Natural-Objekte pflegen und ausführen

Ein Objekt ist Bestandteil einer Anwendung. Eine Natural-Anwendung besteht aus einem Satz Objekte, die interagieren, um eine bestimmte Aufgabe zu erfüllen.

Zu den für das Einrichten und Verwalten einer Natural-Anwendung zur Verfügung stehenden Objekten gehören Natural-Objekte und Nicht-Natural-Objekte.

Nicht-Natural-Objekte sind Objekte, die nicht mit einer Natural-Entwicklungsfunktion erstellt wurden und die außerhalb einer Natural- und Adabas-Umgebung gespeichert werden. Beispiele für Nicht-Natural-Objekte sind Bitmaps, XML-Sourcen, HTML-Dateien, DL/I-Subdateien und Predict-Regeln.

Dieser Abschnitt enthält allgemeine Informationen zu Natural-Objekten und beschreibt die Schritte, die zum Erstellen, Pflegen, Löschen und Ausführen eines Objekts erforderlich sind.

Alle Aktionen, die an einem Natural-Objekt ausgeführt werden können, erfolgen mit Hilfe von Natural-Kommandos und/oder Natural-Menüfunktionen. Informationen zur Benutzung von Kommandos und Menüfunktionen finden Sie im Abschnitt *Kommandos und Menü-Funktionen benutzen*.

Verwandte Themen im Abschnitt Libraries benutzen:

- *Objekte in einer Library auflisten*
- *Objekte in einer Library suchen*
- *Objekte in einer Library löschen*

Dieser Abschnitt behandelt folgende Themen:

- Natural-Objekte — Einführung
- Natural-Editoren oder Utilities benutzen
- Objekte auswählen und anzeigen
- Objekte erstellen und bearbeiten
- Objekte prüfen und testen
- Objekte speichern und katalogisieren
- Objekt-Verzeichnis-Informationen anzeigen
- Objekte kopieren
- Objekte drucken
- Objekte umbenennen
- Objekte verschieben

- Objekte löschen
- Programme ausführen

Natural-Objekte — Einführung

Folgende Merkmale sind typisch für ein Natural-Objekt:

- Es wird in einer Library in einer Natural-Systemdatei gespeichert.
- Es besteht aus einem katalogisierten Objekt und/oder einem Source-Objekt.
- Es wird mit einem der Natural-Editoren oder einer der Natural-Utilities erstellt.

Dieser Abschnitt behandelt folgende Themen:

- Katalogisiertes Objekt
- Source-Objekt
- Objekttypen

Katalogisiertes Objekt

Ein katalogisiertes Objekt ist die ausführbare (kompilierte) Form eines Natural-Objekts. Erstellt wird es durch den Natural-Compiler und gespeichert wird es als Objektmodul in einer Natural-Systemdatei. Unter dem Katalogisieren versteht man das Kompilieren des Sourcecodes und das Erstellen eines katalogisierten Objekts. Ausgelöst wird dieser Vorgang durch Absetzen des Natural-Systemkommandos `CATALOG` oder `STOW`.

Zur Ausführungszeit wird das katalogisierte Objekt in den Natural Buffer Pool geladen und dann durch das Natural-Laufzeitsystem ausgeführt. Natural-Objekte können nur dann ausgeführt werden oder sich gegenseitig referenzieren, wenn sie als katalogisierte Objekte in einer Natural-Systemdatei gespeichert worden sind.

Ein katalogisiertes Objekt kann nicht geändert oder dekompiliert werden.

Source-Objekt

Ein Source-Objekt (oder ein gespeichertes Objekt) enthält die menschenlesbare Form eines Natural-Sourcecodes. Der Sourcecode wird als Source-Objekt in einer Natural-Systemdatei gespeichert. Dies geschieht durch Absetzen des Systemkommandos `SAVE` oder `STOW`.

Um den in einem Source-Objekt enthaltenen Sourcecode ausführen zu können, müssen Sie den Sourcecode kompilieren, um generierten Objektcode zu erzeugen, der vom Natural-Laufzeitsystem interpretiert und ausgeführt werden kann.

Verwandte Themen:

- *Natural-Systemdateien* - *System-Architektur*-Dokumentation
- *Editors*-Dokumentation
- *Utilities*-Dokumentation

Objekttypen

In einer Natural-Anwendung können mehrere Arten von Natural-Objekten zum Einsatz kommen, um eine effiziente Anwendungsstruktur zu erstellen und um speziellen Programmierungs- und Anwendungserfordernissen gerecht zu werden. Zu den wichtigsten Natural-Objekten gehören Programme, Subprogramme, Routinen und Data Areas. Eine vollständige Übersicht und eine Beschreibung aller verfügbaren Objekttypen finden Sie im Abschnitt *Objekttypen* im *Leitfaden zur Programmierung*.

Informationen zu DDMs siehe *Natural-DDMs* in der *System-Architektur*-Dokumentation und im Abschnitt *Zugriff über Datendefinitionsmodule* im *Leitfaden zur Programmierung*.

Natural-Editoren oder Utilities benutzen

Wenn Sie ein Natural-Objekt anlegen, warten oder löschen, benutzen Sie entweder einen Natural-Editor oder eine Natural-Utility.

Es gibt Wartungsfunktionen, die nicht für alle Objekttypen benutzt werden können. Zum Beispiel können Sie keine Objekte des Typs Adapter bearbeiten.

Ein Natural-Editor wird für alle Objekttypen aufgerufen, die beim Systemkommando `EDIT` oder auf dem Bildschirm **Development Functions** angegeben werden können. Je nach angegebenem Objekttyp ruft Natural den passenden Editor auf: den Programm-Editor, den Data-Area-Editor oder den Map-Editor. So ruft Natural zum Beispiel für ein Objekt des Typs Programm automatisch den Programm-Editor auf.

Eine Natural-Utility wird bei Objekten benutzt, die entweder zusätzliche Verwaltungsarbeiten erfordern oder die nicht in einer Library gewartet werden (z.B. DDMs). Eine Utility hat einen eigenen Editor.

Die folgende Tabelle enthält eine Aufstellung der Natural-Objekttypen und der zugehörigen Editoren bzw. Utilities:

Objekttype	Editor oder Utility
Local Data Area, Global Data Area, Parameter Data Area	Data-Area-Editor
Map (Bildschirm-Layout) Map-Editor-Profil Device-Profil	Map-Editor
Programm Subprogramm Subroutine Copycode Helproutine Class Text Programm-Editor-Profil	Programm-Editor
Error Message (Fehlermeldung)	SYSERR-Utility
Data Definition Module (DDM)	SYSDDM-Utility
Command Processor Source	SYSNCP-Utility
Parameterprofil	SYSPARM-Utility
Debug-Umgebung	Debugger

Verwandte Themen:

- *Editors*-Dokumentation
- *Utilities*-Dokumentation

Dieser Abschnitt behandelt folgende Themen:

- Einen Natural-Editor aufrufen
- Eine Natural-Utility aufrufen
- Persönliches Editor-Profil erstellen

Einen Natural-Editor aufrufen

Um einen Natural-Editor aufzurufen

- Benutzen Sie das Systemkommando `EDIT`.

  Ein Beispiel für die Benutzung des Systemkommandos `EDIT` finden Sie unter *Beispiel für ein Systemkommando*.

  Oder:
  Rufen Sie im Natural-Hauptmenü (**Main Menu**) das Menü **Development Functions** auf (siehe *Natural-Hauptmenü (Main Menu)*) und wählen Sie entweder die Funktion **Create Object** oder **Edit**

Object.

Ein Beispiel für das Aufrufen eines Editors finden Sie unter *Beispiel für eine Menü-Funktion*.

Verwandtes Thema:

- *EDIT - Systemkommandos*-Dokumentation

Eine Natural-Utility aufrufen

Um eine Natural-Utility aufzurufen

- Geben Sie eines der folgenden Systemkommandos ein:

  ```
  SYSERR
  ```

  (für Fehlermeldungen)

  ```
  SYSDDM
  ```

  (für DDMs)

  ```
  SYSNCP
  ```

  (für Kommandoprozessor-Sourcen)

  ```
  SYSPARM
  ```

  (für Parameterprofile)

  ```
  TEST
  ```

  (für Debug-Umgebungen)

  Oder:
  Rufen Sie im Hauptmenü (**Main Menu**) das entsprechende Menü auf und wählen Sie die entsprechende Utility aus:

  Maintenance and Transfer Utilities für SYSERR, SYSDDM und SYSNCP,

  Development Environment Settings für SYSPARM und

Debugging and Monitoring Utilities für TEST.

Verwandtes Thema:

- *Natural-Hauptmenü (Main Menu)*

Persönliches Editor-Profil erstellen

Wenn Sie mit dem Natural-Programm-Editor oder mit dem Data-Area-Editor arbeiten, können Sie die Profilfunktion des Editors benutzen, um sich die aktuellen Einstellungen anzeigen zu lassen und persönliche Profileinstellungen vorzunehmen, die wirksam werden, wenn Sie Sourcecode bearbeiten.

Um die Editorprofileinstellungen anzuzeigen oder zu ändern

1. Geben Sie folgende Kommando in der Kommandoeingabezeile des Programm-Editors oder des Data-Area-Editors ein:

   ```
   PROFILE
   ```

2. Drücken Sie EINGABE.

   Der Bildschirm **Editor Profile** erscheint.

   Informationen zu den Feldern und Optionen auf diesem Bildschirm finden Sie unter *Editor Profile* in der *Editors*-Dokumentation.

Objekte auswählen und anzeigen

Sie können sich ein Source-Objekt anzeigen lassen, um den Sourcecode zu lesen oder zu kopieren, ohne dabei das Source-Objekt zu ändern. Der Sourcecode des angegebenen Objekts wird dann im Arbeitsbereich des zugehörigen Editors angezeigt.

Dazu können Sie entweder ein Objekt aus einer Liste auswählen oder den Namen des Objekts, das Sie anzeigen möchten, angeben.

Dieser Abschnitt beschreibt, wie Sie den Sourcecode mit dem Systemkommando `LIST` anzeigen können. Alternativ zu `LIST` können Sie die Funktion **List Object(s)** im Menü **Development Functions** benutzen. Siehe *Natural-Hauptmenü (Main Menu)*.

Um ein Objekt aus einer Liste von Objekten auszuwählen

1. Rufen Sie die Funktion **LIST Objects in a Library** auf. Siehe Schritte 1 und 2 von *Um Objekte mit LIST aufzulisten* .

2. Geben Sie in der Spalte **Cmd** neben dem gewünschten Objekt folgendes Kommando ein:

   ```
   LI
   ```

3. Drücken Sie EINGABE.

   Der Sourcecode des ausgewählten Objekts wird angezeigt.

Um den Sourcecode eines angegebenen Objekts anzuzeigen

1. Geben Sie folgendes Kommando ein:

   ```
   LIST object-name
   ```

   Dabei ist *object-name* der Name des anzuzeigenden Objekts.

   Wenn Sie für *object-name* keinen Namen angeben, wird der zurzeit im Arbeitsbereich befindliche Sourcecode angezeigt.

2. Drücken Sie EINGABE.

   Der Sourcecode des angegebenen Objekts wird angezeigt.

Verwandte Themen:

- *Objekte in einer Library auflisten*
- *LIST - Systemkommandos*-Dokumentation

Objekte erstellen und bearbeiten

Dieser Abschnitt beschreibt die Schritte, die zum Erstellen und Bearbeiten eines Natural-Objekts mit einem Natural-Editor erforderlich sind. Informationen zu den zuvor erwähnten Utilities finden Sie in der *Utilities*-Dokumentation.

- Aktuelle Umgebung prüfen
- Programmiermodus einstellen
- Natural-Programmiersprache benutzen
- Sourcecode erstellen
- Source-Objekt bearbeiten
- Objekttyp festlegen

Aktuelle Umgebung prüfen

Erstellt wird ein Natural-Objekt in der aktuellen Library in der aktuellen Systemdatei. Bevor Sie damit beginnen, ein Objekt zu erstellen oder zu bearbeiten, sollten Sie prüfen, ob Sie in der Library angemeldet sind, in der Sie das Objekt speichern oder auffinden möchten.

Eine Anleitung, wie Sie eine Library zuordnen und zwischen Libraries umschalten, finden Sie unter *Standard-Library-Zuweisung* und *Einloggen in eine Library*.

Programmiermodus einstellen

Falls erforderlich, können Sie den Programmiermodus von Structured Mode auf Reporting Mode umschalten; siehe *Um den Programmiermodus zu wechseln*.

Weitere Informationen zum Programmiermodus finden Sie auch im Abschnitt *Natural-Programmiermodi* im *Leitfaden zur Programmierung*.

Natural-Programmiersprache benutzen

Die Natural-Programmiersprache besteht aus Statements, Systemfunktionen und Systemvariablen.

Natural-Statements sind Programmieranweisungen, die zum Erstellen einer Natural-Programmsource verwendet werden.

Natural-Systemfunktionen werden beispielsweise eingesetzt, um mathematische Funktionen auszuführen.

Natural-Systemvariablen sind Standardvariablen, die von Natural zur Verfügung gestellt und erzeugt werden. Systemvariablen werden beispielsweise eingesetzt, um das Datum und die Uhrzeit zu erhalten.

Verwandte Themen:

- *Statements*-Dokumentation (Übersicht)
- *Systemfunktionen*-Dokumentation
- *Systemvariablen*-Dokumentation

Sourcecode erstellen

Dieser Abschnitt beschreibt an einem Beispiel, wie Sie Sourcecode erstellen. Verwendet wird das Systemkommando `EDIT` und der Programm-Editor. Darüber hinaus sind Beispiele für Editorkommandos und Anleitungen zum Navigieren in einer Source vorhanden.

Alternativ zum Systemkommando `EDIT` können Sie die Funktion **Create Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

Um neuen Sourcecode einzugeben

1. Geben Sie folgendes Kommando ein:

   ```
   EDIT object-type
   ```

   Dabei ist *object-type* der Typ des Objekts, das Sie anlegen wollen.

   Beispiel: Um ein Objekt des Typs Programm anzulegen, müssen Sie Folgendes eingeben:

```
EDIT PROGRAM
```

Wenn Sie für *object-type* keine Angabe machen, wird standardmäßig der Programm-Editor aufgerufen.

(Siehe auch *Objekttyp festlegen*.)

2. Drücken Sie EINGABE.

   Es erscheint der Arbeitsbereich des Programm-Editors. In der obersten Zeile wird der Typ des anzulegenden Objekts (hier: `Programm`) angezeigt, zum Beispiel:

```
>                                      > +  Program              Lib SYSTEM
  All    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
    0010
    0020
    0030
    0040
    0050
    0060
    0070
    0080
    0090
    0100
         ....+....1....+....2....+....3....+....4....+....5....+... S 0    L 1
```

3. Sollte der Arbeitsbereich nicht leer sein, geben Sie am Eingabeaufforderungszeichen des Editors (>) folgendes Editorkommando ein:

```
CLEAR
```

   und drücken Sie EINGABE.

   `CLEAR` löscht den gesamten Inhalt des Arbeitsbereichs.

4. Die Eingabezeilen sind nummeriert. Beginnen Sie in der ersten Zeile (`0010`) des leeren Arbeitsbereichs, indem Sie den Sourcecode mit Hilfe der Copy- und Paste-Funktionen einer Terminal-Emulation (z. B. Entire Connection) einfügen oder ihn selbst eintippen (siehe auch *Objekte kopieren*). Wenn Sie die automatisch stattfindende Umwandlung von Klein- in Großbuchstaben nicht wünschen, können Sie die Standardeinstellung im Editor-Profil entsprechend ändern; siehe *General Defaults* in der *Editors*-Dokumentation.

5. Wenn alle angezeigten Zeilen im Eingabebereich des Editors vollgeschrieben sind, geben Sie folgendes Editorkommando ein:

```
ADD
```

und drücken Sie EINGABE.

Das Editorkommando ADD bewirkt, dass weitere neun Leerzeilen hinzugefügt werden. In die Programm-Source werden allerdings nur die Zeilen aufgenommen, in die Sie tatsächlich etwas schreiben. Frei gelassene Zeilen werden beim nächsten Drücken von EINGABE entfernt. Sie können diese Standardeinstellung im Editor-Profil ändern, siehe *Editor Defaults* in der *Editors*-Dokumentation. Die zur Verfügung stehenden Programm-Editor-Kommandos sind in der *Program Editor*-Dokumentation beschrieben.

Um in einer Source zu blättern

1. Um an den Anfang des Sourcecodes zurückzukehren, geben Sie folgendes Editorkommando ein:

```
TOP
```

2. Um an das Ende des Sourcecodes zu gelangen, geben Sie folgendes Editorkommando ein:

```
BOT
```

3. Um im Sourcecode eine Seite nach unten zu blättern, drücken Sie PF8 oder EINGABE.
4. Um im Sourcecode eine Seite nach oben zu blättern, drücken Sie PF7.

Die zur Verfügung stehenden Programm-Editor-Kommandos sind im Abschnitt *Editor Commands for Positioning* in der *Program Editor*-Dokumentation beschrieben.

Source-Objekt bearbeiten

Sobald der Sourcecode als Source-Objekt gespeichert worden ist (siehe *Objekte speichern und katalogisieren*), rufen Sie durch Angabe des Source-Object-Namens den zugehörigen Natural-Editoren auf.

Um den Sourcecode eines Source-Objekts zu bearbeiten

1. Geben Sie folgendes Kommando ein:

```
EDIT object-name
```

Dabei ist *object-name* der Name eines vorhandenen Source-Objekts, das sich in der aktuellen Library in der aktuellen Systemdatei befindet.

2. Drücken Sie EINGABE.

Der Sourcecode des angegebenen Source-Objekts wird im Arbeitsbereich des zugehörigen Editors im Ändern-Modus angezeigt.

Alternativ zu EDIT können Sie die Funktion **Edit Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

Alternativ to EDIT können Sie auch das Systemkommando READ benutzen wie in *Objekte kopieren* beschrieben.

Verwandtes Thema:

- *EDIT - Systemkommandos*-Dokumentation

Objekttyp festlegen

Der Objekttyp wird beim Anlegen eines Objekts angegeben (die Standardeinstellung ist Programm) oder automatisch beim Einlesen eines vorhandenen Source-Objekts in den Arbeitsbereich festgelegt. Wenn Sie mit dem Programm-Editor oder dem Data-Area-Editor arbeiten, können Sie den Objekttyp jederzeit ändern, indem Sie das Editorkommando SET TYPE absetzen. Das gilt natürlich nur für Objekttypen, die mit demselben Editor bearbeitet werden.

Um den Objekttyp zu ändern

1. Geben Sie folgendes Editorkommando ein:

   ```
   SET TYPE object-type
   ```

   Zum Beispiel:

   ```
   SET TYPE SUBPROGRAM
   ```

2. Drücken Sie EINGABE.

   Der mit dem Kommando angegebene neue Objekttyp wird am Bildschirm angezeigt (hier: Subprogram).

Verwandte Themen:

- *SET TYPE - Program Editor*-Dokumentation

  SET TYPE - Data Area Editor-Dokumentation

Objekte prüfen und testen

Beim Kompilieren (Katalogisieren) des Sourcecodes wird eine Syntaxprüfung vorgenommen und ausführbarer Objektcode erzeugt. Sie können den im Arbeitsbereich enthaltenen Sourcecode kompilieren, ohne ihn zuerst als katalogisiertes Objekt gespeichert zu haben (siehe *Objekte speichern und katalogisieren*). Zusätzlich können Sie das Kompilieren des Sourcecodes für Objekte des Typs Programm mit der Ausführung des Programms kombinieren. Siehe auch *Programme ausführen*.

Um Sourcecode zwecks Syntaxprüfung zu kompilieren

1. Geben Sie folgendes Systemkommando ein:

```
CHECK
```

2. Drücken Sie EINGABE.

   Wird kein Syntaxfehler gefunden, dann wird der im Arbeitsbereich enthaltene Sourcecode kompiliert.

Um Sourcecode zwecks Programmausführung zu kompilieren

1. Geben Sie folgendes Systemkommando ein:

```
RUN
```

2. Drücken Sie EINGABE.

   Wird kein Syntaxfehler gefunden, dann wird der im Arbeitsbereich enthaltene Sourcecode kompiliert, und der generierte Code wird ausgeführt.

Verwandte Themen:

- *CHECK - Systemkommandos*-Dokumentation
- *RUN - Systemkommandos*-Dokumentation

Online-Hilfe bei Syntaxfehlern

Wenn nach dem Kompilieren keine Fehlermeldung erscheint, war die Sourcecode-Kompilierung erfolgreich.

Falls Natural jedoch beim Kompilieren einen Syntaxfehler feststellt, erscheint am Bildschirm eine Fehlermeldung. Außerdem wird die Programmzeile, die den Fehler enthält, mit einem E (Error) markiert und hell hervorgehoben, zum Beispiel:

```
>                                     > +  Program     PGM01    Lib SYSTEM
       ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
  0250   RD1. READ EMPLOYEES-VIEW BY NAME
  0260       STARTING FROM #NAME-START
  0270       THRU #NAME-END
  0280 *
  0290     IF LEAVE-DUE >= 20
  0300       PERFORM MARK-SPECIAL-EMPLOYEES
  0310     ELSE
  0320       RESET #MARK
  0330     END-IF
  0340 *
E 0350     DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20 #MARK
  0360 *
  0370   END-READ
  0380 *
  0390   IF *COUNTER (RD1.) = 0
  0400     REINPUT 'PLEASE TRY ANOTHER NAME'
  0410   END-IF
  0420 *
  0430 END-REPEAT
  0440 *
       ....+....1....+....2....+....3....+....4....+....5....+... S 49   L 25
NAT0305 Text string must begin and end on the same line.
```

Sie müssen den Fehler berichtigen, um das betreffende Objekt kompilieren zu können. Handelt es sich um einen Syntaxfehler, können Sie den Sourcecode nur als Source-Objekt speichern (siehe den folgenden Abschnitt). Informationen zum vorliegenden Fehler sowie zu Abhilfemaßnahmen erhalten Sie, wenn Sie die Online-Hilfe-Funktion benutzen.

Um Hilfe zu Fehlermeldungen zu erhalten

1. Benutzen Sie eines der folgenden Kommandos:

   ```
   HELP nnnn
   ```

   or

   ```
   ? nnnn
   ```

   Dabei steht *nnnn* für die vierstellige Fehlernummer.

   Zum Beispiel:

   ```
   HELP NAT0305
   ```

2. Drücken Sie EINGABE.

Es erscheint der Bildschirm **Natural System Message** mit einer Erläuterung zu dem angegebenen Fehler.

Weitere Informationen zur Online-Hilfe finden Sie unter *Hilfe zu Natural-Meldungen*.

Objekte speichern und katalogisieren

Sie können den zurzeit als Source-Objekt im Arbeitsbereich enthaltenen Sourcecode speichern, indem Sie das Systemkommando SAVE benutzen. SAVE bewirkt kein Katalogisieren (Kompilieren) des Sourcecodes und führt folglich auch keine Syntaxprüfung durch.

Den zurzeit im Arbeitsbereich enthaltenen Sourcecode können Sie als Source-Objekt *und* als katalogisiertes (kompiliertes) Objekt speichern, indem Sie das Systemkommando STOW benutzen.

Sie können den zurzeit im Arbeitsbereich enthaltenen Sourcecode nur katalogisieren und als katalogisiertes Objekt speichern, indem Sie das Systemkommando CATALOG benutzen. CATALOG bewirkt *keine* Speicherung des Sourcecodes als Source-Objekt, das bearbeitet werden kann.

Um Sourcecode als Source-Objekt zu speichern

1. Geben Sie folgendes Kommando am Editor-Eingabeaufforderungszeichen ein:

   ```
   SAVE object-name
   ```

   Dabei ist *object-name* der Name des Source-Objekts, das Sie anlegen wollen. Der Name des Objekts muss eindeutig sein und muss den Namenskonventionen für Objekte entsprechen.

2. Drücken Sie EINGABE.

   Der Sourcecode wird als Source-Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert.

Um Sourcecode als Source-Objekt und/oder katalogisiertes Objekt zu speichern

1. Geben Sie am Editor-Eingabeaufforderungszeichen eines der folgenden Kommandos ein:

   ```
   STOW object-name
   ```

   oder

   ```
   CATALOG object-name
   ```

   Dabei ist *object-name* der Name des Source-Objekts und/oder des katalogisierten Objekts, das Sie anlegen wollen. Der Name des Objekts muss eindeutig sein und muss den Namenskonventionen für Objekte entsprechen.

2. Drücken Sie EINGABE.

   Wenn Sie `STOW` benutzen, wird der Sourcecode als Source-Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert. Zusätzlich wird der generierte Objektcode als katalogisiertes Objekt in derselben Library in derselben Systemdatei gespeichert.

   Wenn Sie `CATALOG` benutzen, wird der Sourcecode nur als katalogisiertes Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert. Dabei wird der Sourcecode nicht als Source-Objekt in der aktuellen Systemdatei mitgespeichert (oder aktualisiert, wenn das Kommando auf einem schon vorhandenen Source-Objekt ausgeführt wird). Der Sourcecode wird nur mit `SAVE` oder `STOW` gespeichert oder aktualisiert.

Eine Beschreibung, wie Sie feststellen können, ob ein Objekt schon als Source-Objekt und/oder katalogisiertes Objekt gespeichert worden ist, finden Sie unter *Um Objekt-Verzeichnisinformationen anzuzeigen*.

Mehrere Objekte kompilieren

Mit dem Systemkommando `CATALL` können Sie mehrere, in der aktuellen Library enthaltene Source-Objekte kompilieren bzw. neu kompilieren.

Um mehrere Objekte zu kompilieren

1. Geben Sie folgendes Systemkommando ein:

   ```
   CATALL
   ```

2. Drücken Sie EINGABE.

   Es erscheint der Bildschirm **Catalog Objects in Library** (siehe Beispiel weiter unten). Hier können Sie die zu verarbeitenden Objekte, das auszuführende Kompilierungskommando und zusätzliche Optionen, z. B. das Erstellen eines Fehlerberichts, angeben.

```
10:10:55              ***** NATURAL CATALL COMMAND *****            2009-05-20
User SAG                - Catalog Objects in Library -         Library SAGTEST

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

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




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

Weitere Informationen zu den Optionen auf dem Bildschirm **Catalog Objects in Library** finden Sie unter *CATALL* in der *Systemkommandos*-Dokumentation.

Verwandtes Thema:

- *Beispiel für eine Kompilierung - System-Architektur*

Objekt-Verzeichnis-Informationen anzeigen

Im Verzeichnis (Directory) zu einem Natural-Objekt finden Sie allgemeine Informationen zu dem Objekt, z. B. Name des Objekts, Name der Library, in es untergebracht ist, und Datum des Anlegens und Änderns.

Um Objekt-Verzeichnisinformationen anzuzeigen

1. Geben Sie folgendes Kommando ein:

```
LIST DIR object-name
```

Zum Beispiel:

```
LIST DIR PGMTEST
```

2. Drücken Sie EINGABE.

   Der Bildschirm **List Directory** erscheint. Beispiel für das Programm PGMTEST:

```
09:34:31               ***** NATURAL LIST COMMAND *****              2009-05-20
User SAG                      - List Directory -                Library SAGTEST

Directory of Program PGMTEST                    Saved on ... 2008-05-14 13:30:33
------------------------------------------------------------------------------
Library .... SAGTEST    User-ID ...... SAG       Mode ....... Structured
TP-System .. COMPLETE   Terminal-ID .. 1     24
Op-System .. MVS/ESA    Transaction .. NAT42
NAT-Ver .... 4.2.4      Code page .... IBM01140
Source size ..........................    1046 Bytes

Directory of Program PGMTEST                    Cataloged on 2006-05-23 16:36:12
------------------------------------------------------------------------------
Library .... SAGTEST    User-ID ...... SAG       Mode ....... Structured
TP-System .. COMPLETE   Terminal-ID .. 1     16
Op-System .. MVS/ESA    Transaction .. NAT42
NAT-Ver .... 4.2.1      Code page .... IBM01140
Used GDA ...            Options ...... PCHECK DBSHORT PSIGNF GFID TQMARK
Size of global data ...       0 Bytes  Size in DATSIZE ......        720 Bytes
Size in buffer pool ...    3416 Bytes
Size of OPT-Code ......       0 Bytes
Initial OPT string ....
                                                              ENTER to continue
```

   Weitere Informationen zum Bildschirm **List Directory** finden Sie unter *Directory-Informationen anzeigen* im Abschnitt *LIST* in der *Systemkommandos*-Dokumentation.

Objekte kopieren

Sie können neue Objekte anlegen, indem Sie entweder den im Arbeitsbereich enthaltenen Sourcecode kopieren oder die Kopierfunktion einer Natural-Utility, z. B. SYSMAIN, verwenden.

Um Sourcecode aus dem Arbeitsbereich zu kopieren

1. Lesen Sie den Sourcecode ein, den Sie kopieren möchten. Benutzen Sie dazu folgendes Kommando:

```
READ object-name
```

   Dabei ist *object-name* der Name des Objekts, das den zu kopierenden Sourcecode enthält.

2. Drücken Sie EINGABE.

   Der Sourcecode des angegebenen Objekts wird in den Arbeitsbereich eingelesen.

3. Geben Sie eines der folgenden Kommandos ein:

```
SAVE object-name
```

oder

```
STOW object-name
```

Dabei ist *object-name* der Name des Objekts, das Sie anlegen möchten.

4. Drücken Sie EINGABE.

   Das neue Objekt wird in der aktuellen Library in der aktuellen Systemdatei als Source-Objekt gespeichert (mit SAVE) und als katalogisiertes Objekt (mit STOW).

Um ein einzelnes oder mehrere Objekte mit SYSMAIN zu kopieren

1. Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe Schritte 1 bis 4 von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.

2. Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. E bei Fehlermeldungen (Error messages).

   Geben Sie im Feld **Function Code** ein C ein (für Copy).

3. Drücken Sie EINGABE.

   Der Bildschirm **Copy Programming Objects** erscheint.

4. Geben Sie im Feld **Code** ein A ein, um alle Arten von Objektmodulen auszuwählen: katalogisierte Objekte und Source-Objekte.

   Überschreiben Sie im Feld **Sel. List** (Selection List) das Y (Yes) mit einem N (No). Y ist die Standardeinstellung.

   Geben Sie im Feld **Object Name** den Namen des Objekts ein, das Sie kopieren möchten, oder geben Sie einen Bereich von Namen an. Durch Eingabe eines Sterns (*) können Sie alle Objektnamen auswählen. Dies ist die Standardeinstellung

   (Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

   Geben Sie im Feld **Source Library** die ID der Library ein, die die zu kopierenden Objekte enthält.

   Geben Sie im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in die Sie die Objekte kopieren wollen.

   Lassen Sie alle übrigen Felder unverändert.

5. Drücken Sie EINGABE.

   Alle Source-Objekte und katalogisierten Objekte werden von der angegebenen Quell-Library in die angegebene Ziel-Library in der aktuellen Systemdatei kopiert. Danach erscheint die folgende Meldung: `Function completed successfully` (Funktion erfolgreich beendet).

Objekte drucken

Sie können den Sourcecode eines Source-Objekts drucken, indem Sie das Systemkommando `LIST` benutzen.

Außerdem können Sie eine Liste der in einer Library enthaltenen Objekte drucken; siehe *Objektliste drucken*.

Um den Sourcecode eines Source-Objekts zu drucken

1. Sie haben die Wahl zwischen den folgenden Methoden:

   - Wählen Sie ein Objekt aus einer Liste aus, indem Sie die Funktion **LIST Objects in a Library** aufrufen; siehe Schritte 1 und 2 von *Um Objekte mit LIST aufzulisten*.

     Geben Sie in der Spalte **Cmd** neben dem gewünschten Objekt folgendes Kommando ein:

     ```
     PR
     ```

     Drücken Sie EINGABE.

   - Oder:

     Geben Sie folgendes Kommando ein:

     ```
     LIST object-name
     ```

     Dabei ist *object-name* der Name des zu druckenden Objekts.

     Drücken Sie EINGABE.

     Der Sourcecode des angegebenen Objekts wird angezeigt.

     Drücken Sie PF2.

   Das Fenster **PRINT** erscheint.

2. Geben Sie im Feld **Destination** einen gültigen Druckernamen ein (ggf. fragen Sie Ihren Natural-Systemadministrator nach dem Namen eines Druckers, der in Ihrer derzeitigen Umgebung zur Verfügung steht). Wenn Sie möchten, können Sie die Seitenlänge ändern (standardmäßig sind 60 Zeilen eingestellt).

3. Drücken Sie EINGABE.

   Es erscheint der Bildschirm **Printout Specification**. Hier können Sie Druckereinstellungen vornehmen, z. B. Anzahl der Ausdrucke.

4. Drücken Sie EINGABE.

   Das angegebene Source-Objekt wird auf dem angegebenen Drucker gedruckt.

Verwandte Themen:

- *Objektliste drucken*
- *LIST - Systemkommandos*-Dokumentation

Objekte umbenennen

Sie können entweder einzelne Objekte mit dem Systemkommando `RENAME` umbenennen oder mehrere Objekte mit Hilfe der Natural-Utility SYSMAIN.

Alternativ zu `RENAME` können Sie die Funktion **Rename Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

Um ein Objekt mit `RENAME` umzubenennen

1. Geben Sie folgendes Kommando ein:

   ```
   RENAME object-name
   ```

   Dabei ist *object-name* der Name des Objekts, das Sie umbenennen wollen.

2. Drücken Sie EINGABE.
3. Das Fenster **Rename Object** erscheint. Im Feld **Name** den Namen steht bereits der Name des mit dem Kommando angegebenen Objekts.
4. Geben Sie im Feld **New Name** den neuen Objektnamen ein.

   Falls gewünscht, können Sie noch im Feld **New Type** einen neuen Objekttyp angeben.

5. Drücken Sie EINGABE.

   Die folgende Meldung erscheint: `Object renamed successfully` (Objekt erfolgreich umbenannt).

Um ein einzelnes oder mehrere Objekte mit Hilfe von SYSMAIN umzubenennen

1. Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe Schritte 1 bis 4 von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
2. Geben Sie im Feld **Object Code** ein `A` (Standardeinstellung) ein, um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. `E` bei Fehlermeldungen (Error messages).

Geben Sie im Feld **Function Code** ein R ein (für Rename).

3. Drücken Sie EINGABE.

   Es erscheint der Bildschirm **Rename Programming Objects**.

4. Geben Sie im Feld **Code** ein A ein, um alle Arten von Objektmodulen auszuwählen: katalogisierte Objekte und Source-Objekte.

   Geben Sie im Feld **Name** den Namen des umzubenennenden Objekts ein oder geben Sie einen Bereich von Namen an (im Beispiel weiter unten: TEST*). Durch Eingabe eines Sterns (*) können Sie alle Objektnamen auswählen. Dies ist die Standardeinstellung.

   (Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

   Wenn Sie nur ein einzelnes Objekt umbenennen wollen, geben Sie im Feld **New Name** einen neuen Namen ein, und überschreiben Sie im Feld **Sel. List** das Y (Yes) durch ein N (No).

   Geben Sie im Feld **Source Library** die ID der Library ein, die die umzubenennenden Objekte enthält.

   Geben Sie ggf. im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in der Sie die Objekte speichern wollen.

   Lassen Sie alle übrigen Felder unverändert.

5. Drücken Sie EINGABE.

   Es erscheint ein Fenster, in dem Sie ein Y (Yes) eingeben können, wenn Sie eine Kopie des Objekts oder der Objekte, die umbenannt werden sollen, behalten möchten.

6. Drücken Sie EINGABE.

   Wenn Sie einen Bereich von Objekten angegeben haben, erscheint das Fenster **Rename Selection** (siehe Beispiel weiter unten) mit einer Liste aller Objekte, die die angegebenen Auswahlkriterien (im Beispiel: TEST*).

   Geben Sie in der Spalte **C** neben dem bzw. den gewünschten Objekten ein A ein, um sowohl Source-Objekte als auch katalogisierte Objekte umzubenennen. Geben Sie in der Spalte **New Name** den neuen Namen ein; siehe unten:

```
16:39:39             ***** NATURAL SYSMAIN UTILITY *****            2009-05-20
 User SAG                     -    Rename Selection  -

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


  C  Name     Type    S/C   New Name      C  Name     Type    S/C   New Name
  -  -------- ------  ---   --------      -  -------- ------  ---   --------
  A  TEST+    Progrm  S     PGMT+___      _  TEST+2   Progrm  S     _________
  A  TESTCHAR Progrm  S/C   CHARTEST      A  TESTDIR  Progrm  S     PGMDIR__
  _  TESTDISP Progrm  S/C   _________      _  TESTDIS2 Progrm  S/C   _________
  _  TESTMMO  Proc    S/C   _________      A  TESTPGM_ Progrm  S/C   PGMTEST_
  _  TESTTEST Progrm  S     _________      _  TESTXXX2 Progrm  S     _________
  A  TEST1    Subpgm  S/C   SUBTEST1      A  TEST10   Subpgm  S/C   SUB10___
  A  TEST2    Subpgm  S/C   SUBTEST2      _  TEST666  Progrm  S/C   _________

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









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

7. Drücken Sie EINGABE.

   Es erscheint eine Spalte **Message Text**, in der neben jedem umbenannten Objekt eine Bestätigungsmeldung angezeigt wird. Je nachdem, ob Sie die Option, eine Originalkopie des umzubenennenden Objekts zu behalten, markiert hatten oder nicht, erscheint entweder die Meldung `Renamed as` oder `Copied as` (Umbenannt als bzw. Kopiert als).

Objekte verschieben

Sie können mit einer Natural-Utility, z. B. SYSMAIN, Objekte von einer Library in eine andere verschieben.

Um Objekte mit Hilfe von SYSMAIN-Menüfunktionen zu verschieben

1. Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe Schritte 1 bis 4 von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.

2. Geben Sie im Feld **Object Code** ein `A` ein (Standardeinstellung), um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. `E` bei Fehlermeldungen (Error messages).

   Geben Sie im Feld **Function Code** ein `M` ein (für Move).

3. Drücken Sie EINGABE.

Es erscheint der Bildschirm **Move Programming Objects**.

4. Geben Sie im Feld **Code** ein `A` ein, um alle Arten von Objektmodulen auszuwählen: Source-Objekte und katalogisierte Objekte.

   Überschreiben Sie im Feld **Sel. List** (Selection List) das `Y` (Yes) durch ein `N` (No). `Y` ist die Standardeinstellung.

   Geben Sie im Feld **Object Name** den Namen des Objekts ein, das Sie verschieben möchten, oder geben Sie einen Bereich von Namen an. Durch Eingabe eines Sterns (*) wählen Sie alle Objektnamen aus. Dies ist die Standardeinstellung.

   (Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

   Geben Sie im Feld **Source Library** die ID der Library ein, die die zu verschiebenden Objekte enthält.

   Geben Sie im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in die Sie das oder die Objekte verschieben wollen.

   Lassen Sie alle übrigen Felder unverändert.

5. Drücken Sie EINGABE.

   Es erscheint ein Bestätigungsfenster.

6. Drücken Sie EINGABE, um die Verschiebeaktion zu bestätigen, oder einen Punkt (.), um die Aktion abzubrechen.

   Wenn die Verschiebeaktion erfolgreich ausgeführt wurde, sind alle angegebenen Source-Objekte und katalogisierten Objekte von der angebenen Quell-Library in die angegebene Ziel-Library verschoben worden, und die folgende Meldung erscheint: `Function completed successfully` (Funktion erfolgreich ausgeführt).

Objekte löschen

Sie können Objekte löschen, indem Sie entweder das Systemkommando `DELETE`, das Systemkommando `LIST` oder eine Natural-Utility, z. B. SYSMAIN, benutzen. Eine Anleitung, wie Sie Objekte mit `LIST` oder SYSMAIN löschen, finden Sie unter *Objekte in einer Library löschen*.

Als Alternative zu `DELETE` können Sie die Funktion **Delete Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

Um ein einzelnes oder mehrere Objekte mit `DELETE` zu löschen

1. Geben Sie eines der folgenden Kommandos ein:

```
DELETE object-name
```

oder

```
DELETE object-name*
```

oder

```
DELETE *
```

Dabei ist:

object-name der Name des zu löschenden Objekts.

*object-name** ein spezieller Bereich auszuwählender Objekte. Zum Beispiel werden mit `TEST*` alle Objekte ausgewählt, die mit der Zeichenfolge TEST beginnen.

Durch Eingabe eines Sterns (*) werden alle in der aktuellen Library in der aktuellen Systemdatei enthaltenen Objekte ausgewählt.

2. Drücken Sie EINGABE.

   - Wenn Sie ein einzelnes Objekt angegeben haben, erscheint das Fenster **DELETE**.

     Geben Sie den Namen des Objekts ein, um die Löschaktion zu bestätigen.

   - Wenn Sie einen Bereich von Objekten eingegeben haben, erscheint das Fenster **Delete Sources and Objects**.

     Geben Sie in der Spalte **M** neben dem oder den zu löschenden Objekten ein `B`, um sowohl das Source-Objekt als auch die katalogisierten Objekte zu löschen.

     Drücken Sie EINGABE.

     Das Fenster **DELETE** erscheint

     Markieren Sie eine Option, indem Sie ein beliebiges Zeichen neben der geünschten Option eingeben.

     Confirm each deletion ruft das Fenster **DELETE** für das erste zu löschende Objekt auf. Geben Sie den Namen des Objekts ein, und drücken Sie EINGABE, um die Löschung zu bestätigen und das Fenster **DELETE** für die Löschung des nächsten Objekts zu öffnen.

     Delete without confirmation bewirkt die sofortige Ausführung des Löschung oder der Löschungen. **Exit (no deletion)** bewirkt den Abbruch des oder der Löschvorgänge.

3. Drücken Sie EINGABE.

   Es erscheint der Bildschirm **Delete Sources and Objects**. Dort wird neben den für die Löschung ausgewählten Objekten eine Meldung. Die Meldung besagt, dass das Objekt gelöscht wurde (`deleted`) bzw. dass der Löschvorgang abgebrochen wurde (`not deleted`)

Verwandtes Thema:

- *DELETE - Systemkommandos*-Dokumentation

Programme ausführen

Sie können ein Objekt des Typs Programm mit einem Systemkommando ausführen. Alle anderen Objekttypen können dagegen nur ausgeführt oder aufgerufen werden, wenn Sie in diesem Programm oder in einem untergeordneten Objekt referenziert werden. Siehe auch *Mehrere Stufen (Levels) aufgerufener Objekte* im *Leitfaden zur Programmierung*.

Um ein Programm auszuführen, können Sie entweder das Systemkommando RUN oder EXECUTE benutzen.

Als Alternative zu EXECUTE können Sie die Funktion **Execute Program** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

RUN führt den zurzeit im Arbeitsbereich befindlichen Sourcecode oder ein in einer Systemdatei gespeichertes katalogisiertes Objekt aus.

EXECUTE führt nur katalogisierte Objekte aus. Im Gegensatz zu RUN werden bei EXECUTE keine letzten Änderungen berücksichtigt, die an dem entsprechenden Code im Arbeitsbereich vorgenommen worden sein können. Denn diese Änderungen werden nur dann berücksichtigt, nachdem das Source-Objekt aktualisiert und neu kompiliert worden ist.

Durch die Ausführung eines katalogisierten Objekts wird der zurzeit im Arbeitsbereich befindliche Sourcecode nicht beeinflusst.

Um ein Programm mit RUN auszuführen

1. Geben Sie folgendes Kommando ein:

   ```
   RUN
   ```

   oder

   ```
   RUN program-name
   ```

   Dabei ist *program-name* der Name des Source-Objekts vom Typ Programm, das in den Arbeitsbereich eingelesen wird.

2. Drücken Sie EINGABE.

   Wenn kein Syntaxfehler gefunden wurde, wird der im Arbeitsbereich enthaltene Sourcecode kompiliert und ausgeführt.

Um ein Programm mit EXECUTE auszuführen

1. Geben Sie folgendes Kommando ein:

   ```
   EXECUTE program-name
   ```

   Dabei ist *program-name* der Name eines katalogisierten Objekts des Typs Programm.

   Das Schlüsselwort EXECUTE ist optional; es genügt, wenn Sie einfach nur den Namen des Programms (*program-name*) angeben. it is sufficient to specify.

2. Drücken Sie EINGABE.

   Das Objekt wird ausgeführt.

Verwandte Themen:

- *RUN - Systemkommandos*-Dokumentation
- *EXECUTE - Systemkommandos*-Dokumentation
- *Objektausführung - System-Architektur*-Dokumentation
- *Suchreihenfolge bei der Objekt-Ausführung*
- *Beispiel für das Laden und Ausführen eines Objekts - System-Architektur*-Dokumentation