

EDIT

Mit dem Systemkommando `EDIT` rufen Sie einen der Natural-Editoren auf, um die Source-Form eines Natural-Programmierobjekts zu editieren.

Es gibt drei verschiedene Formen der Kommandosyntax. Diese sind nachfolgend in getrennten Abschnitten dokumentiert.

- Syntax 1
- Syntax 2
- Syntax 3

Verwandtes Kommando: `READ`

Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural benutzen*.

Syntax 1

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

object-type

Folgende Objekttypen können editiert werden:

{	<u>C</u> CLASS	}
{	4	}
	<u>C</u> OPYCODE	
	<u>G</u> LOBAL	
	<u>H</u> ELPROUTINE	
	<u>L</u> OCAL	
{	<u>M</u> AP	}
	<u>P</u> ARAMETER	
	<u>P</u> ROGRAM	
{	<u>S</u> UBPROGRAM	}
{	N	}
	<u>S</u> UBROUTINE	
{	<u>T</u> EXT	}

Welcher Editor aufgerufen wird, hängt von dem zu editierenden Objekt ab:

- Ist das zu editierende Objekt eine Local Data Area, Global Data Area oder Parameter Data Area, wird der Data Area Editor aufgerufen.
- Ist das zu editierende Objekt eine Map oder eine Helproutine, die eine Map ist, wird der Map Editor aufgerufen.
- Ist das zu editierende Objekt eine Klasse, wird der Program Editor aufgerufen.
- Alle anderen Objekttypen (PROGRAM, SUBPROGRAM, SUBROUTINE, HELPROUTINE, COPYCODE, TEXT, DESCRIPTION) werden im Programm-Editor editiert.

Anmerkung:

Das Textobjekt DESCRIPTION ist eine Programmbeschreibung, die im Predict-Datendiktionär gespeichert ist und gepflegt wird; diese Objekte können nur editiert werden, wenn Predict installiert ist.

Die Objekttypen werden im *Leitfaden zur Programmierung* beschrieben. Die Editoren werden in der *Editors*-Dokumentation beschrieben.

Wenn Sie den Namen des Objekts, das Sie editieren möchten, angeben, brauchen Sie keinen Objekttyp anzugeben.

object-name

Mit dem Systemkommando EDIT geben Sie den Namen des Objekts an, das Sie editieren möchten. Die maximale Länge des Objektnamens beträgt 8 Zeichen.

Natural lädt dann das Objekt in den Arbeitsbereich des entsprechenden Editors, wo Sie es editieren können. Wenn Sie das Objekt anschließend unter demselben Namen speichern wollen, brauchen Sie bei einem anschließenden SAVE-, CATALOG- oder STOW-Kommando keinen Namen anzugeben.

Falls der Arbeitsbereich des Editors nicht leer ist, wird das angegebene Objekt in den entsprechenden Editor geladen.

Anmerkung:

Bei EDIT DESCRIPTION muß der *object-name* dem betreffenden Natural-Member-Namen in der Predict-Programmdefinition entsprechen.

Falls Sie keinen *object-name* angeben und es befindet sich kein Objekt im Arbeitsbereich, erhalten Sie den leeren Programm-Editor-Schirm, in dem Sie ein Programm erstellen können.

library-id

Befindet sich das Objekt in einer anderen Library als der, in der Sie gerade arbeiten, so müssen Sie die *library-id* der Library angeben, in der das zu editierende Objekt enthalten ist.

Anmerkung:

Eine *library-id*, die mit SYS beginnt, darf nicht angegeben werden (Ausnahme: SYSTEM).

Wenn Natural Security aktiv ist, können Sie keine *library-id* angeben, d.h. Sie können nur Objekte aus ihrer aktuellen Library editieren

Syntax 2

<code>EDIT [*] { * }</code> <code> [<i>object-type</i>] { <i>object-name</i> }</code>

Wenn Sie den Namen des Objekts, das Sie editieren möchten, nicht wissen, haben Sie mit dieser Form des EDIT-Kommandos die Möglichkeit, von einer Liste von Objekten das gewünschte Objekt auszuwählen.

EDIT *	Liefert Ihnen eine Liste aller Objekte, die in Ihrer aktuellen Library gespeichert sind.
EDIT <i>object-type</i> *	Liefert Ihnen eine Liste aller Objekte des angegebenen Typs aus Ihrer aktuellen Library.

Um ein Objekt aus einem bestimmten Bereich von Objekten auszuwählen, können Sie Stern-Notation (*) und Wildcard-Notation (?) für den *object-name* verwenden, und zwar in der gleichen Weise wie beim Systemkommando LIST beschrieben.

Syntax 3

<code>EDIT FUNCTION <i>subroutine-name</i></code>

Mit dem Kommando EDIT FUNCTION können Sie eine Subroutine unter ihrem internen Namen (also dem Namen, unter dem sie aufgerufen wird, nicht dem Namen, unter dem das Objekt gespeichert ist, in dem sie enthalten ist) zum Editieren aufrufen.

Der *subroutine-name* darf maximal 32 Zeichen lang sein.

Beispiel:

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

Angenommen, obige Subroutine ist unter dem Objektnamen CHCKSUB gespeichert, dann haben Sie folgende Möglichkeiten, um die Subroutine CHECK-PARAMETERS aufzurufen:

Entweder mit

<code>EDIT S CHKSUB</code>

oder mit

<code>EDIT F CHECK-PARAMETERS</code>
