

EXAMINE

Dieses Kapitel behandelt folgende Themen:

- Syntax 1 — EXAMINE
- Syntax 2 — EXAMINE TRANSLATE
- Syntax 3 — EXAMINE für Unicode-Grapheme
- Beispiele

Verwandte Statements: ADD | COMPRESS | COMPUTE | DIVIDE | MOVE | MOVE ALL | MULTIPLY | RESET | SEPARATE | SUBTRACT

Gehört zur Funktionsgruppe: *Arithmetische Funktionen und Datenzuweisungen*

Syntax 1 — EXAMINE

```

EXAMINE [DIRECTION-clause]
           [FULL [VALUE [OF]]] { operand1
                                   }
                                   { SUBSTRING (operand1,operand2,operand3) }
           [POSITION-clause]
           [FOR] [FULL [VALUE [OF]]] [PATTERN] operand4
           [DELIMITERS-option]
           {[DELETE-REPLACE-clause] [GIVING-clause]}

```

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Syntax-Beschreibung — Syntax 1

Das Statement EXAMINE dient dazu, den Inhalt eines alphanumerischen oder binären Feldes (oder eines Bereiches von Feldern innerhalb eines Arrays) nach einer bestimmten Zeichenkette abzusuchen und um

- zu zählen, wie oft eine bestimmte Zeichenkette vorkommt;
- die Byte-Position zurückzugeben, an der eine gesuchte Zeichenkette zuerst erscheint;
- die signifikante Länge des Inhalts eines Feldes zurückzugeben, d.h. die Feldlänge ohne nachfolgende Leerzeichen;
- die Ausprägungsnummer (Indices) eines Array-Feldes zurückzugeben, wo eine Zeichenkette zuerst gefunden wurde;

- eine Zeichenkette auszutauschen;
- eine Zeichenkette zu löschen.

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition				
<i>operand1</i>	C*	S	A			A	U											ja	nein
<i>operand2</i>	C	S						N	P	I	B*							ja	nein
<i>operand3</i>	C	S						N	P	I	B*							ja	nein
<i>operand4</i>	C	S	A			A	U				B							ja	nein

* *operand1* darf nur eine Konstante sein, wenn Sie die GIVING-Klausel verwenden, aber nicht, wenn Sie die DELETE-REPLACE-Klausel verwenden.

* Format B von *operand2* und *operand3* kann nur mit einer Länge von kleiner gleich 4 verwendet werden.

Syntax-Element-Beschreibung:

<i>DIRECTION-clause</i>	<p>Suchrichtung:</p> <p>Mit dieser Klausel legen Sie die Suchrichtung fest. Weitere Informationen siehe <i>DIRECTION-Klausel</i> weiter unten.</p>
<i>operand1</i>	<p>Zu untersuchendes Feld:</p> <p><i>operand1</i> ist das Feld, dessen Inhalt untersucht werden soll.</p> <p>Ist <i>operand1</i> eine dynamische Variable, kann deren Länge über eine REPLACE-Operation auf einen höheren oder niedrigeren Wert gesetzt werden; durch eine DELETE-Operation kann deren Länge auf "0" gesetzt werden. Die aktuelle Länge einer dynamischen Variablen kann über die Systemvariable *LENGTH ermittelt werden.</p> <p>Sie finden allgemeine Informationen über dynamische Variablen im Abschnitt <i>Große und dynamische Variablen/Felder</i>.</p>
<i>POSITION-clause</i>	<p>POSITION-Klausel:</p> <p>Mit dieser Klausel können Sie für die Untersuchung eine Start- und Endeposition innerhalb von <i>operand1</i> angeben (oder den Substring von <i>operand1</i>). Weitere Informationen siehe <i>POSITION-Klausel</i> weiter unten.</p>
<i>operand4</i>	<p>Suchwert:</p> <p><i>operand4</i> ist der für die Untersuchung zu verwendende Wert.</p> <p>Weitere Informationen zu <i>operand4</i> und <i>operand6</i> siehe <i>operand6</i>, der in der unten beschriebenen <i>DELETE REPLACE-Klausel</i> benutzt wird.</p>

FULL	<p>FULL-Option:</p> <p>Wenn Sie für einen Operanden FULL angeben, so wird der gesamte Wert, einschließlich nachfolgender Leerstellen, verarbeitet; ohne FULL werden dem Wert nachfolgende Leerstellen bei der Verarbeitung ignoriert.</p>
SUBSTRING	<p>SUBSTRING-Option:</p> <p>Normalerweise wird der ganze Inhalt des Feldes untersucht, und zwar vom Anfang des Feldes bis zum Ende bzw. bis zum letzten signifikanten Zeichen.</p> <p>Die Option SUBSTRING ermöglicht es Ihnen, nur einen bestimmten Teil des Feldes zu untersuchen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand1</i>) zunächst die erste Stelle (<i>operand2</i>) und dann die Länge (<i>operand3</i>) des Feldteils, der untersucht werden soll, an.</p> <p>Um zum Beispiel die 5. bis einschließlich 12. Stelle eines Feldes #A zu untersuchen, geben Sie folgendes an:</p> <pre>EXAMINE SUBSTRING(#A , 5 , 8) .</pre> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Wenn Sie <i>operand2</i> weglassen, wird ab Anfang des Feldes untersucht. 2. Wenn Sie <i>operand3</i> weglassen, wird ab der angegebenen Stelle (<i>operand2</i>) bis zum Ende des Feldes untersucht. 3. Wenn SUBSTRING in Verbindung mit einer dynamischen Variable benutzt wird, verhält sich das Feld wie eine Variable fester Länge, d.h. die Länge (*LENGTH) ändert sich nicht als Ergebnis der EXAMINE-Operation, ungeachtet der Tatsache, ob eine DELETE- oder REPLACE-Operation ausgeführt wurde oder nicht.

PATTERN	<p>PATTERN-Option:</p> <p>Wenn Sie das Feld nach einem Wert absuchen möchten, der Variablen enthält, d.h. Platzhalter für Stellen, die bei der Suche nicht berücksichtigt werden sollen, verwenden Sie die Option PATTERN. <i>operand4</i> kann dann die folgenden Platzhalter für nicht zu untersuchende Stellen enthalten:</p> <ul style="list-style-type: none"> ● Ein Punkt (.), Fragezeichen (?) oder Unterstrich (_) steht für eine einzelne Stelle, die nicht untersucht werden soll. ● Ein Stern (*) oder Prozentzeichen (%) steht für eine beliebige Anzahl von Stellen, die nicht untersucht werden sollen. <p>Beispiel:</p> <p>Mit PATTERN 'NAT*AL' könnten Sie ein Feld nach jedem Wert, in dem NAT und AL vorkommt, absuchen, ganz gleich, welche und wieviele andere Zeichen zwischen NAT und AL stehen (dies würde z.B. auf die Werte NATURAL und NATIONAL zutreffen, aber auch auf NATAL).</p>
DELIMITERS-option	<p>DELIMITERS-Option:</p> <p>Diese Option wird zum Suchen eines Wertes benutzt, der Delimiter darstellt. Einzelheiten, siehe <i>DELIMITERS-Option</i> weiter unten.</p>
DELETE-REPLACE-clause	<p>DELETE REPLACE-Klausel:</p> <p>Die DELETE-Option dieser Klausel wird zum Löschen jedes Suchwertes (<i>operand4</i>) benutzt, der in <i>operand1</i> gefunden wird.</p> <p>Die REPLACE-Option wird zum Austauschen jedes in <i>operand1</i> gefundenen Suchwertes (<i>operand4</i>) durch den in <i>operand6</i> angegebenen Wert benutzt.</p> <p>Siehe <i>DELETE REPLACE-Klausel</i> weiter unten.</p>
GIVING-clause	<p>GIVING-Klausel:</p> <p>Siehe <i>GIVING-Klausel</i> weiter unten.</p>

DIRECTION-Klausel

Diese Klausel bestimmt die Suchrichtung.

<p>DIRECTION { FORWARD BACKWARD } <i>operand8</i></p>
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition		
<i>operand8</i>	C	S				A1										ja	nein

Syntax-Element-Beschreibung:

FORWARD	Wenn Sie FORWARD angeben, wird der Feldinhalt von links nach rechts untersucht.
BACKWARD	Wenn Sie BACKWARD angeben, wird der Feldinhalt von rechts nach links untersucht.
<i>operand8</i>	Wenn Sie <i>operand8</i> angeben, wird die Suchrichtung durch den Inhalt von <i>operand8</i> bestimmt. <i>operand8</i> muss mit Format/Länge A1 definiert werden. Wenn <i>operand8</i> ein "F" enthält, dann ist die Suchrichtung FORWARD; wenn <i>operand8</i> ein "B" enthält, dann ist die Suchrichtung BACKWARD. Alle anderen Werte sind ungültig und werden zurückgewiesen: wenn <i>operand8</i> eine Konstante ist, wird der Wert zur Kompilierzeit zurückgewiesen; wenn <i>operand8</i> eine Variable ist, wird der Wert zur Laufzeit zurückgewiesen.

Anmerkung:

Wenn die DIRECTION-Klausel nicht angegeben ist, wird die Standardrichtung FORWARD benutzt.

POSITION-Klausel

Mit dieser Klausel können Sie für die Untersuchung eine Start- und Endeposition innerhalb von *operand1* angeben (oder den Substring von *operand1*).

[[STARTING] FROM [POSITION] <i>operand9</i>] [{ ENDING AT } [POSITION] <i>operand10</i>] [{ THRU }]
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition		
<i>operand9</i>	C	S				N	P	I								ja	nein
<i>operand10</i>	C	S				N	P	I								ja	nein

Syntax-Element-Beschreibung:

FROM <i>operand9</i>	<i>operand9</i> definiert die Startposition für die Untersuchung.
ENDING AT / THROUGH <i>operand10</i>	<i>operand10</i> definiert die Endeposition für die Untersuchung.

Startposition (*operand9*) und Endeposition (*operand10*) sind relativ zu *operand1* oder dem Substring von *operand1* und werden beide verarbeitet.

Die Suche beginnt an der Startposition und endet an der Endeposition.

Wenn Start- und/oder Endeposition nicht angegeben sind, gelten die Standardwerte für die Position. Der Wert wird durch die Suchrichtung bestimmt:

Richtung	Standardstartposition	Standardendeposition
FORWARD	1 (erstes Zeichen)	Länge von <i>operand1</i> (letztes Zeichen)
BACKWARD	Länge von <i>operand1</i> (letztes Zeichen)	1 (erstes Zeichen)

Mit dieser Lösung ist EXAMINE BACKWARD . . . identisch mit EXAMINE BACKWARD . . . FROM *LENGTH(. . .) THRU 1 und funktioniert wie erwartet.

Anmerkung:

Eine Suche wird nicht durchgeführt wenn die Suchrichtung FORWARD ist und die Startposition größer als die Endeposition ist, oder wenn die Suchrichtung BACKWARD ist und die Startposition kleiner als die Endeposition ist.

DELIMITERS-Option

{ ABSOLUTE [WITH DELIMITERS] [WITH DELIMITERS] <i>operand5</i> }
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate				Referenzierung erlaubt	Dynam. Definition
<i>operand5</i>	C	S			A			B	ja	nein

Syntax-Element-Beschreibung:

ABSOLUTE	<p>Absolute Suche:</p> <p>Standardmäßig gilt die Option ABSOLUTE; d.h. die zu suchende Zeichenkette wird auch gefunden, wenn sie von anderen Zeichen umgeben und Teil einer längeren Zeichenkette ist.</p>
WITH DELIMITERS	<p>Suchwert mit beliebigen Begrenzungszeichen:</p> <p>Mit WITH DELIMITERS wird ein Wert gesucht, dem je ein Leerzeichen oder irgendein anderes Zeichen, das weder ein Buchstabe noch eine Ziffer ist, vor- und nachgestellt ist.</p>
WITH DELIMITERS <i>operand5</i>	<p>Suchwert mit bestimmten Begrenzungszeichen:</p> <p>Mit WITH DELIMITERS <i>operand5</i> wird ein Wert gesucht, der von dem/den in <i>operand5</i> angegebenen Zeichen eingegrenzt ist.</p>

DELETE REPLACE-Klausel

[AND] { DELETE [FIRST] REPLACE [FIRST] [WITH] [FULL [VALUE [OF]]] <i>operand6</i> }
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition				
<i>operand6</i>	C	S	A			A	U						B					ja	nein

Syntax-Element-Beschreibung:

DELETE	Lösch-Option: Dient zum Löschen der ersten (oder aller) Ausprägung(en) des Suchwertes (<i>operand4</i>) im Inhalt von <i>operand1</i> .
REPLACE	Austausch-Option: Wird benutzt zum Austauschen der ersten (oder aller) Ausprägung(en) des Suchwertes (<i>operand4</i>) in <i>operand1</i> durch den in <i>operand6</i> angegebenen Austauschwert.
FIRST	Löschen/Austauschen des ersten identischen Werts: Wenn Sie das Schlüsselwort FIRST angeben, wird nur der erste identische Wert gelöscht/ausgetauscht.

Anmerkungen:

1. Wenn die REPLACE-Operation zur Generierung von mehr Zeichen führt als in *operand1* passen, erhalten Sie eine Fehlermeldung.
2. Wenn *operand1* eine dynamische Variable ist, kann eine REPLACE-Operation dazu führen, dass seine Länge vergrößert oder verkleinert wird; eine DELETE-Operation kann dazu führen, dass seine Länge auf Null (0) gesetzt wird. Die aktuelle Länge einer dynamischen Variablen kann mittels der Systemvariable *LENGTH ermittelt werden. Allgemeine Informationen zu dynamischen Variablen siehe *Dynamische Variablen benutzen*.
3. Falls ein Laufzeitfehler auftritt, bleibt das geprüfte Feld unverändert.

Suchen und Ersetzen mit mehreren Werten

Der Suchwert (*operand4*) und der Ersetzungswert (*operand6*) können auch als Array-Felder definiert werden. Dadurch ist es mit nur einem EXAMINE-Statement möglich, mehrere unterschiedliche Muster in dem geprüften Feld (*operand1*) zu ersetzen. Der Such- und der Ersetzungsoperand brauchen nicht die gleiche Anzahl an Ausprägungen zu haben. Es muss lediglich die Übertragungskompatibilität zwischen diesen Feldern gewährleistet sein, d.h. *operand4 := operand6* muss eine gültige Operation sein; siehe auch *Zuweisungen bei Arrays im Leitfaden zur Programmierung*.

Die Operationslogik für die Suche mit mehreren Werten arbeitet wie folgt:

- Das zu prüfende Feld (*operand6*) wird nur einmal durchlaufen, entweder von links nach rechts für Richtung FORWARD oder von rechts nach links für Richtung BACKWARD.
- Die Werte im Such-Array (*operand4*) werden, beginnend mit der ersten Position, auf Übereinstimmung geprüft, und zwar einer nach dem anderen, wobei mit der Array-Ausprägung mit dem niedrigsten Index begonnen wird.
- Wird kein Suchwert gefunden, wird der Vergleich auf der nächsten Feldposition fortgesetzt.
- Wenn eines der gesuchten Muster in einem geprüften Feld (*operand1*) gefunden wird, dann wird es durch den Wert des Ersetzungs-Arrays (*operand6*) ersetzt, das das übereinstimmende Muster in *operand4* belegt, wenn eine Operation *operand4:=operand6* ausgeführt würde.
- Nachdem die Ersetzung eines Musters erfolgt ist, wird der Vergleichsvorgang unmittelbar nach dem eingefügten Wert mit der ersten Ausprägung für das Such-Array fortgesetzt. Das bedeutet, dass ein schon einmal ersetztes Muster übersprungen wird und kein zweites Mal mehr ersetzt werden kann.

Beispiel 1:

Dieses Beispiel zeigt die Ersetzung des Kleiner-als-Zeichens (<), des Größer-als-Zeichens (>) und des Zeichens für das Kaufmännische Und (&) durch die entsprechenden HTML-Zeichen '<', '>' und '&'.

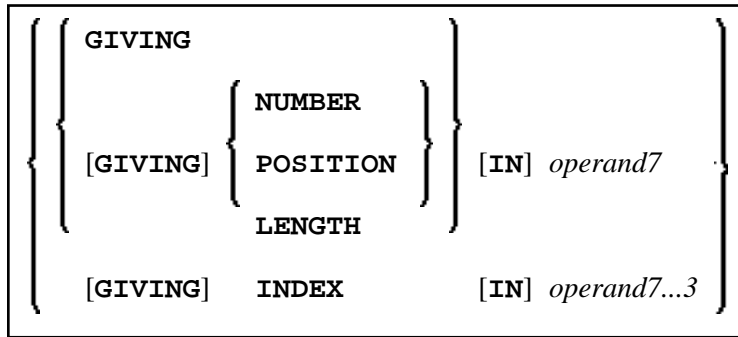
```
DEFINE DATA LOCAL
1 #HTML (A/1:3) DYNAMIC INIT <'&lt;', '&gt;', '&amp;'>
1 #TAB (A/1:3) DYNAMIC INIT <'<', '>', '&'>
1 #DOC(A) DYNAMIC /* document to be replaced
END-DEFINE
#DOC := 'a&lt;&lt;b&amp;b&gt;c&gt;'
WRITE #DOC (AL=30) 'before'
/* Replace #DOC using #HTML to #TAB (n:1 replacement)
EXAMINE #DOC FOR #HTML(*) REPLACE #TAB(*)
/* '&lt;' is replaced by '<' (4:1 replacement)
/* '&gt;' is replaced by '>' (4:1 replacement)
/* '&amp;' is replaced by '&' (5:1 replacement)
WRITE #DOC (AL=30) 'after'
END
```

Beispiel 2:

Dieses Beispiel zeigt die Ersetzung des Musters 'AA', 'Aa' und 'aA' durch '++', des Musters 'BB', 'Bb' und 'bB' durch '--' und des Musters 'CC', 'Cc' und 'cC' durch '**'.

```
DEFINE DATA LOCAL
1 #SV (A2/1:3,1:3) INIT (1,V) <'AA', 'BB', 'CC'>
(3,V) <'Aa', 'Bb', 'Cc'>
(2,V) <'aA', 'bB', 'cC'>
1 #RV (A2/1:3) INIT <'++', '--', '**'>
1 #STRING (A20) INIT <'AAABbbbbBCCCcccCaaaA'>
END-DEFINE
DISPLAY #STRING /* shows 'AAABbbbbBCCCcccCaaaA'
EXAMINE #STRING FOR #SV(*,*)
AND REPLACE WITH #RV(*)
DISPLAY #STRING /* shows '++A--bb--***c**aaa+'
END
```


GIVING-Klausel



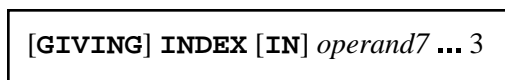
Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
operand7	S	N P I	ja	ja

Syntax-Element-Beschreibung:

GIVING	Wenn nur das Schlüsselwort GIVING angegeben wird, entspricht dies GIVING NUMBER (Voreinstellung).
NUMBER	Wird benutzt, um die Zahl zu erhalten, wie oft der zu suchende Wert (<i>operand4</i>) in dem Feld (<i>operand1</i>) gefunden wird, dessen Inhalt überprüft werden soll.
POSITION	Mit GIVING POSITION erhalten Sie die Byte-Position, die der erste gefundene Wert (<i>operand4</i>) innerhalb von <i>operand1</i> (bzw. des Substrings von <i>operand1</i>) innehat.
LENGTH	Mit GIVING LENGTH erhalten Sie die Länge von <i>operand1</i> (bzw. des Substrings von <i>operand1</i>), nachdem alle DELETE- bzw. REPLACE-Operationen abgeschlossen sind. Nachfolgende Leerzeichen werden ignoriert.
<i>operand7</i>	Die Anzahl der Ausprägungen des Suchwertes. Wenn auch die Option REPLACE FIRST oder DELETE FIRST benutzt wird, ist die Zahl nicht größer als 1.
INDEX <i>operand7...3</i>	Siehe <i>GIVING INDEX-Option</i> weiter unten.

GIVING INDEX-Option



Diese Option steht nur zur Verfügung, wenn das zu überprüfende zugrundeliegende Feld ein Array-Feld ist.

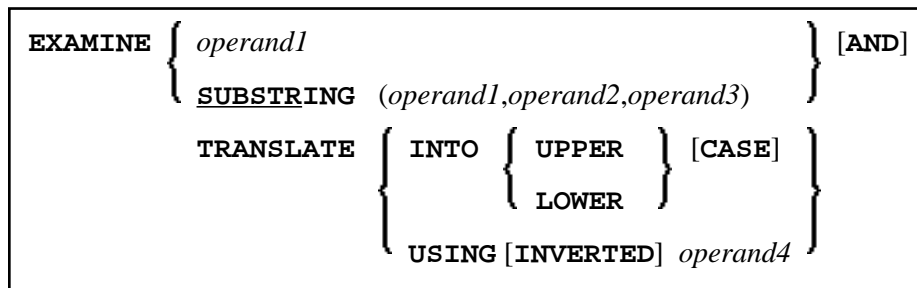
Syntax-Element-Beschreibung:

INDEX	Mit GIVING INDEX erhalten Sie die Nummer der Ausprägung (Index) von <i>operand1</i> , in der der erste gefundene Wert (<i>operand4</i>) enthalten ist.
operand7...3	<i>operand7</i> muss genauso oft angegeben werden wie <i>operand1</i> Dimensionen hat (höchstens dreimal). <i>operand7</i> enthält Null, wenn der gesuchte Wert in keiner der Ausprägungen enthalten ist.

Anmerkung:

Falls der Indexbereich von *operand1* die Ausprägung Null enthält (z.B. 0 : 5), ist der Wert 0 in *operand7* zweideutig. In diesem Falle sollte eine zusätzliche GIVING NUMBER-Klausel verwendet werden, um festzustellen, ob der gesuchte Wert tatsächlich vorkommt oder nicht.

Syntax 2 — EXAMINE TRANSLATE



Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Syntax-Beschreibung — Syntax 2

Das Statement EXAMINE TRANSLATE dient dazu, die in einem Feld enthaltenen Zeichen in Groß- oder Kleinschreibung oder in andere Zeichen umzusetzen.

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate								Referenzierung erlaubt	Dynam. Definition			
<i>operand1</i>		S	A			A					B				ja	nein
<i>operand2</i>	C	S				N	P	I			B*				ja	nein
<i>operand3</i>	C	S				N	P	I			B*				ja	nein
<i>operand4</i>		S	A			A					B				ja	nein

*Format B von *operand2* und *operand3* kann nur mit einer Länge von kleiner gleich 4 verwendet werden.

Syntax-Element-Beschreibung:

EXAMINE <i>operand1</i>	Umsetzung des kompletten Feldinhalts: <i>operand1</i> ist das Feld, dessen Inhalt umgesetzt werden soll.
EXAMINE SUBSTRING <i>operand1 operand2 operand3</i>	Umsetzung von Teilen des Feldinhalts: Normalerweise wird der Inhalt des gesamten Feldes umgesetzt. Die Option SUBSTRING ermöglicht es Ihnen, nur einen bestimmten Teil des Feldes umzusetzen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand1</i>) zunächst die erste Stelle (<i>operand2</i>) und dann die Länge (<i>operand3</i>) des Feldteils, der umgesetzt werden soll, an. Um zum Beispiel die 5. bis einschließlich 12. Stelle eines Feldes #A umzusetzen, geben Sie folgendes an: EXAMINE SUBSTRING(#A,5,8) AND TRANSLATE ... Anmerkung: Wenn Sie <i>operand2</i> weglassen, wird ab Anfang des Feldes umgesetzt. Wenn Sie <i>operand3</i> weglassen, wird ab der ersten Stelle bis zum Ende des Feldes umgesetzt.
TRANSLATE INTO UPPER CASE	Umsetzung in Großbuchstaben: Der Inhalt von <i>operand1</i> wird in Großbuchstaben umgesetzt.
TRANSLATE INTO LOWER CASE	Umsetzung in Kleinbuchstaben: Der Inhalt von <i>operand1</i> wird in Kleinbuchstaben umgesetzt.
TRANSLATE USING <i>operand4</i>	Zu benutzende Umsetzungstabelle: <i>operand4</i> ist die Umsetzungstabelle, die für die Zeichenumsetzung verwendet werden soll. Die Tabelle muss Format/Länge A2 oder B2 haben. Anmerkung: Falls für ein umzusetzendes Zeichen in der Umsetzungstabelle mehr als eine Umsetzung definiert ist, gilt die jeweils letzte Umsetzung.
INVERTED	Wenn Sie das Schlüsselwort INVERTED angeben, wird die Umsetzungstabelle (<i>operand4</i>) in umgekehrter Richtung verwendet, d.h. die Umsetzungsrichtung wird umgekehrt.

Syntax 3 — EXAMINE für Unicode-Grapheme

<pre> EXAMINE [FULL [VALUE [OF]]] { <i>operand1</i> SUBSTRING(<i>operand1</i>,<i>operand2</i>,<i>operand3</i>) } [POSITION-clause] [FOR] [CHARPOSITION <i>operand4</i>] [CHARLENGTH <i>operand5</i>] [GIVING] POSITION IN <i>operand6</i> [[GIVING] LENGTH IN <i>operand7</i>] </pre>
--

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Syntax-Beschreibung - Syntax 3

Unter einem Graphem versteht ein Benutzer normalerweise ein Zeichen. In den meisten Fällen ist eine UTF-16 Code-Einheit (= U-Formatzeichen) ein Graphem, allerdings kann ein Graphem auch aus mehreren Code-Einheiten bestehen. Beispiele sind: eine Folge von einem Basiszeichen gefolgt von Kombinationszeichen oder einem Ersatz-Paar. Weitere Informationen zu Graphemen und anderen Unicode-Begriffen entnehmen Sie dem Dokument *The Unicode Standard* unter <http://www.unicode.org/>.

Das Statement EXAMINE für U-Format-Operanden spricht im Allgemeinen Code-Einheiten an. Allerdings ist es bei CHARPOSITION- und CHARLENGTH-Klauseln möglich, die Startposition und Länge (als Code-Einheiten) einer Graphem-Sequenz zu erhalten. Die zurückgegeben Code-Einheitswerte können dann in anderen Statements/Klauseln benutzt werden, für die Code-Einheitsoperanden erforderlich sind (zum Beispiel in einer SUBSTRING-Klausel).

Weitere Informationen zur Syntax des EXAMINE-Statements, siehe auch *Unicode and Code Page Support* in *Natural Programming Language*, Abschnitt *Statements, EXAMINE*.

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition			
<i>operand1</i>	C	S	A			U											ja	nein
<i>operand2</i>	C	S					N	P	I	B*							ja	nein
<i>operand3</i>	C	S					N	P	I	B*							ja	nein
<i>operand4</i>	C	S	A				N	P	I								ja	nein
<i>operand5</i>	C	S	A				N	P	I								ja	nein
<i>operand6</i>	C	S					N	P	I								ja	nein
<i>operand7</i>	C	S					N	P	I								ja	nein

* Format B von *operand2* und *operand3* kann nur mit einer Länge von kleiner gleich 4 benutzt werden.

Syntax-Element-Beschreibung:

FULL	<p>FULL-Option:</p> <p>Wenn FULL für einen Operanden angegeben wird, wird der gesamte Wert, einschließlich der nachfolgenden Leerzeichen abgearbeitet. Wenn FULL nicht angegeben wird, werden nach folgende Leerzeichen im Operanden ignoriert.</p>
-------------	--

<p>SUBSTRING <i>operand1 operand2 operand3</i></p>	<p>SUBSTRING-Option:</p> <p>Normalerweise wird der ganze Inhalt des Feldes untersucht, und zwar vom Anfang des Feldes bis zum Ende bzw. bis zum letzten signifikanten Zeichen.</p> <p>Die Option SUBSTRING ermöglicht es Ihnen, nur einen bestimmten Teil des Feldes zu untersuchen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand1</i>) zunächst die erste Stelle (<i>operand2</i>) und dann die Länge (<i>operand3</i>) des Feldteils an, der untersucht werden soll. <i>operand2</i> und <i>operand3</i> werden als Code-Einheiten angegeben.</p> <p>Um zum Beispiel die 5. bis einschließlich 12. Stelle eines Feldes #A zu untersuchen, geben Sie folgendes an:</p> <pre>EXAMINE SUBSTRING (#A, 5, 8)</pre> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Wenn Sie <i>operand2</i> weglassen, wird ab Anfang des Feldes (Position 1) untersucht. 2. Wenn Sie <i>operand3</i> weglassen, wird ab der angegebenen Stelle bis zum Ende des Feldes untersucht. 3. Wenn SUBSTRING in Verbindung mit einer dynamischen Variable benutzt wird, verhält sich das Feld wie eine Variable fester Länge, d.h. die Länge (*LENGTH) ändert sich nicht als Ergebnis der EXAMINE-Operation, ungeachtet der Tatsache, ob eine DELETE- oder REPLACE-Operation ausgeführt wurde oder nicht.
<p>POSITION-clause</p>	<p>POSITION-Klausel:</p> <p>Bereichswerte für FROM und THRU werden in Form von Code-Einheiten angegeben. Weitere Informationen siehe <i>POSITION-Klausel</i> unter Syntax 1.</p>
<p>CHARPOSITION <i>operand4</i></p>	<p>CHARPOSITION-Klausel:</p> <p><i>operand4</i> legt die Startposition (als Unicode-Grapheme) der Graphem-Sequenz fest. Die entsprechende Position wird in <i>operand6</i> in Form von Code-Einheiten zurückgegeben. Diese Klausel kann weggelassen werden, wenn die CHARLENGTH-Klausel angegeben wird; in diesem Fall ist die Startposition 1.</p>

CHARLENGTH <i>operand5</i>	CHARLENGTH-Klausel: <i>operand5</i> legt die Länge der Graphem-Sequenz (als Unicode-Grapheme) fest. Die Länge der Graphem-Sequenz wird in Form von Code-Einheiten in <i>operand7</i> zurückgegeben. Diese Klausel kann weggelassen werden, wenn die CHARPOSITION-Klausel angegeben wird; in diesem Fall wird die Länge von der Startposition bis zum Ende der Zeichenkette zurückgegeben.
GIVING POSITION IN <i>operand6</i>	GIVING POSITION-Klausel: <i>operand6</i> enthält (als Code-Einheiten) die Startposition der von <i>operand4</i> und <i>operand5</i> definierten Graphem-Sequenz. Wenn <i>operand1</i> weniger als <i>operand4</i> Grapheme hat, wird Null (0) zurückgegeben. Diese Klausel kann weggelassen werden, wenn die GIVING LENGTH-Klausel angegeben wird.
GIVING LENGTH IN <i>operand7</i>	GIVING LENGTH-Klausel: <i>operand7</i> enthält (als Code-Einheiten) die Länge der mit <i>operand4</i> und <i>operand5</i> definierten Graphem-Sequenz. Wenn <i>operand1</i> weniger als <i>operand4+operand5</i> Grapheme hat, wird 0 zurückgegeben. Diese Klausel kann weggelassen werden, wenn die GIVING POSITION-Klausel angegeben wird.

Anmerkungen:

1. Es muss entweder die CHARPOSITION- oder die CHARLENGTH-Klausel oder beide angegeben werden.
2. Es muss entweder die GIVING POSITION- oder die GIVING LENGTH-Klausel oder beide angegeben werden.

Beispiele

- Beispiel 1 — EXAMINE
- Beispiel 2 — EXAMINE SUBSTRING, PATTERN, TRANSLATE
- Beispiel 3 — EXAMINE TRANSLATE
- Beispiel 4 — EXAMINE für Unicode-Grapheme

Beispiel 1 — EXAMINE

```
** Example 'EXMEX1': EXAMINE
*****
DEFINE DATA LOCAL
1 #TEXT      (A40)
1 #A         (A1)
1 #START     (N2)
1 #NMB1      (N2)
1 #NMB2      (N2)
1 #NMB3      (N2)
```

```

1 #NMBEX2 (N2)
1 #NMBEX3 (N2)
1 #NMBEX4 (N2)
1 #POSEX5 (N2)
1 #LGHEX6 (N2)
1 #NMBEX7 (N2)
1 #NMBEX8 (N2)
END-DEFINE
*
WRITE 'EXAMPLE 1 (GIVING NUMBER, WITH DELIMITER) '
MOVE 'ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C- ' TO #TEXT
ASSIGN #A = 'A'
EXAMINE #TEXT FOR #A GIVING NUMBER #NMB1
EXAMINE #TEXT FOR #A WITH DELIMITER GIVING NUMBER #NMB2
EXAMINE #TEXT FOR #A WITH DELIMITER '.' GIVING NUMBER #NMB3
WRITE NOTITLE '=' #NMB1 '=' #NMB2 '=' #NMB3
*
WRITE / 'EXAMPLE 2 (WITH DELIMITER, REPLACE, GIVING NUMBER) '
WRITE '=' #TEXT
EXAMINE #TEXT FOR #A WITH DELIMITER '-' REPLACE WITH '*'
GIVING NUMBER #NMBEX2
WRITE '=' #TEXT '=' #NMBEX2
*
WRITE / 'EXAMPLE 3 (REPLACE, GIVING NUMBER) '
WRITE '=' #TEXT
EXAMINE #TEXT ' ' REPLACE WITH '+' GIVING NUMBER #NMBEX3
WRITE '=' #TEXT '=' #NMBEX3
*
WRITE / 'EXAMPLE 4 (FULL, REPLACE, GIVING NUMBER) '
WRITE '=' #TEXT
EXAMINE FULL #TEXT ' ' REPLACE WITH '+' GIVING NUMBER #NMBEX4
WRITE '=' #TEXT '=' #NMBEX4
*
WRITE / 'EXAMPLE 5 (DELETE, GIVING POSITION) '
WRITE '=' #TEXT
EXAMINE #TEXT '+' DELETE GIVING POSITION #POSEX5
WRITE '=' #TEXT '=' #POSEX5
*
WRITE / 'EXAMPLE 6 (DELETE, GIVING LENGTH) '
WRITE '=' #TEXT
EXAMINE #TEXT FOR 'A' DELETE GIVING LENGTH #LGHEX6
WRITE '=' #TEXT '=' #LGHEX6
*
*
NEWPAGE
*
MOVE 'ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C- ' TO #TEXT
*
ASSIGN #A = 'A B C'
ASSIGN #START = 6
*
WRITE / 'EXAMPLE 7 (SUBSTRING, GIVING NUMBER) '
WRITE '=' #TEXT
EXAMINE SUBSTRING(#TEXT,#START,9) FOR #A GIVING NUMBER #NMBEX7
WRITE '=' #TEXT '=' #NMBEX7
*
WRITE / 'EXAMPLE 8 (PATTERN, GIVING NUMBER) '
WRITE '=' #TEXT
EXAMINE #TEXT FOR PATTERN '-A-' GIVING NUMBER #NMBEX8
WRITE '=' #TEXT '=' #NMBEX8
*
END

```

Ausgabe des Programms EXMEX1:

```

EXAMPLE 1 (GIVING NUMBER, WITH DELIMITER)
#NMB1:  4 #NMB2:  3 #NMB3:  1

EXAMPLE 2 (WITH DELIMITER, REPLACE, GIVING NUMBER)
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-
#TEXT: ABC  A B C  .A.  .B.  .C.  -*  -B-  #NMBEX2:  1

EXAMPLE 3 (REPLACE, GIVING NUMBER)
#TEXT: ABC  A B C  .A.  .B.  .C.  -*  -B-
#TEXT: ABC+++A+B+C+++A.++B.++C.++++-*---B-  #NMBEX3:  18

EXAMPLE 4 (FULL, REPLACE, GIVING NUMBER)
#TEXT: ABC+++A+B+C+++A.++B.++C.++++-*---B-
#TEXT: ABC+++A+B+C+++A.++B.++C.++++-*---B-+  #NMBEX4:  1

EXAMPLE 5 (DELETE, GIVING POSITION)
#TEXT: ABC+++A+B+C+++A.++B.++C.++++-*---B-+
#TEXT: ABCABC.A..B..C.-*--B-  #POSEX5:  4

EXAMPLE 6 (DELETE, GIVING LENGTH)
#TEXT: ABCABC.A..B..C.-*--B-
#TEXT: BCBC...B..C.-*--B-  #LGHEX6:  18

EXAMPLE 7 (SUBSTRING, GIVING NUMBER)
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-  #NMBEX7:  1

EXAMPLE 8 (PATTERN, GIVING NUMBER)
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-  #NMBEX8:  1

```

Beispiel 2 — EXAMINE SUBSTRING, PATTERN, TRANSLATE

```

** Example 'EXMEX2': EXAMINE TRANSLATE
*****
DEFINE DATA LOCAL
1 #TEXT (A50)
1 #TAB (A2/1:10)
1 #START (N2)
END-DEFINE
*
MOVE 'ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C-' TO #TEXT
*
MOVE 'AX' TO #TAB(1)
MOVE 'BY' TO #TAB(2)
MOVE 'CZ' TO #TAB(3)
*
*
WRITE 'EXAMPLE 1 (USING TRANSLATION TABLE)'
WRITE '=' #TEXT
EXAMINE #TEXT TRANSLATE USING #TAB(*)
WRITE NOTITLE '=' #TEXT
*
WRITE / 'EXAMPLE 2 (USING INVERTED TRANSLATION TABLE)'
WRITE '=' #TEXT
EXAMINE #TEXT TRANSLATE USING INVERTED #TAB(*)
WRITE NOTITLE '=' #TEXT
*
WRITE / 'EXAMPLE 3 (USING SUBSTRING, LOWER CASE)'
WRITE '=' #TEXT

```



```

ASSIGN #START = 13
EXAMINE SUBSTRING(#TEXT,#START,15) TRANSLATE INTO LOWER CASE
WRITE '=' #TEXT
END
    
```

Ausgabe des Programms EXMEX2:

```

EXAMPLE 1 (USING TRANSLATION TABLE)
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C-
#TEXT: XYZ  X Y Z  .X.  .Y.  .Z.  -X-  -Y-  -Z-

EXAMPLE 2 (USING INVERTED TRANSLATION TABLE)
#TEXT: XYZ  X Y Z  .X.  .Y.  .Z.  -X-  -Y-  -Z-
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C-

EXAMPLE 3 (USING SUBSTRING, LOWER CASE)
#TEXT: ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C-
#TEXT: ABC  A B C  .a.  .b.  .c.  -A-  -B-  -C-
    
```

Beispiel 3 — EXAMINE TRANSLATE

```

** Example 'EXMEX2': EXAMINE TRANSLATE
*****
DEFINE DATA LOCAL
1 #TEXT  (A50)
1 #TAB   (A2/1:10)
1 #START (N2)
END-DEFINE
*
MOVE 'ABC  A B C  .A.  .B.  .C.  -A-  -B-  -C-' TO #TEXT
*
MOVE 'AX' TO #TAB(1)
MOVE 'BY' TO #TAB(2)
MOVE 'CZ' TO #TAB(3)
*
*
WRITE 'EXAMPLE 1 (USING TRANSLATION TABLE)'
WRITE '=' #TEXT
EXAMINE #TEXT TRANSLATE USING #TAB(*)
WRITE NOTITLE '=' #TEXT
*
WRITE / 'EXAMPLE 2 (USING INVERTED TRANSLATION TABLE)'
WRITE '=' #TEXT
EXAMINE #TEXT TRANSLATE USING INVERTED #TAB(*)
WRITE NOTITLE '=' #TEXT
*
WRITE / 'EXAMPLE 3 (USING SUBSTRING, LOWER CASE)'
WRITE '=' #TEXT
ASSIGN #START = 13
EXAMINE SUBSTRING(#TEXT,#START,15) TRANSLATE INTO LOWER CASE
WRITE '=' #TEXT
END
    
```

Ausgabe des Programms EXMEX2:

EXAMPLE 1 (USING TRANSLATION TABLE)

```
#TEXT: ABC   A B C   .A. .B. .C.   -A- -B- -C-
#TEXT: XYZ   X Y Z   .X. .Y. .Z.   -X- -Y- -Z-
```

EXAMPLE 2 (USING INVERTED TRANSLATION TABLE)

```
#TEXT: XYZ   X Y Z   .X. .Y. .Z.   -X- -Y- -Z-
#TEXT: ABC   A B C   .A. .B. .C.   -A- -B- -C-
```

EXAMPLE 3 (USING SUBSTRING, LOWER CASE)

```
#TEXT: ABC   A B C   .A. .B. .C.   -A- -B- -C-
#TEXT: ABC   A B C   .a. .b. .c.   -A- -B- -C-
```

Beispiel 4 — EXAMINE für Unicode-Grapheme

Dieses Beispiel veranschaulicht die Analyse einer Unicode-Zeichenkette mit den Zeichen ä und ü. Beide Zeichen sind als Basiszeichen, gefolgt von einem Kombinationszeichen festgelegt: ä ist als U+0061, gefolgt von U+0308 kodiert, und ü ist als U+0075, gefolgt von U+0308 kodiert.

```
DEFINE DATA LOCAL
1 #U (U20)
1 #START (I2)
1 #POS (I2)
1 #LEN (I2)
END-DEFINE
#U := U'AB'-UH'00610308'-U'CD'-UH'00750308'-U'EF'
*
REPEAT
  #START := #START + 1
  EXAMINE #U FOR CHARPOSITION #START
                CHARLENGTH      1
                GIVING POSITION IN #POS
                LENGTH IN #LEN
*
  INPUT (AD=0) MARK POSITION #POS IN FIELD *#U
  '          UNICODE-STRING:' #U      (AD=MI)
// '          CHARACTER NO.:' #START  (EM=9)
/ ' STARTS AT BYTE POSITION:' #POS     (EM=9)
/ '          AND THE LENGTH IS:' #LEN  (EM=9)
WHILE #POS NE 0
END-REPEAT
END
```

Ausgabe:

Großrechner-Umgebungen:	Windows-, UNIX- und OpenVMS-Umgebungen (mit Natural Web I/O Interface):
<p>UNICODE-STRING: ABa?CDu?EF</p> <p>CHARACTER NO.: 1</p> <p>STARTS AT BYTE POSITION: 1</p> <p>AND THE LENGTH IS: 1</p>	<p>UNICODE-STRING: ABäCDüEF</p> <p>CHARACTER NO.: 1</p> <p>STARTS AT BYTE POSITION: 1</p> <p>AND THE LENGTH IS: 1</p>
<p>Drücken Sie die Eingabetaste um fortzufahren.</p>	
<p>UNICODE-STRING: ABa?CDu?EF</p> <p>CHARACTER NO.: 2</p> <p>STARTS AT BYTE POSITION: 2</p> <p>AND THE LENGTH IS: 1</p>	<p>UNICODE-STRING: ABäCDüEF</p> <p>CHARACTER NO.: 2</p> <p>STARTS AT BYTE POSITION: 2</p> <p>AND THE LENGTH IS: 1</p>
<p>Drücken Sie die Eingabetaste um fortzufahren.</p>	
<p>Beachten Sie, dass das Zeichen in Position 3 eine Kombinationszeichenfolge ist und zwei Code-Einheiten lang ist.</p>	
<p>UNICODE-STRING: ABa?CDu?E</p> <p>CHARACTER NO.: 3</p> <p>STARTS AT BYTE POSITION: 3</p> <p>AND THE LENGTH IS: 2</p>	<p>UNICODE-STRING: ABäCDüEF</p> <p>CHARACTER NO.: 3</p> <p>STARTS AT BYTE POSITION: 3</p> <p>AND THE LENGTH IS: 2</p>
<p>Und so weiter.</p>	<p>Und so weiter.</p>