DEFINE CLASS</code>-Statements benutzt, um einer <i>Class</i> eine <i>GUID</i> zuzuweisen. Eine <i>GUID</i> wird durch eine alphanumerische Konstante dargestellt, die mit dem Datenbereich-Editor (<i>Data Area Editor</i>) erzeugt werden kann. Der <i>Natural Studio Class Builder</i> weist jeder neuen <i>Class</i> automatisch eine neue <i>GUID</i> zu.</p> |
| Class Name | <p>Name der <i>Class</i>, der im Operanden <i>class-name</i> im <code>DEFINE CLASS</code>-Statement definiert wird. Dieser Name wird im <code>CREATE OBJECT</code>-Statement benutzt, um Objekte dieser <i>Class</i> zu erstellen. Siehe auch <code>DEFINE CLASS</code> in der <i>Statements</i>-Dokumentation.</p> |
| Class Module Name | <p>Name des <i>Natural</i>-Moduls, in dem eine <i>Natural Class</i> definiert ist.</p> |
| Client Stub | <p>Siehe RPC Stub.</p> |
| CLOB | <p>Abkürzung für <i>Character Large Object</i>. Datentyp zur Speicherung sehr langer Zeichenketten in Datenbanken.</p> |
| Cluster | <p><i>VSAM</i>: A data set defined to <i>VSAM</i>. A cluster can be a key-sequenced data set (KSDS), an entry-sequenced data set (ESDS), or a relative-record data set (RRDS).*</p> |
| Code Point | <p>Wert im Unicode Code Space, d.h. U+000000 - U+10FFFF.</p> |
| Code Page | <p>Siehe auch Unicode.</p> |

| | |
|---|---|
| Code Unit | Kleinste Bit-Kombination, die eine Einheit kodierten Textes zur Verarbeitung oder zum Austausch darstellen kann. |
| COM = Component Object Model | Von Microsoft stammendes <i>Framework</i> zum Entwickeln und Unterstützen von Programmkomponentenobjekten. COM bietet einen Satz an Schnittstellen, die es <i>Clients</i> und <i>Servers</i> ermöglichen, innerhalb desselben Rechners miteinander zu kommunizieren. Komponenten von verschiedenen Maschinen können mittels <i>DCOM</i> kombiniert werden. |
| Compiler | Der <i>Natural Compiler</i> übersetzt in Quellcode vorliegende Programme in Pseudo-Code. Siehe auch <i>Natural Optimizer Compiler</i> . |
| Com-plete | Universaler, vollständig dialogorientierter <i>Online-Transaction Processing (OLTP) Monitor</i> . |
| Compound Application (Verbund-Anwendung) | Siehe <i>Application</i> . |
| Component Browser | <i>Windows</i> : Ermöglicht es, ActiveX-Komponenten zu betrachten, die zur Entwicklung von NaturalX-Anwendungen zur Verfügung stehen. Merkmale sind: <ul style="list-style-type: none">■ Verfügbare ActiveX-Komponenten, ihre Bereitstellung und duale Schnittstellen werden aufgelistet.■ Datentypen werden auf Natural-Formate abgebildet.■ Hilfedateien externer Komponenten sind direkt zugänglich.■ Natural-Programmierbeispiele werden automatisch generiert. |
| Conceptual File (Konzeptionelle Datei) | <i>Predict</i> : Dateiobjekte des Typs <i>Conceptual</i> werden für Entwurfszwecke in der Frühphase der Anwendungsentwicklung benutzt. |
| Configuration Files (Konfigurationsdateien) | <i>Windows und UNIX</i> : Das Basisverzeichnis für FNAT/FUSER kann durch Nachsehen in den folgenden Konfigurationsdateien gefunden werden: <ul style="list-style-type: none">■ Lokale Konfigurationsdatei - Natural.INI Enthält <i>Natural Buffer Pool</i>-Zuweisungen und Installationszuweisungen. Befindet sich für jede installierte Natural-Version im entsprechenden Verzeichnis <i>etc</i>.■ Globale Konfigurationsdatei - NATCONF.CFG Enthält DBMS-Zuweisungen, Verzeichnis-Server-Zuweisungen, Druckerprofil-Zuweisungen und Systemdatei-Zuweisungen. |

| | |
|---|--|
| Conflict Management (Konfliktmanagement) | <p><i>Predict</i>: Erste Phase des <i>Coordinator Check Cycle</i>. Bei Predict Version 3.3 und höher haben alle Objekte eine interne Kennung (<i>Internal ID</i>). Konfliktmanagement bedeutet die Lösung von Konflikten, die von dieser internen Kennung herrühren. Ein potenzieller Konflikt läge vor, wenn zwei Objekte dieselbe Objektkennung, aber verschiedene interne Kennungen haben.</p> <p>Konflikte aufgrund der internen Kennung müssen im <i>Coordinator FDIC</i> gelöst werden, bevor die Daten in das <i>Main FDIC</i> geladen oder importiert werden können.</p> |
| Con-form | Textformatierungssoftware, die automatisch zusammen mit <i>Connect</i> installiert wird. |
| Con-nect | Bürokommunikationssystem, das eine Vielzahl an Unterstützungsfunktionen für Bürotätigkeiten bietet, z.B. elektronische Post, Textverarbeitung, Dokumentenverwaltung und Zeitplanung. |
| Connection (Verbindung) | <p><i>Predict</i>: Eine <i>Connection</i> zwischen einem externen Objekt und einem <i>Documentation</i>-Objekt wird hergestellt, wenn entweder das externe Objekt von dem <i>Documentation</i>-Objekt generiert worden ist oder das <i>Documentation</i>-Objekt vom externen Objekt aufgenommen worden ist. Die meisten verbundenen externen Objekte und <i>Documentation</i>-Objekte können explizit getrennt werden.</p> <p>Auf der Predict-Seite wird die Verbindung mit dem Implementierungszeiger, einem Attribut von <i>Documentation</i>-Objekten, realisiert.</p> |
| Consistency Check (Konsistenzprüfung) | <i>Predict</i> : Dritte Phase des <i>Coordinator Check Cycle</i> . Zu übertragende Objekte werden auf logische Konsistenz geprüft, z.B. daraufhin, dass eine Dateinummer nur einmal in einer Datenbank auftritt. Es werden die gleichen Prüfungen durchgeführt wie in den Pflegefunktionen (<i>Maintenance</i>). |
| Constants (Konstanten) | <p>Natural unterstützt die folgenden Typen von Konstanten:</p> <ul style="list-style-type: none">■ Numerische Konstante■ Alphanumerische Konstante■ Datums- und Zeit-Konstante■ Hexadezimale Konstante■ Logische Konstante■ Fließkomma-Konstante■ Attribut-Konstante■ Handle-Konstante |

| | |
|--|---|
| | Siehe <i>Benutzerkonstanten im Leitfaden zur Programmierung</i> . |
| Coordinator Check Cycle (Coordinator-Prüfzyklus) | <i>Predict</i> : Wenn Objekte mit dem <i>Predict Coordinator</i> geladen oder importiert werden, müssen Sie einen Prüfzyklus durchlaufen, der folgende Phasen umfasst: Konfliktmanagement (<i>Conflict Management</i>), Sicherheitsüberprüfung (<i>Security</i>) und Konsistenzprüfung (<i>Consistency Check</i>). Falls in einer dieser Phasen Konflikte festgestellt werden, müssen diese im <i>Main FDIC</i> oder im <i>Coordinator FDIC</i> gelöst werden, bevor die Lade- und Import-Funktion fortgesetzt werden kann. |
| Coordinator FDIC | <i>Predict</i> : Beim <i>Coordinator FDIC</i> handelt es sich um eine <i>Predict</i> -Datei, die zur zeitweiligen Speicherung von Daten benutzt wird, wenn <i>Coordinator</i> -Funktionen benutzt werden. Die Daten in der <i>Coordinator FDIC</i> -Datei werden gelöscht, sobald die Funktion erfolgreich ausgeführt worden ist. |
| Coupled Field | <i>Predict</i> : Felder, die von Feldern in <i>Standard Files</i> oder <i>Master Files</i> kopiert werden, werden gekoppelte Felder (<i>Coupled Fields</i>) genannt. |
| Cross-Reference List | <i>Predict</i> : Eine von der <i>Predict</i> -Funktion X im Menü Retrieval and Active Retrieval erzeugte Liste, die alle Objekte zeigt, mit denen ein gegebenes Objekt per <i>Association</i> verknüpft ist. |
| CSA = Common System Area / Common Storage Area | <ol style="list-style-type: none"> 1. <i>CICS</i>: A common system area is a major <i>CICS</i> storage control block that contains areas and data required for the operation of <i>CICS</i>.* <p>Siehe auch ECSA.</p> <ol style="list-style-type: none"> 2. <i>z/OS</i>: Ein gemeinsamer Speicherbereich, der für alle Adressräume in <i>z/OS</i> zur Verfügung steht. |
| CSECT = Control Section | That part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.* |
| CSV = Comma-Separated Values (Dateiformat) | Dateien mit Statistiktabellen werden im Dateiformat CVS (mit durch Komma getrennten Werten) gespeichert, um getrennte Spalten zur Bearbeitung in Tabellenkalkulationsprogrammen (<i>Spreadsheet</i>) zu erhalten. |
| CUI | Abkürzung für <i>Character User Interface</i> , d.h. zeichenbasierte Benutzeroberfläche. |
| Current Objects | <i>Predict</i> : Einige Recherchetypen (z.B. <i>Files</i> mit <i>Children</i>) liefern Berichte (<i>Reports</i>) sowohl zu den Attributen von Objekten, die die |

angegebenen Auswahlkriterien erfüllen, als auch zu Attributen von Objekten, die einen Bezug zu diesen Objekten haben. Bei der Angabe von Optionen werden die Objekte, die die angegebenen Auswahlkriterien erfüllen, als die aktuellen Objekte (*Current Objects*) bezeichnet.

Current Virtual Machine /
Current Network

Predict: Die Objekte *Network* und *Virtual Machine* dokumentieren die Hardware- und Betriebssystemumgebung eines Informationsverarbeitungssystems. *Predict* speichert eine aktuelle *Virtual Machine* und ein aktuelles *Network*, das dann benutzt wird

- als Standard-*Parent* für Datenbank- und *Virtual Machine*-Objekte,
- als Zielumgebung für die Funktionen **Generate/ Incorporate/ Compare** und einige **AOS**-Funktionen, die aus dem *Predict*-Menü **Special Functions** aufgerufen werden können.

Die Angaben für aktuelle *Virtual Machine* und aktuelles *Network* sind in den *Predict*-Standardwerten enthalten.

D

Database (Datenbank)

Predict: *Predict-Documentation*-Objekte des Typs *Database* dokumentieren eine Sammlung von physischen und/oder logischen Dateien. Wie Dateien zusammengefasst werden, um ein Objekt des Typs *Database* zu bilden, hängt vom dokumentierten Datenbanktyp ab.

Dataspace (Datenraum)

Predict: *Predict-Documentation*-Objekte des Typs *Dataspace* dokumentieren *Db2 Tablespace*s und *SQL/DS DBspace*s.

Data Set (Dataset)

Sammlung von Daten außerhalb eines Programms, d.h. eine Datei .

Data Control Language
(DCL)

Siehe [DCL](#).

Data Definition Object
(Datendefinitionsobjekt)

Predict: Externe Objekte, die mit Generierungsfunktionen erstellt wurden, z.B. für die *Adabas Compression Utility*, COBOL Copy Code, Datendefinitionsmodule (DDMs), Db2-Datenbanken, Verifizierungsregeln.

Data Definition Language

Siehe [DDL](#).

Data Definition Module -
DDM (Datendefinitionsmodul)

Siehe [DDM](#).

| | |
|---|--|
| Data Dictionary Object | <i>Predict:</i> Siehe Object . |
| Data View (Datensicht) | Eine Datensicht (<i>Data View</i>) referenziert Datenbankfelder, die in einem Datendefinitionsmodul (<i>Data Definition Module/DDM</i>) definiert sind. Eine Datensicht kann alle oder einige der Felddefinitionen im DDM enthalten. Sie wird einem <code>DEFINE DATA</code> -Statement oder einem Datenbereich (<i>Data Area</i>) definiert. |
| DB | Abkürzung für <i>Database</i> (Datenbank). |
| DBA | Abkürzung für <i>Database Administrator</i> (Datenbankadministrator). |
| DBCS=Double-Byte Character Set | Zeichensatz, bei dem jedes Zeichen durch zwei Bytes dargestellt wird. Typische Verwendung zur Darstellung von Symbolen von Sprachen (z.B. Chinesisch, Japanisch), die mehr Codes benötigen, als bei Ein-Byte-Zeichensätzen (SBCS) zur Verfügung stehen. |
| DBD = Database Description | <i>DL/I:</i> A description of the physical characteristics of a DL/I database. It defines the structure, segment keys, physical organization, names, access method, devices and other details of the database.* |
| DBID = Database Identification (Datenbankkennung) | Die Datenbanknummer. |
| DB2 = DATABASE 2 | Relationales Datenbankmanagementsystem (RDBMS) der IBM, das auf verschiedenen Plattformen verwendet wird. Die Daten werden in Tabellen (<i>Tables</i>) verwaltet und in Tabellendateien (<i>Tablespaces</i>) gespeichert. |
| DCL = Data Control Language | <i>SQL:</i> Die <i>Data Control Language</i> handhabt Datensicherheitsaspekte durch Zurverfügungstellung von Statements zum Erteilen und Widerrufen von Privilegien. |
| DCOM = Distributed Component Object Model | Komponenten-Technologie von Microsoft, die es Objekten auf verschiedenen Rechnern gestattet, über gemeinsame Protokolle, einschließlich Internet- und web-basierte Protokolle, miteinander zu kommunizieren. <i>DCOM</i> erweitert COM zu einem verteilten Komponenten-Software-Modell, das angibt, wie Software-Komponenten in einer verteilten Umgebung miteinander kommunizieren. |
| DDE = Dynamic Data Exchange | Von Microsoft definiertes Protokoll, das es verschiedenen Anwendungen ermöglicht, Daten auszutauschen. |
| DDL = Data Definition Language | Dient zum Anlegen, Ändern und Löschen von SQL-Datenstrukturen. |
| DDM = Data Definition Module | Logische Definition einer physischen Datenbankdatei, die von Natural-Objekten referenziert wird. DDMs enthalten Informationen |

über die einzelnen Felder in der Datei, die für ihre Benutzung innerhalb von Objekten relevant sind. Auf Großrechnern werden DDMs in der Systemdatei FDIC gespeichert. Unter Windows und UNIX werden sie in der Systemdatei FUSER in der relevanten Bibliothek gespeichert. Informationen zur DDM-Struktur siehe *Zugriff über Datendefinitionsmodule* im Kapitel *Datenbankzugriffe* im *Leitfaden zur Programmierung*.

Siehe Kapitel *SYSDDM Utility* in der *Editoren-Dokumentation*.

Debugger

Siehe *Natural Debugger*.

Development Server File

Datenbankdatei (zurzeit FDIC auf dem Großrechner), in der Anwendungsbeschreibungen physisch gespeichert werden. Siehe *Application Description*.

DIGITAL Command Language (DCL)

Siehe *DCL*.

Distributed Component Object Model (DCOM)

Siehe *DCOM*.

DL/I = Data Language/One

Zugriffsverfahren zur Veränderung von hierarchischen Datenbanken.

DLL

Abkürzung für *Dynamic Link Library*.

DML = Data Manipulation Language

Sprache zur Veränderung von *SQL*-Datenstrukturen.

DSECT

Abkürzung für *Dummy Control Section*.

DTD = Document Type Definition (Dokumenttypdefinition)

Schema-spezifische Spezifikationsmethode für SGML- und XML-Dokumente. DTDs sind entweder im Dokument enthalten oder gehören zur externen Untermenge des Dokuments und werden dann aus der Dokumententypdeklaration des Dokuments heraus per URL referenziert. Bekannte DTDs sind zum Beispiel DocBook, CML, IBTWSH und HTML. *dtd2html* generiert HTML-Dokumentation für SGML DTDs. Bei XML werden DTDs durch die neue XML-Schema-Spezifikationsmethode ersetzt.

Dynamic Definition

Beliebige Definition von Natural-Variablen, die nicht Teil einer `DEFINE DATA`-Statement-Klausel ist.

Dynamischer Parameter

Werden beim Start von Natural durch Angabe von Einzelparametern und/oder einer Parameterdatei zugewiesen und gelten für die aktuelle Natural-Sitzung.

Dynamic Variable (Dynamische Variable) Durch Verwendung von Variablen mit dem Attribut `DYNAMIC` können in Natural große, binäre und alphanumerische Datenstrukturen verarbeitet werden, ohne dass zur Entwicklungszeit eine Platzbeschränkung definiert werden muss. Dynamische Variablen werden ohne Längenangabe definiert. Die Speicherzuweisung erfolgt zur Ausführungszeit, und zwar entweder implizit, wenn die dynamische Variable als Zieloperand benutzt wird, oder explizit, mittels `EXPAND`- oder `RESIZE`-Statement. Dynamische Variablen können nur in einem `DEFINE DATA`-Statement definiert werden. Siehe *Dynamische Variablen im Leitfaden zur Programmierung*.

E

ECSA = Extended Common System Area z/OS: A major element of z/OS virtual storage above the 16 MB line. This area contains pageable system data areas that are addressable by all active virtual storage access spaces. It duplicates the common system area (**CSA**) which exists below the 16 MB line.*

EDIT Line (EDIT-Zeile) *Predict*: Zeile im unteren Bildschirmbereich der Funktionen **Add** (Anlegen), **Copy** (Kopieren) und **Modify** (Ändern). In dieser Zeile wird festgelegt, ob bestimmte Attribute angelegt oder geändert werden sollen. Wenn in Eingabefeldern in der **EDIT**-Zeile ein **Y** eingegeben wird, dann wird ein Predict-Editor aufgerufen.

Elementary Field (Elementarfeld) *Adabas*: Ein Feld mit einem Wert (Skalar) im Gegensatz zu einem Feld mit mehreren Werten (**multiples Feld** oder **Periodengruppe**).
Predict: Siehe **Field**. In früheren Versionen von Predict wurde der Predict-Objektyp *Field* als *Elementary Field* bezeichnet.

ELPA = Extended Link Pack Area z/OS: A major element of z/OS virtual storage above the 16 MB line. It duplicates the link pack area (**LPA**).*

Endian Mode *Windows und UNIX*: Architekturtyp, für den der Compiler-Typ **GP** generiert. Es gibt zwei Arten:

Big Endian
Das höherwertige Byte wird im Speicher an der niedrigsten Adresse abgelegt und das niederwertige Byte an der höchsten Adresse (zuerst kommt das größere Ende).

Little Endian
Das niederwertige Byte der Zahl wird im Speicher an der niedrigsten Adresse abgelegt und das höherwertige Byte an der höchsten Adresse (zuerst kommt das kleinere Ende.)

| | |
|---------------------------------|---|
| | Festgelegt durch den Session-/Profilparameter <code>ENDIAN</code> . Der <code>ENDIAN</code> -Parameter wird benutzt, um das Ausführen von portierbaren GPs zu beschleunigen. |
| EntireX Broker | Der <i>EntireX Broker</i> steuert die Kommunikation zwischen verteilten Anwendungskomponenten innerhalb der IT-Architektur. Der <i>EntireX Broker</i> unterstützt verschiedene Kommunikationsmodelle: synchron, asynchron, <i>Client/Server</i> , <i>Peer-to-Peer</i> , <i>Publish</i> und <i>Subscribe</i> . Die Nachfolgetechnologie ist <i>EntireX</i> . |
| Entire Connection | Produkt zum Abwickeln der Kommunikation zwischen Host-Rechner und PC. <i>Entire Connection</i> bietet eine Großrechner-Terminal-Emulation, eine hochrangige Anwendungsprogrammierschnittstelle, einen bedienerlosen Arbeitsstationsbetrieb, Datenkonvertierungsfunktionen und verwandte Kommunikationsaufgaben. |
| Entire DB Engine | <i>Entity Relationship</i> -Datenbank, die auf <i>Adabas</i> basiert. |
| Entire Net-Work | <i>Middleware</i> , die gemeinsame Kommunikationsdienste für Datenbankanwendungen, Abfrage-Tools, kooperative Verarbeitungsanwendungen, Anwendungsentwicklungs-Tools und sonstige Software bietet. |
| Entire Operations | Produkt zur Planung, Ausführung und Steuerung von Job-Netzwerken. Bietet alle Funktionen, die zur Verarbeitung von Jobs im Online- oder Batch-Modus erforderlich sind. |
| Entire Output Management | Produkt zur Verarbeitung beliebiger Arten von Druckausgabedaten in heterogenen Client/Server-Umgebungen, ohne zu den Anwendungen oder Programmen, mit denen die Daten erstellt wurden, wechseln zu müssen. |
| Entire System Server | Ein in sich geschlossenes Software-Paket, das Betriebssystem-Dienste in einer Natural-Umgebung zur Verfügung stellt. Wenn der Entire System Server auf Rechnern installiert ist, die mit <i>Entire Net-Work</i> verlinkt sind, unterstützt er verteilte Rechnerumgebungen, die heterogene Betriebssysteme umfassen können. |
| Entire Systems Management (ESM) | Name einer Produktlinie. |
| EntireX | Middleware-Technik für <i>Open-Enterprise Computing</i> . Bietet eine leistungsfähige Infrastruktur zum Entwickeln von verteilten Anwendungen, deren Komponenten auf verschiedenste Betriebssystemumgebungen portiert werden können. EntireX ermöglicht die Integration von Alt-Anwendungen (z.B. auf dem Großrechner) und <i>GUI Front-Ends</i> . |

| | |
|--|--|
| EntireX Broker Stub | Schnittstelle zwischen der <i>Natural RPC</i> -Laufzeit und der <i>EntireX Broker</i> -Transportschicht, die arrangierte Daten zwischen <i>Client</i> und <i>Server</i> austauscht. |
| EntireX DCOM | Produkt zur Entwicklung von DCOM -Komponenten in einer heterogenen Netzwerkumgebung. |
| EntireX Manager | Wurde ersetzt durch den <i>System Management Hub</i> . |
| Entry-Sequenced Data Set (ESDS) | Siehe <i>ESDS</i> . |
| ESDS = Entry-Sequenced Data Set | <i>VSAM</i> : A data set whose records are physically in the same order in which they were put in the data set. It is processed by addressed direct access or addressed sequential access and has no index. New records are added at the end of the data set.* |
| ET = END TRANSACTION | Natural-Statement, das abgesetzt wird, um die Beendigung einer Datenbanktransaktion zu bestätigen. Siehe auch <i>Transaction</i> und <i>BACKOUT TRANSACTION (BT)</i> . |
| Event-driven Programming (Ereignisgesteuerte Programmierung) | <i>Windows</i> : Ereignisgesteuerte Programmierung gestattet es, Anwendungen durch Eingaben zu steuern, die über die grafische Benutzerschnittstelle entgegengenommen werden. Somit hängt die Reihenfolge, in der Code ausgeführt wird, davon ab, welche Ereignisse auftreten, was wiederum davon abhängt, was der Benutzer tut. |
| Export | <i>Predict</i> : Die <i>Coordinator</i> -Funktion Export wird verwendet, um <i>Predict</i> -Daten in einer Form zu exportieren, die es gestattet, Daten mit anderen Systemen auszutauschen. Außerdem ist es möglich, Text von Attributen eines <i>Predict</i> -Objekts (z.B. erweiterte Beschreibung (<i>Description</i>) oder Unterabfrage einer Datei (<i>File</i>) oder von der Ausgabe einer anzeigeorientierten <i>Predict</i> -Funktion (z.B. Retrieval oder Active Retrieval) zu einem externen Ziel zu exportieren (z.B. <i>Natural</i> , ein <i>Con-nect</i> -Dokument oder eine <i>PC-ASCII</i> -Datei). |
| External Class (Externe Class) | <p>Eine <i>NaturalX Class</i> kann eine lokale, eine interne oder eine externe <i>Class</i> sein. Dies hängt davon ab, wie die <i>Class</i> registriert wurde. Eine externe <i>Class</i> ist eine <i>Class</i>, die als DCOM Class mit der Option <i>ES</i> (<i>ExternalSingle</i>) oder <i>EM</i> (<i>ExternalMultiple</i>) des <i>REGISTER</i>-Kommandos registriert worden ist.</p> <p>Weitere Informationen siehe <i>Local Class</i> und <i>Internal Class</i>.</p> |
| External Object (Externes Objekt) | <i>Predict</i> : Objekte von Anwendungen (Datenverarbeitungssystemen), die in <i>Predict</i> mit <i>Documentation</i> -Objekten dokumentiert werden. Externe Objekte können von <i>Documentation</i> -Objekten mit <i>Genera-</i> |

te-Funktionen generiert und *Documentation*-Objekte können von externen Objekten mit *Incorporate*-Funktionen erstellt werden. *Compare*-Funktionen werden zum Vergleichen von externen Objekten und *Documentation*-Objekten benutzt.

External Interface (Externes Interface) *Windows*: Ein externes Interface ist ein Interface, das in einem Interface-Modul definiert ist, das von der *Class* einbezogen wird.

Extract *Predict*: *Extract* ist ein vordefinierter Objekttyp in *Predict*, der zwei Hauptfunktionen erfüllt:

- Logisches Einordnen von Objekten in Gruppen,
- Erstellen eines Satzes von Objekten, die mit dem *Predict Coordinator* übertragen werden sollen.

F

FDIC *Systemdatei*, die Verweistabellendaten (*XRef Data*) und DDMs enthalten. Wenn *Predict* in Ihrer Umgebung installiert ist, dann ist FDIC die *Predict*-Diktionärdatei. Wenn *Predict* nicht in Ihrer Umgebung installiert ist, dann ist FDIC identisch mit FNAT oder FUSER und enthält auf dem Großrechner nur DDMs. FDIC wird mit dem Profilparameter `FDIC` angegeben. Siehe *Parameter-Referenz*-Dokumentation.

FDICX *VSAM*: Alternativer Indexpfad für VSAM-Basis-*Cluster FDIC*.

FDT = Field Definition Table *Adabas*: Tabelle mit Felddefinitionen einer *Adabas*-Datei. Gibt an, wie sie von *Adabas* gespeichert und benutzt werden.

Field (Feld) *Predict*: *Predict*-Objekte des Typs *Field* dokumentieren die kleinste logische Referenzeinheit innerhalb einer Datei, siehe *File*.

Field Definition Table (FDT) Siehe *FDT*.

File (Datei) *Predict*: Ein *Predict*-Objekt des Typs *File* enthält die Definition einer Sammlung von Feldern.

File Coupling (Verkoppeln von Dateien) *Adabas*: *File Coupling* dient zur Auswahl von Datensätzen aus einer Datei, die in Bezug zu Datensätzen in einer anderen Datei stehen (mit diesen verkoppelt sind). Zum Verkoppeln von Dateien werden *Adabas*-Deskriptoren benutzt. Dateien können entweder physisch (*Hard Coupling* über *ISN*-Listen) oder logisch (*Soft Coupling* durch Auswerten des Suchergebnisses dynamisch zur Laufzeit mittels `FIND`-Statement) miteinander verkoppelt werden

| | |
|---------------------------------|---|
| File Relation (Datei-Beziehung) | <i>Predict</i> : Ein Predict-Objekt des Typs <i>File Relation</i> dokumentiert eine logische oder physische Beziehung zwischen zwei <i>Files</i> , die unter Verwendung von Feldern (<i>Fields</i>) in den Dateien (<i>Files</i>) hergestellt wird. In früheren Versionen von Predict wurden <i>File Relations</i> als <i>Relationships</i> bezeichnet. Der Name wurde aus Gründen der Konsistenz mit anderen Produkten geändert. |
| FNAT | Natural- Systemdatei , in der Natural-Systemobjekte und Parameterprofile gespeichert werden. Wird mit dem Profilparameter FNAT angegeben, siehe <i>Parameter-Referenz-Dokumentation</i> . |
| FNR = File Number (Dateinummer) | Normalerweise die Dateinummer einer Natural- Systemdatei . |
| Front-End | <i>Windows und UNIX</i> : Stellt Bildschirm- und Druckerausgaben bereit. <i>Großrechner</i> : Siehe Front-End Stub . |
| Front-End Stub | <i>z/OS</i> : Eine Komponente der Natural Server-Umgebung, die vom <i>Natural Db2 Stored Procedures Server</i> und dem <i>Natural Development Server</i> benutzt wird. Siehe <i>Natural as a Server under z/OS</i> . Der <i>Front-End Stub</i> interagiert zwischen dem <i>Client/Server Protocol</i> und dem <i>Natural Server Front-End</i> . |
| FSEC | Natural Security- Systemdatei . |
| FSEQ | Abkürzung für <i>File Sequence</i> . |
| FSPool | Natural Advanced Facilities Spool-Systemdatei. Siehe auch System File . |
| FUSER | Natural-Systemdatei, in der alle von Benutzern generierten Objekte gespeichert werden. Wird mit dem Profilparameter FUSER angegeben, siehe <i>Parameter-Referenz-Dokumentation</i> . Siehe auch System File . |

G

| | |
|--|---|
| GDA = Global Data Area (Globaler Datenbereich) | Ein Natural-Objekt, das Datenelemente enthält, die von mehreren Natural-Objekten gemeinsam genutzt werden können. Weitere Informationen siehe <i>Global Data Area</i> im <i>Leitfaden zur Programmierung</i> . |
| Generation | <i>Predict</i> : Externe Datendefinitionsobjekte können aus Predict File-Objekten mit Generierungsfunktionen erzeugt werden. |

| | |
|---|--|
| Generated Program - GP (Generiertes Programm)) | <p><i>Natural</i>: Das ausführbare (katalogisierte) Objekt, das vom Compiler generiert wird.</p> <p><i>Großrechner</i>: Ein auf dem Großrechner katalogisiertes Objekt ist nur auf dem Großrechner ausführbar. Auf dem Großrechner besteht die Möglichkeit, ein katalogisiertes Objekt mit dem <i>Natural Optimizer Compiler</i> zu optimieren.</p> <p><i>Windows und UNIX</i>: Ab Natural Version 5.1 sind generierte Programme über alle Windows- und UNIX-Plattformen portierbar, die von Natural unterstützt werden (nicht auf dem Großrechner). Außerdem kann, falls erforderlich, der <i>Endian Mode</i> für ein generiertes Programm angegeben werden.</p> <p><i>Synonyms</i>: Der Begriff „katalogisiertes Objekt“ wird oft als Synonym für <i>GP</i> verwendet. In <i>Predict Application Control</i> wird der Begriff <i>loadable</i> (ladbar) verwendet.</p> <p>Siehe auch <i>Source Object</i>, <i>Object</i> und <i>Object Types</i>.</p> |
| Generation Task (Generierungs-Task) | <p><i>Predict</i>: Die Generierung von Datendefinitionsobjekten kann problemlos erfolgen, indem mehrere Generierungsaufgaben in einen Implementierungsplan aufgenommen werden. Der <i>Generation Task</i> ist der Aufruf einer Predict-Generierungsfunktion.</p> |
| Global Parameter (Globaler Parameter) | Siehe <i>Session-Parameter</i> . |
| GP | Siehe <i>Generated Program</i> . |
| GUI | Abkürzung für <i>Gaphical User Interface</i> , d.h. grafische Benutzeroberfläche. |
| GUID (Globally Unique Identifier) | Konstante, bei der sichergestellt ist, dass sie im <i>COM/DCOM Model</i> weltweit eindeutig ist. Wird verwendet, um <i>Classes</i> und deren <i>Interfaces</i> in einem <i>Network</i> eindeutig zu kennzeichnen. Wenn eine <i>Natural Class</i> als <i>DCOM Class</i> registriert werden soll, muss der <i>Class</i> und jedem ihrer <i>Interfaces</i> eine <i>GUID</i> zugewiesen werden. In Natural wird eine <i>GUID</i> durch eine alphanumerische Konstante dargestellt, die im Datenbereich-Editor (<i>Data Area Editor</i>) generiert werden kann. Der <i>Natural Studio Class Builder</i> weist jeder neuen <i>Class</i> automatisch eine neue <i>GUID</i> zu. |

H

| | |
|--|---|
| Hexadecimal Notation (Hexadezimal-Notation) | Die folgende Tabelle enthält eine Übersicht über die Hexadezimal-Notationen, die für hexadezimale Formate in der Natural-Dokumen- |
|--|---|

tation und in Dokumenten, die Natural-Add-On-Produkte betreffen, verwendet werden:

| Notation | Beispiel | Erklärung | Natural-Dokum |
|----------|----------------------------|---|--|
| H | H '5A3 ' | Verwendet in der Programmiersprache Natural (<i>Natural Programming Language</i> - NPL). | <i>Leitfaden zur Programmierung und sonstige NPL-bezogene Referenz-Dok</i> |
| X | X '5A3 ' | Am weitesten verbreitet bei Großrechner-Betriebssystemen. Verwendet in Assembler-Sprache, PL/I, Cobol, JCL, Skripten, Kommandos und an anderen Stellen. | <i>Natural for z/OS-Dokum Installation, C Monitor Inter</i> |
| 0x | 0x5A3 | Verwendet in der Programmiersprache C und deren syntaktischen Abkömmlingen. | <i>Natural for W (z.B. Installat Operations), Developmen Natural Web</i> |
| UH | UH '00C4007000660065006C ' | Verwendet in der Programmiersprache Natural (NPL - <i>Natural Programming Language</i>) zur Definition von Unicode-Konstanten. | <i>Unicode and Support-Dok Siehe U Con Abschnitt Na Format U for Data.</i> |

HF, HFS = Hierarchical File System (Hierarchisches Dateisystem)

UNIX-Dateisystem, das bei z/OS *UNIX Services* zur Verfügung steht.

HTML

Abkürzung für *Hypertext Markup Language* (Hypertext-Auszeichnungssprache).

HTTP

Abkürzung für *Hypertext Transfer Protocol*.

HTTP Cookies

HTTP Cookies stellen dem Server einen Mechanismus zur Verfügung, der es ermöglicht, Informationen über das System der Client-Anwendung zu speichern und abzurufen. Damit können web-basierte Anwendungen Informationen über gewählte Artikel, Nutzungspräferenzen, Registrierung und sonstige Informationen speichern, die später wieder abgerufen werden können. Verwendung im Statement REQUEST DOCUMENT. Weitere Informationensiehe REQUEST DOCUMENT in der *Statements*-Dokumentation.

HTTPS Abkürzung für *Hypertext Transfer Protocol Secure* (Hypertext-Übertragungsprotokoll). Stellt eine zusätzliche Schicht zwischen *HTTP* und dem *TCP/IP Protocol Stack* (TCP/IP-Protokollstapel) dar.

I

ICU = International Components for Unicode Siehe auch IBM's Website unter <http://www-01.ibm.com/software/globalization/icu/>.

ID = Identifier (Identifikator, Kennung) *Predict*: Abgesehen von einer Ausnahme werden Predict-Objekte durch zwei Attribute eindeutig gekennzeichnet: Objekttyp und Identifikator (*ID*). Objekte verschiedenen Typs können daher dieselbe ID haben. Objekt des Typs *Field* werden durch drei Attribute eindeutig gekennzeichnet: Objekttyp, Identifikator (*ID*) des *File*, zu dem sie gehören, und Identifikator des *Field* selbst. *Field*-Objekte können deshalb dieselben Identifikator haben.

IIS Abkürzung für *Internet Information Services* (Dienstplattform des Unternehmens Microsoft).

Import *Predict*: Die *Coordinator*-Funktion **Import** dient dazu, Daten von einem Übertragungsmedium in eine Predict-Umgebung zu übertragen. Ermöglicht außerdem den Import von Text aus verschiedenen externen Quellen, z.B. Natural, einer PC-ASCII-Datei oder einem Connect-Dokument, in das Text-Attribut eines Predict-Objekts, z.B. die erweiterte Beschreibung (*Description*) eines Predict-Objekts oder die Unterabfrage (*Subquery*) einer Datei (*File*).

IMS Siehe [*Information Management System*](#).

IMS DB = IMS Database Manager z/OS: Bestandteil von IMS: Das *IMS Database Management System* von IBM, das Zugang zu [*DL/I*](#)-Datenbanken ermöglicht. Siehe auch [*Information Management System*](#).

IMSDC = IMS Data Communication Obsolet. Ersetzt durch [*IMS TM*](#).

IMS TM = IMS Transaction Manager z/OS: Bestandteil von IMS: Das *IMS Online Transaction Processing System* von IBM. Siehe auch [*Information Management System*](#).

Incorporation (Eingliederung) *Predict*: Objekte im Predict-Datendiktionär können durch Eingliederung von externen Objekten (z.B. existierende Natural-Datendefinitionsmodule (DDMs) oder Adabas-Felddefinitionstabellen) erstellt werden.

| | |
|---|--|
| Information Management System (IMS) | Any of several system environments available with Database Manager (<i>IMS DB</i>) and Transaction Manager (<i>IMS TM</i>), capable of managing complex databases and terminal networks.* |
| INPL = Initial Natural Program Load | Natural Utility, die zum Laden oder Durchsuchen von Natural-Modulen oder DDMs aus <i>Datasets</i> (zum Beispiel Natural INPL-Bänder) aus der Arbeitsdatei 1 (<i>Work File 1</i>) benutzt wird. Darüber hinaus steht eine <i>Natural Security</i> -Wiederherstellungsfunktion (Recover) zur Verfügung, die es ermöglicht, eine Initialisierung der <i>Natural Security</i> -Umgebung zu erzwingen. |
| Instance (Instanz) | Beim objektorientierten Programmierungsmodell werden Datenstrukturen und Funktionen (<i>Methods</i>) zusammen in Objekte gepackt. Jedes Objekt gehört zu einer <i>Class</i> , die die interne Struktur des Objekts und seiner <i>Interfaces</i> , <i>Properties</i> und <i>Methods</i> beschreibt. Wenn ein Objekt zu einer bestimmten <i>Class</i> gehört, wird es auch als <i>Instance</i> (Instanz) dieser <i>Class</i> bezeichnet. |
| Interface (Schnittstelle) | <p><i>Interfaces</i> werden von <i>Classes</i> benutzt, um <i>Clients</i> mit <i>Services</i> (Diensten) zu versorgen. Ein <i>Interface</i> ist eine Sammlung von <i>Methods</i> und <i>Properties</i>. Ein <i>Client</i> greift auf diese Dienste zu, indem er ein Objekt der <i>Class</i> anlegt und die <i>Methods</i> und <i>Properties</i> seiner <i>Interfaces</i> benutzt.</p> <p>Ein <i>Interface</i> wird folgendermaßen definiert:</p> <ul style="list-style-type: none"> ■ Definition der <i>INTERFACE</i>-Klausel zwecks Angabe des <i>Interface</i>-Namens. ■ Definition der <i>Properties</i> des <i>Interface</i> mit <i>PROPERTY</i>-Definitionen. ■ Definition der <i>Methods</i> des <i>Interface</i> mit <i>METHOD</i>-Definitionen. |
| Interface GUID | Wenn eine <i>NaturalX Class</i> als <i>DCOM Class</i> registriert werden soll, muss der <i>Class</i> und jedem ihrer <i>Interfaces</i> eine <i>GUID</i> zugewiesen werden, um sicherzustellen, dass die <i>Interfaces</i> in einem Netzwerk zweifelsfrei identifiziert werden können. Die Zuordnung der <i>GUID</i> zu einem <i>Interface</i> erfolgt in der <i>ID</i> -Klausel des <i>INTERFACE</i> -Statement. In Natural wird eine <i>GUID</i> durch eine alphanumerische Konstante dargestellt, die im Datenbereich-Editor (<i>Data Area Editor</i>) generiert werden kann. Der <i>Natural Studio Class Builder</i> weist jeder neuen <i>Class</i> automatisch eine neue <i>GUID</i> zu. |
| Interface Inheritance (Interface-Vererbung) | Interface-Vererbung bedeutet, dass verschiedenen <i>Classes</i> dieselben <i>Interfaces</i> gegeben werden, jedoch die <i>Interfaces</i> in den verschiedenen <i>Classes</i> unterschiedlich implementiert werden. Dadurch wird es möglich, Client-Programme zu schreiben, die nur auf diesen <i>Interfaces</i> |

| | |
|---|---|
| | beruhen und in der Lage sind, mit jeder <i>Class</i> zu funktionieren, die diese <i>Interfaces</i> hat. |
| Interface Module (Interface-Modul) | Ein <i>Natural Copycode</i> -Modul, das <i>Interfaces</i> definiert. Das <i>Interface Module</i> kann in einer <i>Class</i> verwendet werden, um die darin enthaltenen <i>Interfaces</i> zu definieren. Die <i>Class</i> kann die <i>Method</i> - und <i>Property</i> -Implementierungen überschreiben, aber alle anderen Einstellungen des <i>Interface</i> werden so benutzt, wie sie im <i>Interface Module</i> definiert sind. |
| Internal Class (Interne Class) | <p>Eine <i>NaturalX Class</i> kann eine lokale, eine interne oder eine externe <i>Class</i> sein. Dies hängt davon ab, wie die <i>Class</i> registriert wurde. Eine interne <i>Class</i> ist eine <i>Class</i>, die mit der Option <i>IM</i> (<i>InternalMultiple</i>) des <i>REGISTER</i>-Kommandos als <i>DCOM Class</i> registriert worden ist. Objekte interner <i>Classes</i> können nicht durch andere Prozesse angelegt werden, auf sie kann aber durch andere Prozesse zugegriffen werden. Dazu muss das Objekt beispielsweise als Rückgabewert einer <i>Method</i> an den Client-Prozess übergeben worden sein.</p> <p>Weitere Informationen siehe Local Class und External Class.</p> |
| Internal Code (Interner Code) | <i>Predict</i> : Jeder <i>Predict</i> -Objekttyp wird intern durch einen internen Code gekennzeichnet. Interne Codes werden von <i>Predict</i> zugewiesen und können vom Benutzer nicht geändert werden. |
| Internal ID (Interne Kennung) | <i>Predict</i> : Ab <i>Predict</i> Version 3.3 haben alle Objekte eine interne Kennung (<i>ID</i>). Diese wird automatisch zugeordnet, wenn ein Objekt hinzugefügt wird. Sie ist weltweit unverwechselbar und verbleibt bei dem Objekt über dessen gesamte Lebensdauer hinweg. |
| Internal Interface (Internes Interface) | Ein <i>Interface</i> , das direkt in der <i>Class</i> definiert ist, oder ein <i>Interface</i> eines <i>Interface Module</i> , das in dem <i>Interface Module</i> definiert ist. |
| I/O Buffer = Input/Output Buffer | Ein Eingabe-/Ausgabe-Pufferspeicherbereich, der die am häufigsten benutzten Daten und Datenbeziehungen enthält. Trägt dazu bei, die physischen Ein- und Ausgabe-Aktivitäten zu minimieren und dadurch Rechnerzeit zu sparen. |
| IOCB | Abkürzung für <i>Input/Output Control Block</i> (Ein-/Ausgabe-Steuerblock). |
| IPL = Initial Program Load | Die Prozedur, die das Laden eines Betriebssystems einleitet. |
| ISAPI = Internet Server Application Programming Interface | Wird vom <i>Natural Web Interface</i> verwendet, um Internet-Dienste in <i>Natural</i> zur Verfügung zu stellen. |

ISN = Internal Sequence Number
Eine eindeutige Kennung für Datensätze variabler Länge, die in einer **Adabas**-Datenbankdatei gespeichert sind. Zu jeder ISN wird eine relative Adabas-Blocknummer (RABN) zugewiesen, die den Block angibt, in dem der Datensatz physisch gespeichert ist.

Isolated Database
Predict: Datenbank, auf die nicht mittels Adabas Star zugegriffen werden kann.

IUPD = INPL Update
Enthält *Fixes* (Programmkorrekturen) für einen **INPL**.

J

JCL = Job Control Language (Jobkontrollsprache)
Betriebssystem z/OS: Steuersprache für Stapelverarbeitungen. Dient dazu, einen **Job** und seine Anforderungen an ein Betriebssystem zu definieren.

Jobs werden z.B. benötigt, um Natural zu installieren und zu pflegen und um Natural im Batch-Modus auszuführen.

JES = Job Entry Subsystem (Jobeingabe-Subsystem)
z/OS: An IBM licensed program that receives jobs into the system and processes all output data produced by the jobs.*

Job
A unit of work defined by a user that is to be accomplished by a computer. Loosely, the term job is sometimes used to refer to a representation of a job. This representation may include a set of computer programs, files, and control statements to the operating system.*

Entire Operations: Der Job stellt einen der zentralen Objekttypen innerhalb des Entire Operations-Systems dar. Ein Job im Sinne von Entire Operations ist eine vom Benutzer definierte Aufgabe, die durch JCL-Anweisungen und Jobkennungen (Job IDs), je nach Betriebssystem unterstützte Skripte oder Dateien, Entire Operations-Unternetzwerke oder Dummy-Jobs oder Natural-Programme ausgeführt wird.

JSON
Abkürzung für JavaScript Object Notation.

K

Kennwort
Siehe **Passwort**.

Keyword (Schlüsselwort)
Natural: Siehe Für Natural reservierte Schlüsselwörter im Leitfaden zur Programmierung.

Predict: Objekte des Typs *Keyword* werden als zusätzliches Mittel zum Referenzieren von Datendiktionärobjekten verwendet. Zu einem gegebenen Objekt können bis zu 32 Schlüsselwörter zugewiesen werden.

KSDS = Key-Sequenced
Data Set

A VSAM file whose records are loaded in key sequence and controlled by an index.*

L

Label

Ein Natural-Statement kann markiert werden, indem man ihm ein *Label* voranstellt. Ein *Label* ist ein willkürlich vergebener Name. Ein mit einem *Label* markiertes Statement kann durch Angabe seines *Label* an einer anderen Stelle im Programm referenziert werden.

Large Variable (Große
Variable)

Windows und UNIX: *Large Variables* für alphanumerische und binäre Daten basierten auf den bekannten Natural-Formaten A und B. Die Einschränkung auf 253 Bytes bei Format A und 126 Bytes bei Format B entfallen damit. Das neue Limit für die Variablengröße beträgt jetzt 1 GB. Diese großen, statischen Variablen und Felder werden bezüglich Definition, Redefinition, Werteraumzuweisung, Konvertierungen, Referenzierung in Statements usw. auf die gleiche Weise wie die herkömmlichen alphanumerischen und binären Variablen und Felder behandelt. Alle Regeln, die die alphanumerischen und binären Formate betreffen, gelten ebenfalls für die großen Formate.

LDA = Local Data Area

Ein Natural-Objekt, das Datenelementdefinitionen enthält, die von mehreren Natural-Objekten benutzt werden können. Weitere Informationen siehe Abschnitt *Local Data Area* im *Leitfaden zur Programmierung*.

Library Structure (Bibliotheksstruktur)

Predict: *Predict*-Objekte des Typs *Library Structure* enthalten Systemobjekte, die die Struktur der *Natural Steplib* dokumentieren.

Link

Predict: *Predict*-Objekte des Typs *Documentation* können mittels *Associations* verknüpft werden (siehe [Association](#)).

Load (Ladevorgang)

Predict: Bei einem Load werden Daten vom [Coordinator FDIC](#) migriert. Die zu ladenden Daten müssen alle drei Phasen des *Coordinator*-Prüfzyklus (siehe [Coordinator Check Cycle](#)) durchlaufen.

Local Class (Lokale Class)

Eine *NaturalX Class* kann eine lokale, eine interne oder eine externe *Class* sein. Dies hängt davon ab, wie die *Class* registriert wurde. Eine lokale *Class* ist eine *Class*, die nicht als [DCOM Class](#) registriert worden ist. Das Anlegen von und der Zugriff auf Objekte lokaler *Classes* können nicht durch andere Prozesse, sondern nur durch Programme

in der aktuellen Natural-Sitzung erfolgen. Weitere Informationen siehe [Internal Class](#) und [External Class](#).

| | |
|--|---|
| Logical File (Logische Datei) | <i>Predict</i> : Eine logische Datei muss nicht notwendigerweise Informationen über die physische Implementierung der Datei enthalten. Ihre Definition einer logischen Datei ist im Grunde die Definition der Dateistruktur. |
| LPA = Link Pack Area | <i>z/OS operating systems</i> : A major element of z/OS virtual storage below the 16MB line. The storage areas that make up the LPA contain all the common reentrant modules shared by the system. The LPA provides economy of real storage by sharing one copy of the modules, protection because LPA code cannot be overwritten even by key 0 programs, and reduced path length because the modules can be branched to. The LPA is duplicated above the 16MB line as the extended link pack area (ELPA).* |
| LSO = Library Search Order (Suchreihenfolge in Bibliotheken) | Die Reihenfolge, in der Natural beim Laden eines Natural-Objekts durchsucht. |
| LU = Logical Unit (Logische Einheit) | Die zugrunde liegende Netzwerk-Entität, die dem Benutzer Zugang zu einem Netzwerk ermöglicht, das auf der <i>Systems Network Architecture</i> (SNA) basiert. |

M

| | |
|--------------------------------|---|
| Main FDIC | <i>Predict</i> : Die Datei <i>Main FDIC</i> ist die Zielumgebung einer <i>Load</i> - oder <i>Import</i> -Operation mit dem <i>Predict Coordinator</i> . |
| Map (Maske) | Ein Bildschirm-Layout (Maske), auf das in einem Programm Bezug genommen wird. Eine Maske liefert Formatierungsinformationen für Bildschirm-Layouts, die in einem Programm referenziert werden, definiert Ein- und Ausgabefelder und weist diesen Feldern Programmvariablen zu. Natural-Objekte des Typs <i>Map</i> werden mit dem Masken-Editor (<i>Map Editor</i>) angelegt. Siehe <i>Natural-Editoren-Dokumentation</i> . |
| Mapping | <i>Single Point of Development</i> : In einer Remote-Entwicklungsumgebung können Sie einen Entwicklungsserver per <i>Mapping</i> auf ihre Windows-Sitzung abbilden und dann Verbindung zu Anwendungen herstellen. |
| Master Field | <i>Predict</i> : Feld in einem <i>Master File</i> . |
| Master File (Definitionsdatei) | <i>Predict</i> : Eine Datei, von der eine Sicht (<i>View</i>) erstellt worden ist. Ein <i>Master File</i> stellt die Zwischenebene der Datenhierarchie innerhalb |

| | |
|---|--|
| | von Predict dar: Die <i>Master File</i> -Felder können Attribute von Standardfeldern erben und mit den Attributwerten in einem <i>Master File</i> werden <i>Views</i> und <i>Userviews</i> auf tieferen Ebenen automatisch aktualisiert (<i>Rippling</i>). |
| Member (Teildatei) | <i>Predict</i> : Natural- oder 3GL-Objekt, für das Verweistabellendaten (<i>XRef Data</i>) existieren. |
| Message Processing Program (MPP) | Siehe MPP . |
| Metadata Structure (Metadaten-Struktur) | <i>Predict</i> : Die Struktur des Predict-Datendiktionärs besteht aus Objekttypen, deren Attributen und assoziierten Typen. Die Struktur kann erweitert werden, indem man neue Objekttypen und Assoziationstypen mit den in Predict vorhandenen Metadata Administration -Funktionen definiert. Die in Predict mitgelieferten vordefinierten Objekttypen können nicht geändert werden. |
| Method | <i>Objektorientierte Programmierung</i> : Eine Funktion, die ein Objekt oder eine <i>Instance</i> (Instanz) einer <i>Class</i> auf Anforderung von einem Client ausführen kann. |
| MPP = Message Processing Program | An IMS application program that is driven by transactions and has access to online IMS databases and message queues.* |
| Multiple-Master View | <i>Predict</i> : Eine Datensicht (<i>View</i>), die von einem oder mehreren <i>Master Files</i> abgeleitet werden kann, z.B. <i>ORACLE View</i> . |
| Multiple-Value Field (Multiples Feld) | <i>Adabas</i> : Ein Feld, das mehrere Werte (oder Ausprägungen) haben kann. Die Anzahl der Ausprägungen eines jeden multiplen Felds kann bis 65.534 Werte oder Ausprägungen in einem einzelnen Datensatz haben. Die maximale Anzahl an Ausprägungen ist abhängig von der Adabas-Version und der Definition der <i>File Definition Table</i> (FDT). In Natural erfolgt der Zugriff auf ein multiples Feld in Adabas dadurch, dass man es als Array definiert. |

N

| | |
|-----------------------------|--|
| NATPARM | <i>Großrechner</i> : Platzhalter für den Namen des Natural- Parametermoduls , das vom Benutzer definiert wird. <i>Windows und UNIX</i> : Siehe Parameterdatei |
| Natural Advanced Facilities | <i>Natural Advanced Facilities</i> besteht aus dem Spool- und Report-Verwaltungssystem <i>NATSPOOL</i> , das es gestattet, Natural-Programmausgaben in einem Spool-System zwischenspeichern und dann physischen Druckern zuzuleiten. |

| | |
|---|--|
| Natural Command (Natural-Kommando) | Natural-Kommandos werden verwendet, um während einer Natural-Sitzung Funktionen auszuführen. Siehe Abschnitt <i>Kommandos und Menü-Funktionen benutzen</i> im Dokument <i>Natural benutzen</i> . |
| Natural Command Processor - NCP (Natural-Kommandoprozessor) | Besteht aus den Komponenten <i>Maintenance</i> (Verwaltung) und <i>Runtime</i> (Laufzeit). Die Utility SYSNCP, siehe <i>Debugger und Dienstprogramme (Utilities)</i> -Dokumentation, ist dabei der Verwaltungsteil, der alle Funktionen zum Definieren und Steuern der Navigation innerhalb einer Anwendung umfasst. Das Statement <code>PROCESS COMMAND</code> (Siehe <i>Statements</i> -Dokumentation) ist der Laufzeitteil mit dem Natural-Programme aufgerufen werden. |
| Natural Configuration Utility | <i>Windows und UNIX</i> : Die <i>Natural Configuration Utility</i> dient zur Pflege folgender Dateien: <ul style="list-style-type: none">■ Globale und lokale Konfigurationsdateien.■ Natural-Parameterdateien. |
| Natural Construct | Ein Anwendungsgenerator, der verschiedenste, äußerst flexible Anwendungsstrukturschablonen generiert und zahlreiche Aspekte der Natural-Software-Produktion automatisiert. Kann sowohl von IT-Experten als auch von Nicht-Experten benutzt werden. |
| Natural Construct Spectrum | Gestattet den Zugang zu Natural auf dem Großrechner aus einer Windows-Umgebung. Unter Verwendung von <i>Natural Construct Spectrum</i> und dem <i>Software Development Kit (SDK)</i> können Anwendungsentwickler alle Komponenten einer Client/Server-Web-Anwendung erstellen, die Funktionen zur Pflege und Anzeige ausführen, sowie Dialoge mit grafischer Benutzeroberfläche (GUI) oder Webseiten, die mit Subprogrammen dieser Natural-Objekte kommunizieren. Die Kommunikation zwischen Server- und Client-Komponenten einer Anwendung erfolgt mittels einer Kombination von <i>Entire Broker</i> und <i>Entire Net-Work</i> (oder <i>Entire Broker</i> , konfiguriert für die Benutzung von TCP/IP) sowie mit Hilfe der Middleware-Komponenten von Construct Spectrum. |
| Natural Debugger | <i>Großrechner/Windows/UNIX</i> : Der <i>Natural Debugger</i> dient folgenden Zwecken: <ul style="list-style-type: none">■ Temporäre Steuerung oder Beeinflussung des Programmflusses einer Natural-Anwendung durch Ändern von Variablen.■ Prüfung des Programmflusses mittels einer Historie der Aufrufe.■ Feststellen von logischen Anwendungsfehlern in einem Natural-Programm durch Überprüfen des Inhalts seiner Variablen, unter |

Verwendung von bedingten Programmstopps (*Break Points*) oder Bedingungen (*Conditions*) für die Programmunterbrechung.

- Permanente Beobachtung von Variablen.

Der Debugger kann auch benutzt werden, um von anderen Entwicklern geschriebene Programme leichter zu verstehen, eigene Anwendungen schneller zu entwickeln und die Logik von Natural besser zu verstehen, was für Erstbenutzer hilfreich sein kann.

Natural Development Server

Gestattet es, die Entwicklungsumgebung von Natural Studio per Mapping auf eine Natural-Remote-Umgebung abzubilden. Dadurch ist es möglich, Natural-Anwendungen in Remote-Umgebungen zu entwickeln und zu testen, ohne den gemeinsamen Arbeitsbereich von Natural Studio zu verlassen.

Weitere Informationen siehe *Natural Development Server*-Dokumentation.

Natural Engineer

Großrechner/Windows: Tool, das es gestattet, Natural-Code zu prüfen, zu pflegen und die Erfüllung definierter Anforderungen sicherzustellen. Ermöglicht Reengineering und Wartung von Natural-Anwendungen mit Blick auf die Zukunft, durch Anwendung von Standards, Aufrechterhaltung von Flexibilität und Vorbereitung auf die Übernahme neuer Technologien.

Natural for Ajax

Natural for Ajax ermöglicht das Erstellen von umfangreichen Internet-Anwendungen, die Ajax-Technologien verwenden (*Ajax = Asynchronous JavaScript and XML*). Dies ermöglicht Natural-Benutzern auf Windows-, UNIX- und Großrechner-Plattformen die Entwicklung und Benutzung von Natural-Anwendungen mit Benutzerschnittstellen auf Browser-Basis (ähnlich wie bei GUI-Arbeitsplatzanwendungen).

Natural ISPF (Integrated Structured Programming Facility)

Natural ISPF ist ein Anwendungsentwicklungs-Tool zum Erstellen, Testen und Pflegen von Anwendungen über den gesamten Produktlebenszyklus.

Natural Nucleus (Natural-Nukleus)

Hauptfunktionskomponente von Natural, die den Systemkern (*Kernel*) umfasst.

Großrechner: Besteht aus dem umgebungsunabhängigen und dem umgebungsabhängigen Nukleus. Der umgebungsunabhängigen kann von mehreren Großrechner-Betriebs- und TP-Systemen benutzt werden. Der umgebungsabhängigen Nukleus enthält Komponenten, die von den Betriebs- und TP-Systemen abhängig sind.

| | |
|-------------------------------------|--|
| | <i>Windows und UNIX:</i> Der einzige Teil von Natural, der vom Betriebssystem abhängig ist. |
| Natural Object Handler | Verarbeitet Objekte zur Verteilung von Anwendungen. Dazu werden die Objekte in der Quellumgebung in Arbeitsdateien entladen und danach aus den Arbeitsdateien in die Zielumgebung entladen. Der <i>Natural Object Handler</i> besteht aus der Utility <i>SYSOBJH</i> , die sich in der Bibliothek <i>SYSOBJH</i> befindet, und der Direktkommando-Schnittstelle. Weitere Informationen siehe <i>Allgemeine Informationen zum Object Handler</i> in der <i>Natural Object Handler</i> -Dokumentation. |
| Natural Optimizer Compiler | <i>Großrechner:</i> Ein <i>Natural Compiler</i> , der, soweit dies möglich ist, Maschinencode generiert (andernfalls generiert er Pseudo-Code). Siehe <i>Natural Optimizer Compiler</i> -Dokumentation. |
| Natural Profiler | <p>Der <i>Natural Profiler</i> ist ein Tool für die dynamische Analyse von Natural-Anwendungen. Immer wenn ein Natural-Ereignis auftritt, sammelt das Tool Profilierungsdaten, z.B. wenn ein Programm gestartet wird oder vor einem Datenbankaufruf. Der <i>Natural Profiler</i> visualisiert die Ereignisdaten in <i>NaturalONE</i> als Ereignisverfolgung (<i>Event Trace</i>). Darüber hinaus werden der Zeitverbrauch und die Häufigkeit der Anwendung der ausgeführten Objekte, Natural-Statements und Programmzeilen angezeigt. Eine mit dem <i>Natural Profiler</i> vorgenommene Analyse ist die Basis für die Leistungsoptimierung einer Natural-Anwendung.</p> <p>Der <i>Natural Profiler</i> ist für UNIX-, Windows- und Großrechnerplattformen verfügbar.</p> |
| Natural Roll Server | <p><i>z/OS:</i> Der <i>Natural Roll Server</i> wird benutzt, um den den Inhalt einer Natural-Sitzung bei bestimmten Ereignissen zu speichern:</p> <ul style="list-style-type: none"> ■ bei Terminal-Ein-Ausgaben in den TP-Monitor-Umgebungen CICS und IMS TM, ■ beim Warten auf die nächste Client-Anforderung in einem NDV-, NWO- oder RPC-Server mit Front-End unter CICS. <p>Siehe auch <i>Natural Roll Server Functionality</i> und <i>Natural Roll Server Operation</i> in der <i>Operations</i>-Dokumentation.</p> |
| Natural RPC (Remote Procedure Call) | Als Implementierung von RPC-Techniken in einer Natural-Umgebung ermöglicht es der Natural RPC, Natural-Subprogramme, Anwendungen, Prozeduren und Objekt-Methoden auf einem RemoteRechner über den EntireX Broker aufzurufen. |

| | |
|---------------------------------------|---|
| | <p>Die Verbindungen zwischen <i>Client</i>, <i>Server</i> und <i>Services</i> (Dienste) werden für jeden <i>Natural Client</i> mit der Natural Utility SYSRPC konfiguriert.</p> <p>Weitere Informationen siehe <i>Natural RPC (Remote Procedure Call)</i>-Dokumentation.</p> |
| Natural Runtime (Natural-Laufzeit) | <ol style="list-style-type: none">1. Die <i>Natural Runtime</i> stellt die Umgebung für die Ausführung von Natural-Anwendungen zur Verfügung.2. Die virtuelle Maschine, die den Natural-Code interpretiert. |
| Natural Security | <p>Umfassendes Sicherheitssystem, mit dem Zugriffe auf die Natural-Umgebung gesteuert, geprüft und verhindert werden können.</p> <p>Weitere Informationen siehe <i>Natural Security</i>-Dokumentation.</p> |
| Natural Session (Natural-Sitzung) | <p>Eine Natural-Sitzung stellt den benutzerspezifischen Natural-Laufzeitkontext dar, der erforderlich ist, damit das Natural-Laufzeitsystem Natural-Programme ausführen kann.</p> |
| Natural Studio | <p><i>Windows</i>: Die GUI-Entwicklungsumgebung von Natural. In <i>Natural Studio</i> sind die früher separaten Editoren und Tools in einer einfach zu benutzenden Arbeitsumgebung integriert.</p> |
| Natural Web Interface | <p>Natural Web Interface bietet komfortablen Web-Zugang für existierende Natural-Anwendungen. Das <i>Natural Web Interface</i> ist ein Bindeglied zwischen einem <i>Web Server (HTTP Server)</i> und Ihrer Natural-Umgebung. Kann auf einer separaten Maschine (z.B. Großrechner) oder auf derselben Maschine wie der <i>HTTP Server</i> (z.B. <i>Netscape Communication Server</i> oder <i>Microsoft IIS</i>) liegen.</p> |
| Natural Web Adaptor Server Extensions | <p>Implementierungen verschiedener Server-Schnittstellen als Teil eines <i>Natural Web Interface</i>, z.B. <i>CGI</i>, <i>ISAPI</i> und <i>NSAPI</i>.</p> |
| NaturalX | <p>Wird verwendet, um objektbasierte Natural-Anwendungen zu schreiben. Auf Windows-Plattformen gestattet es <i>NaturalX</i>, verteilte Natural-Anwendungen unter Verwendung verteilter Objekt-Techniken (zurzeit DCOM) zu erstellen.</p> |
| NaturalX Class | <p>Ausgehend vom objektbasierten Programmierungsansatz werden <i>NaturalX Classes</i> benutzt, um Datenstrukturen (<i>Objects</i>) mit entsprechender Funktionalität (<i>Methods</i>) zu kapseln.</p> <p>Die interne Struktur der Objekte eines <i>Class Object</i> wird mittels einer <i>Object Data Area</i> definiert. Die <i>Methods</i> einer <i>Class</i> werden als Unterprogramme (<i>Subprograms</i>) implementiert.</p> |

NaturalX Classes können mit dem Natural-Kommando REGISTER bei [DCOM](#) bekannt gemacht werden. Danach kann auf sie in einem Netzwerk (*Network*) zugegriffen werden.

Classes können *internal*, *external* oder *local* sein.

| | |
|--|--|
| NaturalX Client | Ein <i>NaturalX Client</i> ist ein Prozess, der <i>NaturalX Objects</i> anlegt oder auf sie zugreift. |
| NaturalX Server | Ein <i>NaturalX Server</i> ist ein Prozess, der eine Natural-Sitzung verwaltet. Die von einem <i>NaturalX Server</i> verwaltete Natural-Sitzung wird benutzt, um <i>COM Objects</i> aufzunehmen. |
| Network (Netzwerk) | <p><i>Entire Operations</i>: Ein Netzwerk, das einen Satz Jobs und allgemeiner Verarbeitungsanweisungen umfasst, um eine bestimmte Aufgabe zu erledigen.</p> <p><i>Predict</i>: Netzwerke definieren zusammen mit Predict-Objekten des Typs <i>Virtual Machine</i> die Hardware- und Betriebssystemumgebung eines Datenverarbeitungssystems.</p> |
| Node (Knoten) | <p><i>Entire Operations</i>: Ein Knoten (<i>Node</i>) ist eine Server-Definition, die sich auf die Maschine bezieht, auf der Anforderungen an das Betriebssystem ausgeführt werden.</p> <p><i>Predict</i>: Predict-Objekte des Typs <i>Node</i> werden zusammen mit Objekten des Typs <i>Server</i> benutzt, um <i>Remote Procedure Calls</i> zu dokumentieren.</p> |
| Node Name (Knotenname) | <p><i>Entire Operations</i>: Ein logischer Kurz- oder Langname für einen numerischen Betriebssystem-Serverknoten.</p> <p><i>Natural RPC</i>: Der Name des Knotens, an den der Remote CALLNAT gesendet wird. Beim <i>EntireX Broker</i> ist der Knotenname der Name des <i>EntireX Broker</i> gemäß Definition im Feld BROKER-ID in der <i>EntireX Broker</i>-Attribut-Datei im Feld BROKER-ID. Siehe auch Natural RPC.</p> |
| NSAPI = Netscape Application Programming Interface | Wird vom Natural Web Interface benutzt, um Internet-Dienste Natural zur Verfügung zu stellen. |

O

| | |
|-----------------|--|
| Object (Objekt) | <i>Objektorientierte Programmierung</i> : Beim objektorientierten Programmierungsmodell werden Datenstrukturen und Funktionen (so genannte <i>Methods</i>) gemeinsam in Objekte gepackt. Jedes Objekt |
|-----------------|--|

gehört zu einer *Class*, die die interne Struktur des Objekts, seiner *Interfaces* und *Methods* beschreibt.

Entire Operations: Ein Einzelbestandteil eines Job-Netzwerks, das erforderlich ist, um die Verarbeitung von Jobs zu verwalten, auszuführen oder zu steuern. Grundlegende Objekte sind Netzwerke, Jobs, Zeitpläne und Eigentümer.

Predict: In Predict werden Informationen in folgenden Objekten gespeichert: Datendiktionärobjekt, Dokumentationsobjekt, Predict-Objekt. Datendiktionärobjekte werden auch als *Predict Objects* oder *Documentation Objects* bezeichnet. Siehe auch [Object Types \(Objekttypen\)](#).

Natural: Siehe auch [Cataloged Object](#), [Source Object](#) und [Object Types](#).

Object Data Area (Objektdatenbereich)

Objektorientierte Programmierung: Im Objektdatenbereich werden die aktuellen Werte aller *Properties* eines Objekts gespeichert. Andere Variablen, die für Clients nicht als *Properties* zugänglich sind, können ebenfalls im Objektdatenbereich definiert werden. Diese Variablen werden von den *Methods* des Objekts benutzt, um den internen Status des Objekts zu verwalten. Die Struktur des Objektdatenbereichs aller Objekte einer *Class* wird in der `OBJECT USING`-Klausel des `DEFINE CLASS`-Statements definiert. Ein Objektdatenbereich wird mit dem Datenbereich-Editor (*Data Area Editor*) als lokaler Datenbereich (*Local Data Area*) angelegt.

Object Data Variable (Objektdatenvariable)

Objektorientierte Programmierung: Jede *Property* benötigt eine Variable im Objektdatenbereich der *Class*, um deren Wert zu speichern. Diese wird als die gespeicherte Objektdatenvariable bezeichnet. Außerdem können noch andere Variablen im Objektdatenbereich definiert werden, die für Clients nicht als *Properties* zugänglich sind, sondern nur von den *Methods* der *Class* benutzt werden.

ODA

Siehe [Object Data Area](#).

Object Locking (Objektsper-
rung)

Windows: Verhindert zeitgleiches Ändern von Programmen in einer *Remote Development*-Umgebung.

Object Types (Objekttypen)

Natural: Beispiele für Natural Objekttypen sind: *Program*, *Map*, *Copycode*, *Text*, *Subprogram*, *Helproutine*, *Subroutine*, *Class*, *Data Areas* (*Global*, *Local*, *Parameter*), *Dialog* und *Function*.

Weitere Informationen siehe Kapitel *Objekte zum Erstellen und Pflegen von Natural-Anwendungen* im Leitfaden zur Programmierung.

Predict: In Predict werden Informationen in folgenden Objekten gespeichert: Datendiktionärobjekt, Dokumentationsobjekt, Predict-Objekt. Datendiktionärobjekte werden auch als *Predict Objects* oder *Documentation Objects* bezeichnet. Predict-Dokumentationsobjekte haben einen bestimmten Typ, z.B. *Database* oder *Program*. Zusätzliche Objekttypen können mit den *Predict Metadata Administration*-Funktionen definiert werden. Jeder Objekttyp hat seine eigenen typspezifischen Attribute.

Object Type Code (Objekttypcode)

Natural: Zu jedem Natural-Programmierobjekttyp gibt es einen unverwechselbaren Objekttypcode:

| Natural-Objekttypcode | Bedeutung |
|-----------------------|------------------------|
| 3 | Dialog |
| 4 | Class |
| 5 | Command Processor |
| 7 | Function |
| 8 | Adapter |
| 9 | Resource |
| A | Parameter Data Area |
| C | Copycode |
| D | Data Definition Module |
| G | Global Data Area |
| H | Helproutine |
| L | Local Data Area |
| M | Map |
| N | Subprogram |
| O | Macro |
| P | Program |
| R | Report |
| S | Subroutine |
| T | Text |
| Z | Recording |

Predict: Jeder Objekttyp innerhalb der Predict-Metastruktur wird durch einen unverwechselbaren Objekttypcode gekennzeichnet, der aus zwei Buchstaben besteht (z.B. *FI* für *File*).

Operating System (Betriebssysteme)

Siehe [Betriebssysteme](#).

Operand

Ein Operand kann Folgendes sein:

- Konstante
- Benutzervariable oder benutzerdefinierte Function
- Feld einer Datensicht (*View Field*)
- Systemvariable oder Systemfunktion

Operanden werden in Verbindung mit den Natural-Statements benutzt. Weitere Informationen siehe *Operandentabelle* in der *Statements*-Dokumentation:.

| | |
|----------------------------|--|
| Output Mode (Ausgabemodus) | <i>Predict: Output Mode</i> und <i>Retrieval Type</i> bestimmen gemeinsam, wie die Daten des Dictionärs durch die <i>Predict Retrieval</i> -Funktionen ausgewertet werden. Zu beachten ist, dass nicht alle <i>Output Modes</i> für alle <i>Retrieval Types</i> verfügbar sind. |
| Owner (Eigentümer) | <i>Entire Operations</i> : Der Eigentümer eines Job-Netzwerks. Jedem <i>Entire Operations</i> -Benutzer wird ein oder mehrere Eigentümer zugewiesen. Der Benutzer kann nur Netzwerke eines ihm zugewiesenen Eigentümers verwalten. <i>Predict</i> : Ein Eigentümer ist im Wesentlichen eine Gruppe von einem oder mehreren Benutzern. Ein Eigentümer kann beispielsweise für eine organisatorische Einheit stehen. Verantwortlichkeiten können in <i>Predict</i> dokumentiert werden, indem ein Eigentümer der Eigentümerliste eines Benutzers und derselbe Eigentümer der Eigentümerliste eines Objekts zugewiesen wird. |

P

| | |
|------------------|--|
| Packagelist | <i>Predict</i> : <i>Predict</i> -Objekte des Typs <i>Packagelist</i> dokumentieren <i>Db2 Packages</i> . |
| Parallel Sysplex | <ol style="list-style-type: none">1. A set of z/OS systems that communicate and cooperate with each other through multisystem hardware components and software services to process customer workloads.*2. A sysplex that uses one or more coupling facilities.* |
| Parameter | Siehe folgende Glossar-Einträge und Verlinkungen: <ul style="list-style-type: none">■ Dynamischer Parameter■ Natural Configuration Utility■ Parameterdatei■ Parametermodul |

- [Druckerprofil](#)
- [Profile](#)
- [Profilparameter](#)
- [Profilparameter-Hierarchy](#)
- [SYSPARM Utility](#)
- [Session-Parameter](#)
- [Statischer Parameter](#)

Natural RPC (Remote Procedure Call): Alle [Natural RPC](#)-Parameter sind in der umgebungsspezifischen *Natural-Parameter-Referenz-Dokumentation* dokumentiert. Bei Natural for z/OS sind diese Parameter im NTRPC-Makro (statische Definition) enthalten oder werden mit dem Profilparameter `RPC` (dynamische Definition) definiert.

Parameter File (Parameter-datei)

Großrechner: Siehe [Parametermodul](#).

Windows und UNIX: Standardmäßig befinden sich die Parameterangaben in der Parameterdatei `NATPARM.SAG` und werden benutzt, um die Eigenschaften Ihrer Natural-Umgebung zu bestimmen. Anfangs enthält die Parameterdatei die ausgelieferten Standardwerte. Wenn Sie Natural nicht mit den Standard-Parameterwerten, sondern mit anderen Werten benutzen möchten, können Sie mit der [Natural Configuration Utility](#) die Standard-Parameterdatei `NATPARM.SAG` ändern und/oder Ihre eigene(n) Parameterdatei(en) erstellen. Die Namen der Parameterdateien müssen 8 Zeichen lang sein und die Namensweiterung `.SAG` haben.

Parameter Module (Parametermodul)

Windows und UNIX: Siehe [Parameterdatei](#).

Großrechner: Enthält alle Profilparameter-Einstellungen, die zum Konfigurieren einer Natural-Umgebung erforderlich sind. Ein Natural-Parametermodul wird während des Installationsvorgangs angelegt und generiert.

Weitere Informationen siehe *Natural Parameter Module* und *Building a Parametermodul* in der *Operations-Dokumentation*.

Parent

Predict: Associations werden verwendet, um zu dokumentieren, wie Predict-Objekte mit anderen Predict-Objekten in Beziehung stehen. Eine *Association* wird hergestellt, indem Objekte mit einem Objekt entweder als *Child* oder als *Parent* verknüpft werden. Beispielsweise ist eine Datenbank *Parent* eines *File*, ein *File* ist *Parent* eines *Field*.

| | |
|---|--|
| Passive Cross-References (Passive Querverweise) | <i>Windows - XRef Evaluation:</i> Die Funktion Passive Cross-References zeigt an, welche Objekte das aktuelle Objekt benutzen. So kann man z.B. feststellen, in welchen Teilen einer Anwendung ein bestimmter <i>Copycode</i> verwendet wird. Die Ergebnisse werden in einer Baumstrukturansicht dargestellt, wobei das referenzierte („benutzte“) Objekt an oberster Stelle angezeigt wird. Bei einigen Objekttypen (z.B. <i>Copycodes</i> , <i>DDMs</i> und <i>Methods</i>) gibt es standardmäßig nur passive Querverweise. |
| Password | <i>Allgemein:</i> Kennwort . <i>Natural:</i> Passwort . Eine Folge von Zahlen, Buchstaben und Sonderzeichen, die der Benutzer beim Anmelden eingeben muss. |
| Passive Help (Passive Hilfe) | <i>Predict:</i> Passive Hilfe liefert beschreibende Informationen zu Funktionen. Kontextspezifische Online-Hilfe kann durch Eingabe eines Fragezeichens (?) in das Feld Retrieval Type/Function eines Menüs oder aus dem Help Main Menu angezeigt werden. |
| PCB= Program Communication Block | DL/I or IMS control block that describes an application program's interface to a DL/I or IMS database or, additionally, for message processing and batch message processing programs, to the source and destination of messages. Siehe auch PSB = Program Specification Block .* |
| PDA = Parameter Data Area (Parameter-Datenbereich) | Ein Natural-Objekt, das benutzt wird, um die Datenelemente zu definieren, die als Parameter an ein <i>Subprogram</i> , eine externe <i>Subroutine</i> oder eine <i>Helproutine</i> übergeben werden. Weitere Informationen siehe <i>Parameter Data Area</i> im <i>Leitfaden zur Programmierung</i> . |
| Periodic Group (Periodengruppe) | <i>Adabas:</i> Eine Periodengruppe definiert eine Gruppe von Feldern (wobei es sich um Elementarfelder (Elementary Fields) und/oder multiple Felder (Multiple-Value Fields) handeln kann), die sich innerhalb eines Datensatzes wiederholen können. Die durch eine Periodengruppe definierten Felder können, abhängig von der Adabas-Version und FDT-Definition, bis zu 65.534 mal pro Datensatz wiederholt werden. In Natural erfolgt der Zugriff auf eine Periodengruppe in Adabas, indem alle Felder als getrennte <i>Arrays</i> definiert werden oder indem eine Gruppe als <i>Array</i> definiert wird. |
| Physical File (Physische Datei) | <i>Predict:</i> Siehe Master File . |
| Placeholder (Platzhalter) | <i>Predict:</i> Wenn ein Objekt, das mit einem anderen Objekt verlinkt ist, geladen oder importiert wird und dabei das referenzierte Objekt nicht geladen oder importiert wird und in der Zielumgebung nicht existiert, wird in der Zielumgebung ein Platzhalter (<i>Placeholder</i>) für |

das referenzierte Objekt hinzugefügt. Zweck dieses Platzhalters ist es, die Objektkennung (*Object ID*) des referenzierten Objekts in der Zielumgebung zu reservieren, damit die in der alten Umgebung vorhandene Verknüpfung in der neuen Umgebung zu einem späteren Zeitpunkt wiederhergestellt werden kann.

Ein *Placeholder* enthält folgende Informationen:

- *Object ID* (Objektkennung)
- *Internal ID* (interne Kennung)
- *Subtype* (Untertyp), falls zutreffend)
- *Transfer status placeholder* (Platzhalter für den Übertragungsstatus)

Plattform

Siehe *Betriebssysteme*.

Alle von Natural und Natural-Add-on-Produkten unterstützten Betriebssysteme sind im Abschnitt *Product Version Availability* auf der *Empower Website* aufgeführt, siehe <https://empower.softwa-reag.com/>.

PLOG = Protection Log (Schutzprotokollierung)

Adabas: Datei, in der Vorher- und Nachher-Abbilder von Datensätzen und anderen Elementen aufgezeichnet werden, wenn an einer Adabas-Datenbank Änderungen vorgenommen werden. Wird verwendet, um nach einem Neustart die Datenbank wiederherzustellen (bis zu letzten beendeten Transaktion oder *ET*).

Plug-in Manager

Windows: Die Natural Studio-Benutzerschnittstelle kann durch *Plug-Ins* erweitert werden, die mit dem *Plug-in Manager* aktiviert und deaktiviert werden können. Ein Teil der Funktionalität von *Natural Studio* selbst wird in Form von *Plug-ins* geleifert. Ein Muster-Plug-In wird in Quellcode in der Bibliothek SYSEXPLG ausgeliefert.

Portable GPs (Portierbare generierte Programme)

Windows und UNIX: GPs, die mit Natural ab Version 5 katalogisiert werden, sind portierbar über jede von Natural unterstützte UNIX- und Windows-Plattform und können dort ohne erneute Kompilierung ausgeführt werden. Siehe auch *Generated Programs*.

Predict

Datendiktionär für die Entwicklung mit Sprachen der vierten Generation, insbesondere Natural. Es handelt sich um einen Zentralspeicher für Anwendungs-Metadaten, dokumentiert und stellt Verweistabellendaten (*XRef Data*) zur Verfügung. Mit Predict ist es möglich, Code automatisch anhand von Definitionen zu generieren und so die Produktivität hinsichtlich Entwicklung und Pflege zu erhöhen.

| | |
|--------------------------------------|---|
| Predict Coordinator | <i>Predict</i> : Eine Utility, die den Austausch von Daten zwischen verschiedenen FDIC-Dateien und zwischen Predict und der <i>Natural Engineering Workbench</i> ermöglicht. Der <i>Predict Coordinator</i> benutzt eine eigene FDIC-Datei, die als <i>Coordinator FDIC</i> bezeichnet wird und als Zwischenspeicher dient. Die Anwendung, die die <i>Coordinator</i> -Funktionen enthält, liegt in der Natural-Bibliothek <code>SYSDICBE</code> . |
| Printer Profile (Druckerprofil) | <i>Windows und UNIX</i> : Profilinformatoren für Drucker werden in der globalen Konfigurationsdatei gespeichert, die die ausgelieferten Druckausgabe-Standard Einstellungen enthält. Die Einstellungen in der globalen Konfigurationsdatei können mit der <i>Natural Configuration Utility</i> geändert werden. |
| Process (Prozess) | Ein Betriebssystemprozess ist ein Arbeitselement, das einen eigenen Speicherraum, Code, Daten und andere Betriebssystem-Ressourcen hat und aus einem oder mehreren <i>Threads</i> besteht. |
| Processing Rule (Verarbeitungsregel) | Eine für ein Maskenfeld (<i>Map Field</i>) definierte Verarbeitungsregel, die den Inhalt des Feldes prüft und abhängig vom vorhandenen Inhalt reagiert. Die Prüfung kann auch mehrere Felder umfassen. Verarbeitungsregeln werden zentral in Predict gespeichert (<i>Free Rules</i>) und/oder <i>DDM</i> -Feldern permanent zugewiesen (<i>Automatic Rules</i>). |
| Profil | <i>Großrechner</i> : Mit der Natural-Utility <code>SYSPARM</code> kann eine Profilparameter-Zeichenkette angegeben und unter einem Profilnamen gespeichert werden. Um das Profil zu verwenden, wird Natural mit dem dynamischen Parameter <code>PROFILE=profile-name</code> gestartet. Die unter diesem Profilnamen gespeicherte Profilparameter-Zeichenkette wird dann als dynamische Parameter an Natural übergeben. Profil bedeutet in diesem Kontext eine aus Profilparametern bestehende Zeichenkette, die unter einem Profilnamen gespeichert wird. |
| Profilparameter | <p>Ein Natural-Profilparameter definiert eine Eigenschaft der Natural-Umgebung.</p> <p>Der Wert für einen Profilparameter wird aus einer der folgenden Quellen übernommen:</p> <ul style="list-style-type: none">■ Zuweisung zur Laufzeit aus einem Session-Parameter, der mit dem Natural-Statement <code>SET GLOBALS</code> oder dem Natural-Systemkommando <code>GLOBALS</code> (höchste Priorität) angegeben ist.■ Dynamische Zuweisung, die für die aktuelle Natural-Sitzung gültig ist. Dies erfolgt durch Angabe eines Einzelparameters beim Natural-Start. |

- Statische Zuweisung, die im Natural-Parametermodul bzw. in der Natural-Parameterdatei angegeben wird (niedrigste Priorität). Auf dem Großrechner ist das das Parametermodul, welches mit dem **Natural-Nukleus** verlinkt ist. Unter Windows und UNIX ist das die Parameterdatei NATPARM.SAG.

Weitere Informationen siehe die plattformspezifische *Natural-Operations*-Dokumentation.

Profilparameter-Hierarchie Natural-Profilparameter werden, wie in der folgende Tabelle dargestellt, auf verschiedenen hierarchisch organisierten Ebenen (mit absteigender Priorität) gesetzt:

| | |
|--|---|
| Während der Sitzung (höchste Priorität) | Entwicklungsumgebungseinstellungen Einstellungen auf Programm-/Statement-Ebene Session-Parameter-Einstellungen <i>Natural Security</i> -Definitionen |
| Dynamisch beim Start der Sitzung | Dynamische Parametereingabe Vordefinierte Parameterprofile Vordefinierte dynamische Parameter-Sätze |
| Statisch | Alternatives Parametermodul bzw. Parameterdatei Natural-Standard-Parametermodul bzw. Parameterdatei |

Profiler Tool für die dynamische Programmanalyse. Misst die Häufigkeit und Dauer von Anweisungen zur Vereinfachung der Programmoptimierung.

Siehe auch *Natural Profiler*.

ProgID (Programmatic Identifier) Die *ProgID* einer **DCOM** Class ist ein aussagekräftiger Name, durch den die Class in Client-Programmen gekennzeichnet wird. Bei *Natural Classes* wird der Name, der im Operanden *class-name* im DEFINE CLASS-Statement definiert ist, als eine *ProgID* in die *Registry* geschrieben, wenn die Class mit dem Kommando REGISTER als DCOM Class registriert wird.

Program (Programm) *Predict*: Ein Predict-Objekt des Typs *Program* dokumentiert Datenverarbeitungsobjekte verschiedenen Typs und verschiedener Sprachen.

| | |
|--|--|
| Program-driven Application | Anwendungen, bei denen die ausführenden Teile des Code durch Programme und nicht durch Ereignisse gesteuert werden. Die Ausführung beginnt mit der ersten Zeile ausführbaren Codes, folgt einem definierten Pfad durch die Anwendung und ruft dabei gemäß Anweisungen in der vordefinierten Abfolge zusätzliche Programme auf. |
| Property (Eigentum) | Attribute eines Objekts, auf die von <i>Clients</i> zugegriffen werden kann. In <i>Natural Classes</i> werden <i>Property</i> -Werte eines Objekts im Objektdatenbereich gespeichert. Deshalb muss jedem <i>Property</i> eine Objektdatenvariable zugewiesen werden. Weitere Informationen siehe Object Data Variable . |
| Property Implementation | Die Objektdatenvariable, die einem <i>Property</i> zugewiesen wird. Weitere Informationen siehe Object Data Variable . |
| Protocol Access Layer | Codierung von Zugangsaufrufen an das und von dem Kommunikationsprotokoll (zurzeit TCP/IP) an den und von dem <i>Client</i> und <i>Server</i> . |
| PSB = Program Specification Block | DL/I or IMS control block that describes databases and logical message destinations available for an application program. A PSB consists of one or more program communication blocks (PCBs).* |
| PU = Physical Unit (Physische Einheit) | SNA : In der <i>Systems Network Architecture</i> (SNA) identifiziert eine physische Einheit (<i>Physical Unit - PU</i>) einen Netzknoten, der Kommunikationssitzungen zwischen logischen Einheiten (<i>Logical Units - LU</i>) unterstützt. |

Q

| | |
|--------------------------|---|
| Qualifier (Kennzeichner) | Die Benutzervariable, die zur Kennzeichnung eines Feldes verwendet wird, wenn es referenziert wird. Um das Feld zu kennzeichnen, geben Sie den Namen des Datenelements der Ebene 1, in dem sich das Feld befindet, und einen Punkt vor dem Feld an. Weitere Informationen siehe <i>Datenstrukturen kennzeichnen im Leitfaden zur Programmierung</i> . |
|--------------------------|---|

R

| | |
|------|--|
| RABN | Abkürzung für <i>Relative Adabas Block Number</i> . |
| RAD | Abkürzung für <i>Rapid Application Development Model</i> . |

| | |
|--|--|
| RBA = Relative Byte Address | VSAM: The displacement in bytes of a stored record or control interval from the beginning of the storage space allocated to the data set to which it belongs.* |
| Registry (Windows-Registrierungsdatenbank) | Zentrales Verzeichnis auf Windows-Plattformen für Konfigurationsdaten, Datenbank für Konfigurationsdateien. Physisch besteht die <i>Registry</i> aus zwei Dateien: <i>System.dat</i> und <i>User.dat</i> . Logisch umfasst die <i>Registry</i> die Konfigurationsdaten, die im <i>Registry Editor</i> angezeigt werden. Die <i>Registry</i> enthält Konfigurationsinformationen aller Art, sowohl für Hardware als auch Software. Beispielsweise enthält die <i>Registry</i> Informationen über <i>DCOM Classes</i> und deren Zuweisung zu Servern. Weitere Informationen siehe Microsoft-Dokumentation. |
| Registry Key (Registrierungsschlüssel) | Wenn eine <i>Class</i> registriert wird, werden Einträge mit Registrierungsschlüsseln in der Windows-System-Registrierungsdatenbank des Servers vorgenommen. Registrierungsschlüssel werden außerdem in der Registrierungsdatenbank des Client hinzugefügt, wenn die Client-Registrierungsdatei ausgeführt. Ausführliche Informationen zu den Registrierungsschlüsseln und deren Verwaltung siehe Registry-Dokumentation für die entsprechende Plattform. |
| Related Objects | <i>Predict</i> : Generischer Begriff für <i>Parent</i> - und <i>Child</i> -Objekte. |
| Relationship | <i>Predict</i> : Siehe <i>File Relation</i> . |
| Report Listing | <i>Predict</i> : <i>Predict</i> -Objekte des Typs <i>Report Listing</i> werden verwendet, um die Transfer-Operationen mit dem <i>Predict Coordinator</i> und den Konvertierungsfunktionen zu protokollieren. |
| Reserved Word (Reserviertes Wort) | Teil der Programmiersprache Natural. Es handelt sich um Buchstabenkombinationen, die den Benutzern nicht zur eigenen Verwendung zur Verfügung stehen. Siehe Abschnitt <i>Für Natural reservierte Schlüsselwörter im Leitfaden zur Programmierung</i> . |
| Restrictions (Einschränkungen) | <i>Predict</i> : Sammlung von <i>Keywords</i> (Schlüsselwörter), <i>Owners</i> (Eigentümer) und <i>Text Strings</i> (Textzeichenketten), die als Auswahlkriterien bei der Rückgewinnung von Informationen zu Objekten verwendet werden. Einschränkungen können in Profilen gespeichert oder zur vorübergehenden Verwendung angegeben werden. |
| Retrieval (Informationsrückgewinnung) | <i>Predict</i> : Untersystem, das anhand von Auswahlkriterien Ausgabeinformationen (Objekte, deren Attribute und Verknüpfungen, s. <i>Associaton</i>) am Terminal oder (im Batch-Modus) für das Spool-System verfügbar macht. |

| | |
|---|---|
| Retrieval Model (Verfahren zur Informationsrückgewinnung) | <i>Predict</i> : Ein <i>Retrieval Model</i> umfasst zwei Teile: eine Informationsrückgewinnungsstruktur und eine Definition des Inhalts sowie des Layout der Berichts (<i>Report</i>), die erstellt werden sollen. |
| Retrieval Operation (Recherche-Operation) | <i>Predict</i> : Zur Informationsrückgewinnung aus dem Datendiktionär müssen ein <i>Retrieval</i> -Typ und ein Ausgabemodus angegeben werden. Diese Einstellungen bestimmen zusammen den Typ der <i>Retrieval Operation</i> , die im Kopfbereich von Berichts (<i>Reports</i>) angezeigt wird. |
| Retrieval-Typ | <i>Predict</i> : Der <i>Retrieval</i> -Typ bestimmt den Typ der Informationen, die <i>Predict</i> aus dem Diktionär recherchieren soll. <i>Retrieval</i> -Typen sind entweder unabhängig, z.B. <i>Object</i> mit <i>Children</i> , oder typspezifisch, z.B. Dateiu Unterschied (<i>Difference of Files</i>). |
| Rippling | <i>Predict</i> : Das automatische Aktualisieren aller abgeleiteten Felder auf niedrigerer hierarchischer Ebene, wenn das entsprechende Feld auf einer höheren hierarchischen Ebene geändert wird. Es gibt folgende hierarchische Ebenen: <ul style="list-style-type: none">■ <i>Standard Files</i>■ <i>Master Files</i>■ <i>Userviews</i> |
| Routine | Sammelbegriff für Natural-Objekttypen, die nicht eigenständig ausgeführt werden können, z.B. <i>Helproutine</i> , <i>Subprogram</i> , <i>Subroutine</i> . |
| RPC = Remote Procedure Call | Client/Server-Kommunikationstechnik, die angewendet wird, um Anwendungen, Prozeduren und Objekt-Methoden auf einem Remote-Rechner über ein Netzwerk aufzurufen. Der Client sendet eine Anforderung an den Server, der den Dienst erbringt. Die Übertragung der Anforderungen erfolgt über <i>Client</i> und <i>Server Stubs</i> . Siehe auch <i>Natural RPC (Remote Procedure Call)</i> . |
| RPC Stub | <i>Natural RPC</i> : Das <i>Client Subprogram</i> , über das das <i>Server Subprogram</i> aufgerufen wird. Der <i>RPC Stub</i> hat den gleichen Namen und enthält die gleichen Parameter wie das entsprechende <i>Server Subprogram</i> . Der <i>RPC Stub</i> empfängt die <i>CALLNAT</i> -Anforderungen auf der Client-Seite, arrangiert die <i>Natural CALLNAT</i> -Parameter und übergibt sie an die <i>RPC Client-Laufzeit</i> , die dann die Parameter an den <i>EntireX Broker Stub</i> des Client übergibt. |

Weitere Informationen siehe *Creating Stub Programs* und *Working with Automatic Natural RPC Execution* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

RRDS = Relative-record
Data Set

VSAM: A data set organization, in which records are of fixed length are accessed by their relative record numbers (**RRN**).*

RRN = Relative Record
Number (Relative Datensatz-
nummer)

VSAM: Eindeutige Kennung für jeden Datensatz, der in einer zugehörigen Datei (**RRDS**, *Relative Record Data Set*) gespeichert wird.

RSL = Record-sharing Level

VSAM: Eine Option, die es gestattet, VSAM-Dateien (*Data Sets*) auf Datensatzebene von mehreren Adressräumen aus über mehrere Systeme hinweg gemeinsam zu nutzen.

Runtime (Laufzeit)

1. Teil des **Natural-Nukleus**, der im Gegensatz zum *Compiler* ein Natural-Programm ausführt.
2. Die Zeit, während der ein Natural-Programm ausgeführt wird.

S

SAA = Systems Application
Architecture

A set of common standards and procedures for working with IBM systems and data. SAA enables different software, hardware and network environments to coexist. It provides bases for designing and developing application programs that are consistent across different systems.*

SBCS = Single-Byte Character
Set

Zeichensatz, bei dem jedes Zeichen durch ein Byte dargestellt wird, im Unterschied zum Doppelbyte-Zeichensatz (**DBCS**).

SCP = System Control Program

Großrechner: Die *Natural SCP*-Umgebung ist Teil des *Natural CICS Interface*. Ein Auszug (*Dump*) wird bei allen SCP-Störfällen erstellt.

Siehe Abschnitt *Natural under CICS Abend Codes and Error Messages* in der *Messages and Codes*-Dokumentation.

SDI

Abkürzung für *Single-Document Interface*.

Selection Criteria (Auswahl-
kriterien)

Predict: Predict-Suchfunktionen können auf Einzelobjekten oder auf Gruppen von Objekten angewendet werden. Suchkriterien werden angegeben, um Objekte auszuwählen, auf die die Suchfunktionen angewendet werden sollen.

Selection Window (Auswahl-
fenster)

Predict: Siehe **Active Help (Aktive Hilfe)**.

| | |
|----------------------------|--|
| Server | <i>Predict</i> : Predict-Objekte des Typs <i>Server</i> werden gemeinsam mit Objekten des Typs <i>Node</i> (Knoten) benutzt, um <i>Remote Procedure Calls</i> zu dokumentieren. |
| Server Front-End | <i>z/OS</i> : Eine Komponente der Natural Server-Umgebung, die vom <i>Natural Db2 Stored Procedures Server</i> und dem <i>Natural Development Server</i> benutzt wird. Siehe <i>Operating Natural - Environment-Specific Information</i> in der <i>Operations</i> -Dokumentation. Es handelt sich um einen funktional erweiterten umgebungsabhängigen Nukleus (Siehe Natural-Nukleus), der für den <i>Front-End Server</i> zusätzliche Funktionalität zur Verfügung stellt, z.B. Initialisierung der Server-Umgebung, Ein- und Ausspeichern der Sitzung und Ausführung von Natural-Objekten. |
| Server ID (Server-Kennung) | Die Server-Kennung ist eine Zeichenkette, die einen <i>NaturalX Server</i> identifiziert. Die Server-Kennung ist ein Schlüssel in der <i>Windows System Registry</i> , dessen Eigentümer Natural ist und der alle <i>Classes</i> zusammenhält, die zu einem gegebenen <i>NaturalX Server</i> gehören. Es ist eine willkürliche, 32 Zeichen umfassende Zeichenkette, die keine Leerzeichen enthält und bei der nicht zwischen Groß-/Kleinschreibung unterschieden wird. Die Server-Kennung wird mit dem Natural-Parameter <code>COMSERVERID=serverid</code> definiert. |
| Server Name (Server-Name) | <i>Natural RPC</i> : Der Name des Servers, auf dem der <code>CALLNAT</code> ausgeführt werden soll. Beim <i>EntireX Broker</i> wird der Name im Feld SERVER in der <i>EntireX Broker-Attribut-Datei</i> definiert. |
| Server Task (Server-Task) | <i>Natural RPC</i> : Ein <i>Server Task</i> ist eine Natural-Task, der Dienste (Unterprogramme) anbietet. Typischerweise handelt es sich dabei um einen <i>Batch Task</i> oder einen asynchronen Task. Die Identifizierung erfolgt mittels Server Name . |
| Service Directory | <i>Natural RPC</i> : Verzeichnis, das Informationen darüber enthält, welcher Server welche Dienste (Unterprogramme) zur Verfügung stellt. Das <i>Service Directory</i> kann entweder lokal auf jedem Client-Knoten vorhanden sein oder es kann auf einem <i>Remote Directory Server</i> liegen, der mit dem Session-Parameter <code>RDS</code> referenziert wird. Generierung erfolgt mit der Utility <code>SYSRPC</code> zur Implementierung des <i>Service Directory</i> . Das <i>Service Directory</i> wird durch das Natural-Unterprogramm <code>NATCLTGS</code> implementiert, das durch die Utility <code>SYSRPC</code> generiert wird. |

| | |
|--|---|
| Session (Sitzung) | <i>Natural</i> : Der benutzerabhängige Natural-Laufzeitkontext, der erforderlich ist, damit das Natural-Laufzeitsystem Natural-Programme für einen einzelnen Benutzer ausführen kann. |
| Session Parameter (Session-Parameter) | <p>Parameter innerhalb der aktuellen Natural-Sitzung, die mit dem Systemkommando <code>GLOBALS</code> (oder einem <code>SET GLOBALS</code>-Statement) zugewiesen werden.</p> <p>Natural-Session-Parameter können in bestimmten Natural-Statements benutzt werden, um Faktoren wie z.B. die Größe eines Berichts und die Art und Weise, wie Felder angezeigt werden sollen, zu steuern.</p> <p>Bei der Installation von Natural setzt der Natural-Administrator diese Parameter auf Standardwerte, die dann für alle Natural-Benutzer gültig sind.</p> <p>Siehe <i>Parameter-Referenz-Dokumentation</i>.</p> <p>Um festzustellen, welche Parameter für Ihre Natural-Sitzung gelten, können Sie das Systemkommando <code>GLOBALS</code> absetzen.</p> |
| Session Variable (Session-Variable) | <i>Predict</i> : Bei vielen Predict-Funktionen müssen Parameterwerte angegeben werden. Predict speichert Parameterwerte temporär in Sitzungsvariablen. Welche Parameterwerte von Sitzungsvariablen genommen werden können, hängt sowohl von der Funktion als auch dem Parametertyp ab. |
| Shared Nucleus (Umgebungsunabhängiger Nukleus) | Synonym für den umgebungsunabhängigen Nukleus. Siehe auch Natural-Nukleus . |
| Single-Master View | <i>Predict</i> : Eine Sicht, die stets nur von einem <i>Master File</i> abgeleitet wird, z.B. eine Adabas-Benutzer-Datensicht (<i>Usserview</i>). |
| Single Point of Development (SPoD) | <i>Windows</i> : Ein ab Natural 5 für Windows realisiertes Konzept einer plattformunabhängigen Anwendungsentwicklung. |
| SIP = Session Information Pool | Dient zur Ablage der Datensätze mit Informationen über die Natural-Sitzung. Wird in Verbindung mit dem <i>Natural Authorized Services Manager</i> (ASM) benutzt. |
| SIR = Session Information Record (Natural) / Session Information Retrieval (IBM) | <i>Natural</i> : Das <i>Natural CICS Interface</i> hält permanent Informationen über alle aktiven Natural-Sitzungen bereit. Zu jeder Sitzung wird ein Sitzungsinformationsdatensatz (<i>Session Information Record - SIR</i>) gepflegt. |

| | |
|--------------------------------------|---|
| | <i>IBM:</i> The function that allows an operator to enable or disable session information retrieval for a particular gateway or for all gateway sessions. When a gateway session ends, trace information about the most recent sequence or FID0 numbers to cross the gateway is passed back to all system services control points (SSCPs) that have enabled SIR for that session or for all sessions. This information can also be passed back to the requesting host.* |
| SIT = System Initialization Table | A CICS table that contains information to initialize and control system functions, module suffixes for selection of user-specified versions of CICS modules and tables, and information used to control the initialization process. You can generate several SITs, using the resource definition macro DFHSIT, and then use the SIT system initialization parameter to select the one that best meets your current requirements at initialization time.* |
| SMA = System Maintenance Aid | Siehe <i>System Maintenance Aid</i> . |
| SMF = System Management Facility | A standard feature of z/OS that collects and records a variety of system and job-related information.* |
| SNA = Systems Network Architecture | The IBM architecture that defines the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information (the users) to be independent of and unaffected by the specific SNA network services and facilities that are used for information exchange.* |
| SOAP = Simple Object Access Protocol | XML-basierte Nachrichtenkonvention, die in Kombination mit dem <i>EntireX XML Wrapper</i> ermöglicht, dass eine Natural-Anwendung durch ein mittels HTTP gesendetes XML-Dokument gestartet wird. |
| Software AG Editor | <i>Großrechner:</i> Der Software AG Editor kann alternativ zum Natural-Programm-Editor benutzt werden. Voraussetzung ist, dass Natural ISPF installiert und Ihr Editor-Profil entsprechend eingestellt ist. (Siehe <i>Editoren</i> -Dokumentation bezüglich Editor-Profilen). |
| Source Field (Source-Feld) | <i>Predict:</i> Feld, das in der Definition eines abgeleiteten Feldes benutzt wird. Abgeleitete Felder sind Super-, Hyper-, Subfield-Deskriptoren und phonetische Deskriptoren. |
| Source Object (Quellcode-Objekt) | <i>Natural:</i> Ein <i>Source Object</i> (oder gespeichertes Objekt) enthält Natural-Quellcode. Ein <i>Source Object</i> wird als Objektmodul in einer Natural-Systemdatei gespeichert. |

| | |
|---------------------------------|--|
| | Siehe auch <i>Cataloged Object</i> , <i>Object</i> und <i>Object Types</i> |
| SPA = Scratch Pad Area | <i>IMS TM</i> : Ein terminal-spezifischer Bereich zur zeitweiligen Speicherung von Daten, die im Laufe von Benutzer-Transaktionen verarbeitet werden. |
| SPoD | Siehe <i>Single Point of Development</i> . |
| SQL = Structured Query Language | A programming language that is used to define and manipulate data in a relational database.* |
| SSA = Segment Search Argument | <i>IMS</i> oder <i>DL/I</i> : The part of a DL/I call that identifies a segment or group of segments to be processed. SSAs may be simple segment names or they may be qualified to include constraints on the values of fields within the named segment types.* |
| Standard Field (Standard-Feld) | <i>Predict</i> : Ein <i>Standard Field</i> in einem <i>Standard File</i> . <i>Standard Files</i> und <i>Standard Fields</i> erzwingen den standardmäßigen Gebrauch von Feldern in verschiedenen Dateien. |
| Standard File (Standard-Datei) | <i>Predict</i> : Ein <i>Standard File</i> dokumentiert des Gesamt-Layouts aller Daten, die innerhalb einer Organisation verarbeitet werden (Datendefinitionen und/oder Firmenstandards). Felder in einem <i>Standard File</i> beziehen sich nicht direkt auf eine implementierte Datenstruktur: Felddefinitionen in einem <i>Standard File</i> beziehen sich auf <i>Master Files</i> verschiedenen Typs. |
| | Siehe <i>Rippling</i> . |
| Startup Transaction - *STARTUP | Die Systemvariable *STARTUP enthält den Namen des Programms, das zum Starten einer Natural-Sitzung benutzt wird. Wenn Sie eine <i>Natural Development Server</i> -Umgebung von <i>NaturalONE</i> aus abbilden („mappen“) (*SERVER-TYPE) und das von *STARTUP gelieferte Startprogramm benutzen, sollten Sie sich vergewissern, dass dieses Programm keine Bildschirm-Ein-/Ausgaben durchführt und korrekt in der Bibliothek beendet wird, in der es enthalten ist. Siehe auch Beschreibung der Systemvariablen *STARTUP. |
| Statements | Siehe <i>Statements</i> -Dokumentation. |
| Statische Parameter | <i>Großrechner</i> : Zugewiesen durch Profilparameter, die im Natural- Parametermodul angegeben sind, welches während der Installation angelegt wird. <i>Windows und UNIX</i> : Die in NATPARM.SAG angegebenen Parametereinstellungen.. |

| | |
|-----------------------------------|---|
| Steplib | Eine <i>Steplib</i> ist eine Natural-Benutzer-Bibliothek (<i>Library</i>) oder System-Bibliothek, die mit der aktuellen Benutzer- oder System-Bibliothek verkettet ist. Eine <i>Steplib</i> kann als einzelner Speicherort für mehrere Objekte benutzt werden, die von verschiedenen Anwendungen gemeinsam genutzt werden. Dadurch wird die redundante Speicherung identischer Objekte vermieden und Anwendungen können besser organisiert werden. Wenn <i>Natural Security</i> installiert ist, kann eine <i>Steplib</i> außerdem verwendet werden, um den den Zugang zu bestimmten Objekten einzuschränken. |
| Storagespace | <i>Predict</i> : Predict-Objekte des Typs <i>Storagespace</i> dokumentieren <i>Db2 Storagegroups</i> . |
| STOW | In Natural bezieht sich der Begriff "Stow" auf das Natural-Systemkommando <i>STOW</i> . Es dient dazu, ein Natural-Objekt (sowohl in Sourceform als auch in Objektform) in einer Natural-Systemdatei zu katalogisieren (kompilieren) und zu speichern. Beide Objekte haben dann denselben Zeitstempel. |
| Stub (Anknüpfungspunkt) | Program, das als Kommunikationsendpunkt (<i>Socket</i>) die Verknüpfung zwischen dem lokalen Anwendungsprogramm und dem <i>Broker</i> herstellt. Siehe auch RPC Stub . |
| Subtype (Untertyp) | <i>Predict</i> : Objekttypen können Untertypen haben. Der Objekttyp <i>File</i> beispielsweise kann die Untertypen <i>Adabas File</i> , <i>Adabas Userview</i> , <i>Db2 Table</i> , <i>Db2 View</i> usw. Die Untertypen des Objekttyps <i>File</i> beispielsweise werden auch als Dateitypen bezeichnet. |
| Super Natural | Produkt, mit dem Endbenutzer Daten aus Großrechner- und PC-Dateien extrahieren und verarbeiten können. Bietet eine menügesteuerte, interaktive Report-Layout-Angabe und flexible, kundenspezifische Anpassung von Endbenutzerprofilen. |
| SVC = Supervisor Call | A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.* |
| Syncpoint = Synchronization Point | In CICS und IMS TM ist ein <i>Syncpoint</i> ein logischer Punkt während der Ausführung eines Anwendungsprogramms, an dem die von einem Programm vorgenommenen Änderungen konsistent und vollständig sind und in der Datenbank festgeschrieben werden können. Die Ausgabe, die bis zu diesem Punkt angehalten wurde, wird an den oder die betreffenden Bestimmungsorte gesendet, die Eingabe wird aus den Nachrichtenwarteschlangen entfernt und die |

| | |
|-----------------------------------|--|
| | Datenbankänderungen werden für andere Anwendungen verfügbar gemacht. |
| SYSARM Utility | <i>Großrechner:</i> Utility zum Anlegen und Pflegen von Parameterprofilen. Sie können eine aus Profilparametern bestehende Zeichenkette angeben, diese unter einem Profilnamen speichern und danach Natural mit dem dynamischen Parameter <code>PROFILE=profile-name</code> aufrufen. Die mit diesem Profil definierten Parameter werden dann als dynamische Parameter an Natural übergeben. |
| SYSRPC Utility | Utility zum Konfigurieren von Diensten und Verbindungen für jeden Natural Client, der mit Remote Procedure Calls arbeitet. Siehe <i>SYSRPC</i> in der <i>Debugger und Dienstprogramme (Utilities)</i> -Dokumentation. |
| System | <i>Predict:</i> Predict-Objekte des Typs <i>System</i> dokumentieren eine Zusammenstellung von Programmen, die eine Anwendung oder einen Teil einer Anwendung bilden. |
| System Command (Systemkommando) | Natural-Systemkommandos führen Funktionen aus, die zum Anlegen, Pflegen oder Ausführen von Natural-Programmierobjekten benötigt werden. Außerdem gibt es Systemkommandos, die zum Überwachen und Verwalten der aktuellen Natural-Umgebung benutzt werden können. |
| System File (Systemdatei) | Die Natural-Systemdateien (FNAT, FUSER) enthalten Informationen, Programme, Module usw., die für das Funktionieren des Natural-Systems erforderlich sind. Produkte wie <i>Predict</i> , <i>Natural Security</i> , <i>Natural Advanced Facilities</i> und <i>Natural for VSAM</i> benötigen eigene Systemdateien (FDIC, FSEC, FSPOOL, FDICX). |
| System Function (Systemfunktion) | In Natural eingebaute Funktionen, die in bestimmten Statements benutzt werden können. Siehe <i>Systemfunktionen</i> -Dokumentation. |
| System Library (Systembibliothek) | Natural-interne Bibliothek, die bei der Installation in der Systemdatei FNAT angelegt wird. Es sind darin keine kundenseitige Änderungen möglich. Alle Natural System Library-Namen beginnen mit den Buchstaben "SYS" (mit einigen Ausnahmen auf dem Großrechner). |
| SYSTEM Library | Die Bibliothek, die durchsucht wird, wenn ein Natural-Objekt weder in der aktuellen Bibliothek noch in den <i>Steplib</i> s gefunden werden kann. Eine <i>SYSTEM Library</i> gibt es sowohl in der FNAT- als auch in der FUSER-Systemdatei. Siehe Steplib . |
| System Maintenance Aid (SMA) | Produkt zum Installieren und Pflegen von z/OS-Produkten. |

| | |
|----------------------------------|--|
| System Management Hub | Produkt- und plattformübergreifendes <i>Management-Framework</i> , dessen Architektur die Integration von <i>Plug-ins</i> existierender Produkte, Trennung von implementierungsspezifischer Verwaltungstechnologien und Wiederverwendung als produktunabhängiges, plattformübergreifendes Framework gestattet. |
| System Program (System-Programm) | <i>Predict</i> : Programme, die nicht als Quellcode verfügbar sind, werden mit Predict-Objekten des Typs <i>Program</i> mit <i>Subtype E (External Object)</i> und <i>Language Z (System Program)</i> dokumentiert. Immer wenn in Predict ein Systemprogramm angelegt wird, werden dazu auch <i>XRef</i> -Daten geschrieben. |
| Systemvariable | Systemvariablen werden verwendet, um Systeminformationen anzuzeigen. Sie können an jedem beliebigen Punkt innerhalb eines Natural-Programms referenziert werden. Siehe <i>Systemvariablen</i> -Dokumentation. |

T

| | |
|---|--|
| TAC | Abkürzung für <i>Transaction Code</i> (Transaktionsschlüssel). |
| Task (Aufgabe) | <i>Großrechner</i> : Einzelne oder mehrere Abfolgen von Anweisungen, die in einer Mehrprogramm- oder Mehrprozessor-Umgebung von einem Steuerprogramm als zusammenhängende Aufgabe behandelt werden, die von einem Rechnersystem erledigt werden sollen. <i>Windows und UNIX</i> : Siehe Prozess . |
| TCB = Task Control Block | <i>z/OS</i> : Vom Betriebssystem zugewiesener Steuerblock, der alle Informationen zu Tasks enthält, die für das System relevant sind. |
| TD = Transient Data (Übergangsdaten) | CICS : Daten die in einer Einrichtung zeitweilig gespeichert werden. Ein Übergangsobjekt wird automatisch gelöscht, wenn es nicht mehr benutzt wird. |
| Terminal ID, TID (Terminalkennung) | Ein Kennzeichen zur eindeutigen Unterscheidung von Datenendgeräten durch das System. |
| Terminal Command (Terminalkommando) | Sie können ein anderes Sonderzeichen als Terminalkommando-Steuerzeichens definieren. Dazu dient der Session-Parameter CF. Siehe <i>Terminalkommando</i> -Dokumentation. |
| Thousands Separator (Tausendertrennzeichen) | Um die Ausgabe großer Ganzzahl-Werte zu strukturieren, ist es üblich, Trennzeichen nach jeweils drei Ziffern einer Ganzzahl einzufügen, um Gruppen von Tausendern voneinander zu trennen. Dieses Trennzeichen wird „Tausendertrennzeichen“ genannt. |

| | |
|--|--|
| Dynamic Thousands Separator (Dynamisches Tausendertrennzeichen) | In Natural ist ein „dynamisches Tausendertrennzeichen“ ein spezifisches Zeichen in der Editiermaske, um die Stelle anzugeben, an der die Tausendertrennzeichen zur Laufzeit eingefügt werden. |
| (Thousands Separator Character (Tausendertrennzeichen) | Das „Tausendertrennzeichen“ ist das Zeichen, das zur Laufzeit als Trennzeichen zwischen jeder dritten Ziffer einer Ganzzahl verwendet werden soll. |
| Thread | <p><i>Großrechner</i>: Speicherbereich, der alle Daten für eine bestimmte Natural-Sitzung enthält.</p> <p><i>Windows und UNIX</i>: Die grundlegende (Speicher-)Einheit, der das Betriebssystem CPU-Zeit zuweist. <i>Threads</i> gestatten paralleles Rechnen in einzelnen oder mehreren Prozessoren, bei denen die Ausführung unabhängig voneinander erfolgt.</p> |
| TP Monitor = Teleprocessing Monitor / Transaction Processing Monitor | <p>z/OS:</p> <p>Ein Steuerprogramm für die Administration und Verwaltung von Anwendungen zur Online-Transaktionsverarbeitungen (OLTP). Die TP-Monitor-Technik ergänzt das Betriebssystem und dient als Zwischenschicht, die den Zugang zu Ressourcen kapselt. Auf diese Weise wird der Benutzer oder Anwendungsprogrammierer vor technischen Einzelheiten auf der Betriebssystemebene, z.B. bezüglich dem Datenbankzugriff, abgeschirmt.</p> |
| Transfer Medium (Transfer-Medium) | <i>Predict</i> : Wenn Daten unter Verwendung des <i>Predict Coordinator</i> übertragen werden, kann das Übertragungsmedium eine PC-Arbeitsdatei, eine Natural-Arbeitsdatei oder eine Datei in einer Datenbank sein. Dabei ist das Übertragungsmedium bei den <i>Coordinator</i> -Funktionen Unload und Export die Zielumgebung; bei den <i>Coordinator</i> -Funktionen Load und Import ist das Übertragungsmedium die Quellumgebung. |
| Transaction (Transaktion, Vorgang) | <ol style="list-style-type: none"> 1. Eingabe von Anwendungsdaten, die von einem einzelnen Benutzer übergeben werden und durch die eine Abfolge von Informationsaustauschvorgängen und die Ausführung von operativen Schritten (Eingabe - Datenverarbeitung - Ausgabe) eingeleitet werden. Um den Ablauf zu starten, wird ein Transaktions-schlüssel benötigt. 2. Natural führt Datenbankaktualisierungen auf der Basis von Transaktionen (Datenbanktransaktionen) aus, was bedeutet, dass alle Datenbankänderungsanforderungen in logischen Transaktions-einheiten verarbeitet werden. Eine logische Transaktion ist die kleinste Arbeitseinheit (gemäß Definition durch den Benutzer), die als Ganzes ausgeführt werden muss, um sicherzustellen, dass |

die in der Datenbank enthaltenen Informationen logisch konsistent sind. Die Statements `END TRANSACTION (ET)` und `BACKOUT TRANSACTION (BT)` bewirken, dass die Datenbank alle Datensätze seit Beginn der Transaktion entsperrt. Siehe auch *Datenbankzugriffe im Leitfaden zur Programmierung*.

| | |
|-----------------------------------|---|
| Translator Database | <i>Predict</i> : Datenbank, die ADASTAR-Übersetzungstabelle enthält, die verwendet wird, um bei der Benutzung von <i>Adabas Star</i> die physische Datei hinter einer Nummer einer logischen Datei zu bestimmen. |
| Tree View (Baumstrukturansicht) | Visuelle, hierarchische Darstellung der Bestandteile. |
| Type Information (Typinformation) | Wenn eine <i>NaturalX Class</i> als <i>DCOM Class</i> registriert wird, dann wird eine <i>Type Library</i> generiert, die mit der <i>Class</i> über einen Eintrag in der <i>Windows System Registry</i> verbunden ist. <i>Clients</i> können die in der <i>Type Library</i> enthaltenen Typinformationen benutzen, um die Beschreibungen von <i>Interfaces</i> , <i>Methods</i> und <i>Properties</i> beim Kompilieren oder zur Laufzeit zu prüfen. |
| Type Library | Wenn eine <i>NaturalX Class</i> als <i>DCOM Class</i> registriert wird, dann wird eine <i>Type Library</i> generiert, die mit der <i>Class</i> über einen Eintrag in der <i>Windows System Registry</i> verbunden ist. <i>Clients</i> können die in der <i>Type Library</i> enthaltenen Typinformationen benutzen, um die Beschreibungen von <i>Interfaces</i> , <i>Methods</i> und <i>Properties</i> beim Kompilieren oder zur Laufzeit zu prüfen. |

U

| | |
|-------------------|---|
| UCB | Abkürzung für <i>Unit Control Block User</i> . |
| UDF | Benutzerdefinierte Funktion. |
| Unicode | <p>Informationen zu Unicode-Begriffen, die in der Natural-Dokumentation verwendet werden, können Sie dem Glossar entnehmen, das für die aktuelle Unicode-Version auf Unicode Home Page veröffentlicht wird:</p> <ul style="list-style-type: none">■ http://www.unicode.org/■ http://www.unicode.org/versions/Unicode4.0.0/b1.pdf |
| Unload (Entladen) | <i>Predict</i> : Es gibt zwei Verfahren, um <i>Predict</i> -Objekte zu entladen: |

■ Auszüge (Extracts) benutzen

Mit Hilfe der Menü-Funktionen können Sie nur Objekte entladen, die in einem Extract enthalten sind. Pro Unload-Operation können bis zu 10 Extracts angegeben werden.

■ Unload-Kommandos benutzen

Außerdem können Sie in der Kommandozeile einzelne Objekte oder Bereiche von Objekten angeben. Der Leistungsumfang der Funktion kann durch Parameter eingeschränkt werden.

URI

Abkürzung für *Uniform Resource Identifier* (Identifikator für Ressourcen im Internet).

Beispiel:

```
http://guest:guestpwd@si15.hq.sag:80/xml/mycar.xml
```

| | |
|-----------------|-----------|
| Protokoll | http |
| Benutzerkennung | guest |
| Passwort | guestpwd |
| Port | 80 |
| Pfad | /xml |
| Datei | mycar.xml |

User (Benutzer)

Entire Operations: In einer Entire Operations-Umgebung handelt es sich um einen definierten Benutzer mit individuellem Benutzerprofil, das festlegt, welche Pflege- und Steuerungsfunktionen für den Benutzer erlaubt sind.

Predict: Predict-Objekte des Typs *User* dokumentieren Personen, die ein System benutzen. Eine logische Verbindung zwischen *Users* und *Documentation Objects* (Dokumentationsobjekten) wird mittels *Owners* (Eigentümern) eingerichtet. Siehe [Objekttypen](#).

URL (einheitlicher Ressourcenzeiger)

Abkürzung für *Uniform Resource Locator*, ein Subtyp des *Uniform Resource Identifier (URI)*. Im gängigen Sprachgebrauch, aber auch in manchen technischen Veröffentlichungen, wird es als Synonym für *URI* benutzt.

User-Defined Variable (Benutzervariable)

Ein Feld, das Sie selbst in einem Programm definieren können. Wird benutzt, um Werte oder Zwischenergebnisse zu speichern, die an einem Punkt in der Programmverarbeitung zur weiteren Verarbeitung oder zur Anzeige erzielt werden.

| | |
|---|--|
| | Eine Benutzervariable definieren Sie, indem Sie ihren Namen und ihr Format/Länge im <code>DEFINE DATA</code> -Statement angeben. Siehe Abschnitt <i>Benutzervariablen</i> im <i>Leitfaden zur Programmierung</i> . |
| User Exit (User-Exit) | Ein <i>User Exit</i> ist ein Punkt in Natural, eine Unterkomponente oder ein Unterprodukt, bei dem die Kontrolle an eine <i>User Exit Routine</i> übergeben wird. |
| User Exit Routine (User-Exit-Routine) | Eine <i>User Exit Routine</i> übernimmt an einem definierten <i>User Exit</i> die Kontrolle, um Daten zu bearbeiten oder Entscheidungen zu treffen. Üblicherweise wird ein Muster eines <i>User Exit</i> in Quellcodeformat ausgeliefert. Die in dem <i>User Exit</i> enthaltenen Anleitungen müssen durch den Benutzer geschrieben oder angepasst werden. Die meisten <i>User Exits Routines</i> nutzen die Vorteile der Programmiersprache Natural; eine kleine Untermenge muss auf Großrechnersystemen in in Assembler und auf Nicht-Großrechnersystemen in C geschrieben werden. |
| User ID, User Identification, UID (Benutzerkennung) | Kennzeichen zur eindeutigen Unterscheidung von Benutzern durch das System. |
| Utility (Dienstprogramm) | Eine <i>Utility</i> ist ein zum Lieferumfang eines Software-Produkts gehörendes Dienstprogramm oder Tool für Systemverwalter oder Benutzer zur Ausführung allgemeiner oder spezifischer, systemnaher Aufgaben. |
| Userview (Benutzer-Datensicht) | <i>Predict</i> : <i>Userviews</i> sind logische Sichten auf <i>Master Files</i> , die zur Verwendung in den Datendeklarationsabschnitten von Programmen definiert werden. Anzahl und Reihenfolge der Felder in einer <i>Userview</i> können von den <i>Master Files</i> abweichen und - innerhalb bestimmter Kompatibilitätsregeln - können bestimmte Attribute von Feldern in <i>Userviews</i> von den entsprechenden Werten auf der physischen/logischen Ebene abweichen. |
| UTC | Abkürzung für „Coordinated Universal Time“, die koordinierte Weltzeit. Sie ist nahezu identisch zu der früher gebräuchlichen „Greenwich Mean Time“ (GMT). |

V

| | |
|-----------------------------------|--|
| Variable | Siehe <i>Systemvariable</i> und <i>Benutzervariable</i> . |
| Verification (Gültigkeitsprüfung) | <i>Predict</i> : Ein <i>Predict</i> -Objekt des Typs <i>Verification</i> dokumentiert die Regeln zur Gültigkeitsprüfung von Feldwerten. Natural-Verarbeitungsregeln können von <i>Predict</i> -Objekten des Typs <i>Verification</i> generiert werden. |

| | |
|--|--|
| Version | <p>Gemäß unserer Maintenance Policy wird die Version (bzw. Release) eines Produkts wie folgt definiert:</p> <ul style="list-style-type: none"> ■ Major Version Die erste Stelle der Version gibt einen <i>Major Release</i> mit bedeutenden neuen Merkmalen und Funktionen und eventuell größeren Änderungen an der Architektur an. ■ Minor Version Die zweite Stelle der Version gibt einen <i>Minor Release</i> mit einigen neuen Merkmalen und Funktionen und Korrekturen (<i>Fixes</i>) an. ■ Service Pack Die dritte Stelle der Version gibt ein <i>Service Pack</i> an, die eine Kumulierung von Korrekturen (<i>Fixes</i>) ist, die als geplantes <i>Release</i> ausgeliefert werden. Es wird in der Versionsnummerierung als <i>SP</i> bezeichnet. ■ Fix Die vierte Stelle der Version gibt ein bestimmtes <i>Fix</i> an, das kumulativ sein kann. <p>In der Produktdokumentation wird die Version häufig in Form eines Platzhalters dargestellt, z.B. <i>vrs</i>, <i>v.r.s</i>, <i>vr</i>, <i>v</i>, <i>nnn</i> oder <i>nn</i>, z.B. in Namen von Dateien oder Modulen. Weitere Informationen zur <i>Maintenance Policy</i> siehe entsprechenden Abschnitt unter https://empower.softwareag.com.</p> |
| View | <i>Predict</i> : Siehe Userview . |
| Virtual Machine | <i>Predict</i> : Zusammen mit Objekten des Typs <i>Network</i> , dokumentieren <i>Predict</i> -Objekte des Typs <i>Virtual Machine</i> die Hardware- und Betriebssystemumgebung eines Datenverarbeitungssystems. |
| VM = Virtual Machine | „A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system.“* |
| VSAM = Virtual Storage Access Method | Eine IBM-Zugriffsmethode zum Pflegen von Datensätzen mit unterschiedlicher Dateiorganisation: <i>Key-Sequenced Data Sets</i> (KSDS), <i>Entry-Sequenced Data Sets</i> (ESDS) oder <i>Relative-Record Data Sets</i> (RRDS). |
| VTAM= Virtual Telecommunications Access Method | Anwendungsprogrammierungsschnittstelle (API) zur Kommunikation mit Datenfernverarbeitungsgeräten mit <i>Systems Network Architecture</i> (SNA) und ihren Benutzern. |

W

| | |
|---------------------------------------|--|
| WTO = Write-to-Operator | „An optional user-coded service that allows a message to be written to the system console operator informing the operator of errors and unusual system conditions that may need to be corrected.“* |
| Work File Format (Arbeitsdateiformat) | Beschreibt die Struktur der Daten, die in der Zieldatei gespeichert werden. Beispielsweise schreibt das Arbeitsdateiformat <code>ENTIRE CONNECTION</code> in zwei Dateien: <code>.NCD</code> und <code>.NCF</code> . |
| Work File Type (Arbeitsdateityp) | Eine in Natural zur Verfügung stehende Datei, die ein bestimmtes Arbeitsdateiformat verwendet. Es gibt beispielsweise zwei verschiedene Arbeitsdateitypen, bei denen Daten im <i>Entire Connection</i> -Format gespeichert werden. Diese Dateitypen sind <code>ENTIRE CONNECTION</code> und <code>TRANSFER</code> . Weitere Informationen siehe Profilparameter <code>WORK</code> . |

X

| | |
|---------------------------------------|---|
| X-Arrays | <i>X-Arrays</i> oder <i>eXtensible Arrays</i> sind <i>Arrays</i> , bei denen man die Anzahl der Ausprägungen zur Laufzeit verändern kann. Dies kann zu einer effizienteren Speichernutzung beitragen. Beispielsweise kann man kurzzeitig eine große Anzahl an Ausprägungen benutzen und dann den Speicherplatz verringern, wenn die Anwendung das <i>Array</i> nicht mehr benutzt. Weitere Informationen siehe <i>X-Arrays</i> im <i>Leitfaden zur Programmierung</i> . |
| XCF = Cross-systems Coupling Facility | „A component that provides functions to support cooperation between authorized programs running within a sysplex.“* Natural benutzt XCF nur in Verbindung mit <i>Parallel Sysplex</i> . |
| XML | Abkürzung für „Extensible Markup Language“. |
| XML Toolkit | <i>Windows</i> : Ermöglicht es Entwicklern, XML-Dokumente innerhalb von Natural zu verarbeiten. Zum <i>XML Toolkit</i> gehört ein Assistent (<i>Wizard</i>), der Natural-Quellcode generiert und Folgendes leistet: <ul style="list-style-type: none">■ Abbilden (<i>Mapping</i>) von Natural-Datendefinitionen auf <i>DTDs</i>,■ Serialisierung,■ Abbilden (<i>Mapping</i>) von <i>DTDs</i> auf Natural-Datendefinitionen,■ Parsen einer XML-Datei und Zuweisen des Inhalts zu einer Natural-Datenstruktur. |
| XRef Data = Active Cross-References | <i>Predict</i> : Verweistabellendaten (<i>XRef Data</i>) werden in <i>Predict</i> für implementierte Objekte gespeichert, und zwar unabhängig von |

Dokumentationsobjekten. Mit den aktiven Datenabfragefunktionen können Sie Informationen aus den XRef-Daten und aus den Dokumentationsobjekten abrufen, um zu ermitteln,

- ob Objekte, die im Datendiktionär dokumentierte Objekte noch nicht implementiert worden sind,
- ob implementierte Members noch nicht dokumentiert worden sind oder
- ob es Abweichungen zwischen Dokumentationsdaten und der Implementierung gibt.

Datensätze mit XRef-Daten für Natural werden von Natural angelegt, wenn ein Natural-Objekt katalogisiert wird. XRef-Daten werden generiert, wenn ein Objekt katalogisiert wird und der Parameter `XREF` entsprechend gesetzt ist.

XRef Evaluation

Windows: Das *XRef Evaluation Plug-In* dient zur Navigation in den Referenzinformationen, die beim Absetzen eines CAT- oder STOW-Kommandos im *Development Server File* angelegt werden. Die Informationen werden in *Natural Studio* in Form einer Baumstrukturansicht angezeigt. Beide Arten von Referenzinformationen (aktive und passive, d.h. referenzierende und referenzierte Referenzen) können angezeigt werden. Das Navigieren innerhalb der Hierarchie der aktiven und passiven Referenzen ist in der Baumstrukturansicht möglich. Siehe *Single Point of Development-Dokumentation*.

Y

Year Sliding Window, YSLW (Gleitendes Jahr-Fenster)

Natural: Mechanismus, bei dem davon ausgegangen wird, dass ein Datum mit einem zweistelligen Jahr innerhalb eines „Fensters“ von 100 Jahren liegt. Innerhalb dieser 100 Jahre wird jede zweistellige Jahreseinstellung eindeutig zu einem spezifischen Jahrhundert zugeordnet, so dass es keine Verwirrung gibt, welches Jahrhundert gemeint ist. Siehe auch *YSLW - Gleitendes oder festes Jahr-Fenster* in der *Natural-Parameter-Referenz-Dokumentation*.

Z

Zap (Korrektur)

Produktfehlerkorrektur zum Ändern des Inhalts eines ausführbaren Moduls an einer angegebenen Adresse. In der IBM-Terminologie als *Patch* bekannt.

zIIP

Der *IBM System z Integrated Information Processor (zIIP)* umfasst zahlreiche Spezialprozessoren zur vereinfachten Integration verschiedenster Arten von Arbeitsbelastungen und zur optimalen Nutzung der Leistungsfähigkeit des Großrechners.