

# MOVE

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

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

Gehört zur Funktionsgruppe: *Arithmetische Funktionen und Datenzuweisungen*

---

## Funktion

Das Statement MOVE dient dazu, den Wert eines Operanden in einen oder mehrere andere Operanden (Felder oder Arrays) zu übertragen.

Ein MOVE-Statement mit mehreren Zieloperanden ist mit den betreffenden einzelnen MOVE-Statements identisch:

```
MOVE #SOURCE TO #TARGET1 #TARGET2
```

ist identisch mit:

```
MOVE #SOURCE TO #TARGET1
MOVE #SOURCE TO #TARGET2
```

Beispiel:

```
DEFINE DATA LOCAL
1 #ARRAY(I4/1:3) INIT <3,0,9>
1 #INDEX(I4)
1 #RESULT(I4)
END-DEFINE
*
#INDEX := 1
MOVE #ARRAY(#INDEX) TO #INDEX      /* #INDEX is 3
                        #RESULT     /* #RESULT is 9
*
#INDEX := 2
MOVE #ARRAY(#INDEX) TO #INDEX      /* #INDEX is 0
                        #ARRAY(3)  /* returns run time error NAT1316
```

Ist *operand2* eine dynamische Variable, kann ihre Länge mit der MOVE-Operation geändert werden. Die aktuelle Länge einer dynamischen Variable kann mittels der Systemvariable \*LENGTH bestimmt werden. Allgemeine Informationen zu dynamischen Variablen siehe den Abschnitt *Dynamische und große Variablen benutzen* im Leitfaden zur Programmierung.

Hat *operand2* das Format C, kann *operand1* auch als (*parameter*) angegeben werden. Es gibt die folgenden gültigen Parameter:

<b>Parameter, die mit dem MOVE-Statement angegeben werden können:</b>		<b>Spezifikation:</b>
		<b>S = auf Statement-Ebene</b>
		<b>E = auf Element-Ebene</b>
AD	Attribute Definition	SE
CD	Color Definition	S

Weitere Informationen zur Datenübertragungs-Kompatibilität und den Regeln für die Datenübertragung finden Sie im Abschnitt *Datenübertragung im Leitfaden zur Programmierung*.

## Weitere Hinweise

Wird ein Datenbankfeld als Ergebnisfeld (*operand2*) verwendet, so ändert sich der Wert des Feldes durch die MOVE-Operation nur programmintern. Der in der Datenbank gespeicherte Feldwert wird davon nicht beeinflusst.

Eine Natural-Systemfunktion darf nur eingesetzt werden, wenn das MOVE-Statement in Verbindung mit einem AT BREAK-, AT END OF DATA- oder AT END OF PAGE-Statement verwendet wird.

Siehe auch Abschnitt *Arithmetische Operationen im Leitfaden zur Programmierung*.

### Anmerkung:

Wenn *operand1* eine Zeitvariable (Format T) ist, wird nur die Zeitkomponente des Variableninhalts übertragen, aber nicht die Datumskomponente (außer bei MOVE EDITED). Siehe *Syntax 4* und *Syntax 5*.

## Syntax-Beschreibung

Das MOVE-Statement bietet mehrere Syntax-Varianten:

- Syntax 1 — MOVE ROUNDED
- Syntax 2 - MOVE SUBSTRING
- Syntax 3 - MOVE BY NAME / POSITION
- Syntax 4 – MOVE EDITED (Editiermaske mit operand2 angegeben)
- Syntax 5 – MOVE EDITED (Editiermaske mit operand1 angegeben)
- Syntax 6 - MOVE LEFT / RIGHT JUSTIFIED
- Syntax 7 - MOVE NORMALIZED
- Syntax 8 - MOVE ENCODED

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

## Syntax 1 — MOVE ROUNDED

<b>MOVE</b> [ <b>ROUNDED</b> ] <i>operand1</i> [( <i>parameter</i> )] <b>TO</b> <i>operand2</i> ...
---

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition	
<i>operand1</i>	C	S	A	N	A	U	N	P	I	F	B	D	T	L	C	G	O	ja	nein
<i>operand2</i>		S	A	M	A	U	N	P	I	F	B	D	T	L	C	G	O	ja	ja

Syntax-Element-Beschreibung:

<b>MOVE ROUNDED</b>	<p><b>Rundung:</b></p> <p>Das Schlüsselwort ROUNDED bewirkt, dass der Wert von <i>operand2</i> auf- bzw. abgerundet wird.</p> <p>ROUNDED wird ignoriert, wenn <i>operand2</i> nicht numerisch ist.</p> <p>Wenn <i>operand2</i> das Format N oder P hat und mehr als einmal angegeben wird, wird ROUNDED bei Zieloperanden mit 7 Stellen hinter dem Dezimalpunkt (Komma) ignoriert.</p> <p>Siehe auch <i>Beispiel 1 – Verschiedene Beispiele für die Benutzung des MOVE-Statements</i>.</p>				
<p>(<i>parameter</i>)</p>	<p><b>Parameter:</b></p> <p>Als <i>parameter</i> können Sie die Option PM=I oder den Session-Parameter DF angeben:</p> <table border="1" data-bbox="448 751 1395 1528"> <tr> <td data-bbox="448 751 521 1528"> <b>PM=I</b> </td> <td data-bbox="521 751 1395 1528"> <p><b>Schreibrichtung:</b></p> <p>Zur Unterstützung von Sprachen, deren Schreibrichtung von rechts nach links verläuft, können Sie die Option PM=I angeben, um den Wert von <i>operand1</i> invers (d.h. von rechts nach links) in <i>operand2</i> zu übertragen.</p> <p>Zum Beispiel hätte als Ergebnis der folgenden Statements das Feld #B den Inhalt ZYX:</p> <pre>MOVE 'XYZ' TO #A MOVE #A (PM=I) TO #B</pre> <p>PM=I kann nur angegeben werden, wenn <i>operand2</i> alphanumerisches Format hat.</p> <p>Nachfolgende Leerzeichen in <i>operand1</i> werden entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt), dann wird der Wert umgedreht und anschließend in <i>operand2</i> übertragen. Falls <i>operand1</i> nicht alphanumerisches Format hat, wird der Wert in alphanumerisches Format umgesetzt, bevor er umgedreht wird.</p> <p>Zur Verwendung von PM=I zusammen mit MOVE LEFT/RIGHT JUSTIFIED siehe MOVE LEFT/RIGHT JUSTIFIED.</p> </td> </tr> <tr> <td data-bbox="448 1528 521 1770"> <b>DF</b> </td> <td data-bbox="521 1528 1395 1770"> <p><b>Datumsformat:</b></p> <p>Wenn <i>operand1</i> eine Datumsvariable und <i>operand2</i> ein alphanumerisches Feld ist, können Sie den Session-Parameter DF als <i>parameter</i> für diese Datumsvariable angeben.</p> <p>Der Session-Parameter DF ist in der <i>ParameterReferenz</i> beschrieben.</p> </td> </tr> </table>	<b>PM=I</b>	<p><b>Schreibrichtung:</b></p> <p>Zur Unterstützung von Sprachen, deren Schreibrichtung von rechts nach links verläuft, können Sie die Option PM=I angeben, um den Wert von <i>operand1</i> invers (d.h. von rechts nach links) in <i>operand2</i> zu übertragen.</p> <p>Zum Beispiel hätte als Ergebnis der folgenden Statements das Feld #B den Inhalt ZYX:</p> <pre>MOVE 'XYZ' TO #A MOVE #A (PM=I) TO #B</pre> <p>PM=I kann nur angegeben werden, wenn <i>operand2</i> alphanumerisches Format hat.</p> <p>Nachfolgende Leerzeichen in <i>operand1</i> werden entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt), dann wird der Wert umgedreht und anschließend in <i>operand2</i> übertragen. Falls <i>operand1</i> nicht alphanumerisches Format hat, wird der Wert in alphanumerisches Format umgesetzt, bevor er umgedreht wird.</p> <p>Zur Verwendung von PM=I zusammen mit MOVE LEFT/RIGHT JUSTIFIED siehe MOVE LEFT/RIGHT JUSTIFIED.</p>	<b>DF</b>	<p><b>Datumsformat:</b></p> <p>Wenn <i>operand1</i> eine Datumsvariable und <i>operand2</i> ein alphanumerisches Feld ist, können Sie den Session-Parameter DF als <i>parameter</i> für diese Datumsvariable angeben.</p> <p>Der Session-Parameter DF ist in der <i>ParameterReferenz</i> beschrieben.</p>
<b>PM=I</b>	<p><b>Schreibrichtung:</b></p> <p>Zur Unterstützung von Sprachen, deren Schreibrichtung von rechts nach links verläuft, können Sie die Option PM=I angeben, um den Wert von <i>operand1</i> invers (d.h. von rechts nach links) in <i>operand2</i> zu übertragen.</p> <p>Zum Beispiel hätte als Ergebnis der folgenden Statements das Feld #B den Inhalt ZYX:</p> <pre>MOVE 'XYZ' TO #A MOVE #A (PM=I) TO #B</pre> <p>PM=I kann nur angegeben werden, wenn <i>operand2</i> alphanumerisches Format hat.</p> <p>Nachfolgende Leerzeichen in <i>operand1</i> werden entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt), dann wird der Wert umgedreht und anschließend in <i>operand2</i> übertragen. Falls <i>operand1</i> nicht alphanumerisches Format hat, wird der Wert in alphanumerisches Format umgesetzt, bevor er umgedreht wird.</p> <p>Zur Verwendung von PM=I zusammen mit MOVE LEFT/RIGHT JUSTIFIED siehe MOVE LEFT/RIGHT JUSTIFIED.</p>				
<b>DF</b>	<p><b>Datumsformat:</b></p> <p>Wenn <i>operand1</i> eine Datumsvariable und <i>operand2</i> ein alphanumerisches Feld ist, können Sie den Session-Parameter DF als <i>parameter</i> für diese Datumsvariable angeben.</p> <p>Der Session-Parameter DF ist in der <i>ParameterReferenz</i> beschrieben.</p>				

## Syntax 2 - MOVE SUBSTRING

MOVE	{	<i>operand1</i>	}	[(parameter)]	TO	{	<i>operand2</i>	}	...
		<u>SUBSTRING</u> ( <i>operand1,operand3,operand4</i> )					<u>SUBSTRING</u> ( <i>operand2,operand5,operand6</i> )		

Operanden-Definitionstabelle:

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

\* Siehe Text.

Syntax-Element-Beschreibung:

<b>MOVE SUBSTRING</b>	<p><b>Teilweise Übertragung:</b></p> <p>Ohne SUBSTRING-Option wird der ganze Inhalt des Feldes übertragen.</p> <p>Die Option SUBSTRING ermöglicht es Ihnen, nur einen bestimmten Teil eines alphanumerischen oder binären Feldes zu übertragen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand1</i>) zunächst die erste Stelle (<i>operand3</i>) und dann die Länge (<i>operand4</i>) des Feldteils an, der übertragen werden soll.</p> <p>Wenn das zugrundeliegende Feldformat von <i>operand1</i>:</p> <ul style="list-style-type: none"> <li>• alphanumerisch (A) oder numerisch (N) ist, gelten die mit <i>operand3</i> oder <i>operand4</i> angegebenen Werte als Byte-Zahlen.</li> <li>• Unicode (U) ist, gelten die mit <i>operand3</i> oder <i>operand4</i> angegebenen Werte als Zahl der Unicode-Codeeinheiten, d.h. als Doppelbytes.</li> </ul> <p>Um zum Beispiel die 5. bis einschließlich 12. Stelle eines Feldes #A in ein Feld #B zu übertragen, würden Sie folgendes angeben:</p> <pre>MOVE SUBSTRING( #A, 5, 8 ) TO #B</pre> <p>Ist <i>operand1</i> eine dynamische Variable, muss der angegebene und zu übertragene Feldteil im Bereich seiner aktuellen Länge sein; sonst tritt ein Laufzeit-Fehler auf.</p> <p>Sie können einen Wert eines alphanumerischen, binären oder numerischen Feldes auch in einen bestimmten Teil des Zielfeldes übertragen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand2</i>) zunächst die erste Stelle (<i>operand5</i>) und dann die Länge (<i>operand6</i>) des Feldteils an, in den der Wert übertragen werden soll.</p> <p>Wenn das zugrundeliegende Feldformat von <i>operand2</i>:</p> <ul style="list-style-type: none"> <li>• alphanumerisch (A) oder numerisch (N) ist, gelten die mit <i>operand5</i> oder <i>operand6</i> angegebenen Werte als Byte-Zahlen.</li> <li>• Unicode (U) ist, gelten die mit <i>operand5</i> oder <i>operand6</i> angegebenen Werte als Zahl der Unicode-Codeeinheiten, d.h. als Doppelbytes.</li> </ul> <p>Um zum Beispiel den Wert eines Feldes #A in die 3. bis einschließlich 6. Stelle eines Feldes #B zu übertragen, würden Sie folgendes angeben:</p> <pre>MOVE #A TO SUBSTRING( #B, 3, 4 )</pre> <p>Wenn <i>operand2</i> eine dynamische Variable ist, darf die erste Stelle (<i>operand5</i>) nicht größer sein als die aktuelle Länge der Variable plus 1; eine höhere erste Stelle würde einen Laufzeit-Fehler zur Folge haben, weil dies zu einer nicht definierten Lücke im Inhalt von <i>operand2</i> führen würde.</p> <p>Wenn <i>operand3</i> bzw. <i>operand5</i> oder <i>operand4</i> bzw. <i>operand6</i> eine binäre Variable ist, kann sie nur mit einer Länge kleiner gleich 4 benutzt werden</p> <p>Wenn Sie <i>operand3</i> bzw. <i>operand5</i> weglassen, wird ab Anfang des Feldes übertragen. Wenn Sie <i>operand4</i> bzw. <i>operand6</i> weglassen, wird ab der angegebenen Stelle (<i>operand3</i> bzw. <i>operand5</i>) bis zum Ende des Feldes übertragen.</p> <p>Wenn <i>operand2</i> eine dynamische Variable ist und die erste Stelle (<i>operand5</i>) der aktuellen Länge der Variable plus 1 entspricht, was bedeutet, dass die MOVE-Operation verwendet wird, um die Länge der Variablen zu vergrößern, muss <i>operand6</i> angegeben werden, um die neue Länge der Variablen zu bestimmen.</p> <p><b>Anmerkung:</b> MOVE mit SUBSTRING-Option ist eine Byte-für-Byte-Übertragung (d.h. die unter <i>Arithmetische Operationen im Leitfaden zur Programmierung</i> beschriebenen Regeln gelten hierbei nicht).</p>
---------------------------	---

## Syntax 3 - MOVE BY NAME / POSITION

MOVE BY { [NAME] POSITION } operand1 TO operand2
---

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	G		ja	nein
<i>operand2</i>	G		ja	nein

Syntax-Element-Beschreibung:

<p><b>MOVE BY NAME</b> <i>operand1</i> TO <i>operand2</i></p>	<p><b>Positionsunabhängige Übertragung:</b></p> <p>Mit dieser Option können Sie einzelne in einer Datenstruktur enthaltene Felder in eine andere Datenstruktur übertragen, und zwar unabhängig von ihrer Position innerhalb der Struktur.</p> <p>Ein Feld kann nur übertragen werden, wenn sein Name in beiden Datenstrukturen vorkommt (dies gilt auch für mit einem REDEFINE-Statement redefinierte Felder sowie Felder, die aus einer Redefinition resultieren). Die einzelnen Felder können jedes beliebige Format haben. Die beiden Operanden können auch Views sein.</p> <p><b>Anmerkung:</b> Die Reihenfolge der einzelnen Übertragungen ergibt sich aus der Reihenfolge der Felder in <i>operand1</i></p> <p>Siehe auch <i>Beispiel 2 – Statement MOVE BY NAME</i>.</p> <p><b>MOVE BY NAME mit Arrays:</b></p> <p>Enthalten die Datenstrukturen Arrays, so werden diese bei der Übertragung intern mit Index (*) versehen; dies kann zu einem Fehler führen, falls die Arrays nicht den Zuweisungsbedingungen für Arrays (siehe Abschnitt <i>Verarbeitung von Arrays im Leitfaden zur Programmierung</i>) entsprechen.</p> <p>Siehe auch <i>Beispiel 3 – MOVE BY NAME mit Arrays</i>.</p>
---	---

<p><b>MOVE BY POSITION</b> <i>operand1</i> TO <i>operand2</i></p>	<p><b>Positionsabhängige Übertragung:</b></p> <p>Mit dieser Option können Sie Werte von Feldern einer Gruppe in Felder einer anderen Gruppe übertragen, und zwar unabhängig von den Namen der Felder.</p> <p>Die Werte werden Feld für Feld von einer Gruppe in die andere übertragen, und zwar in der Reihenfolge, in der die Felder definiert sind (Felder, die aus einer Redefinition resultieren, werden dabei nicht berücksichtigt).</p> <p>Die einzelnen Felder können jedes beliebige Format haben. Die Anzahl der Felder in beiden Gruppen muss gleich sein; auch die Level-Struktur und die Array-Dimensionen der Felder müssen einander entsprechen. Format- Umsetzungen erfolgen entsprechend der im Natural Statements-Handbuch beschriebenen Regeln für arithmetische Operationen. Die beiden Operanden können auch Views sein.</p> <p>Siehe auch <i>Beispiel 4 – MOVE BY POSITION</i>.</p>
---	--

#### Syntax 4 – MOVE EDITED (Editiermaske mit *operand2* angegeben)

<p><b>MOVE EDITED</b> <i>operand1</i> TO <i>operand2</i> { (EM=value) } { (EMU=value) }</p>
---

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition		
<i>operand1</i>	C	S	A		A	U								B					ja	nein
<i>operand2</i>		S	A		A	U	N	P	I	F	B	D	T	L					ja	ja

Syntax-Element-Beschreibung:



<b>MOVE EDITED</b>	<p><b>Übertragung mit Editiermaske:</b></p> <p>Ist für <i>operand2</i> eine Editiermaske definiert, so wird der Wert von <i>operand1</i> unter Verwendung dieser Editiermaske in das Feld <i>operand2</i> übertragen.</p> <p>Die Editiermaske kann als eine Eingabe-Editiermaske für <i>operand2</i> angesehen werden, die dazu dient anzugeben, an welchen Stellen in dem alphanumerischen Inhalt von <i>operand1</i> der signifikante Input für <i>operand2</i> zu finden ist.</p> <p>Wenn die Editiermaske mehr Zeichen oder Ziffern enthält, als in <i>operand2</i> vorhanden sind, erfolgt eine entsprechende Abschneidung. Die Länge von <i>operand1</i> darf nicht kleiner sein als die von der Editiermaske dargestellte Länge des Eingabewertes. Wenn die Länge von <i>operand1</i> die Länge der Editiermaske übersteigt, werden alle darüber hinaus gehenden Daten ignoriert.</p> <p>Unter der Voraussetzung, dass die Länge von <i>operand1</i> nicht die Länge der Editiermaske übersteigt, kann man die Operation</p> <pre>MOVE EDITED operand1 TO operand2 (EM=value)</pre> <p>als Ausführung der folgenden Operation ansehen:</p> <pre>STACK TOP DATA operand1 INPUT operand2 (EM=value)</pre> <p>Siehe auch <i>Beispiel 1 – Verschiedene Beispiele für die Benutzung des MOVE-Statements.</i></p>
<b>EM</b>	<p><b>Parameter für Editiermaske:</b></p> <p>Einzelheiten zu Editiermasken finden Sie unter dem Session-Parameter EM in der <i>Parameter-Referenz.</i></p>
<b>EMU</b>	<p><b>Parameter für Editiermaske:</b></p> <p>Einzelheiten zu Editiermasken finden Sie unter dem Session-Parameter EMU in der <i>Parameter-Referenz.</i></p>

### Syntax 5 – MOVE EDITED (Editiermaske mit *operand1* angegeben)

<b>MOVE EDITED</b> <i>operand1</i> { <b>(EM=value)</b> } <b>TO</b> <i>operand2</i> { <b>(EMU=value)</b> }
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition	
<i>operand1</i>	C	S	A	N	A	U	N	P	I	F	B	D	T	L				ja	nein
<i>operand2</i>		S	A		A	U					B							ja	ja

Syntax-Element-Beschreibung:

<b>MOVE EDITED</b>	<p><b>Übertragung mit Editiermaske:</b></p> <p>Ist für <i>operand1</i> eine Editiermaske definiert, wird diese Editiermaske auf <i>operand1</i> angewandt und der Wert anschließend in <i>operand2</i> übertragen.</p> <p>Die Editiermaske kann als eine Ausgabe-Editiermaske für <i>operand1</i> angesehen werden, die dazu dient, eine alphanumerische Zeichenkette mit dem/der durch die Editiermaske festgelegten Layout/Länge zu erstellen. Außer den aus <i>operand1</i> stammenden Datenzeichen oder -ziffern können Sie zusätzliche ausschmückende Zeichen in die Ausgabe-Zeichenkette aufnehmen.</p> <p>Wenn die Editiermaske mehr Zeichen referenziert, als in <i>operand1</i> vorhanden sind, erfolgt eine entsprechende Abschneidung. Die Länge der erstellten Ausgabe-Zeichenkette (die sich nach Anwendung der Editiermaske aus dem Wert von <i>operand1</i> ergibt) darf nicht die Länge von <i>operand2</i> übersteigen.</p> <p>Unter der Voraussetzung, dass die Länge von <i>operand2</i> nicht kleiner ist als die Länge der Editiermaske, kann man die Operation</p> <pre>MOVE EDITED <i>operand1</i> (EM=value) TO <i>operand2</i></pre> <p>als eine Operation</p> <pre>WRITE <i>operand1</i> (EM=value)</pre> <p>betrachten, bei der die Ausgabe nicht auf den Schirm erfolgt, sondern in die Variable <i>operand2</i> geschrieben wird.</p> <p>Siehe auch <i>Beispiel 1 – Verschiedene Beispiele für die Benutzung des MOVE-Statements.</i></p>
<b>EM</b>	<p><b>Parameter für Editiermaske:</b></p> <p>Einzelheiten zu Editiermasken finden Sie unter dem Session-Parameter EM in der <i>Parameter-Referenz.</i></p>
<b>EMU</b>	<p><b>Parameter für Editiermaske:</b></p> <p>Einzelheiten zu Editiermasken finden Sie unter dem Session-Parameter EMU in der <i>Parameter-Referenz.</i></p>

**Syntax 6 - MOVE LEFT / RIGHT JUSTIFIED**

(Wertübertragung mit links- oder rechtsbündiger Ausrichtung)

<b>MOVE</b> { <b>LEFT</b> } { <b>RIGHT</b> }	<b>[JUSTIFIED]</b> <i>operand1</i> [( <i>parameter</i> )] <b>TO</b> <i>operand2</i>
---	---

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate													Referenzierung erlaubt	Dynam. Definition		
<i>operand1</i>	C	S	A	N	A	U	N	P	I	F	B	D	T	L						ja	nein
<i>operand2</i>		S	A		A	U													ja	ja	

Syntax-Element-Beschreibung:

<b>MOVE LEFT / RIGHT JUSTIFIED</b>	<p><b>Ausrichtung:</b></p> <p>Mit dieser Option können Sie festlegen, ob der übertragene Wert in <i>operand2</i> links- oder rechtsbündig ausgerichtet werden soll.</p> <p>MOVE LEFT/RIGHT JUSTIFIED kann nicht benutzt werden, wenn <i>operand2</i> eine dynamische Variable ist.</p>
<b>MOVE LEFT JUSTIFIED</b>	<p><b>Linksbündig:</b></p> <p>Bei MOVE LEFT JUSTIFIED werden vorangestellte Leerzeichen in <i>operand1</i> entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt), bevor der Wert linksbündig in <i>operand2</i> übertragen wird. Der Rest von <i>operand2</i> wird dann gegebenenfalls mit Leerzeichen aufgefüllt. Falls der Wert länger als <i>operand2</i> ist, wird der Wert rechts abgeschnitten.</p>
<b>MOVE RIGHT JUSTIFIED</b>	<p><b>Rechtsbündig:</b></p> <p>Bei MOVE RIGHT JUSTIFIED werden nachfolgende Leerzeichen in <i>operand1</i> entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt), bevor der Wert rechtsbündig in <i>operand2</i> übertragen wird. Der Rest von <i>operand2</i> wird dann gegebenenfalls mit Leerzeichen aufgefüllt. Falls der Wert länger als <i>operand2</i> ist, wird der Wert links abgeschnitten.</p> <p>Siehe auch <i>Beispiel 1 – Verschiedene Beispiele für die Benutzung des MOVE-Statements</i>.</p>
<i>parameter</i>	<p><b>Invertierte Anzeigerichtung:</b></p> <p>Wenn Sie MOVE LEFT/RIGHT JUSTIFIED in Verbindung mit PM=I verwenden, erfolgt die Übertragung in folgenden Schritten:</p> <ol style="list-style-type: none"> <li>1. Falls <i>operand1</i> nicht alphanumerisches Format hat, wird der Wert in alphanumerisches Format umgesetzt.</li> <li>2. Nachfolgende Leerzeichen in <i>operand1</i> werden entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt).</li> <li>3. Bei LEFT JUSTIFIED werden außerdem vorangestellte Leerzeichen in <i>operand1</i> entfernt (auf Großrechnern werden Leerzeichen und binäre Nullen entfernt).</li> <li>4. Der Wert wird umgedreht und dann in <i>operand2</i> übertragen.</li> <li>5. Gegebenenfalls wird <i>operand2</i> mit Leerzeichen aufgefüllt oder der Wert abgeschnitten (vgl. oben).</li> </ol>

## Syntax 7 - MOVE NORMALIZED

Das Statement MOVE NORMALIZED konvertiert eine Unicode-Zeichenkette in die Unicode Normalization Form C (NFC). Die sich daraus ergebende Unicode-Zeichenkette enthält keine Kombinationssequenzen für Zeichen mehr, die als vordefinierte Zeichen zur Verfügung stehen.

Wenn das Format des Zieloperanden selbst kein Unicode ist, findet eine implizite Konvertierung von Unicode in das Codepage-Format des Zieloperanden statt – während dieser Konvertierung wird die Standard-Codepage (siehe Systemvariable \*CODEPAGE) benutzt.

Weitere Informationen zum Statement MOVE NORMALIZED siehe Abschnitt *Statements* in der *Unicode and Code Page Support*-Dokumentation.

Syntax-Diagramm:

```
MOVE NORMALIZED operand1 TO operand2
```

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate													Referenzierung erlaubt	Dynam. Definition										
<i>operand1</i>	C	S	A																							ja	nein	
<i>operand2</i>		S	A															A	U								ja	ja

Syntax-Element-Beschreibung:

<b>MOVE NORMALIZED</b>	<p><b>Konvertierung von Unicode-Feldern:</b></p> <p>Diese Option dient zum Konvertieren von Unicode-Feldern mit potenziell unnormalisiertem Inhalt in die Unicode Normalization Form C (NFC).</p> <p>Diese zusammengesetzte Form einer Unicode-Zeichenkette enthält keine Kombinationssequenzen für Zeichen, die als vordefinierte Zeichen zur Verfügung stehen.</p> <p>Siehe auch <a href="http://www.unicode.org/reports/tr15/#Canonical_Composition_Examples">http://www.unicode.org/reports/tr15/#Canonical_Composition_Examples</a> (Normalization Forms D and C Examples).</p> <p>Beispiel:</p> <pre>MOVE NORMALIZED #SCR TO #TGT</pre>
<i>operand1</i>	<p><b>Zu konvertierende Unicode-Zeichenkette:</b></p> <p><i>operand1</i> enthält die zu konvertierende Unicode-Zeichenkette.</p>
<i>operand2</i>	<p><b>Konvertierte Zeichenkette:</b></p> <p><i>operand2</i> (Zieloperand) dient zur Aufnahme der konvertierten Zeichenkette.</p>

Beispiel:

Einige Codepunkte haben unterschiedliche Darstellungen im Unicode. Zum Beispiel der deutsche Buchstabe Ä: die aufgelöste Darstellung im Unicode ist U+0041, gefolgt von U+0308, bei der ein Kombinationszeichen (U+0308) benutzt wird; eine andere Darstellung ist das vordefinierte Zeichen U+00C4. Das Statement MOVE NORMALIZED konvertiert die Unicode-Darstellung mit

Kombinationszeichen in eine standardisierte Unicode-Darstellung mittels vordefinierter Zeichen, wo möglich.

### Syntax 8 - MOVE ENCODED

Dieser Abschnitt erläutert die Syntax des Statements MOVE ENCODED.

Weitere Informationen zum Statement MOVE ENCODED siehe Abschnitt *Statements* in der *Unicode and Code Page Support*-Dokumentation.

Syntax-Diagramm:

```

MOVE ENCODED
  operand1 [[IN] CODEPAGE operand2] TO
  operand3 [[IN] CODEPAGE operand4]
  [GIVING operand5]
    
```

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition	
<i>operand1</i>	C	S	A		A	U	B											ja	nein
<i>operand2</i>		S			A	U												ja	nein
<i>operand3</i>		S			A	U	B											ja	ja
<i>operand4</i>		S	A		A	U												ja	nein
<i>operand5</i>		S						I4										ja	ja

Syntax-Element-Beschreibung:

<b>MOVE ENCODED</b>	<p><b>Codepage-Umsetzung:</b></p> <p>Das Statement <code>MOVE ENCODED</code> setzt eine in einer Codepage kodierte Zeichenkette in die äquivalente Zeichenkette einer anderen Codepage um.</p> <p><b>Anmerkung:</b> Natural verwendet für die Unicode-Konvertierung die International Components for Unicode (ICU) Library. Weitere Informationen siehe <a href="http://icu.sourceforge.net/userguide/">http://icu.sourceforge.net/userguide/</a>.</p>
<i>operand1</i>	<p><b>Umzusetzende Zeichenkette:</b></p> <p><i>operand1</i> enthält die umzusetzende Zeichenkette.</p>
<b>[IN] CODEPAGE operand2</b>	<p><b>Codepage der umzusetzenden Zeichenkette:</b></p> <p>Die Codepage von <i>operand1</i> kann nur angegeben werden, wenn <i>operand1</i> das Format A oder B hat. Siehe Anmerkung 1.</p>
<b>TO operand3</b>	<p><b>Umgesetzte Zeichenkette:</b></p> <p>Wenn das Ergebnis der Umsetzung nicht in das Zielfeld (<i>operand3</i>) passt, wird das Ergebnis aufgefüllt bzw. abgeschnitten, und als Füllzeichen wird das Leerzeichen der sich daraus ergebenden Codepage benutzt.</p> <p>Wenn das Zielfeld als eine dynamische Variable definiert wird, ist kein Auffüllen oder Abschneiden erforderlich, da die Länge der dynamischen Variablen automatisch an die Länge des Ergebnisses der Konvertierung angepasst wird.</p>
<b>[IN] CODEPAGE operand4</b>	<p><b>Codepage der umgesetzten Zeichenkette:</b></p> <p>Die Codepage von <i>operand3</i> kann nur angegeben werden, wenn <i>operand3</i> das Format A oder B hat. Siehe Anmerkung 1.</p>
<b>GIVING operand5</b>	<p><b>GIVING-Klausel:</b></p> <p>Ohne das Schlüsselwort <code>GIVING</code> wird im Falle eines Fehlers eine Natural-Fehlermeldung zurückgegeben. Wenn das Schlüsselwort <code>GIVING</code> benutzt wird, gibt <i>operand5</i> eine 0 oder den Natural-Fehlercode anstatt der Natural-Fehlermeldung zurück.</p> <p>Wenn der Zieloperand abgeschnitten wird, wird keine Natural-Fehlermeldung ausgegeben, aber wenn das Schlüsselwort <code>GIVING</code> benutzt wird, enthält <i>operand5</i> einen entsprechenden Fehlercode, um auf ein Abschneiden hinzuweisen.</p>

**Anmerkungen:**

1. Wenn kein Codepage-Operand angegeben wird, dann wird die voreingestellte (Default-)Codepage (Wert der Systemvariablen `*CODEPAGE`) benutzt.
2. Wenn der Session-Parameter `CPCVERR` in dem Statement `SET GLOBALS` oder in dem Systemkommando `GLOBALS` auf `ON` gesetzt ist, wird ein Fehler ausgegeben, wenn mindestens ein Zeichen des Ausgangsfeldes nicht ordnungsgemäß in die Ziel-Codepage umgesetzt werden konnte, sondern im Zielfeld durch ein Ersetzungszeichen ersetzt wurde.

Beispiele:

```
MOVE ENCODED A-FIELD1 TO A-FIELD2
```

Ungültig: Dies führt zu einem Syntaxfehler, da die Codepage-Namen standardmäßig übernommen werden und für *operand1* und *operand3* identisch sind.

```
MOVE ENCODED A-FIELD1 CODEPAGE 'IBM01140' TO A-FIELD2 CODEPAGE 'IBM01140'
```

Ungültig: Dies führt zu einem Fehler, da die kodierten Codepage-Namen für *operand1* und *operand3* identisch sind.

```
MOVE ENCODED A-FIELD1 CODEPAGE 'IBM01140' TO A-FIELD2 CODEPAGE 'IBM037'
```

Gültig: Die Zeichenkette in A-FIELD1 (kodiert in IBM01140) wird in A-FIELD2 (kodiert in IBM037) konvertiert.

```
MOVE ENCODED U-FIELD TO U-FIELD
```

Ungültig: Dies führt zu einem Fehler, da mindestens ein Operand vom Format A oder B sein muss.

```
MOVE ENCODED U-FIELD TO A-FIELD
```

Gültig: Die Unicode-Zeichenkette in U-FIELD wird angesichts der Tatsache, dass sie in UTF-16 kodiert ist, in das alphanumerische A-FIELD in der Standard-Codepage (\*CODEPAGE) konvertiert.

```
MOVE ENCODED A-FIELD TO U-FIELD
```

Gültig: Die Zeichenkette in A-FIELD wird angesichts der Tatsache, dass sie in der Standard-Codepage (\*CODEPAGE) kodiert ist, in das Unicode-Feld U-FIELD konvertiert.

```
MOVE ENCODED A100-FIELD CODEPAGE 'IBM1140' TO A50-FIELD CODEPAGE 'IBM037'
```

Gültig: Die Konvertierung erfolgt mittels der betreffenden Codepages von A100-FIELD (Format/Länge: A100) in A50-FIELD (Format/Länge: A50). Der Zieloperand wird abgeschnitten. Keine Natural-Fehlermeldung wird zurückgegeben.

```
MOVE ENCODED A100-FIELD CODEPAGE 'IBM1140' TO A50-FIELD CODEPAGE 'IBM037' GIVING RC-FIELD
```

Gültig: Die Konvertierung erfolgt mittels der betreffenden Codepages von A100-FIELD (Format/Länge: A100) in A50-FIELD (Format/Länge: A50). Das Ziel wird abgeschnitten. Da eine GIVING-Klausel angegeben wird, erhält das RC-FIELD einen Fehlercode, der darauf hinweist, dass in diesem Fall Werte abgeschnitten wurden.

## Beispiele

- Beispiel 1 — Verschiedene Beispiele für die Benutzung des MOVE-Statements
- Beispiel 2 — MOVE BY NAME
- Beispiel 3 — MOVE BY NAME mit Arrays
- Beispiel 4 — MOVE BY POSITION



## Beispiel 1 — Verschiedene Beispiele für die Benutzung des MOVE-Statements

```

** Example 'MOVEX1': MOVE
*****
DEFINE DATA LOCAL
1 #A (N3)
1 #B (A5)
1 #C (A2)
1 #D (A7)
1 #E (N1.0)
1 #F (A5)
1 #G (N3.2)
1 #H (A6)
END-DEFINE
*
MOVE 5 TO #A
WRITE NOTITLE 'MOVE 5 TO #A'          30X '=' #A
*
MOVE 'ABCDE' TO #B #C #D
WRITE 'MOVE ABCDE TO #B #C #D'      20X '=' #B '=' #C '=' #D
*
MOVE -1 TO #E
WRITE 'MOVE -1 TO #E'                28X '=' #E
*
MOVE ROUNDED 1.995 TO #E
WRITE 'MOVE ROUNDED 1.995 TO #E'    18X '=' #E
*
*
MOVE RIGHT JUSTIFIED 'ABC' TO #F
WRITE 'MOVE RIGHT JUSTIFIED 'ABC'' TO #F'      10X '=' #F
*
MOVE EDITED '003.45' TO #G (EM=999.99)
WRITE 'MOVE EDITED ''003.45'' TO #G (EM=999.99)'  4X '=' #G
*
MOVE EDITED 123.45 (EM=999.99) TO #H
WRITE 'MOVE EDITED 123.45 (EM=999.99) TO #H'    6X '=' #H
*
END

```

Ausgabe des Programms MOVEX1:

```

MOVE 5 TO #A                #A:      5
MOVE ABCDE TO #B #C #D     #B: ABCDE #C: AB #D: ABCDE
MOVE -1 TO #E              #E:     -1
MOVE ROUNDED 1.995 TO #E   #E:      2
MOVE RIGHT JUSTIFIED 'ABC' TO #F  #F:   ABC
MOVE EDITED '003.45' TO #G (EM=999.99) #G:    3.45
MOVE EDITED 123.45 (EM=999.99) TO #H #H: 123.45

```

## Beispiel 2 — MOVE BY NAME

```

** Example 'MOVEX2': MOVE BY NAME
*****
DEFINE DATA LOCAL
1 #SBLOCK
  2 #FIELD A (A10) INIT <'AAAAAAAAAA'>
  2 #FIELD B (A10) INIT <'BBBBBBBBBB'>
  2 #FIELD C (A10) INIT <'CCCCCCCCCC'>
  2 #FIELD D (A10) INIT <'DDDDDDDDDD'>
1 #TBLOCK
  2 #FIELD1 (A15) INIT <' '>

```

```

2 #FIELD1 (A10) INIT <' '>
2 #FIELD2 (A10) INIT <' '>
2 #FIELD3 (A10) INIT <' '>
2 #FIELD4 (A20) INIT <' '>
2 #FIELD5 (A10) INIT <' '>
END-DEFINE
*
MOVE BY NAME #SBLOCK TO #TBLOCK
*
WRITE NOTITLE 'CONTENTS OF #TBLOCK AFTER MOVE BY NAME:'
  // '=' #TBLOCK.#FIELD1
  / '=' #TBLOCK.#FIELD2
  / '=' #TBLOCK.#FIELD3
  / '=' #TBLOCK.#FIELD4
  / '=' #TBLOCK.#FIELD5
*
END

```

Inhalt von #TBLOCK nach der MOVE BY NAME-Verarbeitung:

```

CONTENTS OF #TBLOCK AFTER MOVE BY NAME:

#FIELD1:
#FIELD2: AAAAAAAAAA
#FIELD3:
#FIELD4: BBBBBBBBBB
#FIELD5:
#FIELD6: CCCCCCCCCC

```

### Beispiel 3 — MOVE BY NAME mit Arrays

```

DEFINE DATA LOCAL
  1 #GROUP1
    2 #FIELD (A10/1:10)
  1 #GROUP2
    2 #FIELD (A10/1:10)
END-DEFINE
...
MOVE BY NAME #GROUP1 TO #GROUP2
...

```

In diesem Beispiel würde das MOVE-Statement intern wie folgt aufgelöst:

```
MOVE #GROUP1.#FIELD (*) TO #GROUP2.#FIELD (*)
```

Wenn ein Teil einer indizierten Gruppe in einen anderen Teil derselben Gruppe übertragen wird, kann dies zu unerwarteten Ergebnissen führen, wie im folgenden Beispiel veranschaulicht.

```

DEFINE DATA LOCAL
  1 #GROUP1 (1:5)
    2 #FIELD1 (N1) INIT <1,2,3,4,5>
    2 REDEFINE #FIELD1
      3 #FIELD2 (N1)
END-DEFINE
...
MOVE BY NAME #GROUP1 (2:4) TO #GROUP1 (1:3)
...

```

In diesem Beispiel würde das MOVE-Statement intern wie folgt aufgelöst:

```
MOVE #FIELDA (2:4) TO #FIELDA (1:3)
MOVE #FIELDB (2:4) TO #FIELDB (1:3)
```

Zunächst wird der Inhalt der Ausprägungen 2 bis 4 von #FIELDA in die Ausprägungen 1 bis 3 von #FIELDA übertragen; d.h. die Ausprägungen erhalten folgende Werte:

Ausprägung:	1.	2.	3.	4.	5.
Wert vorher:	1	2	3	4	5
Wert nachher:	2	3	4	4	5

Dann wird der Inhalt der Ausprägungen 2 bis 4 von #FIELDB in die Ausprägungen 1 bis 3 von #FIELDB übertragen; d.h. die Ausprägungen erhalten folgende Werte:

Ausprägung:	1.	2.	3.	4.	5.
Wert vorher:	2	3	4	4	5
Wert nachher:	3	4	4	4	5

### Beispiel 4 — MOVE BY POSITION

```
DEFINE DATA LOCAL
  1 #GROUP1
    2 #FIELD1A (N5)
    2 #FIELD1B (A3/1:3)
    2 REDEFINE #FIELD1B
      3 #FIELD1BR (A9)
  1 #GROUP2
    2 #FIELD2A (N5)
    2 #FIELD2B (A3/1:3)
    2 REDEFINE #FIELD2B
      3 #FIELD2BR (A9)
END-DEFINE
...
MOVE BY POSITION #GROUP1 TO #GROUP2
...
```

In diesem Beispiel wird der Inhalt von #FIELD1A in #FIELD2A übertragen, und der Inhalt von #FIELD1B in #FIELD2B; die Felder #FIELD1BR und #FIELD2BR sind davon nicht betroffen.