

Alten Debugger benutzen

Der alte Debugger kann erscheinen, wenn eine alte Version von Natural auf dem Entwicklungs-Server installiert ist; siehe *Allgemeine Informationen*.

Dieses Kapitel behandelt die folgenden Themen:

- Natural-Objekte vorbereiten
 - Debugger starten
 - Debugger verlassen
 - Debugger bedienen
 - Source-Fenster des Debuggers
 - Funktionsleiste Watchvariables
 - Funktionsleiste Variables
 - Funktionsleiste Watchpoints and Breakpoints
-

Natural-Objekte vorbereiten

Um den vollständigen Funktionsbereich des Natural Debuggers auszunutzen, müssen Sie den folgenden Natural-Profilparameter entweder dynamisch oder in Ihrer Natural-Parameterdatei angeben:

SYMGEN ist auf "ON" gesetzt

Wenn ein Objekt katalogisiert oder in Source- und Objekt-Form gespeichert wird und SYMGEN auf "ON" gesetzt ist, wird eine Symboltabelle als Bestandteil des generierten Programms generiert. Da diese Tabelle die Informationen enthält, die sich auf die für dieses Objekt aktiven Variablen beziehen, können Variablen nicht aufgerufen werden, ohne dass SYMGEN angegeben wird, obwohl es noch möglich ist, das Objekt auszutesten.

Debugger starten

Der Debugger kann nur auf in Source- und Objekt-Form gespeicherte oder katalogisierte Natural-Programme und Dialoge angewandt werden.

▶ Debugger starten

- Geben Sie das folgende Natural-Kommando ein:

```
DEBUG objektname
```

wobei *objektname* der Name des Natural-Objekts ist, das Sie austesten möchten.

In der Titelleiste erscheint einer der folgenden Einträge:

- **[break]**
Wenn "[break]" in der Titelleiste erscheint, hat der Debugger die Kontrolle.
- **[waiting]**
Wenn "[waiting]" in der Titelleiste erscheint, hat die gerade ausgetestete Natural-Anwendung die Kontrolle.

Wenn der Remote-Debugger im Betriebssystem Windows aktiv wird, erscheinen die folgenden Informationen in der Titelleiste: "Debugging Remote Natural Client (*\\knotenname::benutzername::prozess-id*)", wobei *knotenname* der Name des Computers ist, auf dem Natural läuft, *benutzername* der Name des Natural-Benutzers ist, und *prozess-id* die Natural Prozess-ID ist.

Das Debugger-Fenster enthält ein Child-Fenster mit einem Source-Listing des angegebenen Objekts, das ausgetestet werden soll.

In Verbindung mit dieser Objekt-Source werden die folgenden Informationen mittels Funktionsleisten angezeigt:

Funktionsleiste	Funktion
Breakpoints and Watchpoints	Diese Funktionsleiste besteht aus zwei Register-Bereichen. Ein Bereich verwaltet Breakpoints, während der andere Watchpoints verwaltet.
Variables	Diese Funktionsleiste zeigt die aktiven Variablen und ihren eigentlichen Inhalt an. Diese Variablen werden unter den folgenden Kategorien angezeigt: Locals, Globals, Systems, AIVs und Contexts (Lokale Variablen, Globale Variablen, Systemvariablen, AIV-Variablen und Kontext-Variablen).
Watchvariables	Diese Funktionsleiste zeigt die vom Benutzer ausgewählten Variablen einer Kategorie an, die in der Funktionsleiste Variables zur Verfügung steht.

Die einzelnen Funktionsleisten sind in allen Einzelheiten im Rest dieses Abschnitts beschrieben.

Debugger verlassen

Sie können den Debugger von einem beliebigen Punkt innerhalb einer Anwendung verlassen, indem Sie entweder Exit (siehe unten) oder die entsprechende Schaltfläche in der Symbolleiste wählen.

Der Debugger wird auch beendet, wenn die Anwendung ohne einen Fehler beendet wird, dann wird der Trace-Cursor (Cursor zur Ablaufverfolgung) in die zuletzt ausgeführte Sourcecode-Zeile platziert.

Im Falle eines Fehlers wird die entsprechende Source im Source-Fenster angezeigt, und der Trace-Cursor (Cursor zur Ablaufverfolgung) wird in die Zeile platziert, die den Fehler verursachte. Es erscheint ein Meldungsfenster mit der passenden Fehlermeldung und der Auswahlmöglichkeit, entweder mit der Verarbeitung fortzufahren oder die Debugging-Session zu beenden. Das Fortsetzen der Debugging-Session kann nützlich sein, wenn zum Beispiel Folgendes vorliegt:

- ihre Anwendung enthält eine beliebige Fehlerverarbeitung (einschließlich Fehlertransaktionen);
- Sie möchten beliebige Variablen anzeigen, bevor Sie Ihre Debugging-Session beenden.

Wenn Sie den Debugger verlassen, werden Ihre Breakpoint-, Watchpoint- und Watchvariablen-Einstellungen automatisch gespeichert, und zwar zusammen mit den Fenster- und Symbolleisten-Einstellungen. Alle diese Einstellungen werden das nächste Mal wiederhergestellt, wenn Sie den Debugger nochmals aufrufen.

Anmerkung:

Wenn Sie im Falle einer Remote-Entwicklung den Debugger in einem Remote-System verlassen, wird die Programmausführung fortgesetzt, aber die Debugging-Kontrolle der Programmausführung wird angehalten.

Befehl Exit (Verlassen)

Mit **Exit** verlassen Sie die Debugging-Session, und die Kontrolle wird an Natural zurückgegeben. Der Befehl **Exit** (Verlassen) steht im ersten Menü im Hauptfenster für jede der fünf Debugger-Hauptfunktionen zur Verfügung.

Debugger bedienen

Bevor das Source-Fenster des Debuggers und andere Haupt-Funktionen detailliert behandelt werden, bietet Ihnen dieser Abschnitt allgemeine Informationen zum Debugger.

- Windows und Menüs
- Schaltflächen in der Symbolleiste
- Tastenkombinationen
- Watchpoints und Breakpoints
- Debugging-Session neu starten

Windows und Menüs

Der Debugger bietet verschiedene Fenster, Funktionsleisten, Symbolleisten und Menüs.

Menübefehle, von denen angenommen wird, dass sie sehr oft benutzt werden, stehen auch als Schaltflächen in den betreffenden Symbolleisten zur Verfügung.

Anstatt die Menüs zu benutzen, können Sie Schaltflächen in der Symbolleiste wählen oder Tastenkombinationen benutzen.

Im Gegensatz zu Natural selbst:

- hat der Debugger keine Kommandozeile.
- enthält das **Tools**-Menü des Debuggers die folgenden Optionen:
 - **Customize** (Anpassen) - diese Option ermöglicht es Ihnen, das Aussehen Ihres Menüs und Ihrer Symbolleiste zu ändern, sowie Tastenkombinationen für häufig benutzte Befehle zu definieren;
 - **Fonts** (Schriftarten) - diese Option ermöglicht es Ihnen, die Schriftart des Source-Fensters zu ändern;
 - **Warning messages** (Warnungsmeldungen) - diese ermöglichen es Ihnen, festzulegen, ob Warnungsmeldungen zu fehlendem Sourcecode oder symbolische Informationen angezeigt werden sollen oder nicht. Eine Meldung, die Sie darüber informiert, ob der gerade angezeigte Sourcecode neuer als das betreffende generierte Programm ist und auch davon beeinträchtigt ist.

Schaltflächen in der Symbolleiste

Die Symbolleisten bieten Ihnen einen schnellen Zugriff auf häufig benutzte Befehle. Um eine Kurzbeschreibung eines Befehls anzuzeigen, platzieren Sie den Mauszeiger auf die betreffende Schaltfläche. Die Beschreibung erscheint in der Statusleiste am unteren Rand des Hauptfensters des Debuggers. Wenn ein Befehl gerade nicht angewandt werden kann, wird die Schaltfläche ausgeschaltet.

Tastenkombinationen

Eine weitere Möglichkeit, einen Debugger-Befehl auszuführen, besteht darin, eine entsprechende Tastenkombination über die Tastatur einzugeben. Standardmäßig sind die folgenden Tastenkombinationen definiert:

Menü	Tastenkombination	Funktion
File (Datei)	STRG+O	Open (Öffnen)
Edit (Editieren)	STRG+F	Find (Suchen)
	F3	Find Next (Suchen Nächstes)
Debug (Debug)	F4	Close (Schließen)
	F5	Go (Weiter)
	F6	Step Over (Objekt mit Step Over überspringen)
	F7	Step In (Objekt mit Step In schrittweise ausführen)
	STRG+F7	Step Out (Objekt mit Step Out verlassen)
	STRG+F6	Run To Cursor (Zum Cursor)
	Alt+*	Show Trace Position (Position zur Ablaufverfolgung zeigen)
	F9	Toggle Breakpoint (Breakpoint ein/ausschalten)
Variables (Variablen)	STRG+M	Modify Variable (Variable ändern)
	STRG+D	Display Variable (Variable anzeigen)
	STRG+V	Add to Watch-Variablen (Zu Watch-Variablen hinzufügen)
	STRG+W	Add to Watchpoints (Zu Watchpoints hinzufügen)

Watchpoints und Breakpoints

Zwei Typen von Einträgen können in einem Programm zu Debugging-Zwecken definiert werden: Watchpoints und Breakpoints. Jeder Watchpoint oder Breakpoint wird in seiner entsprechenden Funktionsleiste angezeigt. Für jeden Watchpoint wird ein Name zugewiesen, der dem Namen der Variable entspricht, zu dem er gehört.

Jeder Watchpoint oder Breakpoint kann jederzeit während einer Debugging-Session über die entsprechenden Kontrollkästchen aktiviert oder deaktiviert werden.

Für jeden Watchpoint oder Breakpoint wird die Anzahl der Events (Ereignisse) angegeben, die sich jedes Mal erhöht, wenn der Debug-Eintrag überschritten wird. Die Anzahl der Ausführungen eines Debug-Eintrags kann aber auf zwei Arten eingeschränkt werden:

1. Eine Anzahl von übersprungenen Einträgen kann angegeben werden, bevor der Watchpoint oder Breakpoint ausgeführt wird. Der Debug-Eintrag wird dann ignoriert, bis die Anzahl der Events (Ereignisse) größer als die Anzahl der angegebenen übersprungenen Einträge ist.
2. Eine maximale Anzahl von Ausführungen kann angegeben werden, so dass der Breakpoint oder Watchpoint ignoriert wird, sobald die Anzahl der gefundenen Events (Ereignisse) die angegebene Anzahl der Ausführungen überschreitet.

Debugging-Session neu starten

Wenn Sie Ihre Debugging-Session neu starten, repositioniert der Debugger zum Anfang der Anwendung, während alle Ihre aktuellen Einstellungen (zum Beispiel Watchpoints oder Breakpoints) beibehalten werden, und alle Zähler sowie die Aufruf-Historie neu initialisiert werden. Folglich ist der erneute Start einer Debugging-Session nützlich, wenn Sie Ihre Anwendung neu starten möchten, ohne die für das Debugging relevanten Einstellungen erneut angeben zu müssen. Sie können Ihre Debugging-Session von einem beliebigen Punkt innerhalb der Anwendung neu starten, indem Sie entweder den Befehl **Restart** oder das entsprechende Symbol in der Symbolleiste wählen. Der Befehl **Restart** ist im Debug-Menü verfügbar.

Anmerkung:

Wenn Sie eine Debugging-Session in einer Remote-Umgebung starten, steht der Befehl **Restart** nicht zur Verfügung, und wenn Sie einen DCOM- oder RPC-Server austeste, werden über den Befehl **Restart** die aufgerufene Methode oder das aufgerufene Subprogramm neu gestartet.

Source-Fenster des Debuggers

Wenn der Debugger aufgerufen wird, erhält er die Kontrolle des angegebenen Natural-Objekts und zeigt die betreffende Source im Source-Fenster an. Wenn die Source nicht zur Verfügung steht, bleibt das Fenster leer. Der Trace-Cursor (Cursor zur Ablaufverfolgung) wird in die erste ausführbare Sourcecode-Zeile platziert.

Wenn ein Benutzer ein neues Objekt öffnet, oder wenn ein Watchpoint oder Breakpoint innerhalb eines anderen Objekts außer dem gerade aktiven Objekt als Treffer (Hit Count) angezeigt wird, wird ein neues Source-Fenster geöffnet, das die Source dieses neuen Objekts anzeigt.

Die folgenden Themen sind im Folgenden erörtert:

- Debug-Menü
- Variables-Menü
- Dialogfelder
- Variablen auswählen
- Text im Source-Fenster markieren
- Display (Anzeigen)
- Modify (Ändern)
- Quick Watch (Schnell überwachen)
- Add Watch (Watch-Variablen hinzufügen)
- Add Watchpoint (Watchpoint hinzufügen)
- File Menu (Menü Datei)

- Edit-Menü

Debug-Menü

Die folgenden Befehle des **Debug**-Menüs stehen in Zusammenhang mit dem Source-Fenster zur Verfügung:

Step Into (Objekt mit Step Into schrittweise ausführen)

Wenn Sie den Befehl **Step Into** wählen, wird der nächste Programmschritt ausgeführt, und der Trace-Cursor (Cursor zur Ablaufverfolgung) wird in die betreffende Sourcecode-Zeile platziert.

Wenn in dieser Sourcecode-Zeile ein weiteres Natural-Objekt aufgerufen oder aufgenommen wird, führt der Debugger in diesem Objekt den nächsten Programmschritt aus.

Step Over (Objekt mit Step Over überspringen)

Wenn Sie den Befehl **Step Over** wählen, wird der nächste Programmschritt ausgeführt, und der Trace-Cursor (Cursor zur Ablaufverfolgung) wird in die betreffende Sourcecode-Zeile platziert. Dieses Mal überspringt der Debugger allerdings ein aufgerufenes oder aufgenommenes Natural-Objekt, hält aber an, wenn dieses Objekt Watchpoints oder Breakpoints enthält.

Step Out (Objekt mit Step Out verlassen)

Wenn Sie den Befehl **Step Out** wählen, kehrt der Debugger zur vorherigen Programmebene zurück, hält aber an, wenn er einen Watchpoint oder Breakpoint vorfindet, bevor diese vorherige Ebene erreicht ist.

Animated Step Into (Schrittweises Ausführen des Programms in Objekt)

Wenn Sie den Befehl **Animated Step Into** wählen, wird das Programm bis zum Ende automatisch Schritt für Schritt ausgeführt. Der Debugger greift auf ein aufgerufenes oder aufgenommenes Natural-Objekt zu.

Animated Step Over (Schrittweises Ausführen des Programms und Objekt überspringen)

Wenn Sie den Befehl **Animated Step Over** wählen, wird das Programm bis zum Ende automatisch Schritt für Schritt ausgeführt. Der Debugger überspringt dabei ein aufgerufenes oder aufgenommenes Natural-Objekt; wenn ein Watchpoint oder Breakpoint gesetzt ist, springt er zu der betreffenden Statement-Zeile und fährt mit der automatischen Abarbeitung fort.

Go (Weiter)

Wenn Sie den Befehl **Go** wählen, wird das Programm bis zum nächsten aktiven Watchpoint oder Breakpoint ausgeführt, und der Trace-Cursor (Cursor zur Ablaufverfolgung) wird in der entsprechenden Sourcecode-Zeile platziert.

Go Until Next Event (Weiter bis zum nächsten Event)

Wenn Sie den Befehl **Go Until Next Event** wählen, dann hat dies dieselben Auswirkungen wie der Befehl **Go** in einer nicht ereignisgesteuerten Anwendung. In einer ereignisgesteuerten Anwendung wird das Objekt aber ausgeführt, bis der nächste Event an die Anwendung gesandt wird; es hält an, wenn ein aktiver Watchpoint oder Breakpoint auftritt, bevor der nächste Event gesandt wird.

Run to Cursor (Weiter zum Cursor)

Wenn Sie den Befehl **Run to Cursor** wählen, wird das Programm ausgeführt, bis die Source-Zeile an der aktuellen Cursor-Position erreicht ist.

Show Trace Position (Position zur Ablaufverfolgung zeigen)

Wenn Sie den Befehl **Show Trace Position** wählen, wird der aktuelle Trace-Cursor (Cursor zur Ablaufverfolgung) angezeigt.

Toggle Breakpoint (Breakpoint ein/ausschalten)

Wenn Sie den Befehl **Toggle Breakpoint** (Breakpoint ein/ausschalten) wählen, wird ein Breakpoint für die aktuelle Trace-Position (Position zur Ablaufverfolgung) zur Funktionsleiste für Breakpoints hinzugefügt. Wenn ein Breakpoint für diese Cursor-Position bereits vorhanden ist, wird er von der Funktionsleiste für Breakpoints entfernt.

Calls (Aufrufe)

Das Untermenü **Calls** (Aufrufe) bietet eine Liste (Historie) mit den zuletzt aufgerufenen Natural-Objekten, einschließlich Copycodes und Inline-Subroutinen. Bis zu 20 Objekte können aufgelistet werden; das zuletzt aufgerufene Objekt erscheint oben in der Liste.

Die Objekt-Liste besteht aus den folgenden Informationen:

- Die Programmebene des aufgerufenen Objekts ohne Copycodes und Inline-Subroutinen.
- Die Programmebene des aufgerufenen Objekts mit Copycodes und Inline-Subroutinen.
- Der Name des aufgerufenen Objekts.
- Der Typ des aufgerufenen Objekts.
- Die Event- und Kontroll-Handle des abzuarbeitenden Event-Handlers (nur bei ereignisgesteuerten Anwendungen).

Die Statusleiste am unteren Rand des Hauptfensters des Debuggers zeigt zusätzliche Informationen zum aufgerufenen Objekt an:

- Der Name des aufrufenden Objekts:

"Natural" wird als aufrufendes Objekt angezeigt, wenn das aufgerufene Objekt das Start-Programm der Anwendung oder ein Programm ist, das vom Natural-Stack aktiviert wurde (einschließlich Fehlertransaktionsprogrammen und Programmen, die von einem RUN-Statement innerhalb der Anwendung aktiviert wurden).

- Die Sourcecode-Zeile, in der das Objekt aufgerufen wurde:

Wenn Sie ein Objekt (außer Natural) aus der Liste wählen, wird die Source des aufrufenden Programms in der Mitte des Source-Fensters angezeigt, wobei der Cursor am Anfang der Zeile platziert wird, in der der Aufruf erfolgte.

Variables-Menü

Das **Variables**-Menü dient dazu, um:

- Den Inhalt von ausgewählten Variablen anzuzeigen.
- Den Inhalt von ausgewählten Variablen zu ändern.
- Den Inhalt der Variable an der aktuellen Trace-Position (Position zur Ablaufverfolgung) per Quick Watch schnell zu überwachen.
- Variablen zur Funktionsleiste für Watch-Variablen hinzufügen.
- Variablen zur Funktionsleiste für Watchpoints hinzufügen.

Dialogfelder

Wenn Sie den Befehl **Display** (Anzeigen), **Modify** (Ändern), **Add Watch** (Watch hinzufügen) oder **Add Watchpoint** (Watchpoint hinzufügen) wählen, erscheint ein Dialogfeld, in dem eine Liste aller lokalen Variablen, globalen Variablen, AIV-Variablen oder Systemvariablen angezeigt wird, die im aktuellen Debugging-Kontext aktiv sind. Die folgenden Kontrollelemente sind Bestandteil dieses Dialogfelds:

- Das Textfeld **Variable**, das die gerade ausgewählte Variable zeigt.
- Das Feld **Line Reference** or **Context ID** (Zeilenreferenz oder Kontext-ID), das die Sourcecode-Zeilenummer der Variable oder Kontext-Variable zeigt, die im Textfeld **Variable** gerade enthalten ist.

Das Feld **Line Reference** (Zeilenreferenz) wird nur angezeigt, wenn die Zeilenreferenz erforderlich ist, um die Auswahl der Variablen eindeutig zu gestalten. Dies ist der Fall, wenn:

- die Variable zu einer Map gehört; dann enthält das Feld die Sourcecode-Zeilenummer des betreffenden, vom Map-Editor generierten RULEVAR-Syntaxelements;
- die Variable entweder eine Datenbank-Variable (nur Reporting Mode) oder eine der folgenden Variablen ist: *ISN, *COUNTER, *NUMBER; dann enthält das Feld die Sourcecode-Zeilenummer der betreffenden Datenbank- Schleife oder des entsprechenden Datenzugriffs-Statements;
- die Variable im Reporting Mode definiert ist, aber ohne ein DEFINE DATA-Statement.

Das Feld **Context ID** (Kontext-ID) wird nur angezeigt, wenn die Variable eine Kontext-Variable ist; dann enthält das Feld die "ctx-Id" (Kontext-ID).

- Das Listenfeld **History List** (Historien-Liste) - nur bei **Display**-Befehl - das die zuletzt ausgewählten Variablen (bis zu 20) enthält, die einen FIFO-Mechanismus (first-in first-out) nutzen.

Die Historien-Liste unterstützt Sie dabei, eine Variable schnell zu lokalisieren, die bereits vorher ausgewählt war. Variablen können von der Historien-Liste genauso ausgewählt werden wie von der Variablen-Liste.

- Das Listenfeld **Variable**, das die betreffende Variablen-Liste enthält.

Variablen auswählen

Wenn Sie eine Variable in der Variablen-Liste eines Dialogfeldes wählen, erscheint sie im entsprechenden **Variable**-Textfeld.

Wenn Sie eine Variable in der Variablen-Liste eines Dialogfeldes wählen, wird ein weiteres Dialogfeld angezeigt (außer beim **Watch**-Befehl).

Wenn Sie eine Array- oder Variablen-Gruppe wählen:

- die einzelnen Array- oder Gruppen-Elemente werden im zweiten Dialogfeld angezeigt (Anzeigen),
- das Array wird im zweiten Dialogfeld zum Ändern angezeigt; Gruppen können nicht geändert werden (Ändern),
- eine entsprechende Fehlermeldung wird angezeigt (Watchpoint).

Sie können auch eine Variable wählen, indem Sie sie zuerst direkt im Source-Fenster markieren und dann entsprechend den Befehl **Display** (Anzeigen), **Modify** (Ändern), **Add Watch** (Watch hinzufügen) oder **Add Watchpoint** (Watchpoint hinzufügen) auswählen. Dann zeigt das Textfeld **Variable** des betreffenden Dialogfeldes genau das Stück Sourcecode an, das Sie markiert haben, welches dann geändert werden kann.

Text im Source-Fenster markieren

Im Source-Fenster können Sie Variablen oder Zeichenketten entweder mit der Maus oder der Tastatur zur Auswahl markieren.

Wenn Sie Text mit der Maus markieren, stellen Sie den Mauszeiger auf das erste auszuwählende Zeichen, ziehen Sie den Zeiger zum letzten Zeichen, das Sie auswählen möchten, und lassen Sie die Maustaste wieder los. Um eine Auswahl abubrechen, treffen Sie diese Wahl irgendwo im Dokument.

Wenn Sie die Tastatur benutzen, um Text zu markieren, werden Cursor-Bewegungstasten benutzt. Stellen Sie den Cursor zuerst auf ein Zeichen, indem Sie eine Pfeiltaste benutzen, drücken Sie dann die UMSCHALT-Taste, halten Sie sie gedrückt, und benutzen Sie die folgenden Tasten zur Textauswahl:

- den NACH-LINKS, um den Bereich links von Ihrer Cursor-Position zu markieren,
- den NACH-RECHTS, um den Bereich rechts von Ihrer Cursor-Position zu markieren,
- die ENDE-Taste, um den Bereich bis zum Ende der Sourcecode-Zeile zu markieren,
- die POS1-Taste, um den Bereich bis zum Anfang der Sourcecode-Zeile zu markieren.

Display (Anzeigen)

Mit diesem Befehl kann eine Variable von der Liste im Dialogfeld zur Anzeige ausgewählt werden, und zwar zusammen mit ihrem aktuellen Inhalt in einem zweiten **Display Variable**-Dialogfeld (Dialogfeld Variable anzeigen), wo Sie zwischen alphanumerischer und binärer Darstellung des Variablenwertes wählen können.

Wenn Sie ein Array, eine Handle-Variable oder eine Gruppe von Variablen auswählen, werden die einzelnen Elemente und ihre Werte im zweiten Dialogfeld aufgelistet. Bei Arrays wird ein beliebiger Variablenindex-Ausdruck ausgewertet.

Die Element-Liste kann erweitert oder verkürzt werden, indem Sie die Schaltfläche **Expand/Contract** (Erweitern/Verkürzen) wählen. Immer wenn die Anzahl der Arrays, Gruppen oder Dialog-Elemente in der Liste ein bestimmtes Anzeige-Limit überschreitet, erscheint eine More-Zeile (Mehr-Zeile), die zur Anzeige weiterer Objekte benutzt werden kann. Als Alternative dazu kann auch der Befehl **Expand** (Erweitern) benutzt werden.

Eine Variable, ein Array oder eine Gruppe von Variablen kann auch zur Anzeige im Dialogfeld **Display Variable** (Variable anzeigen) ausgewählt werden, indem Sie die Variable, das Array oder die Gruppe von Variablen mit der linken Maustaste direkt im Source-Fenster wählen.

Modify (Ändern)

Mit diesem Befehl kann eine Variable von der Liste im Dialogfeld zur Anzeige ausgewählt werden, und zwar zusammen mit ihrem aktuellen Wert in einem zweiten **Modify Variable**-Dialogfeld (Variable ändern), wo ihr Wert geändert werden kann.

Wenn Sie eine Systemvariable ändern möchten, werden nur Systemvariablen im ersten Dialogfeld angezeigt, die geändert werden können.

Wenn Sie ein Array ändern möchten, wird nur sein Name angezeigt, aber es erscheinen keine Werte im zweiten Dialogfeld. Der Wert, den Sie eingeben, ist dann gültig für alle Array-Elemente.

Gruppen von Variablen können nicht zum Ändern ausgewählt werden.

Quick Watch (Schnell überwachen)

Bei diesem Befehl erscheint ein Dialogfeld, das den Inhalt der Variablen an der aktuellen Cursor-Position anzeigt.

Add Watch (Watch-Variablen hinzufügen)

Bei diesem Befehl können Variablen, Arrays oder Gruppen von Variablen von der Liste im Dialogfeld ausgewählt werden, um sie zur Funktionsleiste Watchvariables (Watch-Variablen) hinzuzufügen.

Add Watchpoint (Watchpoint hinzufügen)

Mit diesem Befehl können einzelne Variablen und einzelne Gruppen oder Array-Elemente von der Liste im Dialogfeld zur Definition eines Watchpoints in einem zweiten Set Watchpoint-Dialogfeld (Watchpoint setzen) ausgewählt werden; Arrays und Gruppen von Variablen können nicht ausgewählt werden.

Das zweite **Set Watchpoint**-Dialogfeld zeigt den Namen des Watchpoints an (der dem Namen der ausgewählten Variablen entspricht), und zwar zusammen mit seiner Zeilenreferenz (wenn vorhanden), und die Namen des betreffenden Natural-Objekts und der entsprechenden Natural-Library.

Anmerkung:

Bei Systemvariablen ist der betreffende Watchpoint nicht an eine spezifische Library und ein spezifisches Objekt angebunden; deshalb sind der Objekt- und Library-Name stets SYSTEM.

Um einen Watchpoint zu definieren, geben Sie die folgenden Elemente in den entsprechenden Feldern an:

- den Status des Watchpoints,
- eine Bedingung für den zu aktivierenden Watchpoint (optional),
- die Anzahl der Sprünge vor der Ausführung des Watchpoints,
- die maximale Anzahl der Ausführungen des Watchpoints.

File Menu (Menü Datei)

Die folgenden Befehle des **File**-Menüs sind in Verbindung mit dem Source-Fenster verfügbar:

Open (Öffnen)

Mit dem Befehl **Open** (Öffnen) können Sie ein weiteres Source-Programm angeben, das in das Source-Fenster geladen werden soll. Es erscheint das Dialogfeld **Open Source** (Source öffnen), in dem Sie den Programm-Namen und den passenden Library-Namen angeben, wenn das Programm nicht in der aktuellen Library (Voreinstellung) enthalten ist.

Sie können auch eine in das Dialogfeld **Open Source** (Source öffnen) zu stellende Zeichenkette auswählen, indem Sie ihren Namen im Source-Fenster markieren und dann den Befehl **Open** (Öffnen) wählen.

Close (Schließen)

Mit dem Befehl **Close** (Schließen) schließen Sie das gerade aktive Source-Fenster. Wenn das Source-Fenster, das Sie gerade schließen, die Trace-Bar (Ablaufverfolgungsleiste) enthält, wird das Fenster auf Symbolgröße verkleinert.

Exit (Verlassen)

Mit dem Befehl **Exit** (Verlassen) verlassen Sie den Debugger und beenden die aktuelle Programmausführung.

Edit-Menü

Die folgenden Befehle des **Edit**-Menüs sind in Verbindung mit dem Source-Fenster verfügbar:

Find (Suchen)

Mit dem Befehl **Find** (Suchen) können Sie im aktiven Fenster nach oben oder unten absuchen, um jede Ausprägung eines angegebenen Wortes oder einer spezifizierten Zeichenkette zu lokalisieren.

Es erscheint das Dialogfeld **Find** (Suchen), in dem Sie den in dem Textfeld Find zu lokalisierenden Text eingeben können. Außerdem können Sie die Optionen **Match Upper/Lower Case** (Groß/Kleinschreibung) und **Whole Words Only** (Nur ganze Wörter) ein- oder ausschalten.

Wenn Sie den Text gefunden haben, wird die erste Ausprägung des angegebenen Textes hell hervorgehoben (ausgewählt), währenddessen Sie in einer Meldung darüber informiert werden, wenn der Text nicht gefunden werden konnte.

Mit der Option **Match Upper/Lower Case** (Groß/Kleinschreibung) können Sie angeben, ob Sie mit der Find-Operation nach einer genauen Übereinstimmung (ON) oder nur nach denselben Zeichen suchen möchten, ungeachtet der Groß- oder Kleinschreibung (OFF).

Mit der Option **Whole Words Only** (Nur ganze Wörter) können Sie angeben, ob Sie mit der Find-Operation nach Ausprägungen suchen möchten, die nur aus ganzen Wörtern bestehen, und nicht Bestandteil einer Zeichenkette sind (ON), oder aber nach allen Ausprägungen des angegebenen Textes, ganzen Wörtern und Teilen einer Zeichenkette (OFF).

Um die Such-Richtung zu ändern, wählen Sie die Schaltfläche **Up** (Nach oben), um in Aufwärtsrichtung zum Anfang des Textes hin zu suchen, oder wählen Sie die Schaltfläche **Down** (Nach unten), um in Abwärtsrichtung zum Ende des Textes hin zu suchen; **Down** ist die Voreinstellung.

Wenn die Find-Operation nicht am Anfang (oder Ende) des Textes beginnt, und der angegebene Text nicht gefunden werden kann, erscheint ein Dialog. Sie können **Yes** wählen, um die Such-Operation am Anfang (oder Ende) des Textes fortzusetzen, oder **No**, um die Suche abzubrechen.

Sie können auch eine Zeichenkette auswählen, die in das Textfeld **Find** (Suchen) gestellt werden soll, indem Sie sie direkt im Source-Fenster markieren und dann den Befehl **Find** (Suchen) wählen.

Find Next (Suchen Nächstes)

Mit diesem Befehl können Sie die vorherige Such-Operation wiederholen und die nächste Ausprägung des mit dem Befehl **Find** angegebenen Textes lokalisieren.

Funktionsleiste Watchvariables

Die Funktionsleiste Watchvariables dient in erster Linie dazu, vorher ausgewählte Variablen zur genaueren und permanenten Beobachtung ihres Inhalts anzuzeigen.

Sie bietet ein Kontextmenü, das entweder die Befehle anzeigt, die in Verbindung mit der gesamten Funktionsleiste benutzt werden können, oder sie zeigt die Befehle an, die mit jeder einzelnen Watch-Variablen benutzt werden können.

Um das Kontextmenü zu öffnen, wählen Sie mit der rechten Maustaste entweder die Funktionsleisten-Titelleiste oder eine bestimmte Watch-Variable.

Funktionsleiste Variables

Die Funktionsleiste Variables zeigt alle Variablen an, die im aktuellen Status der Programmausführung zur Verfügung stehen. Alle Variablen sind in unterschiedliche Kategorien unterteilt. Diese Kategorien sind lokale Variablen, globale Variablen, Systemvariablen, AIV-Variablen und Kontext-Variablen. Sie können zwischen diesen Kategorien wechseln, indem Sie das entsprechende Register am unteren Rand der Funktionsleiste wählen. Um den Inhalt zu ändern, wählen Sie das Inhaltsfeld einer bestimmten Variablen aus. Für einige Systemvariablen gilt nur Lesezugriff (read-only), und sie können deshalb nicht geändert werden.

Die Funktionsleiste Variables bietet ein Kontextmenü, das entweder die Befehle anzeigt, die in Verbindung mit der gesamten Funktionsleiste benutzt werden können, oder aber die Befehle anzeigt, die mit jeder einzelnen Variablen benutzt werden können.

Um das Kontextmenü zu öffnen, wählen Sie mit der rechten Maustaste entweder die Funktionsleisten-Titelleiste oder eine bestimmte Variable.

Funktionsleiste Watchpoints and Breakpoints

Die Funktionsleiste Watchpoints and Breakpoints dient dazu, Watchpoints und Breakpoints hinzuzufügen und zu verwalten. Sie können zwischen den Watchpoints und Breakpoints umschalten, indem Sie das betreffende Register am unteren Rand der Funktionsleiste wählen.

Watchpoints

Wenn Sie Watchpoints benutzen, können Sie rasch "unzulässige" Änderungen an Natural-Variablen durch Fehler enthaltende Objekte entdecken.

Standardmäßig dienen Watchpoints dazu, den Debugger anzuweisen, die Ausführung von Natural-Objekten zu unterbrechen, wenn sich der Inhalt einer Variablen ändert. Wenn Sie aber einen bestimmten Wert für die Variable zusammen mit einem Watchpoint-Operator beim Setzen eines Watchpoints angeben, kann eine Bedingung gesetzt werden, die den Watchpoint nur aktiviert, wenn die Bedingung wahr ist.

Eine Variable gilt als geändert, wenn entweder ihr aktueller Wert vom Wert abweicht, der aufgezeichnet wurde, als der Watchpoint zuletzt gestartet wurde, oder wenn er vom Anfangswert abweicht.

Um einen Watchpoint zeitweise zu deaktivieren, entfernen Sie das Häkchen aus dem Kontrollkästchen des betreffenden Watchpoint-Eintrags.

Das Watchpoint-Register dieser Funktionsleiste bietet ein Kontextmenü, das entweder die Befehle anzeigt, die in Verbindung mit dem gesamten Register benutzt werden können, oder das die Befehle anzeigt, die mit jedem einzelnen Watchpoint benutzt werden können.

Um das Kontextmenü zu öffnen, wählen Sie mit der rechten Maustaste entweder die Register-Titelleiste oder einen bestimmten Watchpoint.

Watchpoint hinzufügen

Ein neuer Watchpoint kann hinzugefügt werden, indem Sie entweder den Befehl **Add Watchpoint** (Watchpoint hinzufügen) vom Menü Variables oder den Befehl **Add** (Hinzufügen) vom Kontextmenü des Watchvariables-Registers auswählen.

Das Dialogfeld **Add Watchpoint** (Watchpoint hinzufügen) ermöglicht es Ihnen, einzelne Variablen, Arrays und einzelne Gruppen-Elemente von der Liste der verfügbaren Variablen auszuwählen. Wenn Sie den Dialog mit der Schaltfläche **OK** schließen, bewirkt, dass das Dialogfeld **Set Watchpoint** (Watchpoint setzen) geöffnet wird, das es Ihnen ermöglicht, eine Bedingung für diesen Watchpoint anzugeben.

Anmerkung:

Bei Systemvariablen ist der betreffende Watchpoint nicht an eine spezifische Library und ein bestimmtes Objekt angebunden; deshalb sind der Objekt- und Library-Name stets SYSTEM.

Dialogfeld Set Watchpoint (Watchpoint setzen)

Das Dialogfeld **Set Watchpoint** (Watchpoint setzen) zeigt den Namen des Watchpoints an (der dem Namen der ausgewählten Variablen entspricht), und zwar zusammen mit seiner Zeilenreferenz/Kontext-ID (wenn vorhanden) und den Namen des betreffenden Natural-Objekts und der entsprechenden Natural-Library.

Anmerkung:

Bei Systemvariablen ist der betreffende Watchpoint nicht an eine spezifische Library und ein bestimmtes Objekt angebunden; deshalb ist der Objekt- und Library-Name stets SYSTEM.

Um einen Watchpoint zu definieren, können Sie die folgenden Elemente in den entsprechenden Feldern angeben:

- Den Status des zu setzenden Watchpoints; gültige Stati sind "active" (aktiv) - Voreinstellung - und "pending" (anstehend).
- Eine Bedingung für den zu aktivierenden Watchpoint (optional).

Sie können einen passenden Wert und Watchpoint-Operator angeben; wenn kein Operator und Wert (d.h. eine Bedingung) angegeben werden, gilt die Voreinstellung (MOD). Eine Beschreibung der einzelnen Watchpoint-Operatoren finden Sie unten).

- Die Anzahl der übersprungenen Werte vor der Ausführung des Watchpoints, wenn er nicht ausgeführt werden soll, bis das Programm eine bestimmte Anzahl von Malen abgelaufen ist; die Voreinstellung ist 0.
- Die maximale Anzahl der Ausführungen des Watchpoints; die Voreinstellung ist 0.

Ein Watchpoint wird erst gesetzt, wenn Sie entweder mit **OK** bestätigen oder EINGABE drücken. Wenn Sie die Schaltfläche **Cancel** (Abbrechen) wählen oder ESC drücken, wird kein Watchpoint gesetzt.

Sobald ein Watchpoint angegeben wurde, bleibt er stehen, bis Sie ihn explizit löschen.

Watchpoint-Operatoren

Watchpoint-Operatoren werden über Optionsfelder gesetzt; die folgenden Watchpoint-Operatoren sind verfügbar:

Operator	Steht für	Beschreibung
MOD	Modifikation	Der Watchpoint wird jedesmal aktiviert, wenn eine Änderung der Variablen auftritt. Voreinstellung.
LT	Kleiner als	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen kleiner als der angegebene Wert ist.
LE	Kleiner gleich	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen kleiner gleich dem angegebenen Wert ist.
GT	Größer als	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen größer als der angegebene Wert ist.
GE	Größer gleich	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen größer gleich dem angegebenen Wert ist.
EQ	Gleich	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen gleich dem angegebenen Wert ist.
NE	Ungleich	Der Watchpoint wird nur aktiviert, wenn der aktuelle Wert der Variablen ungleich dem angegebenen Wert ist.

Breakpoints

Ein Breakpoint ist ein Punkt, an dem die Kontrolle an den Benutzer zurückgegeben wird, während ein Natural-Objekt ausgeführt wird.

Um einen Breakpoint zeitweise zu deaktivieren, entfernen Sie das Häkchen aus dem Kontrollkästchen des betreffenden Breakpoint-Eintrags.

Das Breakpoint-Register dieser Funktionsleiste bietet ein Kontextmenü, das entweder die Befehle anzeigt, die in Verbindung mit dem gesamten Register benutzt werden können, oder das die Befehle anzeigt, die mit jedem einzelnen Breakpoint benutzt werden können.

Um das Kontextmenü zu öffnen, wählen Sie mit der rechten Maustaste entweder die Register-Titelleiste oder einen bestimmten Breakpoint.

Breakpoint hinzufügen

Mit dem Befehl **Add** (Hinzufügen) können Sie einen neuen Breakpoint definieren. Das Dialogfeld **Add Breakpoint** (Breakpoint hinzufügen) wird angezeigt, in dem Sie den Breakpoint definieren, indem Sie die folgenden Elemente in den betreffenden Feldern angeben:

- Den Status des zu setzenden Breakpoints; gültige Stati sind "active" (aktiv) - Voreinstellung - und "pending" (anstehend).
- Der Name des Natural-Objekts, das den Breakpoint enthalten soll; der standardmäßige Objekt-Name ist der Name des aktuell sich im Source-Fenster befindlichen Objekts.
- Der Name der Natural-Library, die das Objekt mit dem Breakpoint enthält; der standardmäßige Library-Name ist der Name der Library, die das sich aktuell im Source-Fenster befindliche Objekt enthält.

- Die Zeilennummer des Sourcecodes des Objekts, in der der Breakpoint ausgeführt werden soll.

Begin (Anfang) bedeutet in diesem Zusammenhang, dass der Breakpoint an der ersten ausführbaren Code-Zeile des angegebenen Objekts gesetzt werden soll; **End**(Ende) bedeutet, dass der Breakpoint an der letzten ausführbaren Code-Zeile des angegebenen Objekts gesetzt werden soll.

- Die Anzahl der übersprungenen Elemente vor der Ausführung des Breakpoints, wenn er erst ausgeführt werden soll, wenn das Programm eine bestimmte Anzahl von Malen abgelaufen ist; die Voreinstellung ist 0.
- Die maximale Anzahl der Ausführungen des Breakpoints; die Voreinstellung ist 0.

Ein Breakpoint wird erst gesetzt, wenn Sie entweder mit **OK** bestätigen oder EINGABE drücken. Wenn Sie die Schaltfläche **Cancel** (Abbrechen) wählen oder ESC drücken, wird kein Breakpoint gesetzt.

Breakpoints können auch direkt im gerade im Source-Fenster enthaltenen Programm gesetzt werden, indem Sie die passende Statement-Zeile mit der rechten Maustaste doppelt anklicken. Auf diese Art wird ein Breakpoint mit allen Standardwerten und der entsprechenden Sourcecode-Zeilenummer definiert. Er kann angezeigt und/oder geändert werden, indem Sie die betreffenden Funktionen benutzen.

Breakpoints können nicht in Kommentarzeilen oder in einer Statement-Zeile außer der ersten Zeile gesetzt werden, wenn ein einzelnes Statement mehr als eine Zeile einnimmt.

Sobald ein Breakpoint definiert wurde, bleibt er erhalten, bis Sie ihn explizit löschen.