

# Funktionsübersicht

# Qualizer

**Analyse-Werkzeug für  
NATURAL, COBOL und JCL  
Version 3.6**



**ATLANTIS**  
Beratungsgesellschaft für  
Informationsverarbeitung mbH



# Funktionsübersicht

# Qualizer

**Analyse-Werkzeug für  
NATURAL, COBOL und JCL  
Version 3.6**



**ATLANTIS**  
Beratungsgesellschaft für  
Informationsverarbeitung mbH

This material contains and is part of a computer software program, which is property of ATLANTIS Beratungsgesellschaft für Informationsverarbeitung mbH, Germany. The program including this material may not be copied, disassembled or reproduced in any other way, electronically or mechanical, without a written permission, neither in whole nor in part. All permitted reproductions have to be indicated with this note.

Das Programm inklusive der vorliegenden Unterlagen dürfen ohne ausdrückliche, schriftliche Erlaubnis weder insgesamt noch auszugsweise vervielfältigt, entschlüsselt oder sonstwie reproduziert werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln, elektronisch oder mechanisch, dies geschieht. Alle genehmigten Reproduktionen müssen mit diesem Hinweis versehen werden.

### **Version 3.6, January 1999**

Document number: QLZRWIN-ID360-0199-D

Copyright © 1998 - 1999 ATLANTIS Beratungsgesellschaft für Informationsverarbeitung mbH.  
All rights reserved.

Qualizer is a registered trademark of ATLANTIS Beratungsgesellschaft für Informationsverarbeitung mbH, Stuttgart, Germany. All other product names used in this document are registered trademarks or trademarks of their respective holders and are accepted.

# Inhalt

<b>KAPITEL 1</b> .....	<b>1</b>
VORBEMERKUNGEN .....	1
<i>Zielgruppe</i> .....	2
<i>Inhalt des vorliegenden Dokuments</i> .....	2
<i>Voraussetzungen</i> .....	2
<i>Sonstige Dokumentation</i> .....	3
<i>Sonstige Informationsquellen</i> .....	3
<i>Support</i> .....	3
<b>KAPITEL 2</b> .....	<b>5</b>
ALLGEMEINE FUNKTIONALITÄT.....	5
<i>Anwendungsfenster</i> .....	6
<i>Werkzengleisten</i> .....	6
<i>Ausgabe der Ergebnisse</i> .....	9
<i>Weitere Utilities</i> .....	9
<b>KAPITEL 3</b> .....	<b>10</b>
FUNKTIONALITÄT DER EINZELNEN ARBEITSBEREICHE.....	10
<i>Qualizer®/Audit für NATURAL, COBOL und JCL</i> .....	11
Einführung in Qualizer/Audit.....	11
Anwendung von Qualizer/Audit.....	12
<i>Qualizer®/Doc</i> .....	13
Einführung in Qualizer/Doc .....	13
Anwendung von Qualizer/Doc .....	14
Gutachten-Fenster („Survey“).....	14
<i>Qualizer®/Info</i> .....	15
Einführung in Qualizer/Info .....	15
<i>Qualizer®/Test</i> .....	26
Einführung in Qualizer/Test.....	26
Anwendung von Qualizer/Test.....	26
<b>KAPITEL 4</b> .....	<b>29</b>
SYSTEMVORAUSSETZUNGEN .....	29
<i>Hardware</i> .....	30
Minimale Anforderungen .....	30
Empfohlene Konfiguration .....	30
<i>Software</i> .....	31
<i>Installation</i> .....	31
Qualizer® .....	31
Win32s .....	31
<b>ANHANG</b> .....	<b>33</b>
ANHANG .....	33
<i>Abbildungsverzeichnis</i> .....	34
<i>Glossar</i> .....	35
STICHWORTVERZEICHNIS .....	I



# Vorbemerkungen

**Inhalt**

Dieses Kapitel enthält Hinweise zum Adressatenkreis dieses Dokuments und Hinweise auf weiterführende Dokumente.

## Inhalt dieses Kapitels

---

ZIELGRUPPE .....	2
INHALT DES VORLIEGENDEN DOKUMENTS .....	2
VORAUSSETZUNGEN .....	2
SONSTIGE DOKUMENTATION .....	3
SUPPORT .....	3

# Vorbemerkungen

## Zielgruppe

Diese erste Einführung in **Qualizer**<sup>®</sup> richtet sich an Verantwortliche für die Software-Qualitätssicherung und den Software-Test sowie den IT-Manager.

## Inhalt des vorliegenden Dokuments

Diese Einführung gibt lediglich einen Überblick über die wesentlichen Funktionen der Software wider. Die komplette Funktionalität wird in den Handbüchern und der Online-Hilfe beschrieben.

Verstehen sie diese Übersicht bitte nur als Hilfsmittel, das Ihnen einen ersten Eindruck von der Leistungsfähigkeit von **Qualizer**<sup>®</sup> vermitteln soll.

Nicht dargestellt wird insbesondere die weitreichende Anpassbarkeit der Auswertungen an die jeweiligen Erfordernisse und hausinternen Standards.

Das Dokument ist folgendermaßen strukturiert:

- 
- Kapitel 1: *Vorbemerkungen* – enthält grundlegende Hinweise zur Benutzung dieses Handbuchs
  - Kapitel 2: *Allgemeine Funktionalität* – beschreibt allgemeingültige Funktionalitäten und Navigationsmöglichkeiten.
  - Kapitel 3: *Funktionalität der einzelnen Arbeitsbereiche* – enthält detailliertere Informationen zu den einzelnen Arbeitsbereichen und den durch **Qualizer**<sup>®</sup> vorgenommenen Analysen.
  - Kapitel 4: *Definition of the E/NAT interface environment* - describes the possibilities for the adaptation of the E/NAT interface to customer specific requirements ("Customizing").
  - Anhang: *Abbildungsverzeichnis* – listet alle Abbildungen
  - Glossar: Enthält einige Begriffsdefinitionen.
- 

## Voraussetzungen

Grundlegende Kenntnisse der Software-Qualitätssicherung sowie der Umgang mit der MS Windows Benutzeroberfläche.

## Sonstige Dokumentation

Weiterführende Informationen sind den Qualizer-Handbüchern zu entnehmen:

- *Qualizer User's Guide*
- *Qualizer/Audit User's Guide*
- *Qualizer/Doc User's Guide*
- *Qualizer/Info User's Guide*
- *Qualizer/Test User's Guide*

## Sonstige Informationsquellen

Ein statisches Dokument kann niemals den einfachen Gebrauch dieses Werkzeugs darstellen. Wir bieten daher allen Interessenten kostenlos eine zeitlich limitierte Testversion des Produkts an, denn nur so läßt sich die interaktive Unterstützung des Werkzeugs vollständig darstellen.

Ferner haben Sie nur so die Möglichkeit, Ihre eigene Anwendung zu analysieren und können so auch die sicherlich hochinteressanten Ergebnisse besser bewerten.

Gerne nennen wir Ihnen auch Referenzen.

## Support

Bitte richten Sie Ihre Anfragen an unseren Support oder Vertrieb unter einer der folgenden Adressen:

ATLANTIS  
Beratungsgesellschaft fuer  
Informationsverarbeitung mbH  
Lueglensheidestrasse 22

70378 Stuttgart

GERMANY

Phone: +49 (711) 95 39 23 – 11

Fax: +49 (711) 95 39 23 – 55

E-mail: [support@atlantis-ip.com](mailto:support@atlantis-ip.com) (Vertrieb)

[qualizer@atlantis-ip.com](mailto:qualizer@atlantis-ip.com) (Technik)

WWW: <http://www.atlantis-ip.com/support/qualizer/>



# Allgemeine Funktionalität

## Leistungsumfang von Qualizer®

Dieses Kapitel zeigt die prinzipielle Arbeitsweise des Produkts und gibt eine kurze Einführung in die Benutzeroberfläche.

## Inhalt dieses Kapitels

---

ANWENDUNGSFENSTER.....	6
WERKZEUGLEISTEN .....	6
AUSGABE DER ERGEBNISSE .....	9
WEITERE UTILITIES.....	9

# Allgemeine Funktionalität

## Anwendungsfenster

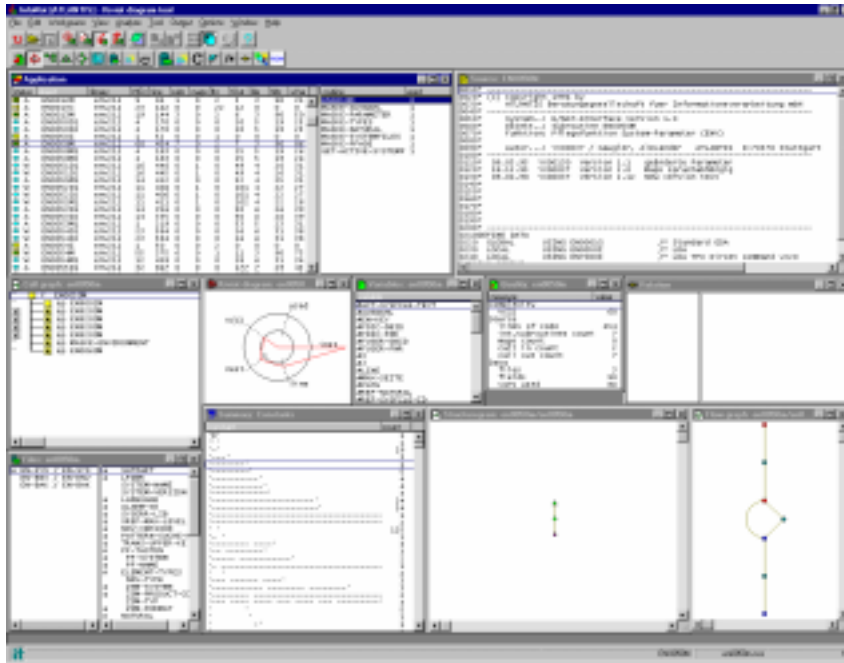


Abbildung 1: Das Qualizer Anwendungsfenster am Beispiel von Qualizer/Info

Das Anwendungsfenster läßt sich abhängig von der jeweiligen Aufgabenstellung frei einteilen und skalieren. Unterhalb der Menüleiste befinden sich zwei Toolbars, mit deren Hilfe alle wesentlichen Funktionen aufgerufen werden.

## Werkzeugleisten

Die obere Leiste dient der allgemeinen Navigation und der Auswahl des Arbeitsbereichs:



Abbildung 2: Die obere Werkzeugleiste

- Öffnen von Batch-Files zur automatisierten Ausführung vordefinierter Abläufe
- Öffnen/Definition von NATURAL libraries
- Komplette Analyse aller geladenen Objekte
- Aufruf Arbeitsbereich AUDIT
- Aufruf Arbeitsbereich DOC
- Aufruf Arbeitsbereich INFO
- Aufruf Arbeitsbereich TEST
- Aufruf Object Editor
- Nächstes Objekt
- Vorheriges Objekt
- Aufruf Renumber des Objekts
- Anzeige Objekt-Beschreibung (inkl. der extrahierten Kommentare aus Programmkopf)
- Ausgabe der Anzeige in eine Datei (als reine Textdatei oder formatiert im RTF-Format).

Darunter befindet sich kontextbezogen eine weitere Leiste, deren Symbole vom gewählten Arbeitsbereich abhängen:

## AUDIT



Abbildung 3: Die untere Werkzeuggestreife mit Qualizer/Audit

- Audit des ausgewählten Objekts
- Audit der gesamten Anwendung
- Anzeige undefinierter Variablen
- Anzeige unbenutzter Variablen
- Bearbeitung der Namenskonventionen der Anwendung (Vorgabe)

## DOC



Abbildung 4: Die untere Werkzeuggestreife mit Qualizer/Doc

- Anzeige der Zusammenfassung für das gewählte Objekt
- Anzeige der Zusammenfassung für die gesamte Anwendung
- Ausgabe der Zusammenfassung(en) der ausgewählten Objekte in eine RTF-Datei
- Ausgabe der Zusammenfassung der Anwendung in eine RTF-Datei
- Ausgabe des Callgraph (Aufrufstruktur) in eine RTF-Datei
- Aufteilung der Anwendung (Application Segmentation)

## INFO



Abbildung 5: Die untere Werkzeuggestreife mit Qualizer/Info

- Anzeige der Qualitäts-Kennzahlen (Objekt)
- Anzeige des Kivi-Diagramms (Objekt)
- Anzeige des CallGraph (Objekt)
- Anzeige des Strukturdiagramms (Objekt)
- Anzeige des Kontrollflußgraph (Objekt)
- Anzeige von Maskenabbildungen (Objekt)
- Anzeige der benutzten Files und Felder (Objekt)
- Anzeige der Variablen (Objekt)
- Anzeige der Konstanten aus Zuweisungen (Objekt)
- Anzeige der benutzten Files und Felder (Anwendung)
- Anzeige der Variablen (Anwendung)
- Anzeige der Konstanten aus Zuweisungen (Anwendung)
- Anzeige der Verwendung von Feldern (Feld, Struktur, Format und Anzahl der Nutzungen)
- Anzeige aller Kommentare
- Anzeige der Referenzen auf andere Objekte
- Anzeige der Auswirkungen gemäß definierter Regeln (z-B. Jahr 2000 oder Euro)
- Aufruf Qualizer/Rem

TEST



Abbildung 6: Die untere Werkzeugleiste mit Qualizer/Test

- Anzeige eines Summaries für die Anwendung
- Anzeige Source und Versionsvergleich
- Anzeige Source mit Flußdiagramm
- Anzeige Testpfad und Testbedingungen
- Lokalisierungs-Modus (Graph → Codezeile(n))
- Ausrichtungs- und Vergrößerungsfunktionen für die Anzeige des Graphen
- Ausgabe auf Drucker

## Ausgabe der Ergebnisse

Die in den Fenstern jeweils dargestellten Ergebnisse können vollständig exportiert und in anderen Anwendungen übernommen werden.

Texte können entweder als reiner Text (.TXT) oder im komfortableren Rich Text Format (.RTF) ausgegeben werden. Das RTF-Format eignet sich besonders für die Übernahme in eine Textverarbeitungsanwendung wie z.B. MS Word. Ferner steht für die Bereitstellung im Intranet eine Option zur Ausgabe im HTML-Format zur Verfügung.

Grafiken werden im vektororientierten Format Windows Meta File (.WMF) gespeichert und erlauben so eine beliebige Skalierung ohne Qualitätsverlust. Das WMF-Format wird von praktisch allen Windows-Anwendungen unterstützt und kann leicht weiterverarbeitet werden.

## Weitere Utilities

Daneben sind einige Utilities verfügbar, die unter anderem folgende Aufgaben unterstützen:

- IMPORT - Extraktion von Objekten aus SYSTRANS-Dateien oder ULDMAIN (NATUNLD)-Dateien
- REMEDY - Qualizer/Rem (Year 2000 Remedy)
- COMPARE - NatCmp (Compare, stand-alone)
- CLEAN - Qualizer/Clean (restructures/builds Data Areas)
- CONV2223 - Umstellungshilfe NATURAL 2.2 nach NATURAL 2.3
- HTMLGEN - Ausgabe von Analyse-Ergebnissen im HTML-Format zur Bereitstellung im Intranet
- Qualizer/DDM -
- Qualizer/LDA -
- STACK - Qualizer/Stk (Analyse von Aufrufen via Stack und/oder LOGON)

# Funktionalität der einzelnen Arbeitsbereiche

**Inhalt**

Dieses Kapitel stellt den wesentlichen Funktionsumfang von **Qualizer**<sup>®</sup> anhand einiger Beispiele dar. Es werden alle wichtigen Funktionszweige und Leistungsmerkmale gezeigt.

## Inhalt dieses Kapitels

---

QUALIZER/AUDIT FÜR NATURAL, COBOL UND JCL .....	11
QUALIZER/DOC .....	13
QUALIZER/INFO .....	15
QUALIZER/TEST .....	26

# Funktionalität der einzelnen Arbeitsbereiche

Alle Arbeitsbereiche können frei definiert werden. Welche und wie viele Fenster angezeigt werden hängt von Ihrer persönlichen Auswahl ab. Je nach Aufgabenstellung sollten Sie nur die tatsächlich relevanten Informationen anzeigen.

Die nachfolgenden Darstellungen enthalten zur besseren Übersicht über den Leistungsumfang des Werkzeugs möglichst viele Fenster bzw. Informationen.

## Qualizer®/Audit für NATURAL, COBOL und JCL

### Einführung in Qualizer/Audit

Qualizer/Audit kann als ein Auditor für Natural-Source benutzt werden. Qualizer/Audit sucht nach Auffälligkeiten in den NATURAL-Source. Die Liste der überprüften Anomalien ist bereits vordefiniert und wird im „Anomalies“-Fenster angezeigt.

### Anomalien

Qualizer/Audit sucht nach folgenden Auffälligkeiten innerhalb der NATURAL Source:

- **Mehrere Statements auf der selben Zeile** – Mehr als ein NATURAL-Statement in einer einzigen Zeile
- **Verweis auf Zeilennummer** – Eine Verweis auf eine Zeilennummer (statt z.B. auf ein Label)
- **FIND SORTED** – Ein FIND-Statement mit der SORTED-Klausel
- **Unbenutzte Subroutines** – Eine interne Subroutine ist zwar definiert, wird aber nicht verwendet
- **Unbenutzte Variablen** – Eine Variable ist zwar definiert, wird aber nicht verwendet
- **Nicht dem Standard entsprechende Namen** – Z. B. eine Variable entspricht nicht den vordefinierten Standards für die Benennung
- **Kurzes DECIDE** – Drei oder weniger WHEN/VALUE-Klauseln in einem DECIDE-Statement werden verwendet
- **NONE fehlt** – NONE-Klausel in einem DECIDE-Statement ist nicht vorhanden
- **Tiefe Verschachtelung** – Verschachtelungstiefe (z.B. eines IF) ist größer als 7.
- **Komplexer logischer Ausdruck** – Ein logischer Ausdruck enthält mehr als 4 AND- und OR-Operatoren
- **FETCH aus einem Subprogram oder einer Subroutine** – Eine externe Subroutine oder ein Subprogram wird mittels FETCH verlassen
- **Nicht-strukturierter ESCAPE**
- **Using int. subroutine name**

## Anwendung von Qualizer/Audit

1. Sie wählen im Fenster „Application“ ein Objekt aus und erhalten eine Liste der internen Subroutinen, die Liste der Auffälligkeiten im Fenster „Anomalies“ und ein Source-Listing im Fenster „Source“. Daneben können auch nicht definierte oder unbenutzte Variablen angezeigt werden. Desweiteren besteht die Möglichkeit, im Fenster „Description“ eine aus den Programm-Kommentaren extrahierte Beschreibung anzuzeigen.

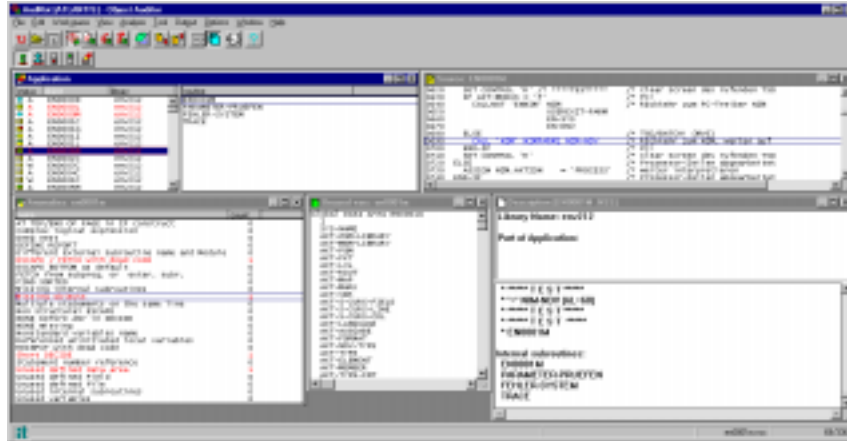


Abbildung 7: Das Anwendungsfenster von Qualizer/Audit

2. Wählen Sie nun im rechten Teil des Fensters „Application“ eine im Objekt definierte Unterroutine („Routine“), so wird das Source-Listing automatisch an den Anfang dieser Routine positioniert.
3. Wählen Sie im Fenster „Anomalies“ eine Auffälligkeit aus, so wird wiederum im Fenster „Source“ auf die entsprechende Stelle im Source-Listing positioniert, die Zeile farbig markiert und Sie können sofort eine Beurteilung vornehmen.

Neben der hier gewählten Darstellung für ein einzelnes Objekt können Sie jederzeit zur Gesamtanalyse der Anwendung wechseln (🔍, 2. Knopf in der Werkzeugleiste) und so die Gesamt-Übersicht erhalten. Wiederum können Sie durch Anwahl von Zeilen in einem anderen Fenster sofort die entsprechende (erste) Stelle im Source-Code anzeigen.

Zwischen den auffälligen Objekten können Sie auch mittels der Knöpfe 🏠 und 🏠 in der ersten Werkzeugleiste direkt springen.

Mittels der Knöpfe 🏠 und 🏠 können Sie

- benutzte aber nicht definierte bzw.
- definierte aber nicht benutzte

Variablen anzeigen.

## Einführung in Qualizer/Doc

Qualizer/Doc ist ein Reverse Engineering Tool, das NATURAL-Programme (Version 2) automatisch dokumentiert. Sowohl der Reporting Mode als auch der Structured Mode werden unterstützt.

Qualizer/Doc stellt folgende Informationen bereit:

- Programm-Listing
- Map-Layouts
- Versionsvergleich zwischen zwei Programmversionen (z.B. aktuelle und Vorversion)
- Zyklomatische Komplexität ("Cyclomatic complexity") der Module
- Automatische Extraktion von Programm- bzw. Funktionsbeschreibungen aus den Programmköpfen
- Liste der internen Subroutinen und ihrer Komplexitäten
- Liste der verwendeten Datenbereiche und Masken
- Nutzung von Files und Views
- Call Graph interner Subroutinen
- Struktogramme
- Anwendungsgutachten

Zur Ermittlung dieser Informationen und Kennzahlen werden alle Objekte einer Anwendung analysiert. Nicht nur die Strukturierung einzelner Objekte sondern auch die Beziehungen zwischen den Objekten werden analysiert (A ruft B ebenso wie **B wird von A gerufen**). Diese Funktionalitäten sind insbesondere bei der Analyse von Auswirkungen von Programm- oder Schnittstellenänderungen hilfreich (vgl. Umstellungen PLZ, Euro oder Jahr 2000).

Im Anschluß an die Analyse stehen folgende Dokumente zur Ausgabe bereit:

1. „*Source Object Summary Pages*“ (1 Seite für jedes NATURAL 2 Objekt)
2. „*Application Summary Page*“ (1 Dokument für die gesamte Anwendung)

Sowohl Aufbau als auch Inhalt der Dokumente können interaktiv an die entsprechenden Bedürfnisse angepaßt werden.

Die Dokumente können

- auf Bildschirm ausgegeben,
  - im ASCII- oder RTF-Format zur Weiterverarbeitung in MS Word oder anderen Textverarbeitungsprogrammen auf Datei gespeichert oder
  - gedruckt
- werden.

## Anwendung von Qualizer/Doc

1. Sie wählen im Fenster „Application“ ein Objekt aus und erhalten das zugehörige Source-Listing sowie das auf dem Re-Engineering-Vorgang basierende Objekt-Dokument.

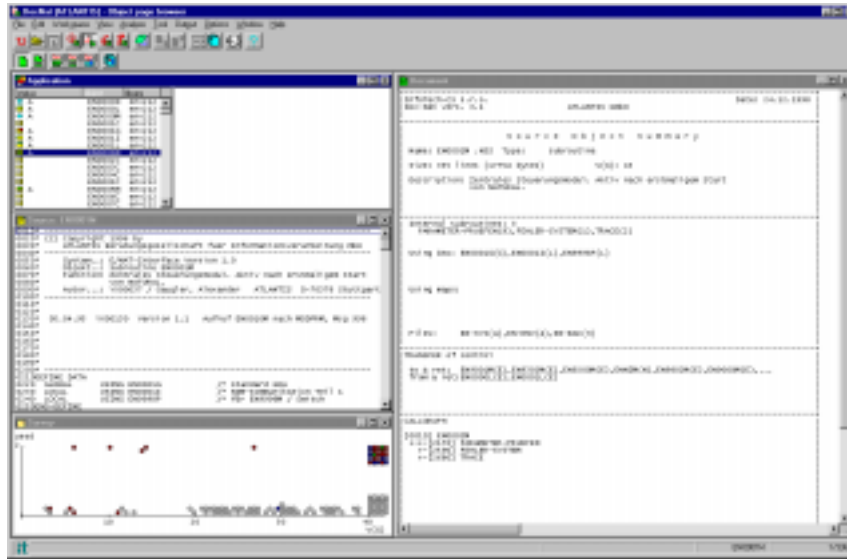


Abbildung 8: Das Anwendungsfenster von Qualizer/Doc

2. Zusätzlich kann das Gutachten für die Gesamt-Anwendung angezeigt werden (→ Fenster „Survey“). Jeder Kreis in diesem Diagramm symbolisiert ein Objekt der Anwendung. Durch Mausclick auf einen der Kreise kann auch hier direkt zu jedem Objekt gesprungen werden und das zugehörige Objekt-Dokument sowie Source-Listing werden angezeigt.

Neben der hier gewählten Darstellung für ein einzelnes Objekt können Sie jederzeit zur Gesamtanalyse der Anwendung wechseln (☐, 2. Knopf in der Werkzeugleiste) und so die Anwendungs-Übersicht erhalten.

Mittels der 3 Knöpfe für RTF-Ausgabe können Sie

- die Dokumente der aktuell ausgewählten Objekte
- das Dokeument der Gesamtanwendung oder
- den nur Call-Graphen des Objekts bzw. der Anwendung

im RTF-Format auf Datei ausgeben.

## Gutachten-Fenster („Survey“)

Cluster-Diagramm der Anwendung welches die Verteilung der Anwendungsobjekte in einem Koordinatensystem zeigt. Die Zuordnungen basieren auf der Objektgröße und der Objekt-Komplexität.

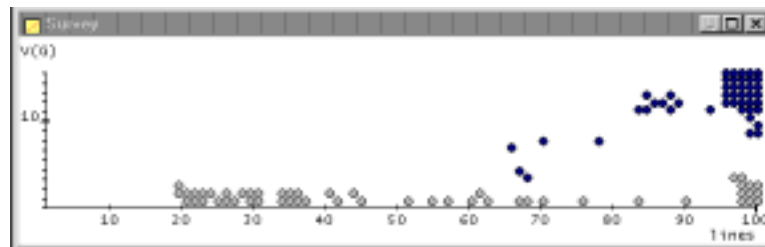


Abbildung 9: Das Surveyfenster von Qualizer/Doc

Je nach Objekttyp werden unterschiedliche Farben zur Markierung benutzt:

- **rot** Programs
- **blau** Subroutines
- **grün** Subprograms
- **grau** andere Objekttypen.

# Qualizer®/Info

## Einführung in Qualizer/Info

Qualizer/Info ist bezüglich der Zahl der Analyseergebnisse das mächtigste Werkzeug. Es umfaßt folgende Analysen und Ausgaben (jeweils auf Ebene der gesamten Anwendung, eines einzelnen Objekts oder einer internen Subroutine):

### Qualität

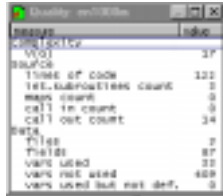


Abbildung 10: Das Quality-Fenster von Qualizer/Info

Kürzel	Beschreibung	Art
V(G)	Zyklomatische Komplexität nach Thomas McCabe.	Komplexität
main	Zyklomatische Komplexität der Hauptroutine.	Komplexität
Habs	Halstead level of abstraction	Komplexität
Hbug	Halstead bug prediction (Anzahl vorhergesagte Programmfehler)	Komplexität
nest	Maximale Verschachtelungstiefe in strukturierten Statements	Komplexität
bFP	Backfired function points (für Programs, Subprograms und Subroutines)	Komplexität
Hlen	Halstead length	Quellcode
line	Anzahl Lines of code (insgesamt, mit Kommentar)	Quellcode
linC	Anzahl Zeilen mit Code gefolgt von Kommentar	Quellcode
comT	Anzahl triviale Kommentarzeilen (nur zur Code-Separierung verwendet, ohne erklärenden Inhalt)	Quellcode
comN	Anzahl Netto-Kommentarzeilen (mit erläuterndem Inhalt)	Quellcode
subr	Anzahl interne Subroutinen	Quellcode
maps	Anzahl Maps	Quellcode
cMap	Maps in USING-Klauseln	Quellcode
fIn	Anzahl Objekte von denen gerufen wird	Quellcode
fOut	Anzahl Objekte die gerufen werden	Quellcode
inst	Anzahl Instruktionen (NATURAL 2 Statements)	Quellcode
tokn	Anzahl Tokens (Schlüsselwörter, Variablenennamen, Labels, Operatoren, Delimiter, Konstanten, Attribute, Kommentare)	Quellcode
adoc	Auto-Dokumentation in % (= [comN] * 100 / [line])	Quellcode
file	Anzahl benutzte Dateien	Daten
flds	Anzahl benutzte Felder	Daten
dVar	Anzahl definierte Lokal-Variablen	Daten
uVar	Anzahl benutzte Lokal-Variablen	Daten
nVar	Anzahl unbenutzte Lokal-Variablen	Daten
dGlo	Anzahl definierte Global-Variablen	Daten
uGlo	Anzahl benutzte Global-Variablen	Daten
nGlo	Anzahl unbenutzte Global-Variablen	Daten

Für jedes Kriterium können folgende Parameter frei definiert werden:

- der Toleranzbereich (von/bis)
- Anzeige auf Anwendungs- und/oder Objektebene
- Aufnahme in Kiviat-Diagramm ja/nein
- und vieles mehr.

### Beschreibung einiger Kriterien in Kürze

#### Zyklomatische Komplexität

Die Zyklomatische Komplexität ist eine Größe zur Messung der strukturellen Komplexität von Programmen. Sie beeinflusst die Testbarkeit, Wartbarkeit und Lesbarkeit von Programmen. Die Basis für die Berechnung bildet der → Kontrollflußgraph.

#### Literaturhinweise

- T. J. McCabe, A Complexity Measure, in IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, December 1976, S. 308-320
- T. J. McCabe, A Complexity Measure, in T.J. MacCabe, Structured Testing, IEEE Computer Society Press 1983, S. 3-15
- T. J. McCabe, A Testing Methodology Using the McCabe Complexity Measure, in T.J. MacCabe, Structured Testing, IEEE Computer Society Press 1983, S. 19-47

#### Halstead Metriken

Die von Halstead vorgeschlagenen Metriken basieren auf der Anzahl der in einem Programm benutzten unterschiedlichen Operanden und Operatoren sowie auf der Gesamtzahl der vorhandenen Operanden und Operatoren.

#### Literaturhinweis

- M. H. Halstead, Elements of Software Science, North-Holland, New York 1977

### Kiviat-Diagramm

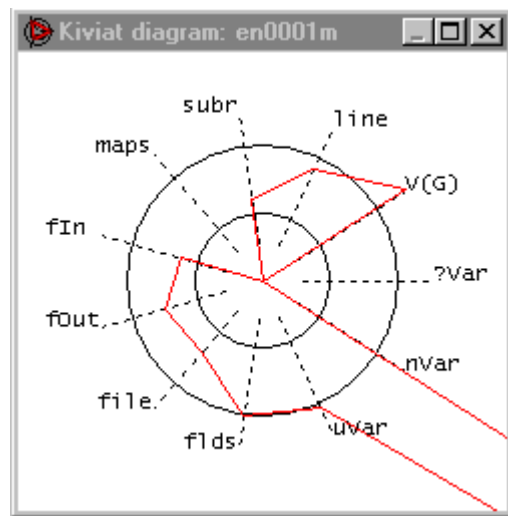


Abbildung 11: Das Kiviat-Diagramm-Fenster von Qualizer/Info

Das Kiviat-Diagramm ist die grafische Repräsentation der Qualitätskriterien (vgl. vorherigen Abschnitt). Der Vorteil dieser grafischen Darstellung liegt in der schnellen optischen Erfassbarkeit von Über- und Unterschreitung vorgegebener Grenzwerte.

Im obigen Beispiel wird schnell deutlich, daß die Obergrenzen für  $V(G)$  und  $nvar$  überschritten wurden.



## Strukturdiagramm

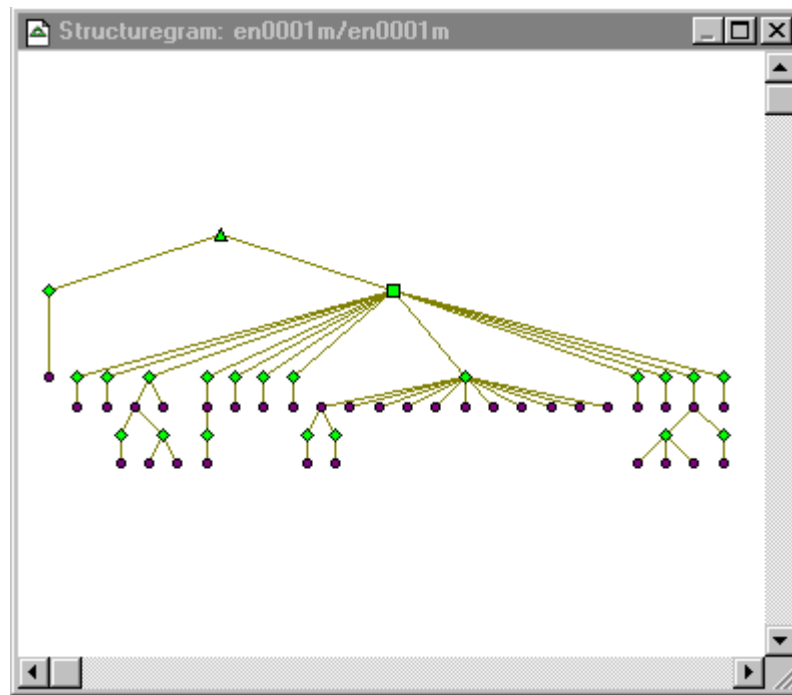


Abbildung 14: Das Struktur-Diagramm-Fenster von Qualizer/Info

Das Strukturdiagramm dient der Bewertung der internen Strukturen eines Objekts. Die Knoten repräsentieren die Statements des Objekts, die Dreiecke die syntaktische Schachtelung der Statements. Nur die Kontroll-Statements gemäß dem gewählten Strukturdiagramm-Typ sind sichtbar.

Verwendete Symbole

- Statements die eine Schleife initiieren
- ◆ Bedingungen
- WHEN, THEN, ELSE Klauseln
- ▲ Sonstige Statements

Durch Klick auf einen der Knoten im Strukturdiagramm wird das Source-Fenster automatisch auf die entsprechende Zeile positioniert.

Über verschiedene Buttons können unterschiedliche Details angezeigt werden:



Abbildung 15: Das Struktur-Diagramm-Werkzeuggestreife

- Nur Kontroll-orientierte Statements
- Kontroll- und Datenbank-orientierte Statements
- Kontroll- Datenbank- und Workfile-orientierte Statements
- Kontroll- und Bildschirm-orientierte Statements
- Kontroll- und Kontrollübergabe-orientierte Statements
- Nur Kontrollübergabe-orientierte Statements

Erläuterung der Statement-Kategorien

- Kontroll-orientiert:  
IF, DECIDE, Statements die Schleifen initiieren, ON ERROR, AT
- Datenbank-orientiert:  
FIND, HISTOGRAM, READ, BROWSE, BACKOUT TRANSACTION,  
END TRANSACTION, GET, GET SAME, GET TRANSACTION, STORE, UPDATE,  
IF NOT FOUND Klauseln
- Workfile-orientiert:  
READ WORK FILE, WRITE WORK FILE, CLOSE WORK FILE
- Bildschirm-orientiert:  
INPUT, MAP, REINPUT, DISPLAY, WRITE
- Kontrollübergabe-orientiert:  
FETCH, CALLNAT, PERFORM, CALL, CALL FILE, CALL LOOP, RUN

## Kontrollflußgraph

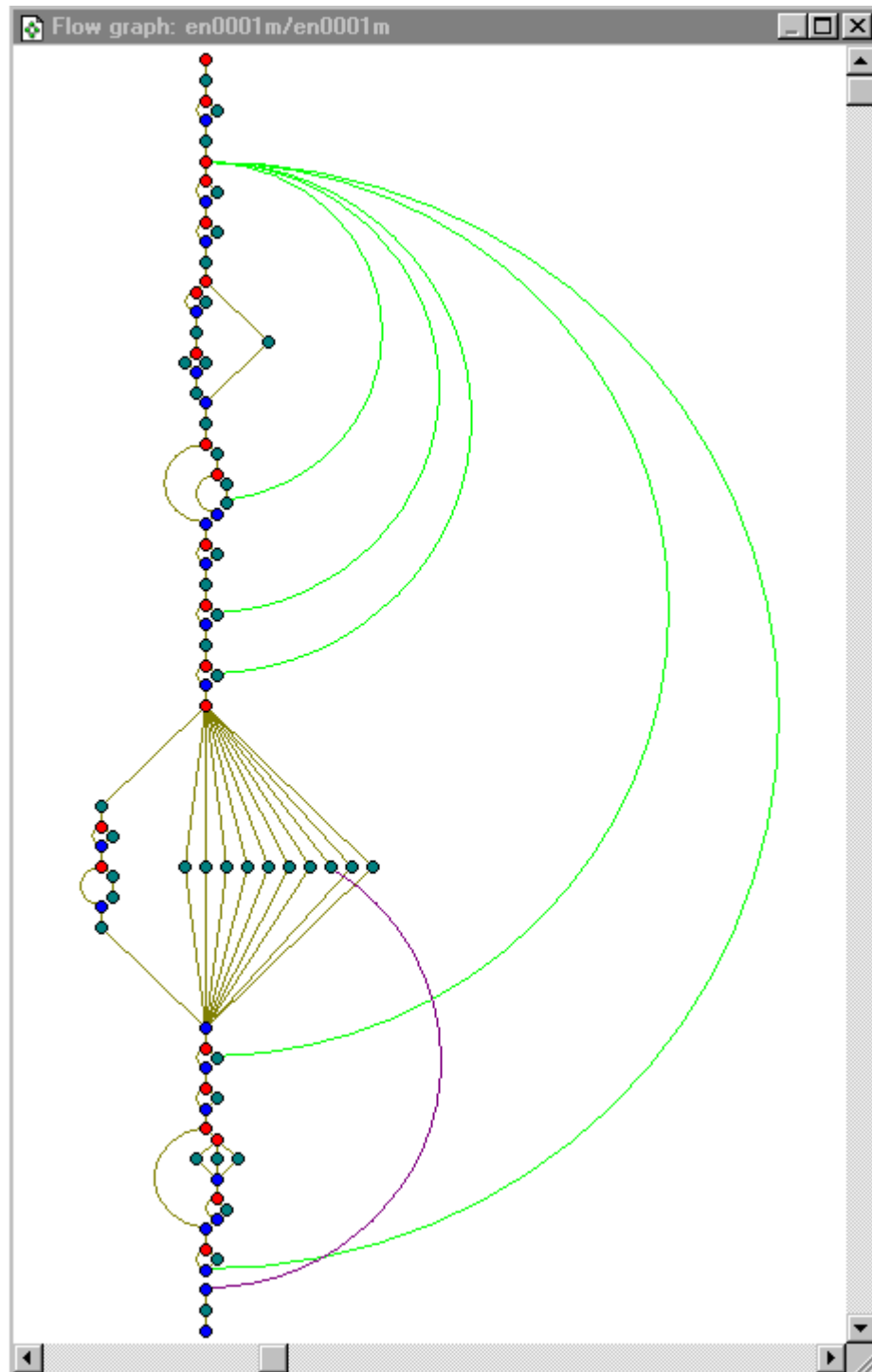


Abbildung 16: Das Kontrollflußgraph-Fenster von Qualizer/Info

Der Kontrollflußgraph ist die grafische Repräsentation des Programmablauf-Plans einer Routine. Er zeigt alle möglichen Pfade durch eine Routine. Der Kontrollflußgraph ist ein gerichteter Graph mit *Knoten* und *Kanten*.

#### Knoten und Kanten

Ein *Knoten* repräsentiert ein Statement oder eine Gruppe fortlaufender, nicht verzweigender Statements der Routine.

Die gerichteten *Kanten* stellen einen möglichen Kontrollfluß zwischen Knoten dar. Gerichtete Kanten werden als *Zweige* bezeichnet.

Eine *Folge* aufeinanderfolgender nicht verzweigender und nicht strukturierter Statements wird als 1 Knoten dargestellt. Diese Zusammenfassung wird als *Segment* oder auch *Block* bezeichnet.

Ein *Segment* ist eine Folge von Knoten mit maximaler Länge, das folgende Eigenschaften besitzt: Es kann ausschließlich über den ersten Knoten des Segments betreten werden und falls der erste Knoten des Segments betreten wird, so werden alle Knoten des Segments in der vorgesehenen Reihenfolge genau einmal durchlaufen.

*Strukturierte Statements* beginnen mit dem Anfangsknoten und enden mit dem Endknoten. Diese beiden Knoten werden auf derselben (gedachten) vertikalen Linie dargestellt. Zwischen dem Anfangs- und Endknoten werden weitere Statements (inklusive Verschachtelungen) dargestellt.

Eine *Routine* wird als *strukturiertes Statement* dargestellt.

Die *Flußrichtung* wird von oben nach unten dargestellt. Der erste Knoten repräsentiert den Anfang der Routine. Für nicht-strukturierte Statements geht der Ablauf im Diagramm nach unten. In strukturierten Statements kann der Ablauf auch nach oben fortgesetzt werden, was durch halbkreisförmige Verbindungslinien dargestellt wird.

### Knotenarten

Der *Startknoten* ist der erste Knoten einer Struktur

Der *Endknoten* ist der letzte Knoten einer Struktur

*Einfache Knoten* repräsentieren nicht-strukturierte und nicht-verzweigende Statements oder eine fortlaufende Gruppe solcher Statements.

### Zweige

Einfache Verbindungen stellen keine Eigenschaften von Strukturen dar. Die Verbindung wird als gerade Linie dargestellt, der Ablauf wird nach unten fortgesetzt.

Schleifen-Verbindungen stellen die Rückkehr an den Anfang einer Struktur dar. Dies sind die einzigen Zweige, bei denen der Ablauf nach oben fortgesetzt wird. In Schleifen (Datenbank, FOR, REPEAT etc.) werden die Zweige als rechter Halbkreis dargestellt. Für ESCAPE TOP erfolgt die Darstellung in abgewinkelter Form (eine Art Dreieck).

Aussprung-Zweige stellen den Sprung an das Ende der Struktur dar und werden als linke Halbkreise dargestellt. Im Falle von ESCAPE BOTTOM erfolgt die Darstellung in abgewinkelter Form (vgl. ESCAPE TOP). Der Ablauf wird nach unten fortgesetzt.

Sprung-Zweige stellen übersprungene Strukturen dar. Sie reichen vom Anfangs-Knoten der Struktur bis zum End-Knoten und werden als linker Halbkreis dargestellt. Der Ablauf geht nach unten.

Wiederum kann durch Klick auf einen der Knoten im Kontrollflußgraph das Source-Fenster automatisch auf die entsprechende Zeile positioniert werden.

Der Kontrollflußgraph ist eines der wichtigsten Hilfsmittel für den → Test.

Literaturhinweis: M. S. Hecht, Flow Analysis of Computer Programs, North-Holland, New York 1977.

## Maps

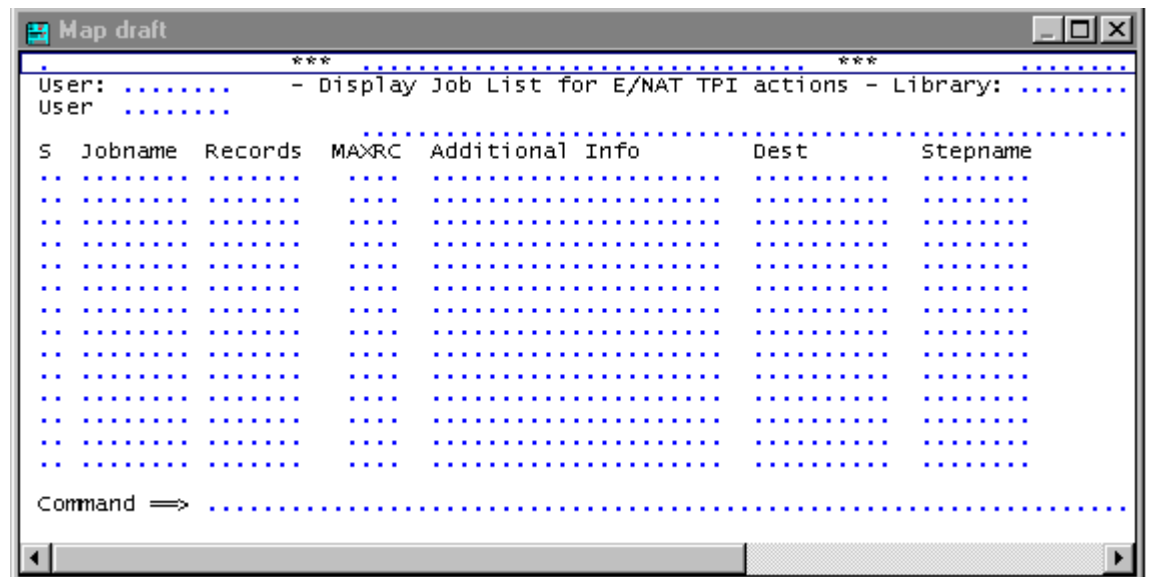


Abbildung 17: Das Maskenbild-Fenster von Qualizer/Info

Dieser Button dient der Darstellung des Maskenabbildes der in einer Routine referenzierten Map(s) oder, falls eine Map direkt ausgewählt ist, der ausgewählten Map.

## Files

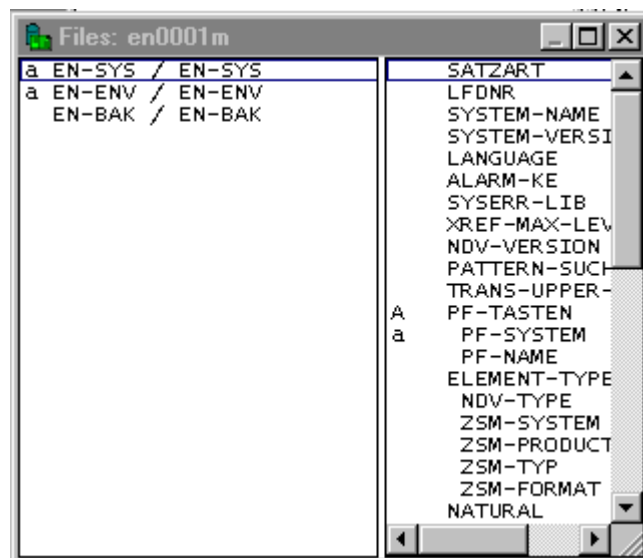


Abbildung 18: Das Files-Fenster von Qualizer/Info

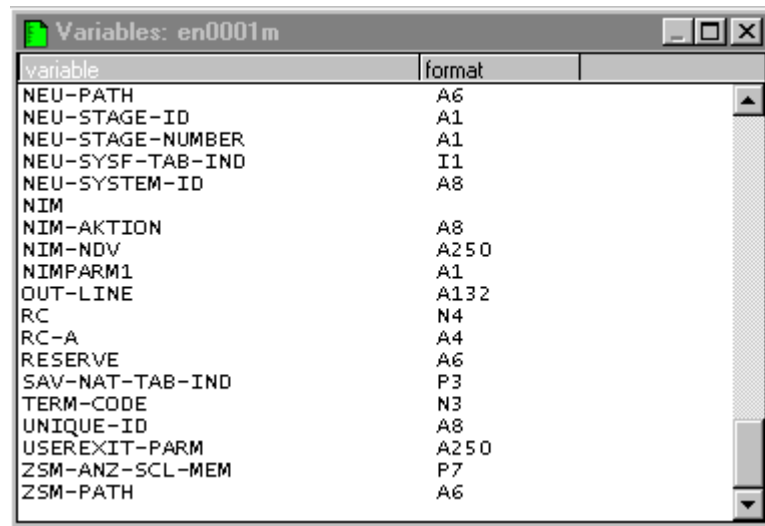
Dieses Tool zeigt eine Liste der Datenbank-Files und Views die im gewählten Objekt definiert sind (auch via externer Data Area). Jeder Eintrag in der Liste wird in der Form *File / View* dargestellt und hat ein Zeichen vorangestellt, welches die Art der Nutzung repräsentiert:

- a** file/view accessed – lesender Zugriff
- u** file/view updated – schreibender Zugriff

Wird ein View ausgewählt, dann werden in der rechten Hälfte des Fensters die definierten View-Felder angezeigt. Präfix **a** bedeutet, daß das Feld tatsächlich benutzt wird, Präfix **A** bedeutet, daß mindestens ein Element einer Struktur benutzt wird.

Auch hier kann wiederum auf ein einzelnes Feld eines Views geklickt werden, wodurch im Source-Fenster der entsprechende Source-Teil angezeigt wird.

## Variablen

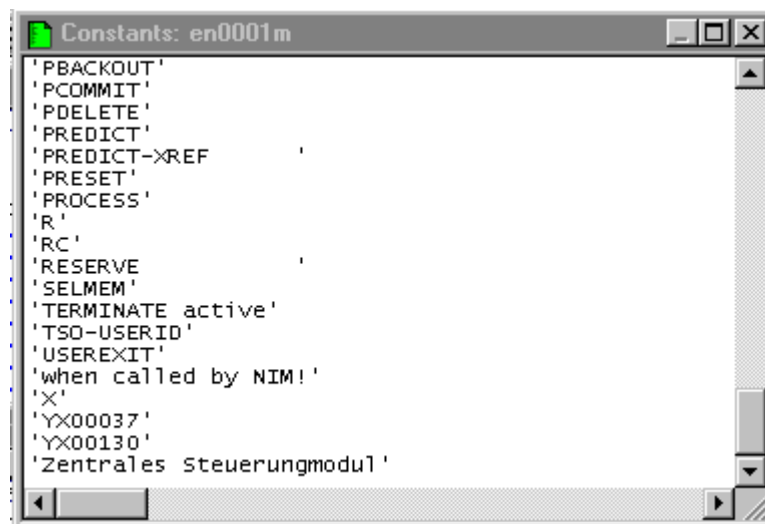


variable	format
NEU-PATH	A6
NEU-STAGE-ID	A1
NEU-STAGE-NUMBER	A1
NEU-SYSF-TAB-IND	I1
NEU-SYSTEM-ID	A8
NIM	
NIM-AKTION	A8
NIM-NDV	A250
NIMPARM1	A1
OUT-LINE	A132
RC	N4
RC-A	A4
RESERVE	A6
SAV-NAT-TAB-IND	P3
TERM-CODE	N3
UNIQUE-ID	A8
USEREXIT-PARM	A250
ZSM-ANZ-SCL-MEM	P7
ZSM-PATH	A6

Abbildung 19: Das Variablen-Fenster von Qualizer/Info

Dieses Werkzeug zeigt die Liste der definierten Variablen inklusive der Format/Längenangabe. Durch Klick auf eine Variable kann wiederum das Source-Fenster auf diejenige Zeile positioniert werden, in der die Variable im Code referenziert wird.

## Konstanten

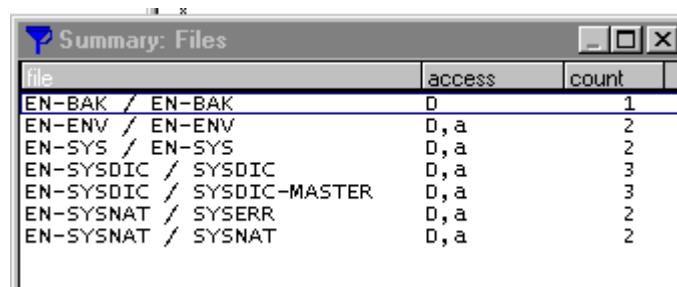


```
'PBACKOUT'
'PCOMMIT'
'PDELETE'
'PREDICT'
'PREDICT->REF'
'PRESET'
'PROCESS'
'R'
'RC'
'RESERVE'
'SELMEM'
'TERMINATE active'
'TSO-USERID'
'USEREXIT'
'when called by NIM!'
'X'
'YX00037'
'YX00130'
'Zentrales Steuerungmodul'
```

Abbildung 20: Das Konstanten-Fenster von Qualizer/Info

Hier werden alle in der gewählten Routine (oder der gesamten Anwendung!) definierten Konstanten (Zahlen und Zeichenketten) aufgelistet. Auch hier kann durch Klick auf eine Konstante das Source-Fenster positioniert werden.

## Zusammenfassung Files – Summary Files



file	access	count
EN-BAK / EN-BAK	D	1
EN-ENV / EN-ENV	D, a	2
EN-SYS / EN-SYS	D, a	2
EN-SYSDIC / SYSDIC	D, a	3
EN-SYSDIC / SYSDIC-MASTER	D, a	3
EN-SYSNAT / SYSERR	D, a	2
EN-SYSNAT / SYSNAT	D, a	2

Abbildung 21: Das Fenster Zusammenfassung Dateien von Qualizer/Info

Dieses Werkzeug listet alle Files / Views, die in der gesamten Anwendung vorhanden (definiert) sind. Das Fenster Relation zeigt die Liste der Objekte, in denen das ausgewählte File benutzt wird:

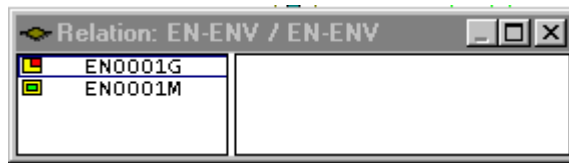


Abbildung 22: Das Fenster Dateireferenzen von Qualizer/Info

Durch Auswahl eines Objekts im Relation-Fenster wird wiederum die zugehörige Source angezeigt und positioniert.

### Zusammenfassung Variablen – Summary Variables

variable	format	count
#ERROR-LINE	N4	1
#ERROR-NR	N4	1
#FUNCTION	A20	2
#I	P7	2
#IND	P7	2
#J	P7	7
#M	P7	7
#MAP-SYSTEM	A8	4
#MAP-TYPES	A32	4

Abbildung 23: Das Fenster Zusammenfassung Variablen von Qualizer/Info

Dieses Werkzeug listet alle Variablen, die in der gesamten Anwendung vorhanden (definiert) sind. Das Fenster Relation zeigt die Liste der Objekte, in denen die ausgewählte Variable benutzt wird:

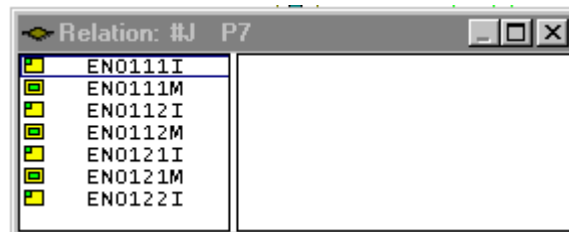


Abbildung 24: Das Fenster Variablenreferenzen von Qualizer/Info

Durch Auswahl eines Objekts im Relation-Fenster wird wiederum die zugehörige Source angezeigt und positioniert.

### Zusammenfassung Konstanten – Summary Constants

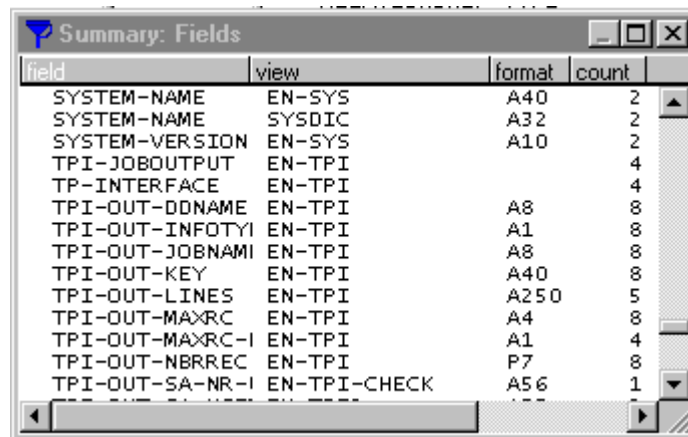
constant	count
'NDV-PRECESSOR-GROU'	1
'NDV-REPLACE	1
'NDV-SIGNOUT-USER	1
'NDV-SUBSYSTEM	1
'NDV-SYSTEM	1
'NDV-T-ENVIRONMENT	1
'NDV-T-STAGE-ID	1
'NDV-T-STAGE-NAME	1
'NDV-T-STAGE-NBR	1
'NDV-T-SYSTEM-ID	1
'NDV-TYPE	1
'NDV-UPDATE-IF-PRES'	1
'NIM'	1
'Nix da zu'	1

Abbildung 25: Das Fenster Zusammenfassung Konstanten von Qualizer/Info

Dieses Werkzeug listet alle Konstanten, die in der gesamten Anwendung vorhanden (definiert) sind. Das Fenster Relation zeigt die Liste der Objekte, in denen die ausgewählte Konstante benutzt wird.

Durch Auswahl eines Objekts im Relation-Fenster wird wiederum die zugehörige Source angezeigt und positioniert.

### Zusammenfassung Felder – Summary Fields



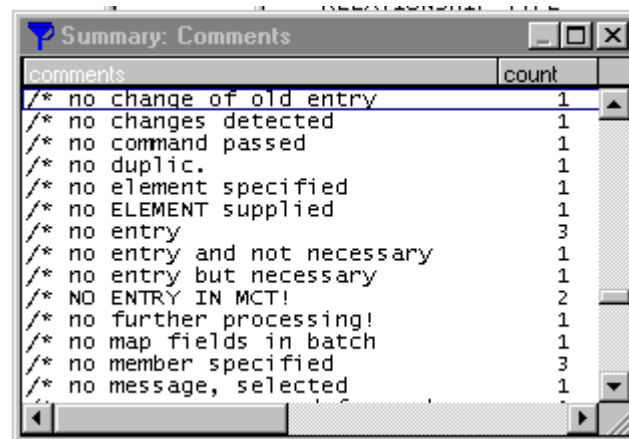
field	view	format	count
SYSTEM-NAME	EN-SYS	A40	2
SYSTEM-NAME	SYSDIC	A32	2
SYSTEM-VERSION	EN-SYS	A10	2
TPI-JOBOUTPUT	EN-TPI		4
TP-INTERFACE	EN-TPI		4
TPI-OUT-DDNAME	EN-TPI	A8	8
TPI-OUT-INFOTYI	EN-TPI	A1	8
TPI-OUT-JOBNAME	EN-TPI	A8	8
TPI-OUT-KEY	EN-TPI	A40	8
TPI-OUT-LINES	EN-TPI	A250	5
TPI-OUT-MAXRC	EN-TPI	A4	8
TPI-OUT-MAXRC-I	EN-TPI	A1	4
TPI-OUT-NBRREC	EN-TPI	P7	8
TPI-OUT-SA-NR-I	EN-TPI-CHECK	A56	1

Abbildung 26: Das Fenster Zusammenfassung Dateifelder von Qualizer/Info

Dieses Werkzeug listet alle View-Felder, die in der gesamten Anwendung vorhanden (definiert) sind. Das Fenster Relation zeigt die Liste der Objekte, in denen das ausgewählte View-Feld benutzt wird.

Durch Auswahl eines Objekts im Relation-Fenster wird wiederum die zugehörige Source angezeigt und positioniert.

### Zusammenfassung Kommentare – Summary Comments



comments	count
/* no change of old entry	1
/* no changes detected	1
/* no command passed	1
/* no duplic.	1
/* no element specified	1
/* no ELEMENT supplied	1
/* no entry	3
/* no entry and not necessary	1
/* no entry but necessary	1
/* NO ENTRY IN MCT!	2
/* no further processing!	1
/* no map fields in batch	1
/* no member specified	3
/* no message, selected	1

Abbildung 27: Das Fenster Zusammenfassung Kommentare von Qualizer/Info

Dieses Werkzeug listet alle Kommentare, die in der gesamten Anwendung vorhanden sind. Das Fenster Relation zeigt die Liste der Objekte, in denen der ausgewählte Kommentar benutzt wird.

Durch Auswahl eines Objekts im Relation-Fenster wird wiederum die zugehörige Source angezeigt und positioniert.

## Qualizer®/Test

### Einführung in Qualizer/Test

Qualizer/Test ist ein Werkzeug, das speziell für den Test und die Wartung von NATURAL-Objekten entwickelt wurde. Es leistet

- Vergleich von Sourcen (neue/alte Version)  
mit Angabe über die Anzahl der geänderten, hinzugefügten und entfernten Zeilen und übersichtliche Gegenüberstellung der Änderungen
- Analyse der Testpfade  
mit Auflistung von zu testenden Bedingungen bzw. Werten (Variableninhalten)

Für jeden ermittelten Testfall kann die entsprechende Stelle in der Source positioniert und angezeigt werden.

Qualizer/Test erleichtert die Testfallermittlung ganz erheblich.

### Anwendung von Qualizer/Test

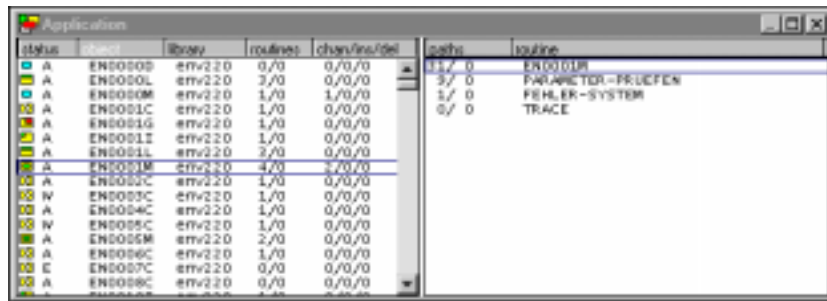


Abbildung 28: Das Fenster Objektauswahl von Qualizer/Test

Im Application-Fenster sehen Sie neben der Liste der Objekte die Anzahl der internen Routinen (Hauptroutine + Anzahl interne Subroutinen) sowie die Änderungen gegenüber der vorherigen Version. Auf der rechten Seite dieses Fensters werden die einzelnen Routinen mit Ihrer Anzahl ermittelter Testfälle gelistet.

Klicken Sie nun auf eine der Routinen erhalten Sie im Test-Fenster zunächst die Liste der Testfälle mit der ermittelten Länge. Wählen Sie einen Testfall aus, so werden die zugehörigen Bedingungen, Werte und Ausdrücke im rechten Teil des Test-Fensters gelistet:

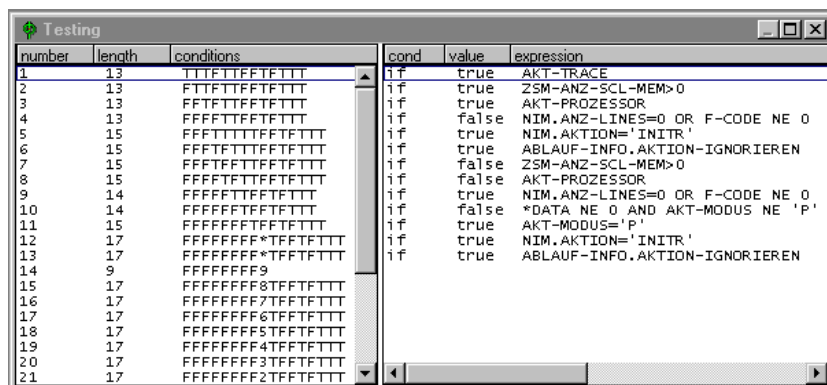


Abbildung 29: Das Fenster Testfälle von Qualizer/Test

Für jeden ermittelten Testfall kann die entsprechende Stelle in der Source positioniert und angezeigt werden. Durch Klick an die gewünschte Stelle im Test-Fenster wird das Source-Fenster automatisch auf die zugehörige Code-Zeile positioniert:



Abbildung 30: Das Fenster Source-Vergleich von Qualizer/Test

Die zugehörige Code-Zeile ist **rot** markiert. Änderungen zwischen aktueller und Vorversion der Source werden zusätzlich **grün** hinterlegt.

Desweiteren können Testpfade und einzelne Bedingungen im → *Kontrollflußgraphen* nochvollzogen werden: Die zum markierten Pfad gehörende Kante wird durch Farbänderung in **blau** hervorgehoben, der zur markierten Bedingung gehörige Knoten wird **gelb** hervorgehoben.



# Systemvoraussetzungen

**Inhalt**

Dieses Kapitel erläutert, in welchen Systemumgebungen Qualizer eingesetzt werden kann und welchem Hardware-Voraussetzungen erfüllt sein sollten bzw. empfohlen werden.

## Inhalt dieses Kapitels

---

HARDWARE.....	30
SOFTWARE .....	31
INSTALLATION .....	31

# Systemvoraussetzungen

## Hardware

### Minimale Anforderungen

Merkmal	Anforderungen
PC	IBM PC oder 100% kompatibler Rechner mit Microsoft Windows 3.1, Windows 95, Windows 98 oder Windows NT 4.0
Prozessor	Intel 386
Hauptspeicher	Windows 3.1 min. 8 MB, Windows 95/98 min. 16 MB, Windows NT min. 32 MB.
Festplattenspeicher	Min. 4 MB freier Plattenplatz (nur Anwendung)
Grafikkarte	Min. Auflösung 800x600 Pixel bei 256 Farben, kompatibel mit MS Windows
Drucker	Kompatibel mit MS Windows
Maus	Kompatibel mit MS Windows

### Empfohlene Konfiguration

Die nachfolgende Aufstellung geht davon aus, daß Sie unter optimalen Voraussetzungen arbeiten möchten. Bessere Eigenschaften sind keinesfalls schädlich, jede Konfiguration zwischen der minimalen und der empfohlenen Konfiguration ist möglich.

Merkmal	Anforderungen
PC	IBM PC oder 100% kompatibler Rechner mit Microsoft Windows 98 oder Windows NT 4.0
Prozessor	Intel Pentium II mit 400 MHz
Hauptspeicher	64 MB.
Festplattenspeicher	4 MB freier Plattenplatz (nur Anwendung) + ca. 40 MB für Sourcen und Spooling
Grafikkarte	Auflösung 1280x1024 Pixel bei 256 Farben, kompatibel mit MS Windows, kompatibler Bildschirm
Drucker	Kompatibel mit MS Windows
Maus	Kompatibel mit MS Windows

## Software

- Windows 3.1 oder höher (Windows 95/98 oder Windows NT)
- Windows muß lokal installiert sein, keine reine Netzwerk-Installation.
- Windows 3.1 bis Windows 98:
  - Minimal 12 MB freier virtueller Speicher. Das Swapfile muß aktiviert sein.
  - Das DOS-Programm SHARE muß über AUTOEXEC geladen sein.
  - In der Datei CONFIG.SYS muß die Angabe "FILES=30" oder höher gesetzt werden.
- Windows 3.1:
  - Microsoft Win32s ist erforderlich. Win32s wird mit Qualizer ausgeliefert.

## Installation

### Qualizer®

Die Installation erfolgt nach den üblichen Verfahren. Details sind der jeweils aktuellsten Dokumentation zu entnehmen.

### Win32s

Qualizer ist eine 32-Bit-Anwendung. Zur Ausführung unter Windows 3.1 muß daher die von Microsoft bereitgestellte Erweiterung Win32s installiert werden (erfolgt automatisch, kostenfrei).



# Anhang

**Inhalt** Der Anhang enthält das Verzeichnis der Abbildungen sowie ein Glossar mit einigen Begriffserläuterungen.

## Inhalt dieses Kapitels

---

ABBILDUNGSVERZEICHNIS.....	34
GLOSSAR.....	35

# Anhang

## Abbildungsverzeichnis

Abbildung 1:	Das Qualizer Anwendungsfenster am Beispiel von Qualizer/Info.....	6
Abbildung 2:	Die obere Werkzeugleiste .....	6
Abbildung 3:	Die untere Werkzeugleiste mit Qualizer/Audit .....	7
Abbildung 4:	Die untere Werkzeugleiste mit Qualizer/Doc.....	7
Abbildung 5:	Die untere Werkzeugleiste mit Qualizer/Info.....	7
Abbildung 6:	Die untere Werkzeugleiste mit Qualizer/Test .....	8
Abbildung 7:	Das Anwendungsfenster von Qualizer/Audit.....	12
Abbildung 8:	Das Anwendungsfenster von Qualizer/Doc .....	14
Abbildung 9:	Das Surveyfenster von Qualizer/Doc .....	14
Abbildung 10:	Das Quality-Fenster von Qualizer/Info .....	15
Abbildung 11:	Das Kiviat-Diagramm-Fenster von Qualizer/Info.....	16
Abbildung 12:	Das Call-Graph-Fenster von Qualizer/Info .....	17
Abbildung 13:	Das Call-Graph-Werkzeugleiste.....	17
Abbildung 14:	Das Struktur-Diagramm-Fenster von Qualizer/Info.....	18
Abbildung 15:	Das Struktur-Diagramm-Werkzeugleiste .....	18
Abbildung 16:	Das Kontrollflußgraph-Fenster von Qualizer/Info .....	20
Abbildung 17:	Das Maskenbild-Fenster von Qualizer/Info .....	22
Abbildung 18:	Das Files-Fenster von Qualizer/Info .....	22
Abbildung 19:	Das Variablen-Fenster von Qualizer/Info .....	23
Abbildung 20:	Das Konstanten-Fenster von Qualizer/Info .....	23
Abbildung 21:	Das Fenster Zusammenfassung Dateien von Qualizer/Info .....	23
Abbildung 22:	Das Fenster Dateireferenzierungen von Qualizer/Info .....	24
Abbildung 23:	Das Fenster Zusammenfassung Variablen von Qualizer/Info.....	24
Abbildung 24:	Das Fenster Variablenreferenzierungen von Qualizer/Info.....	24
Abbildung 25:	Das Fenster Zusammenfassung Konstanten von Qualizer/Info.....	24
Abbildung 26:	Das Fenster Zusammenfassung Dateifelder von Qualizer/Info.....	25
Abbildung 27:	Das Fenster Zusammenfassung Kommentare von Qualizer/Info.....	25
Abbildung 28:	Das Fenster Objektauswahl von Qualizer/Test .....	26
Abbildung 29:	Das Fenster Testfälle von Qualizer/Test .....	26
Abbildung 30:	Das Fenster Source-Vergleich von Qualizer/Test .....	27

# Glossar

## Halstead Metriken

Die von Halstead vorgeschlagenen Metriken basieren auf der Anzahl der in einem Programm benutzten unterschiedlichen Operanden und Operatoren sowie auf der Gesamtzahl der vorhandenen Operanden und Operatoren.

## Zyklomatische Komplexität

Die Zyklomatische Komplexität ist eine Größe zur Messung der strukturellen Komplexität von Programmen. Sie beeinflusst die Testbarkeit, Wartbarkeit und Lesbarkeit von Programmen. Die Basis für die Berechnung bildet der  $\rightarrow$  Kontrollflußgraph.

*McCabe* mißt die Graphkomplexität eines Programmes anhand der Kanten ( $e$ ), Knoten ( $n$ ) und Teilgraphen ( $p$ ). Seine Formel für die zyklomatische Komplexität eines Programmablaufgraphen  $V(G)$  lautet:

$$V(G) = e - n + (2 \times p)$$

Dies ist eines der einfachsten und meistbenutzten Maße für Programmkomplexität auf der Basis der Graphentheorie.



# Stichwortverzeichnis

## A

Anomalien • 11  
 Anomalies • 12  
 Anwendung • 14  
 Anwendungsfenster • 6  
 Anwendungsgutachten • 13  
 Application • 12  
*Application Summary Page* • 13  
 ASCII • 13  
 AT • 19  
 Audit • 11, 12  
 AUDIT  
   Werkzeugleiste • 7  
 Aufrufstruktur • 17

## B

BACKOUT TRANSACTION • 19  
 Beschreibungen  
   extrahieren • 13  
 Bildschirm-orientierte Statements • 18  
*Block* • 20  
 bug prediction • 15

## C

CALL • 19  
 Call Graph • 13  
 Call-Graph • 17  
 CALLNAT • 19  
 CLEAN • 9  
 CLOSE WORK FILE • 19  
 COBOL • 11  
 COMPARE • 9  
 Cyclomatic complexity • 13

## D

Dateien • 15  
 Datenbank-orientierte Statements • 18  
 Datenbereich • 13  
 Datenbereiche • 13  
**DECIDE** • 11, 19  
 DISPLAY • 19  
 Doc • 13  
 DOC  
   Werkzeugleiste • 7

## E

ELSE • 18  
 END TRANSACTION • 19  
**ESCAPE** • 11  
 ESCAPE BOTTOM • 21  
 ESCAPE TOP • 21  
 Extraktion • 13

## F

Feld • 25  
 Felder • 15  
**FETCH** • 11, 17, 19  
 File • 13, 22, 23  
 FIND • 19  
**FIND SORTED** • 11  
 FOR • 21  
 function point  
   backfired • 15

## G

GET • 19  
 GET TRANSACTION • 19  
 Gutachten • 13

## H

Halstead • 15  
 Halstead length • 15  
 Halstead Metriken • 16  
 Hecht • 21  
 HISTOGRAM • 19  
 HTML • 9  
 HTML-Format • 9  
 HTMLGEN • 9

## I

IF • 19  
 IF NOT FOUND • 19  
 IMPORT • 9  
 Info • 15  
 INFO  
   Werkzeugleiste • 7  
 INPUT • 19  
 Instruktionen

Anzahl • 15  
Intranet • 9

**J**

JCL • 11

**K**

Kante • 20  
Kennzahlen • 13  
Kiviat-Diagramm • 16  
Knoten • 20  
Knotenarten • 21  
Kommentar • 15, 25  
    trivial • 15  
**Komplexer logischer Ausdruck** • 11  
Komplexität • 14  
    zyklomatische • 13, 15  
Konstante • 23, 24  
Kontrollflußgraph • 20  
Kontroll-Statements • 18  
Kontrollübergabe-orientierte Statements • 18

**L**

Label • 11  
Lines of code • 15

**M**

Map • 15, 22  
MAP • 19  
Map-Layout • 13  
Maske • 13  
McCabe • 15

**N**

NATURAL • 11

**O**

Objektgröße • 14  
ON ERROR • 19

**P**

PERFORM • 19  
Programmablauf-Plan • 20  
Programmfehler • 15  
Programmkopf • 13  
Programm-Listing • 13

**Q**

Qualität • 15  
Qualizer/DDM • 9  
Qualizer/LDA • 9

**R**

READ • 19  
READ WORK FILE • 19  
Re-Engineering • 14  
REINPUT • 19  
Rekursion • 17  
Relation • 24  
REMEDY • 9  
REPEAT • 21  
Rich Text Format • 9  
RTF • 9, 13, 14  
RUN • 19

**S**

*Segment* • 20  
SHARE.EXE • 31  
*Source Object Summary Pages* • 13  
STACK • 9  
STORE • 19  
Strukturdiagramm • 18  
Subroutine  
    interne • 13, 15  
**Subroutines**  
    **unbenutzte** • 11  
Support • 3  
Survey • 14

**T**

Test • 26  
TEST  
    Werkzeugleiste • 8  
Testfall • 26  
Testfalllänge • 26  
Testfallermittlung • 26  
Testpfad • 26  
THEN • 18  
Toleranzbereich • 16  
TXT • 9

**U**

Umstellungshilfe • 9  
UPDATE • 19  
USING • 15  
Utilities  
    weitere • 9

**V**

V(G) • 15  
Variable • 23, 24  
Variablen • 12  
    global • 15  
    lokal • 15  
**Verschachtelung** • 11  
Verschachtelungstiefe • 15  
Versionsvergleich • 13  
View • 13  
Virtual Memory • 31

**W**

Werkzeugleisten • 6  
WHEN • 18  
Windows Meta File • 9  
WMF • 9  
Workfile-orientierte Statements • 18  
WRITE • 19  
WRITE WORK FILE • 19

**Z**

**Zeilennummer** • 11  
Zweig • 21  
Zyklomatische Komplexität • 13, 16

