

WRITE

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax 1 — Dynamische Formatierung
- Syntax 1 – Beschreibung
- Syntax 2 — Vordefinierte Form/Map benutzen
- Syntax 2 — Beschreibung
- Beispiele

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

Verwandte Statements: AT END OF PAGE | AT TOP OF PAGE | CLOSE PRINTER | DEFINE PRINTER | DISPLAY | EJECT | FORMAT | NEWPAGE | PRINT | SKIP | SUSPEND IDENTICAL SUPPRESS | WRITE TITLE | WRITE TRAILER

Gehört zur Funktionsgruppe: *Erstellen von Ausgabe-Reports*

Funktion

Das Statement WRITE dient dazu, Ausgaben in Freiformat zu erzeugen, die nicht bereits vorformatiert sind (vgl. DISPLAY-Statement).

Das WRITE-Statement unterscheidet sich vom DISPLAY-Statement in folgenden Punkten:

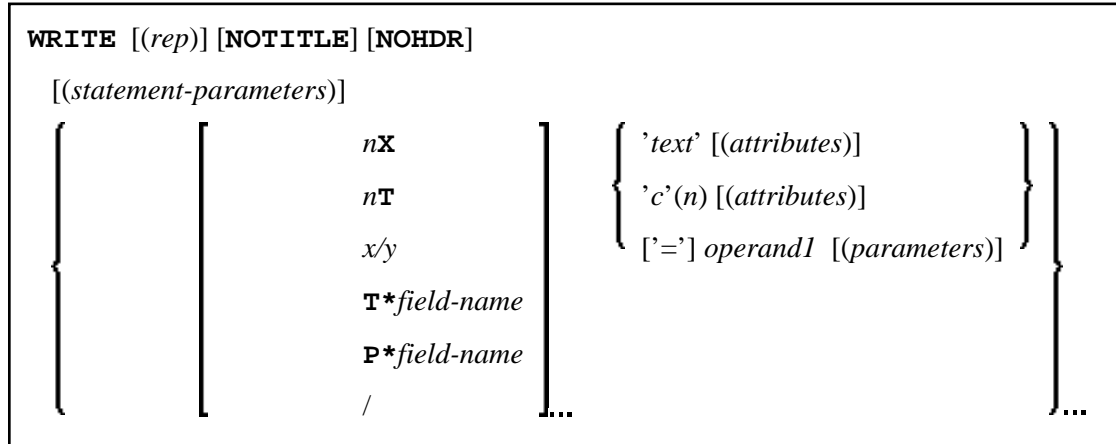
- Passt ein Feld bzw. Textelement nicht mehr in eine Zeile, wird es automatisch in der nächsten Zeile ausgegeben. Ein Feld bzw. Textelement wird nicht auf zwei Zeilen verteilt.
- Es werden keine Standard-Spaltenüberschriften erzeugt. Die Ausgabelänge der Felder richtet sich nach der Länge der tatsächlich ausgegebenen Feldwerte.
- Mehrere Werte/Ausprägungen eines Arrays werden nicht untereinander sondern nebeneinander ausgegeben.

Siehe auch die folgenden Themen im *Leitfaden zur Programmierung*:

- *Steuerung der Ausgabe von Daten*
- *Statements DISPLAY und WRITE*
- *Index-Notation für multiple Felder und Periodengruppen*

- *Beispiel für DISPLAY VERT mit WRITE-Statement*
- *Layout einer Ausgabeseite*

Syntax 1 — Dynamische Formatierung



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

Syntax 1 – Beschreibung

Operanden-Definitionstabelle:

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

Syntax-Element-Beschreibung:

| | |
|---------------------|--|
| (<i>rep</i>) | <p>Report-Spezifikation:</p> <p>Mit der Notation (<i>rep</i>) kann ein bestimmter Report angegeben werden, wenn ein Programm mehrere Ausgaben erzeugen soll.</p> <p>Es kann ein Wert von 0 bis 31 oder ein logischer Name, der mit einem DEFINE PRINTER-Statement zugewiesen wurde, angegeben werden.</p> <p>Wenn (<i>rep</i>) nicht angegeben wird, bezieht sich das WRITE-Statement auf den ersten Report (Report 0).</p> <p>Wenn diese Druckdatei für Natural als PC definiert wird, wird der Report auf den PC heruntergeladen, siehe <i>Beispiel 5</i>.</p> <p>Informationen, wie Sie das Format eines mit Natural erstellten Ausgabe-Reports steuern, finden Sie im Abschnitt <i>Steuerung der Ausgabe von Daten im Leitfaden zur Programmierung</i>.</p> |
|---------------------|--|

| | |
|----------------|--|
| NOTITLE | <p>Unterdrückung der Standard-Kopfzeile:</p> <p>Natural generiert für jede über ein WRITE-Statement ausgegebene Seite eine Kopfzeile. Diese Kopfzeile enthält die laufende Seitennummer, Uhrzeit und Datum. Die Uhrzeit wird zu Beginn der Programmausführung gesetzt. Die Ausgabe dieser Standard-Kopfzeile kann durch Angabe des Schlüsselwortes NOTITLE oder durch ein WRITE TITLE-Statement unterdrückt werden.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> ● Ausgabe einer Standard-Kopfzeile: <code>WRITE NAME</code> ● Ausgabe einer eigenen Kopfzeile: <code>WRITE NAME WRITE TITLE 'user-title'</code> ● Ausgabe ohne Kopfzeile: <code>WRITE NOTITLE NAME</code> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Wenn die NOTITLE-Option verwendet wird, gilt sie für alle DISPLAY-, PRINT- und WRITE-Statements im selben Objekt, die Daten auf denselben Report schreiben. 2. Natural prüft, wann ein Seitenvorschub erforderlich ist, <i>bevor</i> ein WRITE-Statement ausgeführt wird. <i>Während</i> der Ausführung eines WRITE-Statements werden keine neuen Seiten mit Kopf- oder Fußzeilen generiert. |
| NOHDR | <p>Unterdrückung der Spaltenüberschrift:</p> <p>Das WRITE-Statement selbst erzeugt keine Spaltenüberschriften. Wenn Sie allerdings das WRITE-Statement zusammen mit einem DISPLAY-Statement verwenden, können Sie mit der Option NOHDR des WRITE-Statements die vom DISPLAY-Statement generierten Spaltenüberschriften unterdrücken.</p> <p>Die NOHDR-Option ist nur relevant, wenn das WRITE-Statement nach einem DISPLAY-Statement steht, die Ausgabe sich insgesamt über mehr als eine Seite erstreckt und die Ausführung des WRITE-Statements zur Ausgabe einer neuen Seite führt.</p> <p>Ohne NOHDR-Option würden auf dieser neuen Seite die DISPLAY-Spaltenüberschriften ausgegeben, mit NOHDR werden sie dort nicht ausgegeben.</p> |

| | |
|--|--|
| <i>statement-parameters</i> | <p>Parameter-Definition auf Statement-Ebene:</p> <p>Unmittelbar nach dem WRITE-Statement können Sie auf Statement-Ebene in Klammern einzelne Session-Parameter angeben. Die Werte dieser Parameter haben dann für das betreffende Statement vor auf übergeordneter Ebene mittels GLOBALS-Kommando, SET GLOBALS- (nur im Reporting Mode) oder FORMAT-Statement gesetzten Parameterwerten.</p> <p>Wenn Sie mehrere Parameter angeben, müssen Sie sie durch ein oder mehrere Leerzeichen voneinander trennen. Eine einzelne Parameterangabe darf sich nicht über zwei Sourcecode-Zeilen erstrecken.</p> <p>Anmerkung: Die hier gültigen Parameter-Einstellungen kommen nur für Variablen-Felder in Betracht und haben keine Auswirkungen auf Text-Konstanten. Wenn Sie Feldattribute für eine Text-Konstante setzen möchten, dann müssen diese explizit für dieses Element gesetzt werden; siehe <i>Parameter-Definition auf Element-Ebene</i>.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>Liste der Parameter</i> ● <i>Beispiel für die Benutzung von Parametern auf Statement- und Element-Ebene</i> ● <i>Beispiel 5 – WRITE-Statement mit '=' und Parametern auf Statement/Element-Ebene.</i> |
| <i>nX, nT, x/y, T*field-name, P*field-name, '=' , /,</i> | <p>Notation Feld-Positionierung:</p> <p>Siehe <i>Feld-Positionierung</i> im Abschnitt <i>Formatierung der Ausgabe</i>.</p> |
| <i>'text', 'c'(n), attributes, operand1, parameters</i> | <p>Text/Attributzuweisung:</p> <p>Siehe <i>Text-, Attribut-Zuweisung, Ausgabe-Elemente</i> im Abschnitt <i>Formatierung der Ausgabe</i>.</p> |

Liste der Parameter

| Parameter, die mit dem WRITE-Statement angegeben werden können: | | Spezifikation |
|---|-------------------------------------|-------------------------|
| | | S = auf Statement-Ebene |
| | | E =auf Element-Ebene |
| AD | Attribute Definition | SE |
| AL | Alphanumeric Length for Output | SE |
| CD | Color Definition | SE |
| CV | Control Variable | SE |
| DF | Date Format | SE |
| DL | Display Length for Output | SE |
| DY | Dynamic Attributes | SE |
| EM | Edit Mask | SE |
| EMU | Unicode Edit Mask | E |
| FL | Floating Point Mantissa Length | SE |
| IS | Identical Suppress | SE |
| LS | Line Size | S |
| MC | Multiple-Value Field Count | S |
| MP | Maximum Number of Pages of a Report | S |
| NL | Numeric Length for Output | SE |
| PC | Periodic Group Count | S |
| PM | Print Mode | SE |
| PS | Page Size * | S |
| SG | Sign Position | SE |
| UC | Underlining Character | S |
| ZP | Zero Printing | SE |

* Der Session-Parameter PS wird nicht berücksichtigt, wenn die Anzahl der Ausprägungen eines Arrays den PS-Wert überschreitet.

Ausführliche Beschreibungen der oben genannten Session-Parameter finden Sie in der *Parameter-Referenz*.

Siehe auch die folgenden Themen im *Leitfaden zur Programmierung*:

- *Spaltenüberschriften zentrieren – der HC-Parameter*
- *Breite von Spaltenüberschriften – der HW-Parameter*

- Füllzeichen für Überschriften – die Parameter FC und GC
- Unterstreichungszeichen für Überschriften – der UC-Parameter

Beispiel für die Parameter-Benutzung auf Statement/Element-Ebene

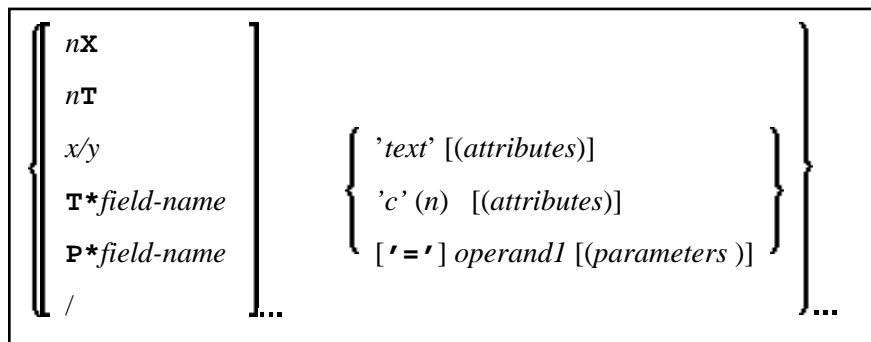
```

DEFINE DATA LOCAL
1 VARI (A4)      INIT <'1234'>          /*      Output
END-DEFINE      /*      Produced
*              /*      -----
WRITE           'Text'                 VARI          /*      Text 1234
WRITE (AD=U)   'Text'                 VARI          /*      Text 1234
WRITE         'Text' (AD=U)           VARI (AD=U)   /*      Text 1234
WRITE         'Text' (AD=U)           VARI          /*      Text 1234
END

```

Siehe auch *Beispiel 5 – WRITE-Statement mit '=' und Parametern auf Statement/Element-Ebene.*

Formatierung der Ausgabe



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

Feld-Positionierung

| | |
|------------------------|--|
| nX | <p>Spaltenabstand:</p> <p>Mit der nX Notation können Sie zwischen zwei Feldern n Leerzeichen einfügen.</p> <p>Beispiel:</p> <pre>WRITE NAME 5X SALARY</pre> <p>Siehe auch:</p> <ul style="list-style-type: none"> • <i>Beispiel 2 – WRITE-Statement mit nX-, nT-Notation</i> (weiter unten) • <i>Spaltenabstand – der SF-Parameter und die Notation nX</i> (im Leitfaden zur Programmierung) |
|------------------------|--|

| | |
|----------------------------|---|
| <i>nT</i> | <p>Tabulator-Einstellungen:</p> <p>Mit der Notation <i>nT</i> setzen Sie Tabulatoren, d.h. die Ausgabe eines Feldes beginnt ab Spalte <i>n</i>. Ein Tabulator, der bereits durch eine andere Ausgabe belegt ist, darf nicht gesetzt werden.</p> <p>In dem folgenden Beispiel wird das Feld NAME ab Spalte 25 und SALARY ab Spalte 50 ausgegeben:</p> <pre>WRITE 25T NAME 50T SALARY</pre> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>Beispiel 2 – WRITE-Statement mit nX-, nT-Notation</i> (weiter unten) ● <i>Tabulator-Notation nT</i> (im Leitfaden zur Programmierung) |
| <i>x/y</i> | <p><i>x/y</i>-Positionierung::</p> <p>Mit der Notation <i>x/y</i> erreichen Sie, dass ein Feld <i>x</i> Zeilen unter der Ausgabe des letzten Statements, und zwar ab Spalte <i>y</i> ausgegeben wird. <i>y</i> darf nicht 0 sein. Eine Spalte, die in derselben Ausgabezeile bereits belegt ist, darf nicht angegeben werden.</p> <p>Siehe auch <i>Positionierungsnotation x/y</i> (im Leitfaden zur Programmierung).</p> |
| <i>T*field-name</i> | <p>Feldbezogene Positionierung:</p> <p>Mit der Notation <i>T*</i> können Sie die WRITE-Ausgabe nach der Position eines in einem vorangegangenen DISPLAY-Statement ausgegebenen Feldes (<i>field-name</i>) ausrichten. Es ist nicht erlaubt, auf eine bereits belegte Position zu positionieren.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>Beispiel 3 – WRITE-Statement mit Notation T*</i> (weiter unten) ● <i>Tabulator-Notation T*field</i> (im Leitfaden zur Programmierung) |
| <i>P*field-name</i> | <p>Feld- und zeilenbezogene Positionierung:</p> <p>Mit der Notation <i>P*</i> können Sie die WRITE-Ausgabe nach der Position und Zeile eines in einem vorangegangenen DISPLAY-Statement ausgegebenen Feldes (<i>field-name</i>) ausrichten. Diese Notation wird vor allem nach DISPLAY VERTICALLY-Statements verwendet. Es ist nicht erlaubt, auf eine bereits belegte Position zu positionieren.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>Beispiel 4 – WRITE-Statement mit Notation P*</i> (weiter unten) ● <i>Tabulator-Notation P*field</i> (im Leitfaden zur Programmierung) |

| | |
|-----|--|
| '=' | <p>Feldinhalt hinter Feldüberschrift:</p> <p>Ein Gleichheitszeichen in Apostrophen ('=') vor einem Feld bewirkt, dass vor dem Feldwert die (im DEFINE DATA-Statement oder im DDM) für das Feld definierte Überschrift ausgegeben wird.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>WRITE-Statement mit '=' , 'text' , '/'</i> ● <i>WRITE-Statement mit '=' und Parametern auf Statement/Element-Ebene</i> |
| / | <p>Zeilenvorschub – Schrägstrich-Notation:</p> <p>Ein Schrägstrich (/) zwischen Feldern/Textelementen bewirkt einen Zeilenvorschub, d.h. die nachfolgenden Felder/ Textelemente werden in der nächsten Zeile ausgegeben.</p> <p>Beispiel:</p> <pre>WRITE NAME / SALARY</pre> <p>Für mehrfachen Zeilenvorschub geben Sie mehrere Schrägstriche an.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>WRITE-Statement mit '=' , 'text' , '/' (weiter unten)</i> ● <i>Zeilenvorschub – die Schrägstrich-Notation (im Leitfaden zur Programmierung)</i> ● <i>Beispiel für Zeilenvorschub in WRITE-Statement (im Leitfaden zur Programmierung)</i> |

Text-, Attribut-Zuweisung, Ausgabe-Elemente

| | |
|--------|--|
| 'text' | <p>Text-Zuweisung:</p> <p>Der in Apostrophen stehende Text wird ausgegeben.</p> <p>Beispiel:</p> <pre>WRITE 'EMPLOYEE' NAME 'MARITAL/STATUS' MAR-STAT</pre> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>WRITE-Statement mit '=' , 'text' , '/' (weiter unten)</i> ● <i>Text Notation, Mit einem Statement zu benutzenden Text definieren - die 'text'-Notation (im Leitfaden zur Programmierung)</i> |
|--------|--|

| | |
|-------------------|--|
| 'c'(n) | <p>Zeichen-Wiederholung:</p> <p>Das in Apostrophen stehende Zeichen (character) wird <i>n</i>-mal unmittelbar vor dem Feldwert ausgegeben.</p> <p>Zum Beispiel:</p> <pre>WRITE '*' (5) '=' NAME</pre> <p>führt zur Ausgabe von</p> <pre>***** SMITH</pre> <p>Siehe auch <i>Text-Notation</i>, <i>Vor einem Feldwert n mal anzuzeigendes Zeichen definieren - die 'c'(n)-Notation</i> (im Leitfaden zur Programmierung).</p> |
| attributes | <p>Felddarstellung und Farbattribute:</p> <p>Es ist möglich, den auszugebenden Feldern/Texten Anzeige- und Farbattribute zuzuordnen. Diese Attribute und die zu benutzende Syntax sind im Abschnitt <i>Ausgabeattribute</i> weiter unten beschrieben.</p> <p>Beispiele:</p> <pre>WRITE 'TEXT' (BGR) WRITE 'TEXT' (B) WRITE 'TEXT' (BBLC)</pre> |
| operand1 | <p>Name des auszugebenden Feldes:</p> <p><i>operand1</i> gibt das Feld an, dessen Inhalt an diese Stelle geschrieben wird.</p> |
| parameters | <p>Parameter-Definition auf Element-Ebene:</p> <p>Unmittelbar nach <i>operand1</i> können Sie auf Element-Ebene in Klammern einzelne Session-Parameter setzen. Diese Parameterwerte haben dann für das betreffende Feld Vorrang vor den mit einem GLOBALS-Kommando, SET GLOBALS- (nur im Reporting Mode) oder FORMAT-Statement oder auf Statement-Ebene gesetzten Parameterwerten.</p> <p>Wenn Sie mehrere Parameter angeben, müssen Sie sie durch ein oder mehrere Leerzeichen voneinander trennen. Eine Parameterangabe darf sich jeweils nicht über zwei Sourcecode-Zeilen erstrecken.</p> <p>Siehe auch:</p> <ul style="list-style-type: none"> ● <i>Liste der Parameter</i> ● <i>Beispiel für die Parameter-Benutzung auf Statement- und Element-Ebene</i> |

Ausgabeattribute

Sie können den ausgegebenen Feldern/Textelementen Anzeige- und Farbattribute zuordnen. Sie können die folgenden Attribute angeben:

| | |
|----------------------|--|
| FORM/MAP | <p>Benutzung des vordefinierten Form/Map-Layouts:</p> <p>Diese Option verwenden Sie, wenn Sie für die Ausgabe eine (mit dem Natural Map Editor erstellte) Map verwenden wollen.</p> <p>WRITE USING MAP bedeutet nicht, dass jedesmal, wenn die Map ausgegeben wird, automatisch eine neue Seite ausgegeben wird.</p> <p>Für den Zeilenabstand muss der Parameter LS um ein Byte größer gesetzt werden als die Zeilenlänge der Map.</p> |
| <i>operand1</i> | <p>Form/Map-Name:</p> <p><i>operand1</i> ist der Name der zu verwendenden Map.</p> |
| <i>operand2</i> | <p>Auszugebendes Feld:</p> <p><i>operand2</i> ist der Name des auszugebenden Feldes bzw. der auszugebenden Felder.</p> <p>Ist <i>operand1</i> eine Konstante und wird <i>operand2</i> nicht angegeben, so werden bei der Kompilierung die Felder aus der Map-Source übernommen.</p> <p>Die Felder müssen bezüglich Anzahl, Reihenfolge, Format, Länge und (bei Arrays) Anzahl der Ausprägungen mit den Feldern in de/mr referenzierten Layout/Map übereinstimmen, sonst tritt ein Fehler auf.</p> |
| NOTITLE/NOHDR | <p>Unterdrückung der Kopfzeile/Spaltenüberschrift:</p> <p>Die Optionen NOTITLE und NOHDR sind unter <i>Syntax 1</i> des WRITE-Statements beschrieben.</p> |

Beispiele

- Beispiel 1 – WRITE-Statement mit '=', 'text', '/'
- Beispiel 2 – WRITE-Statement mit nX-, nT-Notation
- Beispiel 3 – WRITE-Statement mit Notation T*
- Beispiel 4 – WRITE-Statement mit Notation P*
- Beispiel 5 – WRITE-Statement mit '=' und Parametern auf Statement/Element-Ebene
- Beispiel 6 – Report-Spezifikation mit für Natural als PC definierter Ausgabedatei

Beispiel 1 – WRITE-Statement mit '=', 'text', '/'

```

** Example 'WRTEX1': WRITE (with '=', 'text', '/')
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 FULL-NAME
    3 FIRST-NAME
    3 MIDDLE-I

```

```

      3 NAME
      2 CITY
      2 COUNTRY
END-DEFINE
*
LIMIT 1
READ EMPL-VIEW BY NAME
/*
  WRITE NOTITLE
    '=' NAME '=' FIRST-NAME '=' MIDDLE-I //
    'L O C A T I O N' /
    'CITY: ' CITY /
    'COUNTRY:' COUNTRY //
/*
END-READ
END

```

Ausgabe des Programms WRTEX1:

```

NAME: ABELLAN                FIRST-NAME: KEPA                MIDDLE-I:

L O C A T I O N
CITY:   MADRID
COUNTRY: E

```

Beispiel 2 – WRITE-Statement mit nX-, nT-Notation

```

** Example 'WRTEX2': WRITE (with nX, nT notation)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY NAME
  WRITE NOTITLE 5X NAME 50T JOB-TITLE
END-READ
END

```

Ausgabe des Programms WRTEX2:

```

ABELLAN                MAQUINISTA
ACHIESON                DATA BASE ADMINISTRATOR
ADAM                    CHEF DE SERVICE
ADKINSON                PROGRAMMER

```

Beispiel 3 – WRITE-Statement mit Notation T*

```

** Example 'WRTEX3': WRITE (with T* notation)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 SALARY (1)
END-DEFINE
*
LIMIT 5
READ EMPL-VIEW BY CITY STARTING FROM 'ALBU'

```

```

DISPLAY NOTITLE CITY NAME SALARY (1)
AT BREAK CITY
/*
WRITE / 'CITY AVERAGE:' T*SALARY (1) AVER(SALARY(1)) //
/*
END-BREAK
END-READ
END

```

Ausgabe des Programms WRTEX3:

| CITY | NAME | ANNUAL SALARY |
|---------------|----------|---------------|
| ALBUQUERQUE | HAMMOND | 22000 |
| ALBUQUERQUE | ROLLING | 34000 |
| ALBUQUERQUE | FREEMAN | 34000 |
| ALBUQUERQUE | LINCOLN | 41000 |
| CITY AVERAGE: | | 32750 |
| ALFRETON | GOLDBERG | 4800 |
| CITY AVERAGE: | | 4800 |

Beispiel 4 – WRITE-Statement mit Notation P*

```

** Example 'WRTEX4': WRITE (with P* notation)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
2 NAME
2 CITY
2 BIRTH
2 SALARY (1)
END-DEFINE
*
LIMIT 3
READ EMPL-VIEW BY CITY FROM 'N'
DISPLAY NOTITLE NAME CITY
VERT AS 'BIRTH/SALARY' BIRTH (EM=YYYY-MM-DD) SALARY (1)
SKIP 1
AT BREAK CITY
WRITE / 'CITY AVERAGE' P*SALARY (1) AVER(SALARY (1)) //
END-BREAK
END-READ
END

```

Ausgabe des Programms WRTEX4:

| NAME | CITY | BIRTH SALARY |
|----------|-----------|---------------------|
| WILCOX | NASHVILLE | 1970-01-01 38000 |
| MORRISON | NASHVILLE | 1949-07-10 36000 |

```

CITY AVERAGE                                37000

BOYER                NEMOURS                1955-11-23
                                      195900

CITY AVERAGE                                195900

```

Beispiel 5 – WRITE-Statement mit '=' und Parametern auf Statement/Element-Ebene

```

** Example 'WRTEX5': WRITE (using '=', statement/element parameters)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 PERSONNEL-ID
  2 PHONE
END-DEFINE
*
LIMIT 2
READ EMPL-VIEW BY NAME
  WRITE NOTITLE (AL=16 NL=8)
    '=' PERSONNEL-ID '=' NAME '=' PHONE (AL=10 EM=XXX-XXXXXXX)
END-READ
END

```

Ausgabe des Programms WRTEX5:

```

PERSONNEL ID: 60008339          NAME: ABELLAN          TELEPHONE: 435-6726
PERSONNEL ID: 30000231          NAME: ACHIESON          TELEPHONE: 523-341

```

Beispiel 6 – Report-Spezifikation mit für Natural als PC definierter Ausgabedatei

```

** Example 'PCDIEX1': DISPLAY and WRITE to PC
**
** NOTE: Example requires that Natural Connection is installed.
*****
DEFINE DATA LOCAL
01 PERS VIEW OF EMPLOYEES
  02 PERSONNEL-ID
  02 NAME
  02 CITY
END-DEFINE
*
FIND PERS WITH CITY = 'NEW YORK'          /* Data selection
  WRITE (7) TITLE LEFT 'List of employees in New York' /
  DISPLAY (7)          /* (7) designates the output file (here the PC).
    'Location'      CITY
    'Surname'       NAME
    'ID'            PERSONNEL-ID
END-FIND
END

```