

# PROCESS PAGE

Dieses Kapitel behandelt folgende Themen:

- Funktion
  - Syntax 1 — PROCESS PAGE
  - Syntax 2 — PROCESS PAGE USING
  - Syntax 3 — PROCESS PAGE UPDATE
  - Syntax 4 — PROCESS PAGE MODAL
  - Beispiele
- 

## Funktion

Das `PROCESS PAGE`-Statement bildet eine allgemeine Schnittstellen-Beschreibung zu einer externen Rendering-Maschine, wie z.B. Natural for Ajax, wobei somit die interne Natural-Datendarstellung mit einer externen Datendarstellung verknüpft wird. Über diese Verknüpfung werden Daten und Ereignisse, aber keine Rendering-Informationen an und von eine/r externen browser-basierte/n Anwendung versandt.

Weitere Informationen entnehmen Sie der *Natural for Ajax*-Dokumentation.

## Syntax 1 — PROCESS PAGE

```
PROCESS PAGE [(parameter)] operand1
  [WITH PARAMETERS
    {[NAME] operand3 [VALUE] operand4 [(parameters)]} ...
  END-PARAMETERS]
[GIVING operand11]
```

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

Gehört zur Funktionsgruppe: *Bildschirmgenerierung für interaktive Verarbeitung*

### Syntax-Beschreibung — Syntax 1

Die Syntax 1 des `PROCESS PAGE`-Statements wird normalerweise nur in einem Natural-Adapter benutzt. Ein Adapter ist ein Natural-Objekt, das die Schnittstelle zwischen dem Natural-Anwendungscode und der Webseite bildet. Er wird automatisch von Natural for Ajax erstellt/aktualisiert, wenn das Layout gespeichert wird.

Operanden-Definitionstabelle:

Operand	Possible Structure		Possible Formats														Referencing Permitted	Dynamic Definition											
<i>operand1</i>	C	S					A	U																			yes	no	
<i>operand2</i>		S	A																							C	no	no	
<i>operand3</i>	C	S					A	U																			yes	no	
<i>operand4</i>	C	S	A				A	U	N	P	I	F	B	D	T	L											yes	yes	
<i>operand5</i>		S	A																							C	no	no	
<i>operand11</i>		S																										yes	yes

Syntax-Element-Beschreibung:

<b><i>parameter</i></b>	<p>Den Parameter CV können Sie in Klammern angeben, um eine oder mehrere der in <i>operand2</i> angegebenen Kontrollvariablen zu referenzieren:</p> <p>(CV=<i>operand2</i>)</p> <p>Siehe auch <i>Logische Bedingungen</i>, <i>MODIFIED-Option</i> im Leitfaden zur <i>Programmierung</i>.</p>	
<b><i>operand1</i></b>	<p>Enthält den Namen des externen Seiten-Layouts.</p>	
<b><i>operand2</i></b>	<p><i>operand2</i> enthält den Namen der Kontrollvariablen, muss Format C haben und entweder ein Skalar oder eine einzelne Array-Ausprägung sein.</p>	
<b><i>operand3</i></b>	<p>Enthält den/die Namen des bzw. der externen Datenfelder, in die oder aus denen <i>operand4</i> übertragen wird.</p>	
<b><i>operand4</i></b>	<p>Enthält den/die Namen des bzw. der externen Natural-Datenfelder, die übertragen werden.</p>	
<b><i>parameters</i></b>	<p>Unmittelbar nach <i>operand4</i> können Sie einen oder mehrere Parameter in Klammern angeben:</p>	
	EM	<p>Während des Datentransfers verwendete Editiermaske.</p> <p>Weitere Informationen siehe Session-Parameter EM in der <i>Parameter-Referenz</i>.</p>
	CV	<p>Den Parameter CV, können Sie angeben, um eine oder mehrere in <i>operand5</i> angegebene Kontrollvariable zu referenzieren:</p> <p>(CV=<i>operand5</i>)</p> <p>Siehe auch <i>Logische Bedingungen</i>, <i>MODIFIED-Option</i> im Leitfaden zur <i>Programmierung</i>.</p>

<p><b><i>operand5</i></b></p>	<p><i>operand5</i> enthält den Namen der Kontrollvariablen. Die Variable must Format C haben.</p> <p>Falls <i>operand4</i> ein Skalar-Ausdruck oder eine einzelne Array-Ausprägung ist, muss <i>operand5</i> folgendes sein:</p> <ul style="list-style-type: none"> <li>● ein Skalar-Ausdruck</li> <li>● oder eine einzelne Array-Ausprägung.</li> </ul> <p>Falls es sich bei <i>operand4</i> um den vollen Bereich eines Array der Dimension 1 handelt, muss <i>operand5</i> folgendes sein:</p> <ul style="list-style-type: none"> <li>● ein Skalar-Ausdruck</li> <li>● oder eine einzelne Array-Ausprägung</li> <li>● oder der volle Bereich eines Array der Dimension 1 mit der gleichen Größe.</li> </ul> <p>Falls es sich bei <i>operand4</i> um den vollen Bereich eines Array der Dimension 2 handelt, muss <i>operand5</i> folgendes sein:</p> <ul style="list-style-type: none"> <li>● ein Skalar-Ausdruck</li> <li>● oder eine einzelne Array-Ausprägung</li> <li>● oder der volle Bereich eines Array der Dimension 2 mit der gleichen Größe in beiden Dimensionen</li> <li>● oder der volle Bereich eines Array der Dimension 1 mit der gleichen Größe, die <i>operand4</i> in Dimension 2 hat.</li> </ul>
<p><b>GIVING</b> <b><i>operand11</i></b></p>	<p><b>GIVING-Klausel:</b></p> <p><i>operand11</i> enthält den Natural-Fehler, wenn die Anfrage nicht ausgeführt werden konnte.</p>

Beispiel eines Adapters, der vom Natural for Ajax erstellt wurde:

```
* PAGE1: PROTOTYPE          --- CREATED BY Natural for Ajax ---
* PROCESS PAGE USING 'XXXXXXX' WITH
* INFOPAGENAME RESULT YOURNAME
DEFINE DATA PARAMETER
1 INFOPAGENAME (U) DYNAMIC
1 RESULT (U) DYNAMIC
1 YOURNAME (U) DYNAMIC
END-DEFINE
*
PROCESS PAGE U'/njxdemos/helloworld' WITH
PARAMETERS
  NAME U'infopagename'
  VALUE INFOPAGENAME
  NAME U'result'
  VALUE RESULT
  NAME U'yourname'
  VALUE YOURNAME
```

**END-PARAMETERS**

```

*
* TODO: Copy to your calling program and implement.
/**( DEFINE EVENT HANDLER
* DECIDE ON FIRST *PAGE-EVENT
* VALUE U'nat:page.end'
* /* Page closed.
* IGNORE
* VALUE U'onHelloWorld'
* /* TODO: Implement event code.
* PROCESS PAGE UPDATE FULL
* NONE VALUE
* /* Unhandled events.
* PROCESS PAGE UPDATE
* END-DECIDE
/**) END-HANDLER
*
END

```

## Syntax 2 — PROCESS PAGE USING

**PROCESS PAGE USING** *operand6*

[ { **WITH** {*operand7*} ... } ]  
**NO PARAMETER**

[**GIVING** *operand11*]

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

Gehört zur Funktionsgruppe: *Bildschirmgenerierung für interaktive Verarbeitung*

### Syntax-Beschreibung — Syntax 2

Diese Syntax wird benutzt, um eine Rich-GUI-Eingabe/Ausgabeverarbeitung mittels eines Objekts des Typs Adapter auszuführen, das aus einem Seiten-Layout angelegt wurde, welches mit Natural for Ajax oder einem ähnlichen Tool erstellt wurde.

Operanden-Definitionstabelle:

Operand	Possible Structure		Possible Formats												Referencing Permitted	Dynamic Definition			
<i>operand6</i>	C	S					A											yes	no
<i>operand7</i>		S	A	G			A	U	N	P	I	F	B	D	T	L	C	yes	yes
<i>operand11</i>		S										I4						yes	yes

Syntax-Element-Beschreibung:

<b>USING</b> <i>operand6</i>	<p><b>Adapter-Name:</b></p> <p>Ruft eine Adapter-Definition auf, die vorher in einer Natural-Systemdatei gespeichert wurde. Siehe auch <i>Verarbeitung einer Rich GUI Page - Adapter</i> im <i>Leitfaden zur Programmierung</i>.</p> <p>Der Adapter-Name (<i>operand6</i>) kann eine 1 bis 8 Zeichen umfassende, alphanumerische Konstante oder Benutzervariable sein. Wenn eine Variable benutzt wird, muss sie vorher definiert worden sein.</p> <p>Der Adapter-Name kann ein Kaufmännisches Und (&amp;) enthalten; zur Ausführungszeit wird dieses Zeichen durch den aktuellen Wert der Natural-Systemvariablen *LANGUAGE ersetzt. Diese Funktion gibt es aus historischen Gründen. Wenn Sie mehrsprachige Adapter benötigen, machen Sie sich die Funktionalität des externen Rendering-Systems (zum Beispiel: Natural for Ajax) zunutze.</p> <p><b>Anmerkung:</b> Bei neuen Anwendungen muss die &amp;-Funktion nicht mehrsprachig sein. Seiten, die zum Beispiel mit Natural for Ajax gestaltet wurden, können mehrsprachige Informationen als Bestandteil des Layout-Designs aufnehmen. Siehe <i>Multi Language Management</i> in der <i>Natural for Ajax</i>-Dokumentation.</p>
<i>operand7</i>	<p><b>Feld-Spezifikation:</b></p> <p>Eine Liste der Datenbank-Felder und/oder Benutzervariablen, die alle vorher definiert sein müssen. Die Felder müssen in Anzahl, Reihenfolge, Format, Länge und (für Arrays) der Anzahl der Ausprägungen mit den Feldern im referenzierten Adapter übereinstimmen; sonst tritt ein Fehler auf.</p> <p>Wenn der Inhalt eines Datenbank-Feldes als Ergebnis der PROCESS PAGE-Verarbeitung geändert wird, wird nur der Wert geändert, wie er in der Data Area abgelegt ist. Um den Inhalt der Datenbank zu ändern, müssen die passenden UPDATE-/STORE-Statements für die Datenbank benutzt werden.</p> <p>Siehe <i>PROCESS PAGE USING</i> mit im Programm definierten Feldern.</p>
<b>NO PARAMETER</b>	Siehe <i>PROCESS PAGE USING</i> ohne Parameter-Liste.
<b>GIVING</b> <i>operand11</i>	<p><b>GIVING-Klausel:</b></p> <p><i>operand11</i> enthält den Natural-Fehler, wenn die Anfrage nicht ausgeführt werden konnte.</p> <p><b>Anmerkung:</b> Die GIVING-Klausel unterbricht die allgemeine Natural-Fehlerbehandlung, wenn ein Fehler auftritt, während das Adapter-Objekt aktiviert oder ausgeführt wird. Anstatt die Natural-Module zurückzuverfolgen, um eine ON ERROR-Klausel zu finden, wird der Natural-Fehlercode an eine Variable (<i>operand11</i>) übergeben, und die Ausführung wird mit dem nächsten Statement fortgesetzt.</p>

## PROCESS PAGE USING ohne Parameter-Liste

Die folgenden Anforderungen müssen erfüllt sein, wenn `PROCESS PAGE USING` ohne Parameter-Liste benutzt wird:

- Der Adapter-Name (*operand7*) muss als eine alphanumerische Konstante (bis zu 8 Zeichen) angegeben werden.
- Der auf diese Art benutzte Adapter muss vor der Kompilierung des Programms erstellt worden sein, welches den Adapter referenziert.
- Die Namen der zu verarbeitenden Felder werden dynamisch aus der Adapter-Quelldefinition zur Kompilierungszeit übernommen. Die sowohl im Programm als auch im Adapter verwendeten Feldnamen müssen identisch sein.
- Alle im `PROCESS PAGE`-Statement zu referenzierenden Felder müssen an diesem Punkt aufrufbar sein.
- Im Structured Mode müssen die Felder vorher definiert worden sein (Datenbank-Felder müssen für Verarbeitungsschleifen oder Views ordnungsgemäß referenziert werden).
- Wenn das Seiten-Layout geändert wird, müssen die den Adapter verwendenden Programme nicht neu katalogisiert werden. Wenn aber die Array-Strukturen oder Namen, Formate/Längen der Felder geändert werden, oder wenn Felder zum Adapter hinzugefügt oder aus ihm gelöscht werden, müssen die den Adapter benutzenden Programme neu katalogisiert werden.
- Der Adapter-Quellcode muss zur Programm-Kompilierung zur Verfügung stehen; sonst kann das `PROCESS PAGE USING`-Statement nicht kompiliert werden.

### Anmerkung:

Wenn Sie das Programm kompilieren möchten, auch wenn der Adapter noch nicht zur Verfügung steht, geben Sie `NO PARAMETER` an. Das `PROCESS PAGE USING`-Statement kann dann kompiliert werden, auch wenn der Adapter noch nicht verfügbar ist.

## PROCESS PAGE USING mit im Programm definierten Feldern

Wenn Sie die Namen der Felder angeben, die innerhalb des Programms (*operand7*) verarbeitet werden sollen, ist es möglich, dass Sie es so einrichten können, dass die Namen der Felder im Programm sich von den Namen der Felder im Adapter unterscheiden.

Die Reihenfolge der Felder im Programm muss mit der Reihenfolge im Adapter übereinstimmen. Wenn Sie Natural-Maps als Adapter-Objekte benutzen, beachten Sie, dass der Map-Editor die Felder so sortiert, wie in der Map angegeben, und zwar in alphabetischer Reihenfolge nach Feldnamen. Weitere Informationen siehe die Beschreibung des Map Editor in der *Editors*-Dokumentation.

Das Programmzeilenkommando `.I (adaptername)` kann benutzt werden, um ein vollständiges `PROCESS PAGE USING`-Statement mit einer Parameter-Liste zu erhalten, die aus den im angegebenen Adapter definierten Feldern abstammt.

Wenn das Layout des Adapters geändert wird, braucht das den Adapter verwendende Programm nicht neu katalogisiert zu werden. Wenn aber Feldnamen, Feldformate/längen oder Array-Strukturen im Adapter geändert werden, oder wenn Felder zum Adapter hinzugefügt oder aus ihm gelöscht werden, dann muss das Programm neu katalogisiert werden.

Zur Ausführungszeit findet eine Überprüfung statt, um sicherzustellen, dass das im Programm angegebene Format und die im Programm angegebene Länge der Felder mit den Feldern übereinstimmt, die im Adapter spezifiziert sind. Falls die beiden Layouts nicht miteinander übereinstimmen, wird eine Fehlermeldung erzeugt.

## Syntax 3 — PROCESS PAGE UPDATE

```
PROCESS PAGE UPDATE [FULL] [event-option]
[GIVING operand11]
```

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

Gehört zur Funktionsgruppe: *Bildschirmgenerierung für interaktive Verarbeitung*

### Syntax-Beschreibung - Syntax 3

Das PROCESS PAGE UPDATE-Statement wird benutzt, um zu einem vorangegangenen PROCESS PAGE-Statement zurückzukehren und es neu auszuführen. Es wird im Allgemeinen benutzt, um von der Ereignisverarbeitung zurückzukehren, da die Dateneingabe-Verarbeitung des vorangegangenen PROCESS PAGE-Statements unvollständig war.

#### Anmerkung:

Es kann kein INPUT-, WRITE-, PRINT- oder DISPLAY-Statement zwischen einem PROCESS PAGE- und seinem entsprechenden PROCESS PAGE UPDATE-Statement ausgeführt werden.

Wenn es erst einmal ausgeführt ist, repositioniert das PROCESS PAGE UPDATE-Statement den Programm-Status bezüglich Subroutine, Sonderbedingungen und die Schleifenverarbeitung, wie sie existierten, als das PROCESS PAGE-Statement ausgeführt wurde (solange der Status des PROCESS PAGE-Statements noch aktiv ist). Wenn die Schleife nach der Ausführung des PROCESS PAGE-Statements initialisiert wurde und sich das PROCESS PAGE UPDATE-Statement innerhalb dieser Schleife befindet, wird die Schleife unterbrochen und dann neu gestartet, nachdem das PROCESS PAGE-Statement als infolge eines PROCESS PAGE UPDATE-Statements neu verarbeitet worden ist.

Wenn eine Hierarchie von Subroutinen nach der Ausführung des PROCESS PAGE-Statements aufgerufen wurde und wenn das PROCESS PAGE UPDATE-Statement innerhalb einer Subroutine ausgeführt wird, verfolgt Natural automatisch alle Subroutinen zurück und setzt den Programm-Status auf den des PROCESS PAGE-Statements zurück.

Es ist aber nicht möglich, es so einzurichten, dass ein PROCESS PAGE-Statement in einer Schleife, einer Subroutine oder einem speziellen Bedingungsblock positioniert wird, und dann das PROCESS PAGE UPDATE-Statement auszuführen, wenn der Status, unter dem das PROCESS PAGE-Statement ausgeführt wurde, bereits beendet worden ist. Eine Fehlermeldung wird erzeugt und die Programmausführung beendet, wenn eine solche Fehlerbedingung festgestellt wird.

Operanden-Definitionstabelle:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
<i>operand11</i>	S	I4	yes	yes

Syntax-Element-Beschreibung:

<b>FULL</b>	<p>Wenn Sie die FULL-Option in einem PROCESS PAGE UPDATE-Statement angeben, wird das entsprechende PROCESS PAGE-Statement vollständig neu ausgeführt:</p> <ul style="list-style-type: none"> <li>• Mit einem normalen PROCESS PAGE UPDATE-Statement (ohne FULL-Option) wird der Inhalt von Variablen, die zwischen dem PROCESS PAGE- und dem PROCESS PAGE UPDATE-Statement geändert wurden, nicht angezeigt; das heisst, alle Variablen auf dem Bildschirm zeigen den Inhalt an, den sie hatten, als das PROCESS PAGE-Statement ursprünglich ausgeführt wurde.</li> <li>• Mit einem PROCESS PAGE UPDATE FULL-Statement werden alle Änderungen, die nach der anfänglichen Ausführung des PROCESS PAGE-Statements erfolgt sind, auf das PROCESS PAGE-Statement angewandt, wenn es neu ausgeführt wird; das heisst, alle Variablen auf dem Bildschirm enthalten die Werte, die sie hatten, als das PROCESS PAGE UPDATE-Statement ausgeführt wurde.</li> </ul>
<i>event-option</i>	<p><b>EVENT-Option:</b></p> <p>Siehe <i>EVENT-Option</i> weiter unten.</p>
<b>GIVING</b> ( <i>operand11</i> )	<p><b>GIVING-Klausel:</b></p> <p><i>operand11</i> enthält den Natural-Fehler, wenn die Anfrage nicht durchgeführt werden konnte.</p>

Beispiel eines Benutzerprogrammfragments:

```
PROCESS PAGE USING "HELLOW-A"
*
/*( DEFINE EVENT HANDLER
DECIDE ON FIRST *PAGE-EVENT
VALUE U'nat:page.end'
  /* Page closed.
  IGNORE
VALUE U'onHelloWorld'
  COMPRESS "HELLO WORLD" YOURNAME INTO RESULT
  PROCESS PAGE UPDATE FULL
NONE VALUE
  /* Unhandled events.
  PROCESS PAGE UPDATE
END-DECIDE
/*) END-HANDLER
```



## EVENT-Option

```

AND SEND EVENT operand8
  [WITH PARAMETERS
    {[NAME] operand9 [VALUE] operand10 [ { (EM=value) } ]}]...
  END-PARAMETERS]
  
```

Mit dieser Option können Sie das externe E/A-System veranlassen, Sonderfunktionen zu starten. Diese Funktionen sind Bestandteil des externen E/A-Systems, oder implementieren Sonderfunktionen im Hinblick auf die Ausgabeverarbeitung, wie z.B. Fokus setzen, Meldungsfelder anzeigen usw.

Operanden-Definitionstabelle:

Operand	Possible Structure			Possible Formats														Referencing Permitted	Dynamic Definition	
<i>operand8</i>	C	S				A	U												yes	no
<i>operand9</i>	C	S				A	U												yes	no
<i>operand10</i>	C	S	A			A	U	N	P	I	F	B	D	T	L				yes	yes

Syntax-Element-Beschreibung:

<b><i>operand8</i></b>	<b>Vom externen E/A-System angefordertes Ereignis:</b>  Abhängig von der Implementierung des externen E/A-Systems sind Ereignisse verfügbar, siehe <i>Sending Events to the User Interface</i> in der <i>Natural for Ajax</i> -Dokumentation.
<b><i>operand9</i></b>	<b>Externer Datenfeld-Name:</b>  <i>operand9</i> enthält den externen Namen der Datenfelder, in die/aus denen <i>operand10</i> übertragen wird.
<b><i>operand10</i></b>	<b>Natural-Datenfelder:</b>  <i>operand10</i> enthält die Natural-Datenfelder, die übertragen werden.
<b>EM=</b>	<b>Editiermaske:</b>  Bei der Datenübertragung verwendete Editiermaske.  Einzelheiten zu Editiermasken siehe Session-Parameter <i>EM</i> in der <i>Parameter-Referenz</i> .

## Syntax 4 — PROCESS PAGE MODAL

<b>PROCESS PAGE MODAL</b> <i>statement ...</i> <b>END-PROCESS</b>
---

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

Verwandte Statements: `ESCAPE` | `PROCESS PAGE`

Gehört zur Funktionsgruppe:

- Schleifenverarbeitung
- Bildschirmgenerierung für interaktive Verarbeitung

## Syntax-Beschreibung — Syntax 4

Das `PROCESS PAGE MODAL`-Statement wird benutzt, um einen Verarbeitungsblock zu initialisieren und die Anzeigedauer eines modalen Rich-GUI-Fensters zu steuern.

### Anmerkung:

Das `PROCESS PAGE MODAL`-Statement ist im Stapelbetrieb nicht gültig.

Wenn gerade der `PROCESS PAGE MODAL`-Statementblock abgearbeitet wird, erscheinen zuerst Daten vom Report 0, die noch nicht angezeigt worden sind.

Die Systemvariable `*PAGE-LEVEL` wird inkrementiert, und das Öffnen einer modalen Seite wird vorbereitet. Das physische Öffnen der modalen Seite wird mit dem nächsten Statement `PROCESS PAGE USING 'adapter' WITH` ausgeführt.

### Anmerkung:

Zwischen einem `PROCESS PAGE MODAL`-Statement und seinem entsprechenden `END-PROCESS`-Statement kann kein sich auf Report 0 beziehendes `PRINT`-, `WRITE`-, `INPUT`- oder `DISPLAY`-Statement ausgeführt werden.

Beim Verlassen des `PROCESS PAGE MODAL`-Statementblocks werden folgende Aktionen ausgeführt:

- Wurde eine modale Seite für diese Ebene geöffnet, wird die modale Seite geschlossen;
- die Systemvariable `*PAGE-LEVEL` wird dekrementiert, und die Systemvariable `*PAGE-EVENT` wird auf den Wert zurückgesetzt, den sie hatte, bevor mit der Abarbeitung des Statement-Block begonnen wurde.

Syntax-Element-Beschreibung:

<i>statement</i>	Anstelle eines Statements müssen Sie in Abhängigkeit von der Situation eines oder mehrere passende Statements angeben. Wenn Sie kein spezifisches Statement angeben möchten, können Sie das <code>IGNORE</code> -Statement angeben.
<b>END-PROCESS</b>	Das für Natural reservierte Wort <code>END-PROCESS</code> muss benutzt werden, um das <code>PROCESS PAGE MODAL</code> -Statement zu beenden.

**Beispiel:**

```
* Name: First Demo/Open modal!
*
PROCESS PAGE USING "EMPTY-A"
*
/*( DEFINE EVENT HANDLER
DECIDE ON FIRST *PAGE-EVENT
  VALUE U'nat:page.end', U'onClose'
    /* Page closed.
    IGNORE
  VALUE U'onNextLevel'
    PROCESS PAGE MODAL
      FETCH RETURN "EMPTY-P"
    END-PROCESS
  PROCESS PAGE UPDATE
  NONE VALUE
  PROCESS PAGE UPDATE
END-DECIDE
/*) END-HANDLER
END
```

**Beispiele**

Weitere Beispiele zur Verwendung des PROCESS PAGE-Statements sind in der Library SYSEXNJX enthalten.