

GET

Im Structured Mode sowie im Reporting Mode mit einem DEFINE DATA LOCAL-Statement gilt die folgende Syntax:

```
GET [IN] [FILE] view-name
    [PASSWORD=operand1]
    [CIPHER=operand2]
    [ { [RECORD] } ] { operand3 }
    [ { [RECORDS] } ] { *ISN [(r)] }
```

Im Reporting Mode ohne DEFINE DATA LOCAL-Statement gilt die folgende Syntax:

```
GET [IN] [FILE] ddm-name
    [PASSWORD=operand1]
    [CIPHER=operand2]
    [ { [RECORD] } ] { operand3 } operand4 ...
    [ { [RECORDS] } ] { *ISN [(r)] }
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkungen
- Syntax-Beschreibung
- Beispiel

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

Verwandte Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET SAME | GET TRANSACTION | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Das Statement GET dient dazu, einen Datensatz mit einer bestimmten Adabas-ISBN (Interne Satz-Nummer) zu lesen.

Bei XML-Datenbanken wird das GET-Statement verwendet, um ein XML-Objekt mit bekannter Objekt-ID zu lesen.

Das GET-Statement löst keine Verarbeitungsschleife aus.

Einschränkungen

- Das GET-Statement kann nicht für SQL-Datenbanken verwendet werden.
- Mit Entire System Server kann das GET-Statement nicht verwendet werden.

Syntax-Beschreibung

Operanden-Definitionstabelle:

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

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

Syntax-Element-Beschreibung:

<i>view-name</i>	<p>View-Name:</p> <p>Im Structured Mode sowie im Reporting Mode mit einem DEFINE DATA LOCAL-Statement ist <i>view-name</i> der Name eines Views, der entweder direkt in einem DEFINE DATA-Statement oder in einer separaten Global oder Local Data Area definiert ist.</p>
<i>dgm-name</i>	<p>DDM-Name:</p> <p>Im Reporting Mode ohne DEFINE DATA LOCAL-Statement ist <i>dgm-name</i> der Name eines DDMs.</p>
PASSWORD=operand1	<p>PASSWORD-Klausel/CYPHER-Klausel:</p> <p>Diese beiden Klauseln sind nur auf Adabas-Datenbanken anwendbar.</p> <p>Die PASSWORD-Klausel dient dazu, ein Passwort anzugeben, um auf Daten einer passwort-geschützten Adabas-Datei zugreifen zu können.</p> <p>Die CIPHER-Klausel dient dazu, einen Cipher-Code (Chiffrierschlüssel) anzugeben, um in chiffrierter Form gespeicherte Adabas-Daten in entschlüsselter Form zu erhalten.</p> <p>Näheres hierzu siehe FIND- und PASSW-Statement.</p>
CIPHER=operand2	
*ISN / operand3	<p>Interne Sequenz-Nummer:</p> <p>Die ISN kann als numerische Konstante oder Benutzervariable (<i>operand3</i>) oder über die Natural-Systemvariable *ISN angegeben werden.</p>
(r)	<p>Statement-Referenz:</p> <p>Die Notation (<i>r</i>) wird benutzt, um das Statement anzugeben, welches das zum ersten Lesen des Datensatzes benutzte FIND- oder READ-Statement enthält.</p> <p>Wenn (<i>r</i>) nicht angegeben wird, bezieht sich das GET-Statement auf die innerste aktive Verarbeitungsschleife.</p> <p>(<i>r</i>) kann als Statement-Label oder Sourcecode- Zeilennummer angegeben werden.</p>
operand4	<p>Referenz auf Datenbank-Felder:</p> <p>Eine spätere Referenzierung von Datenbankfeldern, die mit einem GET-Statement gelesen wurden, kann entweder unter Angabe eines Statement-Labels oder über die Sourcecode-Zeilenummer des GET-Statements erfolgen.</p>

Beispiel

```

** Example 'GETEX1': GET
*****
DEFINE DATA LOCAL
1 PERSONS VIEW OF EMPLOYEES

```

```

2 PERSONNEL-ID
2 NAME
2 FIRST-NAME
1 SALARY-INFO VIEW OF EMPLOYEES
2 NAME
2 CURR-CODE (1:1)
2 SALARY (1:1)
*
1 #ISN-ARRAY (B4/1:10)
1 #LINE-NR (N2)
END-DEFINE
*
FORMAT PS=16
LIMIT 10
READ PERSONS BY NAME
MOVE *COUNTER TO #LINE-NR
MOVE *ISN TO #ISN-ARRAY (#LINE-NR)
DISPLAY #LINE-NR PERSONNEL-ID NAME FIRST-NAME
/*
AT END OF PAGE
INPUT / 'PLEASE SELECT LINE-NR FOR SALARY INFORMATION:' #LINE-NR
IF #LINE-NR = 1 THRU 10
    GET SALARY-INFO #ISN-ARRAY (#LINE-NR)
    WRITE / SALARY-INFO.NAME
           SALARY-INFO.SALARY (1)
           SALARY-INFO.CURR-CODE (1)
END-IF
END-ENDPAGE
/*
END-READ
END

```

Ausgabe des Programms GETEX1:

Page 1

05-01-13 13:17:42

#LINE-NR	PERSONNEL ID	NAME	FIRST-NAME
1	60008339	ABELLAN	KEPA
2	30000231	ACHIESON	ROBERT
3	50005800	ADAM	SIMONE
4	20008800	ADKINSON	JEFF
5	20009800	ADKINSON	PHYLLIS
6	20012700	ADKINSON	HAZEL
7	20013800	ADKINSON	DAVID
8	20019600	ADKINSON	CHARLIE
9	20008600	ADKINSON	MARTHA
10	20005700	ADKINSON	TIMMIE

PLEASE SELECT LINE-NR FOR SALARY INFORMATION: 1

ABELLAN 1450000 PTA