

# Verarbeitung von Anwendungsfehlern

Dieser Abschnitt beschreibt die zwei grundlegenden Verfahren, die Natural für die Behandlung von Anwendungsfehlern bietet: Standardverarbeitung und anwendungsspezifische Verarbeitung. Darüber hinaus beschreibt es, auf welche Weise die anwendungsspezifische Verarbeitung von Fehlern ermöglicht werden kann: durch das Kodieren eines `ON ERROR`-Statement-Blocks innerhalb eines Programmierobjekts oder durch den Einsatz eines separaten Fehlertransaktionsprogramms.

Schließlich enthält dieser Abschnitt noch eine Übersicht über die Natural-Funktionalität, mit der Sie die Fehlerverarbeitung durch Natural konfigurieren, Informationen über einen Fehler abrufen oder einen Anwendungsfehler verarbeiten oder bereinigen können.

Dieses Kapitel behandelt folgende Themen:

- Natural's Standard-Fehlerverarbeitung
- Anwendungsspezifische Fehlerverarbeitung
- Verwendung eines `ON ERROR`-Statement-Blocks
- Verwendung eines Fehlertransaktionsprogramm
- Funktionalität für die Fehlerverarbeitung

Informationen zur Behandlung von Fehlern in einer Natural RPC-Umgebung siehe *Handling Errors in the Natural Remote Procedure Call*-Dokumentation.

---

## Natural's Standard-Fehlerverarbeitung

Wenn in einer Natural-Anwendung ein Fehler auftritt, geht Natural standardmäßig folgendermaßen vor:

1. Natural beendet die Ausführung des zurzeit laufenden Anwendungsobjekts;
2. Natural gibt eine Fehlermeldung aus;
3. Natural kehrt zum Kommandoingabe-Modus zurück.

"Kommandoingabe-Modus" bedeutet, dass in Abhängigkeit von Ihrer jeweiligen Natural-Konfiguration das Natural-Hauptmenü, die `NEXT`-Zeile oder ein benutzerdefiniertes Einstiegsmenü erscheinen kann.

Die angezeigte Fehlermeldung enthält die Natural-Fehlernummer, den zugehörigen Meldungstext sowie das betroffene Natural-Objekt und die Nummer der Zeile, in der der Fehler aufgetreten ist.

Da die Ausführung des Anwendungsobjekts beendet wird, kann der Status von anhängigen Datenbanktransaktionen von Maßnahmen betroffen sein, die durch die Einstellung der Profilparameter `ETEOP` und `ETIO` bedingt sind. Falls Natural (infolge der Einstellungen dieser Parameter) kein `END TRANSACTION`-Statement ausgegeben hat, dann wird bei der Rückkehr in den Kommandoingabe-Modus ein `BACKOUT TRANSACTION`-Statement ausgegeben.

## Anwendungsspezifische Fehlerverarbeitung

Wenn die Standard-Fehlerverarbeitung nicht den Erfordernissen Ihrer Anwendung entspricht, können Sie die Verarbeitung anwendungsspezifisch anpassen. Mögliche Gründe hierfür können beispielsweise sein:

- Informationen zum Fehler sollen zwecks weiterer Untersuchung durch den Anwendungsentwickler gespeichert werden.
- Die Ausführung der Anwendung soll, falls möglich, nach der Fehlerbehebung fortgesetzt werden.
- Es ist eine spezifische Transaktionsbehandlung nötig.

Da nach Auftreten eines Fehlers die Ausführung des betroffenen Anwendungsobjekts beendet wird, kann der Status von anhängigen Datenbanktransaktionen von Maßnahmen betroffen sein, die durch die der Profilparameter ETEOP und ETIO ausgelöst werden. Deshalb muss die weitere Transaktionsbehandlung (mittels `END TRANSACTION-` oder `BACKOUT TRANSACTION-`Statement) über die Fehlerverarbeitung der Anwendung erfolgen.

Um eine anwendungsspezifische Fehlerverarbeitung zu ermöglichen, haben Sie folgende Möglichkeiten:

- Sie können innerhalb eines Programmierobjekts einen `ON ERROR-`Statement-Block kodieren.
- Sie können ein separates Fehlertransaktionsprogramm verwenden.

Diese Möglichkeiten werden in den folgenden Abschnitten behandelt.

## Verwendung eines `ON ERROR-`Statement-Blocks

Sie können das `ON ERROR-`Statement verwenden, um zur Ausführungszeit auftretende Fehler in einer Anwendung an der Stelle abzufangen, an der ein Fehler auftritt.

Aus einem solchen `ON ERROR-`Statement-Block heraus kann die Ausführung der Anwendung auf der aktuellen Ebene oder auf einer übergeordneten Ebene wieder aufgenommen werden.

Außerdem können Sie ein `ON ERROR-`Statement in mehreren Objekten einer Anwendung angeben, um Fehler, die auf untergeordneten Ebenen aufgetreten sind, zu verarbeiten. Auf diese Weise können Sie die Fehlerverarbeitung exakt auf die Erfordernisse Ihrer Anwendung zuschneiden.

### Verlassen eines `ON ERROR-`Statement-Blocks

Um einen `ON ERROR-`Statement-Block zu verlassen, können Sie eines der folgenden Statements angeben:

- `RETRY`

Die Ausführung der Anwendung wird auf der aktuellen Ebene wieder aufgenommen.

- `ESCAPE ROUTINE`

Es wird davon ausgegangen, dass die Fehlerverarbeitung abgeschlossen ist, und die Ausführung der Anwendung wird auf der übergeordneten Ebene wieder aufgenommen.

- `FETCH`

Es wird davon ausgegangen, dass die Fehlerverarbeitung abgeschlossen ist, und das beim `FETCH`-Statement angegebene Programm wird ausgeführt.

`STOP`

Natural stoppt die Ausführung des betroffenen Programms, beendet die Anwendung und kehrt zum Kommandoingabe-Modus zurück.

- `TERMINATE`

Die Ausführung der Natural-Anwendung wird gestoppt, und die Natural-Session wird beendet.

### Fehlerverarbeitungsregeln

- Wird die Ausführung des `ON ERROR`-Statement-Blocks nicht durch eines der oben genannten Statements beendet, dann wird der Fehler an das Natural-Objekt auf der übergeordneten Ebene durchgereicht, damit er durch einen dort vorhandenen `ON ERROR`-Statement-Block verarbeitet wird.
- Falls keines der Objekte auf einer der übergeordneten Ebenen einen `ON ERROR`-Statement-Block enthält, falls aber ein Fehlertransaktionsprogramm (wie im folgenden Abschnitt beschrieben) angegeben ist, erhält dieses Fehlertransaktionsprogramm die Kontrolle.
- Falls keines der Objekte auf einer der übergeordneten Ebenen einen `ON ERROR`-Statement-Block enthält und falls dort kein Fehlertransaktionsprogramm angegeben ist, dann greift die Standard-Fehlerverarbeitung von Natural wie oben beschrieben.

## Verwendung eines Fehlertransaktionsprogramm

Sie können an den folgenden Stellen ein Fehlertransaktionsprogramm angeben:

- Im Profilparameter `ETA`.
- Im Natural Security Library Profile, falls Natural Security installiert ist; siehe *Components of a Library Profile* in der *Natural Security*-Dokumentation.
- Innerhalb eines Natural-Objekts, indem Sie dort mittels eines `ASSIGN`-, `COMPUTE`- oder `MOVE`-Statements den Namen des Fehlertransaktionsprogramms der Systemvariablen `*ERROR-TA` als Wert zuordnen.

Wenn Sie während der Natural-Session den Namen eines Fehlertransaktionsprogramms der Systemvariablen `*ERROR-TA` zuweisen, dann wird durch diese Zuweisung ein mittels Profilparameter `ETA` angegebenes Fehlertransaktionsprogramm ersetzt. Aber ganz gleich, ob Sie den Profilparameter `ETA` verwenden oder der Systemvariablen `*ERROR-TA` einen Wert zuweisen, die Namen von Fehlertransaktionsprogramm werden nicht gespeichert und werden von Natural nicht für die verschiedenen Ebenen der Aufruf-Hierarchie wieder hergestellt. Darum wird, wenn Sie den Namen des Programms in einem Natural-Objekt der Systemvariablen `*ERROR-TA` zuweisen, dieses Programm aufgerufen, um jeden Fehler zu verarbeiten, der nach dieser Zuweisung in der aktuellen Natural-Session auftritt.

Einerseits wird also, wenn Sie ein Fehlertransaktionsprogramm mit dem Profilparameter `ETA` angeben, eine Fehlertransaktion für die ganze Natural-Session definiert, ohne dass die Notwendigkeit besteht, innerhalb von Natural-Objekten Einzelzuweisungen vorzunehmen. Andererseits bietet das Verfahren, ein Programm der Systemvariablen `*ERROR-TA` zuzuweisen, mehr Flexibilität und gestattet es Ihnen zum Beispiel, in verschiedenen Zweigen der Anwendung verschiedene Fehlertransaktionsprogramme zu benutzen.

Wenn die Systemvariable `*ERROR-TA` zurückgesetzt (leer) wird, dann wird wie oben beschrieben Natural's Standard-Fehlerverarbeitung durchgeführt.

Wenn ein Fehlertransaktionsprogramm angegeben ist und ein Anwendungsfehler auftritt, wird die Ausführung der Anwendung beendet. Das angegebene Fehlertransaktionsprogramm erhält dann die Kontrolle, um eine der folgenden Maßnahmen auszuführen:

- Analyse des Fehlers
- Protokollieren der Fehlerinformationen
- Beenden der Natural-Session
- Fortsetzen der Anwendungsausführung durch Aufrufen eines Programms mittels `FETCH`-Statement.

Da das Fehlertransaktionsprogramm die Kontrolle so erhält, als wenn es im Kommandoeingabe-Modus eingegeben worden wäre, ist es nicht möglich, die Ausführung der Anwendung in einem der Natural-Objekte, die zum Zeitpunkt des Auftretens des Fehlers aktiv waren, wieder aufzunehmen.

Wenn der Natural-Profilparameter `SYNERR` auf `ON` gesetzt ist und zur Laufzeit ein Syntaxfehler auftritt, erhält das Fehlertransaktionsprogramm auch die Kontrolle.

Fehlertransaktionsprogramme müssen sich in einer Library befinden, in der Sie zurzeit angemeldet sind, oder in einer aktuellen Steplib Library.

Wenn ein Fehler auftritt, führt Natural ein `STACK TOP DATA`-Statement aus und legt folgende Informationen oben auf dem Stack ab:

Feldinhalt	Format/Länge
Natural-Fehlernummer	N4, wenn der Session-Parameter SG auf OFF gesetzt ist; N5, wenn SG=ON
Nummer der Zeile, in welcher der Fehler aufgetreten ist	N4
Status Code:	A1
C Kommandoverarbeitungsfehler (Die Zeilennummer ist dann 0.)	
L Logon-Verarbeitungsfehler (Die Zeilennummer ist dann 0.)	
O Objekt-(Ausführungs-)Zeitfehler	
R Fehler auf einem Remote Server (in Verbindung mit dem Natural RPC)	
S Nicht korrigierbarer Syntaxfehler	
Name des Natural-Objekts, in dem der Fehler aufgetreten ist	A8
Nummer der Programmebene, auf der der Fehler aufgetreten ist	N2

Wenn der Natural-Profilparameter SYNERR auf ON gesetzt ist und zur Laufzeit ein Syntaxfehler auftritt, dann ist die Nummer der Programmebene Null und die folgenden Informationen werden zusätzlich auf den Stack gelegt:

Feldinhalt	Format/Länge
Position des den Fehler verursachenden Bestandteils in der Source-Zeile	N3
Länge des verursachenden Bestandteils	N3

Diese Informationen können in einem Fehlertransaktionsprogramm mittels eines INPUT-Statements abgefragt werden.

Beispiel:

```

DEFINE DATA LOCAL
1 #ERROR-NR          (N5)
1 #LINE             (N4)
1 #STATUS-CODE      (A1)
1 #PROGRAM          (A8)
1 #LEVEL           (N2)
1 #POSITION-IN-LINE (N3)
1 #LENGTH-OF-ERRTOKEN (N3)
END-DEFINE
IF *DATA > 6 THEN      /* SYNERR = ON and a syntax error occurred
INPUT
  #ERROR-NR
  #LINE
  #STATUS-CODE
  #PROGRAM

```

```

#LEVEL
#POSITION-IN-LINE
#LENGTH-OF-ERRTOKEN
ELSE
  INPUT          /* other error
#ERROR-NR
#LINE
#STATUS-CODE
#PROGRAM
#LEVEL
END-IF
WRITE #STATUS-CODE
* DECIDE ON FIRST VALUE OF STATUS-CODE
* ... /* process error
* END-DECIDE
END

```

Einige der oben auf dem Stack abgelegten Informationen entsprechen den Inhalten von Systemvariablen, welche in einem ON ERROR-Statement-Block zur Verfügung stehen.

Feldinhalt	Entsprechende Systemvariable im ON ERROR-Statement-Block
Natural-Fehlernummer	*ERROR-NR
Nummer der Zeile, in welcher der Fehler aufgetreten ist	*ERROR-LINE
Name des Natural-Objekts, in dem der Fehler aufgetreten ist	*PROGRAM
Nummer der Programmebene, auf der der Fehler aufgetreten ist	*LEVEL

### Regeln unter Natural Security

Wenn Natural Security installiert ist, gelten zusätzliche Regeln für die Verarbeitung von Fehlern, die beim Anmelden auftreten. Weitere Informationen siehe *Transactions* in der *Natural Security*-Dokumentation.

## Funktionalität für die Fehlerverarbeitung

Natural bietet Ihnen umfangreiche Funktionalität, die Sie im Zusammenhang mit der Fehlerverarbeitung verwenden können. Sie können damit

- das Verhalten von Natural bei der Fehlerverarbeitung konfigurieren,
- Informationen über aufgetretene Fehler abrufen,
- Unterstützung bei der Verarbeitung dieser Fehler anfordern,
- Unterstützung bei der Bereinigung von Anwendungsfehlern erhalten.

Diese Funktionalität umfasst spezielle:

- Profilparameter
- Systemvariablen
- Terminalkommandos
- Systemkommandos
- Anwendungsprogrammierschnittstellen (APIs)

### Profilparameter

Folgende Profilparameter beeinflussen das Verhalten von Natural im Fehlerfall:

Profilparameter	Zweck
CPCVERR	Konvertierungsfehler
DBGERR	Automatischer Start des Debuggers bei einem Laufzeitfehler
ETA	Fehlertransaktionsprogramm
ETEOP	Ausgabe eines END TRANSACTION-Statements bei einem End of Program
ETIO	Ausgabe eines END TRANSACTION-Statements bei Terminal-Eingabe/Ausgabe
RCFIND	Handhabung des Response Code 113 beim FIND-Statement
RCGET	Handhabung des Response Code 113 beim GET-Statement
SYNERR	Überwachung von Syntaxfehlern

### Systemvariablen

Die folgenden anwendungsbezogenen Systemvariablen können verwendet werden, um einen Fehler zu lokalisieren oder um den Namen des Programms zu erhalten bzw. anzugeben, das die Kontrolle im Fehlerfall erhalten soll:

Systemvariable	Inhalt
*ERROR-LINE	Source-Zeilenummer des Statements, das den Fehler verursacht hat. Siehe Beispiel 1.
*ERROR-NR	Fehlernummer des Fehlers, der eine abzuarbeitende ON ERROR-Bedingung verursacht hat.
*ERROR-TA	Name des Programms, das die Kontrolle im Fehlerfall erhalten soll. Siehe Beispiel 2.
*LEVEL	Nummer der Ebene des Natural-Objekts, auf der der Fehler aufgetreten ist.
*LIBRARY-ID	Name der Library, in welcher der Benutzer zurzeit angemeldet ist.
*PROGRAM	Name des Natural-Objekts, das zurzeit ausgeführt wird. Siehe Beispiel 1.

## Beispiel 1:

```

...
/*
ON ERROR
  IF *ERROR-NR = 3009 THEN
    WRITE 'LAST TRANSACTION NOT SUCCESSFUL'
      / 'HIT ENTER TO RESTART PROGRAM'
    FETCH 'ONEEX1'
  END-IF
  WRITE 'ERROR' *ERROR-NR 'OCCURRED IN PROGRAM' *PROGRAM
    'AT LINE' *ERROR-LINE
  FETCH 'MENU'
END-ERROR
/*
...

```

## Beispiel 2:

```

...
*ERROR-TA := 'ERRORTA1'
/* from now on, program ERRORTA1 will be invoked
/* to process application errors
...
MOVE 'ERRORTA2' TO *ERROR-TA
/* change error transaction program to ERRORTA2
...

```

Weitere Informationen zu diesen Systemvariablen finden Sie in den entsprechenden Abschnitten der *Systemvariablen*-Dokumentation.

## Terminalkommandos

Das folgende Terminalkommando beeinflusst das Verhalten von Natural im Fehlerfall:

Terminalkommando	Zweck
%E=	Fehlerbehandlung ein-/ausschalten

## Systemkommandos

Die folgenden Systemkommandos liefern zusätzliche Informationen über eine Fehlersituation bzw. dienen zum Aufrufen von Utilities zur Fehlerbereinigung bei oder zum Protokollieren von Datenbankaufrufen:

Systemkommando	Zweck
LASTMSG	Anzeige von zusätzlichen Informationen zu der zuletzt aufgetretenen Fehlersituation.
RPCERR	Anzeige der letzten Natural-Fehlernummer und -meldung, wenn sie sich auf den Natural RPC bezog.
TECH	Anzeige von technischen und sonstigen Informationen über Ihre Natural-Session, zum Beispiel Informationen über den zuletzt aufgetretenen Fehler.
TEST	Ruft den Debugger auf.
TEST DBLOG	Ruft die DBLOG-Utility auf, die zum Protokollieren von Datenbankaufrufen verwendet wird.

## Anwendungsprogrammierschnittstellen

Die folgenden Anwendungsprogrammierschnittstellen (APIs) stehen generell zur Verfügung, um zusätzliche Informationen über eine Fehlersituation abzurufen oder um eine Fehlertransaktion einzurichten.

API	Zweck
USR0040N	Art des letzten Fehlers abfragen
USR1016N	Fehlerebene bei Copycode zeigen
USR1037N	Informationen über Natural ABEND-Daten
USR1041N	Beispielprogramm für eine Fehlertransaktion (*ERROR-TA)
USR2001N	Informationen über letzten Fehler auslesen
USR2006N	Ausführliche Informationen zur Meldung holen
USR2007N	Informationen zum Standard-RPC-Server setzen/abrufen
USR2010N	Informationen über Datenbankfehler zeigen
USR2026N	Technische Informationen holen (TECH)
USR2030N	Dynamische Fehler-Teile :1;... lesen
USR2034N	System- bzw. Benutzer-Fehlermeldungstext von FNAT bzw. FUSER auslesen
USR3320N	Fehler-Kurztext auf FNAT bzw. FUSER suchen
USR4214N	Information über Programmebene zeigen

Weitere Informationen siehe *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

Für SQL-Aufrufe stehen die folgenden Anwendungsprogrammierschnittstellen zur Verfügung:

API	Zweck
NDBERR	Liefert Diagnose-Informationen über den zuletzt ausgeführten SQL-Aufruf.
NDBNOERR	Unterbindet die Fehlerbehandlung durch Natural im Falle von Fehlern, die durch den nächsten SQL-Aufruf verursacht werden.

Weitere Informationen siehe :

- *Interface Subprograms* in der *Natural for DB2*-Dokumentation.
- *Interface Subprograms* in der *Natural for SQL/DS*-Dokumentation.
- *Interface Subprograms* in der *Natural SQL Gateway*-Dokumentation.