

# Datumsinformationen verarbeiten

Dieses Kapitel behandelt verschiedene Aspekte der Behandlung von Datumsinformationen in Ihren Natural-Anwendungen.

Folgende Themen werden behandelt:

- Editiermasken für Datumsfelder and Datumssystemvariablen
  - Standard-Editiermaske für Datum — der DTFORM-Parameter
  - Datumsformat für alphanumerische Darstellung — der DF-Parameter
  - Datumsformat für Ausgabe — der DFOUT-Parameter
  - Datumsformat für Stack — der DFSTACK-Parameter
  - Year Sliding Window — der YSLW-Parameter
  - Kombinationen von DFSTACK und YSLW
  - Year Fixed Window
  - Datumsformat für Standard-Seitenüberschriften — der DFTITLE-Parameter
- 

## Editiermasken für Datumsfelder and Datumssystemvariablen

Wenn Sie den Wert eines Datumsfeldes in einer bestimmten Form ausgeben möchten, geben Sie normalerweise für das Feld eine Editiermaske an. Mit der Editiermaske bestimmen Sie Zeichen für Zeichen, wie die Ausgabe aussehen soll.

Falls Sie das aktuelle Datum in einer bestimmten Form verwenden möchten, brauchen Sie hierfür kein Datumsfeld mit einer entsprechenden Editiermaske zu definieren; stattdessen können Sie einfach eine *Datumssystemvariable* verwenden. Natural bietet verschiedene Datumssystemvariablen, die alle das aktuelle Datum in unterschiedlichen Darstellungsformen enthalten. Bei manchen dieser Darstellungsformen ist die Jahreszahl-Angabe zweistellig, bei manchen vierstellig.

Weitere Informationen sowie eine Liste aller Datumssystemvariablen finden Sie in der *Systemvariablen*-Dokumentation.

## Standard-Editiermaske für Datum — der DTFORM-Parameter

Der Profilparameter DTFORM bestimmt das Standardformat, das für Datumsangaben als Teil von Natural-Standard-Reporttiteln, für Datumskonstanten und für Datumseingaben gilt.

Dieses Datumsformat bestimmt die Reihenfolge der Angaben für Tag, Monat und Jahr sowie die Trennzeichen, die zwischen diesen Angaben stehen müssen.

Mögliche DTFORM-Einstellungen sind:

Einstellung	Datumsformat <sup>*</sup>	Beispiel
DTFORM=I	<i>yyyy-mm-dd</i>	2005-12-31
DTFORM=G	<i>dd.mm.yyyy</i>	31.12.2005
DTFORM=E	<i>dd/mm/yyyy</i>	31/12/2005
DTFORM=U	<i>mm/dd/yyyy</i>	12/31/2005

\* *dd* = day/Tag, *mm* = month/Monat, *yyyy* = year/Jahr.

Der DTFORM-Parameter kann im Natural-Parametermodul oder dynamisch beim Aufruf von Natural gesetzt werden. Standardmäßig gilt DTFORM=I.

## Datumsformat für alphanumerische Darstellung — der DF-Parameter

Wenn eine Editiermaske angegeben ist, wird die Darstellung des Feldwertes durch die Editiermaske bestimmt. Wenn keine Editiermaske angegeben ist, wird die Darstellung des Feldwertes durch den Session-Parameter DF in Kombination mit dem DTFORM-Profilparameter bestimmt.

Mit dem DF-Parameter können Sie eine der folgenden Datumsdarstellungen wählen:

<b>DF=S</b>	8-Byte-Darstellung mit 2-stelliger Jahreskomponente und Begrenzungszeichen ( <i>yy-mm-dd</i> ).
<b>DF=I</b>	8-Byte-Darstellung mit 4-stelliger Jahreskomponente ohne Begrenzungszeichen ( <i>yyyymmdd</i> ).
<b>DF=L</b>	10-Byte-Darstellung mit 4-stelliger Jahreskomponente und Begrenzungszeichen ( <i>yyyy-mm-dd</i> ).

Bei jeder Darstellung wird die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die zu verwendenden Begrenzungszeichen durch den DTFORM-Parameter bestimmt.

Standardmäßig gilt DF=S (außer bei INPUT-Statements; siehe unten).

Der DF-Parameter wird bei der Kompilierung ausgewertet.

Der DF-Parameter gilt bei folgenden Statements und in folgenden Situationen:

- FORMAT
- INPUT, DISPLAY, WRITE und PRINT auf Statement- und Elementebene (Feldebene)
- MOVE, COMPRESS, STACK, RUN und FETCH auf Elementebene (Feldebene)

Bei Angabe in einem dieser Statements hat der DF-Parameter folgende Auswirkung:

Statement	Auswirkung des DF-Parameters
DISPLAY, WRITE, PRINT	Wenn der Wert einer Datumsvariablen mit einem dieser Statements ausgegeben wird, wird der Wert in alphanumerische Darstellung umgesetzt, bevor er ausgegeben wird. Der DF-Parameter bestimmt, welche Darstellung dafür verwendet wird.
MOVE, COMPRESS	Wenn der Wert einer Datumsvariablen mit einem MOVE- oder COMPRESS-Statement in ein alphanumerisches Feld übertragen wird, wird der Wert in alphanumerische Darstellung umgesetzt, bevor er übertragen wird. Der DF-Parameter bestimmt, welche Darstellung dafür verwendet wird.
STACK, RUN, FETCH	Wenn der Wert einer Datumsvariablen auf dem Stack abgelegt wird, wird er in alphanumerische Darstellung umgesetzt, bevor er auf dem Stack abgelegt wird. Der DF-Parameter bestimmt, welche Darstellung dafür verwendet wird.  Dasselbe gilt, wenn eine Datumsvariable als Parameter in einem FETCH- oder RUN-Statement angegeben ist (da diese Parameter ebenfalls über den Stack übergeben werden).
INPUT	Wenn eine Datumsvariable in einem INPUT-Statement verwendet wird, bestimmt der DF-Parameter, in welcher Form ein Wert in dieses Feld eingegeben werden muss.  Wenn dagegen eine Datumsvariable, für die kein DF-Parameter angegeben ist, in einem INPUT-Statement verwendet wird, kann das Datum entweder mit 2-stelliger Jahresangabe und Begrenzungszeichen oder mit 4-stelliger Jahresangabe ohne Begrenzungszeichen eingegeben werden. Auch in diesem Fall werden die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die zu verwendenden Begrenzungszeichen durch den DTFORM-Parameter bestimmt.

### Anmerkung:

Bei DF=S stehen nur 2 Stellen für die Jahresangabe zur Verfügung; d.h. selbst wenn der Datumswert das Jahrhundert enthielte, gingen diese Informationen bei der Umsetzung verloren. Um die Jahrhundert-Angabe zu behalten, setzen Sie DF=I oder DF=L.

## Beispiele für DF-Parameter beim WRITE-Statements

Diese Beispiele gehen von DTFORM=G aus.

```
/* DF=S (default)
WRITE *DATX    /* Output has this format: dd.mm.yy
END

FORMAT DF=I
WRITE *DATX    /* Output has this format: ddmmyyyy
END
```

```

FORMAT DF=L
WRITE *DATX /* Output has this format: dd.mm.yyyy
END

```

## Beispiel für DF-Parameter beim MOVE-Statement

Dieses Beispiel geht von DTFORM=E aus.

```

DEFINE DATA LOCAL
  1 #DATE (D) INIT <D'31/12/2005'>
  1 #ALPHA (A10)
END-DEFINE
...
MOVE #DATE TO #ALPHA /* Result: #ALPHA contains 31/12/05
MOVE #DATE (DF=I) TO #ALPHA /* Result: #ALPHA contains 31122005
MOVE #DATE (DF=L) TO #ALPHA /* Result: #ALPHA contains 31/12/2005
...

```

## Beispiel für DF-Parameter beim STACK-Statement

Dieses Beispiel geht von DTFORM=I aus.

```

DEFINE DATA LOCAL
  1 #DATE (D) INIT <D'2005-12-31'>
  1 #ALPHA1(A10)
  1 #ALPHA2(A10)
  1 #ALPHA3(A10)
END-DEFINE
...
STACK TOP DATA #DATE (DF=S) #DATE (DF=I) #DATE (DF=L)
...
INPUT #ALPHA1 #ALPHA2 #ALPHA3
...
/* Result: #ALPHA1 contains 05-12-31
/*          #ALPHA2 contains 20051231
/*          #ALPHA3 contains 2005-12-31
...

```

## Beispiel für DF-Parameter beim INPUT-Statement

Dieses Beispiel geht von DTFORM=I aus.

```

DEFINE DATA LOCAL
  1 #DATE1 (D)
  1 #DATE2 (D)
  1 #DATE3 (D)
  1 #DATE4 (D)
END-DEFINE
...
INPUT #DATE1 (DF=S) /* Input must have this format: yy-mm-dd
      #DATE2 (DF=I) /* Input must have this format: yyyymmdd
      #DATE3 (DF=L) /* Input must have this format: yyyy-mm-dd
      #DATE4 /* Input must have this format: yy-mm-dd or yyyymmdd
...

```

## Datumsformat für Ausgabe — der DFOUT-Parameter

Der Session- bzw. Profilparameter DFOUT gilt nur für Datumsfelder in INPUT-, DISPLAY-, PRINT- und WRITE-Statements, für die keine Editiermaske angegeben ist und für die kein DF-Parameter gilt.

Bei Datumsfeldern, die mit einem INPUT-, DISPLAY-, PRINT- oder WRITE-Statement ausgegeben werden und für die weder eine Editiermaske angegeben noch ein DF-Parameter gesetzt ist, bestimmt der Profil/Session-Parameter DFOUT die Form, in der die Feldwerte angezeigt werden.

Mögliche DFOUT-Einstellungen sind:

<b>DFOUT=S</b>	Datumsvariablen werden mit 2-stelliger Jahreskomponente und mit durch den DTFORM-Parameter bestimmten Begrenzungszeichen angezeigt ( <i>yy-mm-dd</i> ).
<b>DFOUT=I</b>	Datumsvariablen werden mit mit 4-stelliger Jahreskomponente und ohne Begrenzungszeichen angezeigt ( <i>yyyymmdd</i> ).

Standardmäßig gilt DFOUT=S. Bei beiden DFOUT-Einstellungen wird die Reihenfolge der Tages-, Monats- und Jahreskomponenten in den Datumswerten durch den DTFORM-Parameter bestimmt.

Die Länge eines Datumsfeldes wird durch die DFOUT-Einstellung nicht beeinflusst, da jede der beiden Datumswert-Darstellungen in ein 8 Byte langes Feld passt.

Der DFOUT-Parameter kann im Natural-Parametermodul dynamisch beim Aufrufen von Natural oder mit dem Systemkommando GLOBALS gesetzt werden. Er wird zur Laufzeit ausgewertet.

### Beispiel:

Dieses Beispiel geht von DTFORM=I aus.

```
DEFINE DATA LOCAL
1 #DATE (D) INIT <D'2005-12-31'>
END-DEFINE
...
WRITE #DATE          /* Output if DFOUT=S is set ...: 05-12-31
                   /* Output if DFOUT=I is set ...: 20051231
WRITE #DATE (DF=L) /* Output (regardless of DFOUT): 2005-12-31
...
```

## Datumsformat für Stack — der DFSTACK-Parameter

Der Session- bzw. Profilparameter DFSTACK gilt nur für Datumsfelder, die in STACK-, FETCH- und RUN-Statements verwendet werden und für die kein DF-Parameter angegeben ist.

Der DFSTACK-Parameter bestimmt die Form, in der die Werte von Datumsvariablen mit einem STACK-, RUN- oder RUN-Statement auf dem Natural-Stack abgelegt werden.

Mögliche DFSTACK-Einstellungen sind:

<b>DFSTACK=S</b>	Datumsvariablen werden mit zweistelliger Jahreskomponente und mit durch den DTFORM-Parameter bestimmten Begrenzungszeichen auf dem Stack abgelegt ( <i>yy-mm-dd</i> ).
<b>DFSTACK=C</b>	Wie DFSTACK=S. Allerdings wird eine Änderung des Jahrhunderts zur Laufzeit abgefangen.
<b>DFSTACK=I</b>	Datumsvariablen werden mit 4-stelliger Jahreskomponente und ohne Begrenzungszeichen auf dem Stack abgelegt ( <i>yyyymmdd</i> ).

Standardmäßig gilt DFSTACK=S. Diese Einstellung bewirkt, dass die Jahrhundert-Informationen nicht mit abgelegt werden (und verlorengehen), wenn ein Datumswert auf dem Stack abgelegt wird.

Wenn dann der Wert vom Stack in eine andere Datumsvariable eingelesen wird, wird entweder als Jahrhundert das aktuelle Jahrhundert genommen oder das Jahrhundert durch die Einstellung des YSLW-Parameters (siehe unten) bestimmt. Das kann dazu führen, dass das Jahrhundert ein anderes ist als im ursprünglichen Datumswert, ohne dass Natural in diesem Fall einen Fehler ausgibt.

DFSTACK=C funktioniert insofern wie DFSTACK=S, dass ein Datumswert ohne Jahrhundert-Informationen auf dem Stack abgelegt wird. Wenn allerdings der Wert vom Stack gelesen wird und das ermittelte Jahrhundert nicht mit dem des ursprünglichen Datumswerts identisch ist (entweder aufgrund des YSLW-Parameters oder weil das ursprüngliche Jahrhundert nicht das aktuelle ist), gibt Natural einen Laufzeitfehler aus.

#### **Anmerkung:**

Dieser Laufzeitfehler wird bereits ausgegeben, sobald der Wert auf dem Stack abgelegt wird.

Mit DFSTACK=I können Sie einen Datumswert in einer Länge von 8 Bytes auf dem Stack ablegen, ohne dass die Jahrhundert-Informationen verlorengehen.

Der DFSTACK-Parameter kann im Natural-Parametermodul (bzw. der -Parameterdatei) dynamisch beim Aufrufen von Natural oder mit dem Systemkommando GLOBALS gesetzt werden. Er wird zur Laufzeit ausgewertet.

#### **Beispiel:**

Dieses Beispiel geht von DTFORM=I und YSLW=0 aus.

```
DEFINE DATA LOCAL
  1 #DATE (D) INIT <D'2005-12-31'>
  1 #ALPHA1(A8)
  1 #ALPHA2(A10)
END-DEFINE
...
STACK TOP DATA #DATE #DATE (DF=L)
...
INPUT #ALPHA1 #ALPHA2
...
/* Result if DFSTACK=S or =C is set: #ALPHA1 contains 05-12-31
/* Result if DFSTACK=I is set .....: #ALPHA1 contains 20051231
/* Result (regardless of DFSTACK) .: #ALPHA2 contains 2005-12-31
...
```

## Year Sliding Window — der YSLW-Parameter

Mit dem Profilparameter `YSLW` können Sie das Jahrhundert eines zweistelligen Jahr-Wertes bestimmen.

Der `YSLW`-Parameter kann im Natural-Parametermodul oder dynamisch beim Aufrufen von Natural gesetzt werden. Er wird zur Laufzeit ausgewertet, wenn ein alphanumerischer Datumswert mit einer zweistelligen Jahreskomponente in eine Datumsvariable übertragen wird. Dies betrifft Datumswerte, die:

- mit der mathematischen Funktion `VAL(field)` verwendet werden,
- mit der Option `IS(D)` in einer logischen Bedingung verwendet werden,
- vom Stack als Eingabedaten gelesen werden,
- in ein Feld als Eingabedaten eingegeben werden.

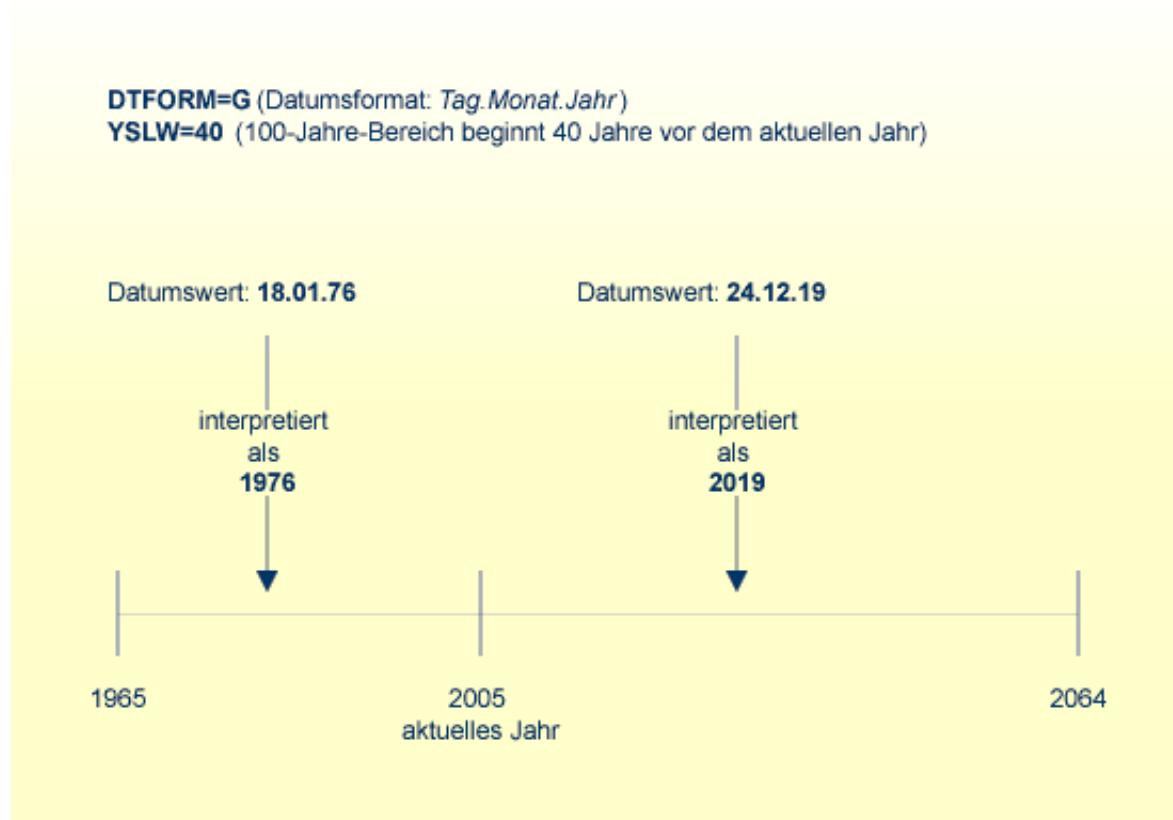
Der `YSLW`-Parameter bestimmt den Bereich von Jahren, der von einem sogenannten "Year Sliding Window" abgedeckt wird. Der "Sliding Window"-Mechanismus geht davon aus, dass ein Datum mit einem zweistelligen Jahr innerhalb eines "Fensters" von 100 Jahren liegt. Innerhalb dieser 100 Jahre kann jeder zweistellige Jahr-Wert eindeutig einem bestimmten Jahrhundert zugeordnet werden.

Mit dem `YSLW`-Parameter legen Sie fest, mit wievielen Jahren in der Vergangenheit der 100-Jahre-Bereich anfangen soll: das erste Jahr des Fensterbereichs ergibt sich aus dem aktuellen Jahr minus dem `YSLW`-Wert.

Mögliche Werte des `YSLW`-Parameters sind 0 bis 99. Der Standardwert ist `YSLW=0`, d.h. der "Sliding Window"-Mechanismus ist nicht aktiv; bei einem zweistelligen Jahr wird dann angenommen, dass es im aktuellen Jahrhundert liegt.

### Beispiel 1:

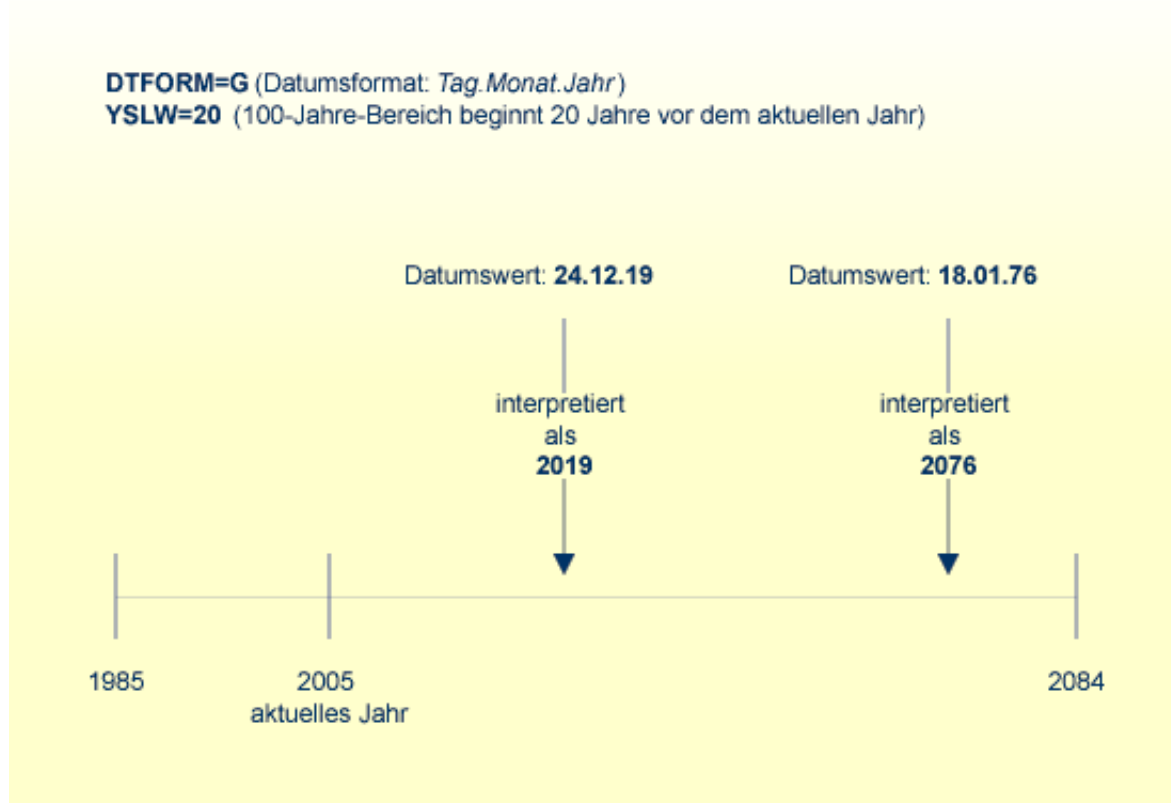
Wenn das aktuelle Jahr 2005 ist und Sie `YSLW=40` angeben, deckt das "Sliding Window" die Jahre 1965 bis 2064 ab. Ein zweistelliger Jahr-Wert `nn` von 65 bis 99 wird dementsprechend als `19nn` interpretiert, während ein zweistelliger Jahr-Wert `nn` von 00 bis 64 als `20nn` interpretiert wird.



### Beispiel 2:

Wenn das aktuelle Jahr 2005 ist und Sie YSLW=20 angeben, deckt das "Sliding Window" die Jahre 1985 bis 2084 ab. Ein zweistelliger Jahr-Wert *nn* von 85 bis 99 wird dementsprechend als 19*nn* interpretiert, während ein zweistelliger Jahr-Wert *nn* von 00 bis 84 als 20*nn* interpretiert wird.





## Kombinationen von DFSTACK und YSLW

Die folgenden Beispiele veranschaulichen die Wirkungen verschiedener Kombinationen der Parameter DFSTACK und YSLW.

### Anmerkung:

Alle Beispiele gehen von DTFORM=I aus.

### Beispiel 1:

Dieses Beispiel geht vom aktuellen Jahr 2005 und folgenden Parametereinstellungen aus:

DFSTACK=S (Standardeinstellung) und YSLW=20

```
DEFINE DATA LOCAL
  1 #DATE1 (D) INIT <D'1956-12-31'>
  1 #DATE2 (D)
END-DEFINE
...
STACK TOP DATA #DATE1 /* century information is lost (year 56 is stacked)
...
INPUT #DATE2           /* year sliding window determines 56 to be 2056
...
/* Result: #DATE2 contains 2056-12-31
   even if #DATE1 is set to <D'2156-12-31'>
```

In diesem Fall ist das "Year Sliding Window" unpassend gesetzt, so dass die Jahrhundert-Informationen sich (unbeabsichtigterweise) ändern.

### Beispiel 2:

Dieses Beispiel geht vom aktuellen Jahr 2005 und folgenden Parametereinstellungen aus:

DFSTACK=S (Standardeinstellung) und YSLW=60

```
DEFINE DATA LOCAL
  1 #DATE1 (D) INIT <D'1956-12-31'>
  1 #DATE2 (D)
END-DEFINE
...
STACK TOP DATA #DATE1 /* century information is lost (year 56 is stacked)
...
INPUT #DATE2           /* year sliding window determines 56 to be 1956
...
/* Result: #DATE2 contains 1956-12-31
           even if #DATE1 is set to <D'2056-12-31'>
```

In diesem Fall ist das "Year Sliding Window" passend gesetzt, so dass die ursprünglichen Jahrhundert-Informationen korrekt wiederhergestellt werden.

### Beispiel 3:

Dieses Beispiel geht vom aktuellen Jahr 2005 und folgenden Parametereinstellungen aus:

DFSTACK=C und YSLW=0 (Standardeinstellung)

```
DEFINE DATA LOCAL
  1 #DATE1 (D) INIT <D'1956-12-31'>
  1 #DATE2 (D)
END-DEFINE
...
STACK TOP DATA #DATE1 /* century information is lost (year 56 is stacked)
...
INPUT #DATE2           /* 56 is assumed to be in current century -> 1956
...
/* Result: RUNTIME ERROR (UNINTENDED CENTURY CHANGE)
```

In diesem Fall ändern sich (unbeabsichtigterweise) die Jahrhundert-Informationen. Allerdings wird diese Änderung durch die Parametereinstellung DFSTACK=C abgefangen.

### Beispiel 4:

Dieses Beispiel geht vom aktuellen Jahr 2005 und folgenden Parametereinstellungen aus:

DFSTACK=C und YSLW=60 (Standardeinstellung)

```

DEFINE DATA LOCAL
  1 #DATE1 (D) INIT <D'2056-12-31'>
  1 #DATE2 (D)
END-DEFINE
...
STACK TOP DATA #DATE1 /* century information is lost (year 56 is stacked)
...
INPUT #DATE2           /* year sliding window determines 56 to be 1956
...
/* Result: RUNTIME ERROR (UNINTENDED CENTURY CHANGE)

```

In diesem Fall ändern sich die Jahrhundert-Informationen aufgrund des "Year Sliding Window". Allerdings wird diese Änderung durch die Parametereinstellung `DFSTACK=C` abgefangen.

## Year Fixed Window

Informationen zu diesem Thema entnehmen Sie der Beschreibung des Profilparameters `YSLW`.

## Datumsformat für Standard-Seitenüberschriften — der DFTITLE-Parameter

Der Session- bzw. Profilparameter `DFTITLE` bestimmt die Form des Datums in einer Standard-Seitenüberschrift (wie sie mit einem `DISPLAY-`, `WRITE-` oder `PRINT-`Statement ausgegeben wird).

<b>DFTITLE=S</b>	Das Datum wird mit zweistelliger Jahreskomponente und Begrenzungszeichen ausgegeben ( <i>yy-mm-dd</i> ).
<b>DFTITLE=L</b>	Das Datum wird mit vierstelliger Jahreskomponente und Begrenzungszeichen ausgegeben ( <i>yyyy-mm-dd</i> ).
<b>DFTITLE=I</b>	Das Datum wird mit vierstelliger Jahreskomponente ohne Begrenzungszeichen ausgegeben ( <i>yyyymmdd</i> ).

Bei jeder dieser Ausgabeformen werden die Reihenfolge der Tages-, Monats- und Jahreskomponenten sowie die verwendeten Begrenzungszeichen durch den `DTFORM`-Parameter bestimmt.

Der `DFTITLE`-Parameter kann im Natural-Parametermodul dynamisch beim Aufrufen von Natural oder mit dem Systemkommando `GLOBALS` gesetzt werden. Er wird zur Laufzeit ausgewertet.

### Beispiel:

Dieses Beispiel geht von `DTFORM=I` aus.

```

WRITE 'HELLO'
END
/*
/* Date in page title if DFTITLE=S is set ...: 05-10-31
/* Date in page title if DFTITLE=L is set ...: 2005-10-31
/* Date in page title if DFTITLE=I is set ...: 20051031

```

**Anmerkung:**

Der DFTITLE-Parameter hat keine Auswirkungen auf benutzerdefinierte Seitenüberschriften, wie sie mit einem `WRITE TITLE`-Statement angegeben werden.