

Natural-Objekte erstellen, pflegen und ausführen

Die Natural-Systemkommandos werden eingesetzt zum Erstellen, Pflegen und Ausführen von Natural-Objekten. In Natural Studio stehen Menübefehle für mehrere Systemkommandos zur Verfügung. Beispielsweise entspricht der Menübefehl **Check** (Prüfen) dem Natural-Systemkommando CHECK.

Alle Systemkommandos, einschließlich denen, für die ein Menübefehl verfügbar ist, können in der Kommandozeile eingegeben werden. Weitere Informationen finden Sie unter *Kommandos in der Kommandozeile absetzen*.

Dieses Kapitel beschreibt die Systemkommandos, die aus einem Menü gewählt werden können. Die folgenden Themen werden behandelt:

- Editoren aufrufen
- Objekte auflisten
- Objekte prüfen
- Objekte speichern
- Objekte in Source- und Objekt-Form speichern
- Objekte katalogisieren
- Objekte in einer Library katalogisieren
- Objekte mit Run ausführen
- Objekte mit Execute ausführen
- Editor-Fenster löschen

Natural-Systemkommandos können auf allen Plattformen ausgeführt werden, auf denen Natural zur Verfügung steht. Detaillierte Informationen zu den verfügbaren Systemkommandos finden Sie in der *Systemkommandos*-Dokumentation.

Anmerkung:

Die Systemkommandos gelten nicht für Ressourcen. Siehe *Ressourcen verwalten*.

Editoren aufrufen

Ein Editor wird aufgerufen, wenn Sie ein neues Natural-Objekt erstellen, oder wenn Sie ein bereits vorhandenes Natural-Objekt öffnen. Ausnahme: ein Editor erscheint nicht für Klassen (weitere Informationen erhalten Sie in den folgenden Abschnitten).

Die folgenden Themen werden erörtert:

- Natural-Objekt erstellen
- Vorhandene Natural-Objekte öffnen

Das entsprechende Systemkommando zum Erstellen und Öffnen eines Natural-Objekts ist `EDIT`.

Informationen zur Programmiersprache finden Sie im *Leitfaden zur Programmierung* und in der *Statements*-Dokumentation.

Siehe auch *Sperren von Objekten* in der Dokumentation *Remote-Entwicklung mit SPoD*, die Informationen zum Sperr-Mechanismus in der lokalen Umgebung und Remote-Umgebung enthält.

Natural-Objekt erstellen

Ein Natural-Objekt wird stets in der Library und Umgebung erstellt, die in der Kommandozeile gerade angegeben ist.

▶ Ein neues Natural-Objekt erstellen

1. Achten Sie darauf, dass die Library, in der Sie das neue Objekt in speichern möchten, markiert ist.
2. Wählen Sie aus dem Menü **Object** den Befehl **New > object-type**. Eine detaillierte Beschreibung jedes unter Natural verfügbaren Objekttyps finden Sie unter *Objekttypen* im *Leitfaden zur Programmierung*.

Anstatt eines Objekttyps können Sie auch den Befehl **Dialog Wizard** (Dialog-Assistent) wählen.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **New Source > object-type**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen sie die entsprechende Schaltfläche in der Symbolleiste (Schaltflächen in der Symbolleiste stehen nur für einige der Objekttypen zur Verfügung).

Oder:

Wenn Sie ein Programm erstellen möchten, drücken Sie `STRG+N` (Tastenkombinationen stehen nicht zum Erstellen anderer Objekttypen zur Verfügung).

Es erscheint ein leeres Editor-Fenster für den ausgewählten Objekttyp. Natural bietet unterschiedliche Editoren für die verschiedenen Objekttypen. Beispielsweise werden Programme und Subprogramme mit dem Programm-Editor editiert, und die verschiedenen Arten von Data-Areas (Local-, Global- und Parameter-Data-Areas) werden mit dem Data-Area-Editor editiert. Detaillierte Informationen zur Benutzung der unterschiedlichen Editoren entnehmen Sie der *Editors*-Dokumentation.

Ausnahmen:

- Wenn Sie den Befehl **Dialog Wizard** gewählt haben, erscheint ein Assistent. Weitere Informationen finden Sie unter *Dialog Wizard* im Abschnitt *Dialog Editor* der *Editors*-Dokumentation.

- Wenn Sie ein DDM erstellen, erscheint zuerst ein Dialog, in dem Sie den Datenbank-Typ auswählen. Weitere Informationen finden Sie unter *Creating DDMs* im Abschnitt *DDM Editor* der *Editors*-Dokumentation.
- Klassen können nur im Logical-View und Flat-View erstellt werden. Wenn Sie eine Klasse erstellen, erscheint im Verzeichnisbaum sofort eine neue Klasse mit dem Standardnamen "NEWCLS". Ein Editor wird nicht aufgerufen. Statt dessen markieren Sie die Klasse, und wählen dann den Befehl **New** aus dem Kontextmenü, um weitere Knoten zu Ihrer Klasse hinzuzufügen (zum Beispiel eine Schnittstelle). Solange Sie die Klasse nicht speichern, steht Sie nicht im Source-Format zur Verfügung. Weitere Informationen finden Sie unter *Class Builder* in der *Editors*-Dokumentation.

▶ Ein neues Natural-Objekt erstellen, indem Sie den Objekttyp vorselektieren

1. Markieren Sie einen Gruppen-Knoten im Logical-View des Library-Workspace oder in einem List-View-Fenster (zum Beispiel den Knoten **Programs**).
2. Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **New**.

Da der Objekttyp durch die vorherige Auswahl bereits festgelegt worden ist, erscheint ein leeres Editor-Fenster für diesen Objekttyp.

Anmerkung:

Wenn Sie den Knoten **Classes** oder **Resources**, oder einen beliebigen Gruppen-Knoten innerhalb dieses Knotens, markieren und dann den Befehl **New** aus dem Kontextmenü wählen, wird eine neue Klasse oder Ressource mit einem Standardnamen zum Verzeichnisbaum hinzugefügt. Ein Editor wird in diesem Fall nicht aufgerufen.

Vorhandene Natural-Objekte öffnen

Wenn Sie ein Natural-Objekt öffnen, wird es in dem entsprechenden Natural-Editor geöffnet.

Ausnahmen:

- Wenn Sie ein generiertes Programm oder einen Dialog doppelt anklicken, für das/den keine Source mehr vorhanden ist, wird das Programm oder der Dialog ausgeführt. Der Befehl **Open** steht in diesem Fall nicht zur Verfügung.
- Wenn Sie den Knoten für eine Klasse öffnen, erscheint ein List-View-Fenster. Um den Inhalt einer Klasse anzusehen, müssen Sie einfach die Knoten im Verzeichnisbaum aufklappen (nur möglich im Logical-View und Flat-View).

Die folgende Beschreibung gilt für alle Natural-Objekte außer Klassen. Sie gilt nur für die Source eines Objekts (und nicht für das generierte Programm, das im File-View sichtbar ist).

▶ Bereits vorhandene Natural-Objekte öffnen

1. Markieren Sie das Natural-Objekt, das Sie öffnen möchten, rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Open** (Öffnen).

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren.

Oder:

Klicken Sie das Natural-Objekt doppelt an.

Oder:

Wenn die Objekt-Symboleiste angezeigt wird, wählen Sie das Natural-Objekt aus, und wählen Sie die folgende Schaltfläche in der Symboleiste.



Objekte auflisten

Sie können sich den Source-Code eines Objekts nur im Lesezugriff anzeigen lassen. Das Objekt wird in dem entsprechenden Editor angezeigt. "Object Listing" erscheint in seiner Titelleiste. Sie können den Inhalt kopieren, aber Sie können ihn nicht ändern. Dies ist hilfreich, wenn Sie zum Beispiel den Inhalt eines Objekts in einer inaktiven Umgebung ansehen möchten, in der es nicht möglich ist, das Objekt zum Editieren zu öffnen.

Anmerkung:

Wenn Sie eine Klasse auflisten, erscheint der Code in einem Editor-Fenster.

Siehe auch die Beschreibung des Systemkommandos `LIST`.

► Objekte auflisten

1. Markieren Sie das/die Natural-Objekt(e), das/die Sie auflisten möchten.

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren.

2. Wählen Sie aus dem Menü **Object** den Befehl **List**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **List**.

Oder:

Wenn die Objekt-Symboleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symboleiste.



Objekte prüfen

Sie können den Source-Code eines Objekts (außer Klassen) auf Syntaxfehler prüfen.

Anmerkung:

Syntax-Prüfungen werden auch durchgeführt, wenn Sie ein Objekt mit dem Befehl **Run** ausführen oder mit dem Befehl **Stow** speichern, oder wenn Sie alle Objekte in einer Library mit dem Befehl **Cat All** katalogisieren.

Siehe auch die Beschreibung des Systemkommandos CHECK.

▶ Objekte prüfen

1. Aktivieren Sie den Editor, der das Objekt enthält, das Sie prüfen möchten.

Oder:

Markieren Sie das Objekt im Library-Workspace oder List-View-Fenster. Wenn ein Editor-Fenster nicht geöffnet wurde, können Sie nur die Schaltfläche in der Symbolleiste oder die Tastenkombination benutzen (siehe unten).

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren.

2. Wählen Sie aus dem Menü **Object** den Befehl **Check**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.

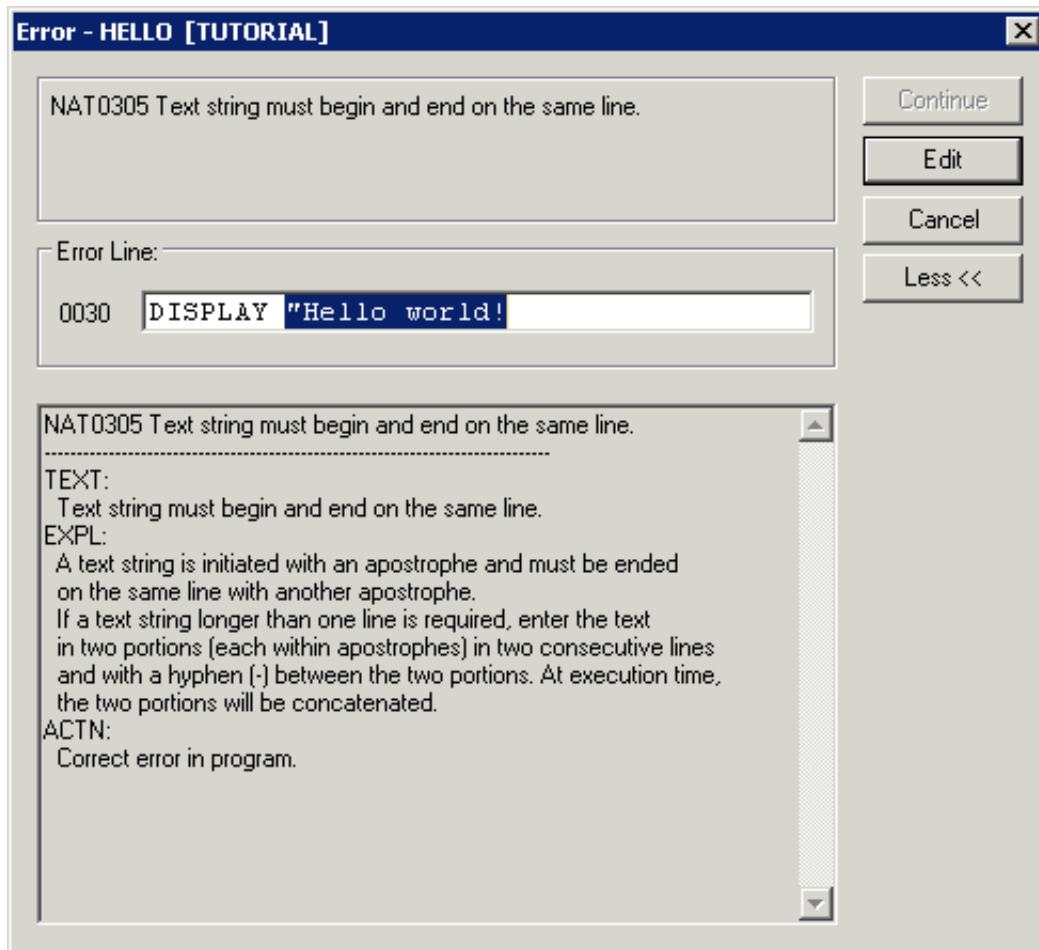


Oder:

Drücken Sie STRG+E.

Wenn keine Fehler gefunden werden, erscheint eine Meldung, die anzeigt, dass die Prüfung erfolgreich war. Die Meldung erscheint entweder in der Statusleiste oder in einem Dialogfeld. Dies ist abhängig von der Einstellung **Display success messages** in den Workspace-Optionen.

Wenn ein Fehler gefunden wurde, erscheint ein Dialogfeld mit Informationen zu dem Fehler.
Beispiel:



3. Korrigieren Sie den Fehler im Dialogfeld.
4. Wählen Sie die Befehlsschaltfläche **Continue** (Weiter), um den nächsten Fehler zu finden.

Wenn kein Fehler mehr gefunden wird, wird das Dialogfeld geschlossen.

Anmerkung:

Anstatt der Befehlsschaltfläche **Continue**, können Sie auch die Schaltfläche **Edit** wählen. Das Dialogfeld wird dann geschlossen, und Sie können den Fehler direkt im Programm-Editor korrigieren.

Objekte speichern

Sie können die Source-Form eines Natural-Objekts speichern. Die Syntax wird nicht überprüft.

Anmerkung:

Sie können ein gespeichertes Objekt mit dem Befehl **Run** ausführen. Solange Sie ein Objekt nicht mit dem Befehl **Stow** speichern, können Sie es nicht mit dem Befehl **Execute** ausführen.

Natural Studio bietet unterschiedliche Befehle zum Speichern. Sie sind weiter unten beschrieben.

- Objekt speichern

- Objekt unter einem neuen Namen speichern
- Alle Objekte speichern

Wenn Sie Natural Studio schließen, und Ihre letzten Änderungen nicht gespeichert worden sind, werden Sie gefragt, ob Sie diese Änderungen speichern möchten.

Siehe auch die Beschreibung des Systemkommandos `SAVE`.

Objekt speichern

Sie können das Source-Objekt in dem aktiven Editor-Fenster speichern.

Klassen werden unterschiedlich behandelt, da sie nicht in einem Editor-Fenster editiert werden.

▶ Ein Objekt (außer einer Klasse) speichern

1. Aktivieren Sie den Editor, der das Objekt enthält, das Sie speichern möchten.
2. Wählen Sie aus dem Menü **Object** den Befehl **Save**.

Oder:

Wenn die Objekt-Symboleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symboleiste.



Oder:

Drücken Sie `STRG+S`.

Anmerkung:

Wenn Sie ein Objekt zum ersten Mal speichern, erscheint das Dialogfeld **Save As** (Speichern unter), und Sie müssen einen Namen für das Objekt angeben. Siehe unten.

▶ Eine Klasse speichern

1. Markieren Sie den Grund-Knoten der Klasse, die Sie speichern möchten.
2. Wählen Sie aus dem Menü **Object** den Befehl **Save**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Save**.

Oder:

Wenn die Objekt-Symboleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symboleiste.



Oder:

Drücken Sie `STRG+S`.

Anmerkung:

Wenn Sie eine Klasse zum ersten Mal speichern, erscheint das Dialogfeld **Save As** (Speichern unter), und Sie müssen einen Modul-Namen angeben. Siehe unten.

Objekt unter einem neuen Namen speichern

Sie können ein neues Objekt erstellen, indem Sie den aktuellen Inhalt des Editor-Fensters in ein neues Objekt kopieren. Das Editor-Fenster für das ursprüngliche Objekt wird automatisch geschlossen.

Bei mehreren Objekttypen ist es auch möglich, eine andere Codepage für das Objekt auszuwählen. Weitere Informationen finden Sie in der Dokumentation *Unicode and Code Page Support*.

Vorsicht:

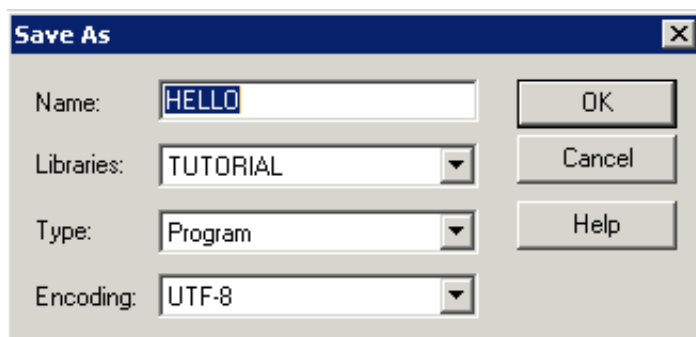
Wenn seit dem letzten Speichern Änderungen erfolgt sind, enthält nur das neue Objekt die Änderungen. Das ursprüngliche Objekt wird geschlossen, ohne die Änderungen zu speichern.

Klassen werden unterschiedlich gehandhabt, da sie nicht in einem Editor-Fenster editiert werden. Wenn eine neue Klasse im Verzeichnisbaum erstellt wird, wird sie nicht automatisch gespeichert. Der Klassen-Name, der 32 Zeichen lang sein kann, erscheint im Logical-View. Der Klassenmodul-Name, d.h. der Name des eigentlichen Natural-Objekts, der bis zu 8 Zeichen lang sein kann, wird erstellt, wenn die Klasse zum ersten Mal gespeichert wird; dieser Name erscheint nicht im Logical-View. Um den Natural-Objektnamen für eine Klasse herauszufinden, können Sie die Eigenschaften der Klasse anzeigen, oder Sie können den Namen im Flat-View oder File-View ersehen. Weitere Informationen finden Sie unter *Class Builder* in der *Editors*-Dokumentation.

▶ Ein Objekt (außer einer Klasse) unter einem anderen Namen speichern

1. Aktivieren Sie den Editor, der das Objekt enthält, das Sie unter einem anderen Namen speichern möchten.
2. Wählen Sie aus dem Menü **Object** den Befehl **Save As**.

Das Dialogfeld **Save As** (Speichern unter) erscheint. Beispiel für ein Programm.



3. Geben Sie einen neuen Namen für das Objekt an.

Siehe auch *Namenskonventionen für Objekte*.

4. Optional. Definieren Sie eine andere Library, indem Sie sie im Dropdown-Listenfeld **Library** auswählen.

- Optional (wenn verfügbar für den aktuellen Objekttyp). Definieren Sie einen anderen Objekttyp, indem Sie ihn im Dropdown-Listefeld **Type** auswählen.

Das Dropdown-Listefeld enthält nur diejenigen Typen, die für den aktuellen Objekttyp zulässig sind. Sie können zum Beispiel ein Programm in ein Subprogramm ändern. Siehe auch *Typ eines Natural-Objekts ändern*.

Anmerkung:

Ein Typ erscheint nicht, wenn Sie eine Map, einen Dialog oder ein DDM speichern.

- Optional (wenn verfügbar für den aktuellen Objekttyp). Definieren Sie eine andere Codepage, indem Sie sie im Dropdown-Listefeld **Encoding** (Kodierung) auswählen.
- Wählen Sie die Befehlsschaltfläche **OK**.

Das neue Objekt wird erstellt. Der Name, die Library und der Objekttyp des neuen Objekts erscheinen jetzt in der Titelleiste des Editor-Fensters.

Das neue Objekt steht in allen Views zur Verfügung (alphabetisch sortiert). Wenn ein Filter gerade aktiv ist, kann es vorkommen, dass das neue Objekt nicht angezeigt wird, da sein Name mit anderen Zeichen anfängt. Weitere Informationen finden Sie unter *Libraries und Objekte filtern*.

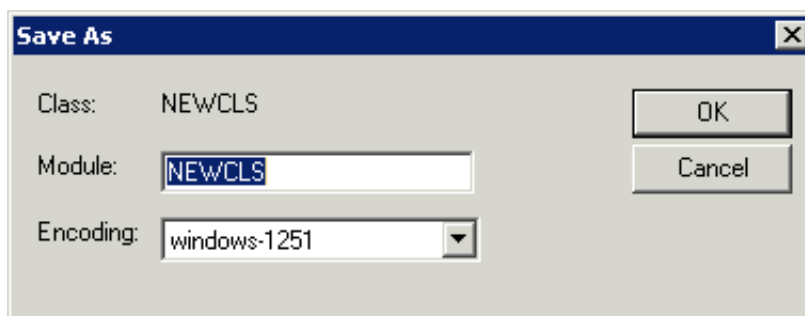
▶ **Eine Klasse unter einem neuen Modul-Namen speichern**

- Markieren Sie den Grund-Knoten der Klasse, die Sie unter einem anderen Namen speichern möchten.
- Wählen Sie aus dem Menü **Object** den Befehl **Save As**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Save As**.

Es erscheint das Dialogfeld **Save As**.



- Geben Sie den neuen Modul-Namen an.

Das Dialogfeld zeigt den aktuellen Modul-Namen für bereits vorhandene Klassen und den Standard-Modulnamen für neue Klassen an. Der Modul-Name kann bis zu 8 Zeichen lang sein.

- Optional. Definieren Sie eine andere Codepage, indem Sie sie im Dropdown-Listefeld **Encoding** (Kodierung) auswählen.

5. Wählen Sie die Befehlsschaltfläche **OK**.

Alle Objekte speichern

Sie können alle Objekte speichern, für die ein Editor-Fenster geöffnet ist. Sie können auch alle Klassen speichern.

▶ Alle Objekte speichern

1. Wählen Sie aus dem Menü **Object** den Befehl **Save As**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.



Oder:

Drücken Sie STRG+UMSCHALT+A.

Die Objekte werden gespeichert. Wenn ein Objekt vorher noch nicht gespeichert wurde, erscheint das Dialogfeld **Save As**.

Objekte in Source- und Objekt-Form speichern

Wenn Sie ein Objekt mit dem Befehl **Stow** speichern, wird es überprüft und kompiliert. Wenn keine Fehler gefunden werden, wird die Source-Form des Objekts gespeichert, und das sich daraus ergebende generierte Programm wird gespeichert (zusätzlich zur Source).

Anmerkung:

Wenn Sie ein Objekt in Source- und Objekt-Form gespeichert haben, können Sie es mit dem Befehl **Execute** ausführen.

Siehe auch die Beschreibung des Systemkommandos **STOW**.

▶ Objekte in Source- und Objekt-Form speichern

1. Aktivieren Sie den Editor, der das Objekt enthält, das Sie in Source- und Objekt-Form speichern möchten.

Oder:

Markieren Sie das Objekt im Library-Workspace oder List-View-Fenster. Wenn kein Editor-Fenster geöffnet wurde, können Sie nur das Kontextmenü, die Schaltfläche in der Symbolleiste oder die Tastenkombination benutzen (siehe unten).

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren.

2. Wählen Sie aus dem Menü **Object** den Befehl **Stow**.

Oder:

Rufen Sie das Kontextmenü für das markierte Objekt auf, und wählen Sie den Befehl **Stow**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.



Oder:

Drücken Sie STRG+T.

Wenn ein Objekt vorher nicht gespeichert wurde, erscheint das Dialogfeld **Stow As**. In diesem Dialogfeld müssen Sie dieselben Informationen angeben wie in dem Dialogfeld **Save As**.

Objekte katalogisieren

Wenn Sie ein Objekt katalogisieren, wird es überprüft und kompiliert. Wenn keine Fehler gefunden werden, wird das resultierende generierte Programm gespeichert.

Im Gegensatz zum Befehl **Stow** wird der Source-Code nicht gespeichert, wenn Sie ein Objekt katalogisieren und dabei zum Beispiel der Editor geöffnet ist und die letzten Änderungen noch nicht gespeichert wurden. Allerdings werden die letzten Änderungen im Editor stets während des Kompilierungsprozesses berücksichtigt, auch wenn sie nicht gespeichert wurden.

Siehe auch die Beschreibung des Systemkommandos CATALOG.

▶ Objekte katalogisieren

1. Markieren Sie das Objekt im Library-Workspace oder List-View-Fenster.

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren.

2. Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Catalog**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.



Objekte in einer Library katalogisieren

Sie können alle Objekte oder ausgewählte Objekte in einer Library katalogisieren. Statt dessen können Sie mit dieser Funktion aber auch Objekte prüfen (CHECK) oder speichern (SAVE oder STOW).

▶ Objekte in einer Library katalogisieren

1. Markieren Sie die Library, in der die Objekte katalogisiert werden sollen.

Tipp:

In einem List-View-Fenster ist es möglich, mehrere Libraries gleichzeitig zu markieren und zu katalogisieren.

2. Wählen Sie aus dem Menü **Library** den Befehl **Cat All**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Cat All**.

Oder:

Wenn die Library-Symbolleiste angezeigt, wählen Sie die folgende Schaltfläche in der Symbolleiste.



Es erscheint das folgende Dialogfeld:

Catalog Objects in Library: TUTORIAL

Starting from: *

Selection

Apply action to all sources

Apply action only to existing modules

Action

Catalog

Renumber source lines

Object Types

<input checked="" type="checkbox"/> DDMs (v)	<input checked="" type="checkbox"/> Maps
<input checked="" type="checkbox"/> Global Data Areas	<input type="checkbox"/> Generate new map source
<input checked="" type="checkbox"/> Local Data Areas	<input checked="" type="checkbox"/> Programs
<input checked="" type="checkbox"/> Parameter Data Areas	<input checked="" type="checkbox"/> Dialogs
<input checked="" type="checkbox"/> Subroutines	<input checked="" type="checkbox"/> Classes (4)
<input checked="" type="checkbox"/> Subprograms (N)	<input checked="" type="checkbox"/> Functions (7)
<input checked="" type="checkbox"/> Help routines	<input checked="" type="checkbox"/> Adapters (8)

Select All Clear All

OK Cancel Help

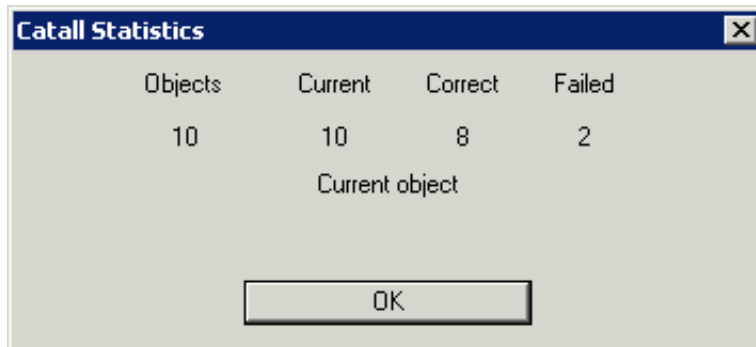
Anmerkung:

Wenn Sie in einer Remote-Entwicklungsumgebung mit SPoD arbeiten, werden die Optionen grau dargestellt, die für diese Umgebung nicht benutzt werden können.

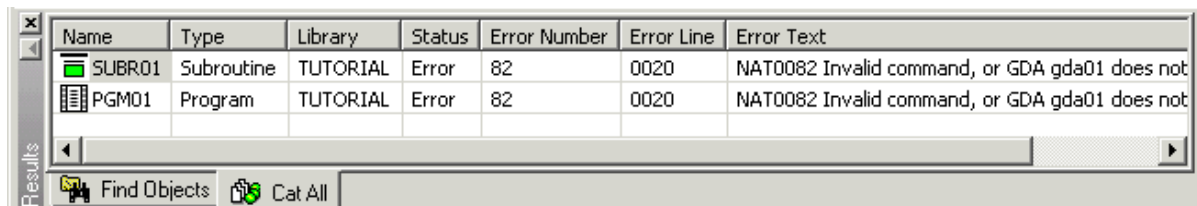
3. Geben Sie die Objekttypen an und die Aktion, die ausgeführt werden soll. Detaillierte Informationen zu den Optionen in diesem Dialogfeld entnehmen Sie der Beschreibung des Systemkommandos CATALOG.

4. Wählen Sie die Befehlsschaltfläche **OK**.

Es erscheint ein Dialogfeld mit Statistikinformationen zu dem ausgeführten Befehl. Beispiel:



Wenn ein Objekt nicht katalogisiert werden konnte, wird es im Results-Fenster aufgelistet. Beispiel:



Die Objekte werden in der Reihenfolge aufgelistet, in der sie verarbeitet wurde. In Verbindung mit jedem Objekt-Namen steht eine Fehlernummer (Ursache, warum das Objekt nicht katalogisiert werden konnte) und eine Fehlerzeile (die Zeilenposition im Source-Code, in der der Fehler gefunden werden kann).

Wenn Sie ein Objekt vom Results-Fenster aus öffnen, wird die Zeile, in der der Fehler aufgetreten ist, automatisch im Editor markiert.

Anmerkung:

Standardmäßig werden nur Fehler im Results-Fenster aufgelistet. Es ist auch möglich, Informationen zu allen katalogisierten Objekten zu zeigen, ungeachtet der Tatsache, ob ein Fehler aufgetreten ist oder nicht. Weitere Informationen finden Sie unter *Workspace-Optionen*.

Objekte mit Run ausführen

Sie können Programme und Dialoge mit dem Befehl **Run** ausführen. Wenn Sie ein solches Objekt mit **Run** ausführen, wird es überprüft und temporär kompiliert. Wenn keine Fehler gefunden werden, wird das Objekt ausgeführt. Ein generiertes Programm wird nicht gespeichert.

Die letzten Änderungen im Editor werden stets während des Kompilierungsprozesses berücksichtigt, auch wenn sie noch nicht gespeichert wurden.

Siehe auch die Beschreibung des Systemkommandos RUN.

▶ Objekte mit Run ausführen

1. Aktivieren Sie den Editor, der das Objekt enthält, das Sie ausführen möchten.
2. Wählen Sie aus dem Menü **Object** den Befehl **Run**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.



Oder:

Drücken Sie STRG+R.

In der lokalen Umgebung erscheint ein Ausgabefenster, wenn ein Natural-Programm Ausgaben auf den Bildschirm schreibt. Weitere Informationen finden Sie unter *Output Window* in der *Operations*-Dokumentation.

In einer Remote-Entwicklungsumgebung erscheint entweder der Natural Web I/O Interface Client oder ein Terminal-Emulationsfenster anstatt eines Ausgabefensters. Weitere Informationen finden Sie unter *Natural Web I/O Interface Client* und *Terminal-Emulation* in der Dokumentation *Remote-Entwicklung mit SPoD*.

Objekte mit Execute ausführen

Sie können Programme und Dialoge mit dem Befehl **Execute** ausführen. Wenn Sie ein solches Objekt mit dem Befehl **Execute** ausführen, wird es überprüft und kompiliert. Wenn keine Fehler entdeckt werden, wird das sich daraus ergebende generierte Programm gespeichert und ausgeführt.

Im Gegensatz zum Befehl **Run**, der stets Ihre letzten Änderungen auf das Objekt anwendet, benutzt der Befehl **Execute** stets die in Source- und Objekt-Form gespeicherte Version Ihres Objekts. Die letzten Änderungen im Editor werden während des Kompilierungsprozesses nicht berücksichtigt.

Siehe auch die Beschreibung des Systemkommandos EXECUTE.

▶ Objekte mit Execute ausführen

1. Markieren Sie das Objekt im Library-Workspace oder List-View-Fenster.

Tipp:

In einem List-View-Fenster ist es möglich, mehr als ein Objekt zu markieren. Wenn Sie mehrere Objekte in einem List-View-Fenster markieren, werden sie eines nach dem anderen ausgeführt. Das erste Objekt im List-View-Fenster wird zuerst ausgeführt.

2. Wählen Sie aus dem Menü **Object** den Befehl **Execute**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Execute**.

Oder:

Wenn die Objekt-Symbolleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symbolleiste.



In der lokalen Umgebung erscheint ein Ausgabefenster, wenn ein Natural-Programm Ausgaben auf den Bildschirm schreibt. Weitere Informationen finden Sie unter *Output Window* in der *Operations*-Dokumentation.

In einer Remote-Entwicklungsumgebung erscheint entweder der Natural Web I/O Interface Client oder ein Terminal-Emulationsfenster anstatt eines Ausgabefensters. Weitere Informationen finden Sie unter *Natural Web I/O Interface Client* und *Terminal-Emulation* in der Dokumentation *Remote-Entwicklung mit SPoD*.

Editor-Fenster löschen

Wenn Sie ein Editor-Fenster löschen, wird es geschlossen, und ein neues leeres Editor-Fenster wird an derselben Position geöffnet. Aus der Titelleiste des Editor-Fensters geht hervor, dass ein Name und eine Library noch nicht definiert sind. Der Objekttyp bleibt derselbe.

Siehe auch die Beschreibung des Systemkommandos CLEAR.

▶ Editor-Fenster löschen

1. Aktivieren Sie das Editor-Fenster, das Sie löschen möchten.
2. Wählen Sie aus dem Menü **Object** den Befehl **Clear**.

Oder:

Wenn die Objekt-Symboleiste angezeigt wird, wählen Sie die folgende Schaltfläche in der Symboleiste.



Anmerkung:

Wenn Ihre letzten Änderungen noch nicht gespeichert wurden, erscheint ein Dialogfeld und Sie werden gefragt, ob Sie diese Änderungen verwerfen möchten. Wenn Sie die Änderungen nicht verwerfen möchten (Befehlsschaltfläche **No**), wird der Löschvorgang angehalten. Sie können dann das Objekt speichern und den Befehl **Clear** noch einmal absetzen.