

**Agilent VEE Pro**

**VEE Pro  
Benutzerhandbuch**



**Agilent Technologies**

## Anmerkungen

© Agilent Technologies, Inc. 2003

Die Vervielfältigung, elektronische Speicherung, Anpassung oder Übersetzung dieses Handbuchs ist gemäß den Bestimmungen des Urheberrechtsgesetzes ohne vorherige schriftliche Genehmigung durch Agilent Technologies verboten.

### Handbuch-Teilenummer

E2120-90013

### Ausgabe

Version 7.0

Eighth edition, February 2004

Printed in USA

Agilent Technologies, Inc.  
815 14 Street SW  
Loveland, CO 80537USA

## Gewährleistung

**Agilent Technologies behält sich vor, die in diesem Handbuch enthaltenen Informationen jederzeit ohne Vorankündigung zu ändern. Agilent Technologies übernimmt keinerlei Gewährleistung für die in dieser Dokumentation enthaltenen Informationen, insbesondere nicht für deren Eignung oder Tauglichkeit für einen bestimmten Zweck. Agilent Technologies übernimmt keine Haftung für Fehler, die in diesem Dokument enthalten sind, und für zufällige Schäden oder Folgeschäden im Zusammenhang mit der Lieferung, Ingebrauchnahme oder Benutzung dieser Dokumentation. Falls zwischen Agilent und dem Benutzer eine schriftliche Vereinbarung mit abweichenden Gewährleistungsbedingungen hinsichtlich der in diesem Dokument enthaltenen Informationen existiert, so gelten diese schriftlich vereinbarten Bedingungen.**

## Technolizenzien

Die in diesem Dokument beschriebene Hardware und/oder Software wird/werden unter einer Lizenz geliefert und dürfen nur entsprechend den Lizenzbedingungen genutzt oder kopiert werden.

## Nutzungsbeschränkungen

Wenn Software für den Gebrauch durch die US-Regierung bestimmt ist, wird sie als „kommerzielle Computer-Software“ gemäß der Definition in DFAR 252.227-7014 (Juni 1995), als „kommerzielle Komponente“ gemäß der Definition in FAR 2.101(a), als „nutzungsbeschränkte Computer-Software“ gemäß der Definition in FAR 52.227-19 (Juni 1987) (oder einer vergleichbaren Agentur- oder Vertragsregelung) ausgeliefert und lizenziert. Nutzung, Vervielfältigung oder Weitergabe von Software unterliegt den standardmäßigen Bestimmungen für kommerzielle Lizenzen von Agilent Technologies. US-Regierung und -Behörden (außer

Verteidigungsministerium) erhalten keine Rechte, die über die Rechte an „nutzungsbeschränkter Computer-Software“ gemäß FAR 52.227-19(c)(1-2) (Juni 1987) hinausgehen. Zur US-Regierung zählende Benutzer erhalten keine Rechte, die über die Rechte an „nutzungsbeschränkter Computer-Software“ gemäß FAR 52.227-14 (Juni 1987) oder DFAR 252.227-7015 (b)(2) (November 1995) hinausgehen, soweit in irgendwelchen technischen Daten anwendbar.

## Sicherheitshinweise

### VORSICHT

Der Hinweis **VORSICHT** macht auf Arbeitsweisen, Anwendungen o. ä. aufmerksam, die bei falscher Ausführung zur Beschädigung des Produkts oder zum Verlust wichtiger Daten führen können. Wenn eine Prozedur mit dem Hinweis **VORSICHT** gekennzeichnet ist, dürfen Sie erst fortfahren, wenn Sie alle angeführten Bedingungen verstanden haben und diese erfüllt sind.

### WARNUNG

Der Hinweis **WARNUNG** macht auf Arbeitsweisen, Anwendungen o. ä. aufmerksam, die bei falscher Ausführung zu Personenschäden, u. U. mit Todesfolge, führen können. Wenn eine Prozedur mit dem Hinweis **WARNUNG** gekennzeichnet ist, dürfen Sie erst fortfahren, wenn Sie alle angeführten Bedingungen verstanden haben und diese erfüllt sind.

# **VEE Pro Benutzerhandbuch**

In diesem Handbuch verwendete Konventionen 4

## In diesem Handbuch verwendete Konventionen

In diesem Handbuch werden folgende typografische Konventionen verwendet:

**Tabelle 1**

<i>Erste Schritte</i>	Text in Kursivschrift wird zur Hervorhebung verwendet.
<b>File</b>	Text in Computerschrift kennzeichnet Text, der auf dem Bildschirm angezeigt wird, z. B. Menübefehle, Funktionen und Schaltflächen.
<i>dir dateiname</i>	In diesem Kontext steht die Computerschrift für ein Argument, das Sie genau wie angegeben eintippen müssen. Der Text in Kursivschrift steht für ein Argument, das Sie durch einen tatsächlichen Wert ersetzen.
<b>File ⇒ Open</b>	Das Zeichen „⇒“ wird als Kurzschrift zur Darstellung der Position der Agilent VEE-Menübefehle verwendet. „ <b>File ⇒ Open</b> “ bedeutet beispielsweise, dass Sie im Menü <b>File</b> den Befehl <b>Open</b> wählen müssen.
Sml   Med   Lrg	Auswahloptionen in Computerschrift, die durch einen senkrechten Strich (   ) getrennt sind, zeigen an, dass Sie eine dieser Optionen wählen müssen.
Drücken Sie die <b>Eingabetaste</b>	In diesem Kontext kennzeichnet die Fettschrift eine Taste auf der Tastatur.
Drücken Sie Strg + O	Steht für eine Kombination von Tasten auf der Tastatur, die gleichzeitig gedrückt werden müssen.

# Inhaltsverzeichnis

## VEE Pro Benutzerhandbuch

In diesem Handbuch verwendete Konventionen .....	4
--	---

## Einführung

Einführung .....	2
Überblick über Agilent VEE .....	3
Vorteile von Agilent VEE bei der Testentwicklung .....	3
Benutzerschnittstellen in Agilent VEE erstellen .....	7
Vorhandene Testprogramme mit Agilent VEE nutzen .....	9
Instrumente mit Agilent VEE steuern .....	10
Testmöglichkeiten mit Hilfe von Agilent VEE verbessern .....	10
Agilent VEE installieren und kennen lernen .....	12
Agilent VEE und E/A-Bibliotheken installieren .....	12
Agilent VEE kennen lernen .....	13
Eine kostenlose Testversion der Software anfordern .....	14
MATLAB Script – Überblick .....	15
Toolbox zur Signalverarbeitung .....	15
Informationen zur MATLAB-Vollversion .....	16
Support zu Agilent VEE anfordern .....	17
Abrufen von Informationen über das World Wide Web .....	17
Zusätzliche Informationsquellen zu MATLAB .....	18

## 1 Die VEE-Entwicklungsumgebung verwenden

Die VEE-Entwicklungsumgebung verwenden .....	20
Überblick .....	21

Interaktion mit Agilent VEE .....	22
Unterstützte Systeme .....	22
Die Maus und die Menüs .....	22
Agilent VEE starten .....	23
Das Agilent VEE-Fenster .....	23
Hilfe aufrufen .....	25
Mit Objekten arbeiten .....	30
Objekte auf dem Arbeitsbereich platzieren .....	30
Die Ansicht von Objekten ändern .....	32
Ein Objektmenü öffnen .....	34
Ein Objekt verschieben .....	35
Ein Objekt duplizieren (klonen) .....	36
Ein Objekt kopieren .....	37
Ein Objekt löschen .....	37
Ein Objekt einfügen .....	38
Die Größe eines Objekts ändern .....	38
Den des Namen (Titel) eines Objekts ändern .....	39
Objekte wählen oder Auswahl aufheben .....	40
Mehrere Objekte wählen .....	41
Alle Objekte wählen/Auswahl aufheben .....	41
Mehrere Objekte kopieren .....	42
Objekte bearbeiten .....	42
Datenlinien zwischen Objekten erstellen .....	43
Datenlinien zwischen Objekten löschen .....	44
Den sichtbaren Teil des Arbeitsbereichs verschieben .....	45
Den Arbeitsbereich löschen .....	46
Standardeinstellungen ändern .....	46
Informationen zu Pins und Anschlüssen .....	49
Einen Anschluss hinzufügen .....	51
Anschlussinformationen bearbeiten .....	52
Einen Anschluss löschen .....	54

Objekte zum Erstellen eines Programms verbinden .....	56
Übung 1-1 Das Programm Display Waveform (Wellenform anzeigen) .....	56
Ein Programm ausführen .....	58
Objekteigenschaften ändern .....	59
Die Anzeige drucken .....	62
Ein Programm speichern .....	63
Agilent VEE beenden .....	65
VEE erneut starten und ein Programm ausführen .....	66
Mehrere Fenster im Arbeitsbereich verwalten .....	67
Die Arbeitsweise von Agilent VEE-Programmen .....	69
Übung 1-2: Datenfluss und Weitergabe anzeigen .....	70
Übung 1-3: Einen Noise Generator hinzufügen .....	71
Übung 1-4: Einen Amplituden-Eingang und einen Real64 Slider hinzufügen .....	73
Kapitel-Checkliste .....	76

## **2 Agilent VEE Programmiertechniken**

Agilent VEE Programmiertechniken .....	78
Übersicht .....	79
Allgemeine Techniken .....	80
Übung 2-1: Erstellen eines UserObject .....	80
Übung 2-2: Ein Dialogfeld für die Benutzereingabe erstellen .....	87
Übung 2-3: Datendateien verwenden .....	90
Übung 2-4: Eine Fensteransicht (Benutzerschnittstelle) erstellen .....	94
Übung 2-5: Mathematische Verarbeitung von Daten .....	96
Die Online-Hilfe .....	102
Verwenden der Hilfefunktion .....	102
Hilfethemen zu einem Objekt anzeigen .....	103
Die Menüposition eines Objekts suchen .....	104
Weitere praktische Übungen mit der Hilfefunktion .....	104

Programme in Agilent VEE debuggen .....	105
Den Datenfluss anzeigen .....	105
Den Ausführungsfluss anzeigen .....	106
Daten an einer Linie untersuchen .....	107
Anschlüsse überprüfen .....	109
Alphanumerischen Anzeigen für die Fehlerbehebung .....	109
Unterbrechungspunkte einsetzen .....	110
Fehlerbehebung .....	112
Die Schaltfläche „Go To“ zum Suchen eines Fehlers .....	112
Den Aufrufstapel zur Fehlersuche nutzen .....	113
Die Ereignisreihenfolge in einem Objekt verfolgen .....	115
Die Ausführungsreihenfolge von Objekten in einem Programm verfolgen .....	117
Schrittweise Ausführung eines Programms .....	118
Suchen eines Objekts in einem komplexen Programm .....	120
Übungen mit Programmen .....	121
2-6: Generieren einer Zufallszahl .....	121
2-7: Eine globale Variable setzen und abrufen .....	122
Agilent VEE-Programme dokumentieren .....	125
Objekten mit dem Dialogfeld Description dokumentieren .....	125
Die Dokumentation automatisch generieren .....	127
Kapitel-Checkliste .....	130

### **3 Einfache Methoden zum Steuern von Instrumenten**

Einfache Methoden zum Steuern von Instrumenten .....	132
Überblick .....	133
Panel-Treiber .....	133
Das Objekt „Direct I/O“ .....	134
PC-Zusatzkarten mit E/A-Bibliothek .....	135
VXIPlug&Play-Treiber .....	135



Instrumente konfigurieren .....	136
Übung 3-1: Ein aktuell nicht vorhandenes Instrument konfigurieren .....	136
Ein Instrument für die Verwendung in einem Programm wählen .....	142
Das physische Instrument zur Konfiguration hinzufügen .....	143
Einen Panel-Treiber verwenden .....	145
Übung 3-2: Die Einstellungen eines Panel-Treibers ändern .....	145
Wechseln zu anderen Fenstern des gleichen Treibers .....	147
Dateneingangs- oder -ausgangsanschlüsse löschen .....	149
Vertiefung .....	149
Das Objekt Direct I/O verwenden .....	150
Übung 3-3: Direct I/O verwenden .....	150
Einen einzelnen Textbefehl an ein Instrument senden .....	151
Eine Ausdrucksliste an ein Instrument senden .....	153
Daten von einem Instrument einlesen .....	154
Upload und Download von Statusangaben zu Instrumenten .....	158
PC-Zusatzkarten .....	160
Data Translation Visual Programming Interface (VPI) .....	160
Amplicon .....	160
PC-Zusatzkarten von ComputerBoards .....	161
ME-DriverSystem von Meilhaus Electronic .....	163
Einen VXIPlug&Play-Treiber verwenden .....	166
Übung 3-4: Einen VXIPlug&Play-Treiber konfigurieren .....	166
Weitere E/A-Funktionen .....	171
Kapitel-Checkliste .....	172
<b>4 .....</b>	<b>Testdaten analysieren und anzeigen</b>
Testdaten analysieren und anzeigen .....	174
Überblick .....	175
Agilent VEE: Datenformen und Datentypen .....	176

Agilent VEE-Analysefunktionen .....	179
Integrierte mathematische Objekte verwenden .....	180
Integrierte Operatoren oder Funktionen aufrufen .....	180
Übung 4-1: Die Standardabweichung berechnen .....	182
Ausdrücke mit dem Objekt Formula erstellen .....	184
Einen Ausdruck mit dem Objekt Formula auswerten .....	185
Eine Agilent VEE-Funktion im Objekt Formula verwenden .....	186
Vertiefung .....	189
MATLAB Script in Agilent VEE verwenden .....	190
Das Objekt MATLAB Script in Agilent VEE verwenden .....	194
Mit Datentypen arbeiten .....	195
Anzeigen von Testdaten .....	198
Testdatenanzeigen anpassen .....	200
Eine Wellenform anzeigen .....	200
Die X- und Y-Skala ändern .....	201
Einen Teil der Wellenform zoomen .....	201
Delta-Markierungen in der Anzeige hinzufügen .....	202
Die Farbe des Trace ändern .....	203
Zur weiteren Übung .....	204
Kapitel-Checkliste .....	205

## **5 Testergebnisse speichern und abrufen**

Testergebnisse speichern und abrufen .....	208
Überblick .....	209
Testergebnisse in Arrays speichern .....	210
Übung 5-1: Ein Array für Testergebnisse erstellen .....	211
Übung 5-2: Werte aus einem Array extrahieren .....	213

Die Objekte To/From File verwenden .....	214
Die Arbeitsweise der E/A-Transaktionen .....	215
E/A-Transaktionsformat .....	216
Übung 5-3: Die Objekte To/From File verwenden .....	218
Eine Textzeichenfolge an eine Datei senden .....	218
Einen Zeitstempel an eine Datei senden .....	220
Ein Real Array an eine Datei senden .....	221
Daten mit dem Objekt „From File“ abrufen .....	223
Records (Datensätze) zum Speichern gemischter Datentypen verwenden .....	227
Einen Record erstellen .....	228
Ein Feld aus einem Record abrufen .....	231
Ein Feld in einem Record setzen .....	233
Einen Record in einem einzigen Arbeitsschritt auflösen .....	236
DataSets zum Speichern und Abrufen von Records verwenden .....	238
Übung 5-5: DataSets verwenden .....	238
Einen Record in einem DataSet speichern und von dort einlesen .....	238
Eine einfache Testdatenbank anpassen .....	243
Übung 5-6: Such- und Sortieroperationen mit DataSets verwenden .....	243
Eine Suchoperation mit DataSets durchführen .....	243
Eine Benutzerschnittstelle für eine Suchoperation erstellen .....	244
Eine Sortieroperation mit einem Record-Feld ausführen .....	250
Kapitel-Checkliste .....	252

## **6 Berichte mit ActiveX bequem erstellen**

Berichte mit ActiveX bequem erstellen .....	254
Überblick .....	255
ActiveX-Automatisierung in Agilent VEE .....	256
Typenbibliotheken zur ActiveX-Automatisierung auflisten .....	256
ActiveX-Programme mit Agilent VEE erstellen und verwenden .....	257

Operationen mit ActiveX-Anweisungen durchführen	258
CreateObject und GetObject verwenden	260
Agilent VEE-Daten an MS Excel senden	261
Übung 6-1: Agilent VEE-Daten an MS Excel senden	261
Erstellen einer Agilent VEE-Vorlage für MS Excel	271
Übung 6-2: Eine Agilent VEE-Vorlage für MS Excel erstellen	271
Vertiefung	273
Die Möglichkeiten mit MS Excel erweitern	274
MS Word für Agilent VEE-Berichte verwenden	276
Übung 6-3: MS Word für Agilent VEE-Berichte verwenden	276
Kapitel-Checkliste	285

## **7 .NET mit VEE verwenden**

.NET mit VEE verwenden	288
Was ist .NET?	289
VEE und .NET Framework	290
.NET-Assembly-Verweise	291
Einen Namespace in VEE importieren	296
VEE und primäre Interop-Assemblies	300
Programmierverfahren	301
Datentypen zwischen .NET und VEE konvertieren	301
Instanzmethode aufrufen	311
Eine freigegebene/statische Methode aufrufen	312
.NET-Programmiertipps	313
Übung 7-1: .NET zur Wahl von Dateien verwenden	314
Übung 7-2: .NET zum Ausführen von DateTime-Operationen verwenden	317
Übung 7-3: .NET zum Abrufen von Dateiinformatoren verwenden	322

.NET- und IVI-Treiber .....	324
Assemblies .....	326
Installieren einer neuen Assembly .....	326
Aktualisieren einer Assembly .....	326
Die VEE-Laufzeitversion weitergeben .....	327
VEE- und .NET-Sicherheit .....	328
.NET-Terminologie .....	329
Assembly .....	329
Primäre Interop-Assembly (PIA) .....	329
Namespace .....	329
Verweis .....	330
Klasse .....	330
Freigegebene oder statische Member .....	330
Instanzmember .....	331
Checkliste für das Kapitel .....	332

## **8 Programme für den PC integrieren**

Integrieren von Programmen in anderen Sprachen .....	334
Überblick .....	335
Die Arbeitsweise des Objekts „Execute Program“ .....	336
Mit dem Objekt „Execute Program“ arbeiten .....	336
Einen Systembefehl verwenden .....	339
Übung 8-1: Einen Systembefehl verwenden .....	339
Schreiben von einfach portierbaren Programmen .....	341
Checkliste für das Kapitel .....	343

## **9 Agilent VEE-Funktionen verwenden**

Agilent VEE-Funktionen verwenden .....	346
Überblick .....	347

Mit Funktionen arbeiten .....	348
Eine Agilent VEE-Funktion definieren .....	348
Die Unterschiede zwischen UserObjects und UserFunctions .....	349
Übung 9-1: Operationen mit UserFunctions .....	350
Erstellen einer UserFunction .....	350
Ändern einer UserFunction .....	353
Eine UserFunction über einen Ausdruck aufrufen .....	355
Den Aufruf einer UserFunction generieren .....	357
UserFunctions und der Program Explorer .....	359
Bibliotheken mit Agilent VEE UserFunctions verwenden .....	361
Übung 9-2: Eine Bibliothek von UserFunctions erstellen und zusammenführen .....	362
Erstellen einer Bibliothek von UserFunctions .....	362
Ein weiteres Programm erstellen und in die Bibliothek einfügen .....	367
Übung 9-3: Bibliotheken importieren und löschen .....	368
Funktionen in großen Programmen finden .....	373
Agilent VEE-Programme zusammenfügen .....	375
Übung 9-4: Ein Programm zur Anzeige von Balkendiagrammen zusammenfügen .....	375
Kapitel-Checkliste .....	377

## **10 Tests sequenzieren**

Tests sequenzieren .....	380
Überblick .....	381
Das Sequencer-Objekt verwenden .....	383
Eine Reihenfolge für die Ausführung eines Tests erstellen .....	384
Übung 10-1: Einen Test konfigurieren .....	384
Einen Test hinzufügen, einfügen oder löschen .....	392
Auf die protokollierten Testdaten zugreifen .....	394

Daten im Sequencer weitergeben .....	397
Übung 10-2: Daten mit einem Eingangsanschluss weitergeben .....	397
Daten mit einer globalen Variablen weitergeben .....	400
Eine Wellenformausgabe mit einer Maske vergleichen .....	404
Daten aus dem Sequencer analysieren .....	409
Übung 10-3: Mehrere Durchläufe von Daten aus dem Sequencer analysieren .....	410
Protokollierte Daten speichern und abrufen .....	413
Übung 10-4: Die Objekte „To/From File“ mit protokollierten Daten verwenden .....	413
Die Objekte „To/From DataSet“ mit protokollierten Daten verwenden .....	414
Kapitel-Checkliste .....	416

## **11 Benutzerschnittstellen verwenden**

Benutzerschnittstellen verwenden .....	418
Überblick .....	419
Wichtige Überlegungen zum Thema Benutzerschnittstellen .....	420
Eine Benutzerschnittstelle erstellen .....	420
Wechseln zwischen der Fensteransicht und der Detailansicht .....	421
Eine Benutzerschnittstelle anpassen .....	422
Benutzerschnittstellenobjekte verwenden .....	424
Farben, Schriftarten und Anzeigerelemente .....	424
Grafikbilder .....	425
Ein Steuerelement für die Benutzereingabe anzeigen .....	426
Eins Dialogfeld für die Benutzereingabe öffnen .....	429
Ein Schalterelement für den Benutzer anzeigen .....	431
Objekte in der Benutzerschnittstelle ausrichten .....	432
Andere Funktionen für die Fenstergestaltung .....	433
Einer Benutzerschnittstelle für die Bedienung mit der Tastatur erstellen .....	433
Bildschirmfarben wählen .....	435

Ein Programm sichern (Laufzeitversion erstellen)	437
Ein Einblendfenster während der Ausführung anzeigen	437
Eine Statusanzeige erstellen	438
Allgemeine Aufgaben beim Erstellen von Benutzerschnittstellen	439
Übung 11-1: Menüs verwenden	439
Übung 11-2: Bitmaps für den Fensterhintergrund importieren	445
Übung 11-3: Eine auffällige Warnung erstellen	447
Übung 11-4: Ein ActiveX-Steuerelement verwenden	453
Übung 11-5: Ein Statusanzeige erstellen	455
Kapitel-Checkliste	460

## **12 Agilent VEE-Programme optimieren**

Agilent VEE-Programme optimieren	462
Überblick	463
Grundlegende Techniken zur Programmoptimierung	464
Setzen Sie bei der Arbeit mit Arrays, wann immer das möglich ist, mathematische Operationen ein	464
Wenn möglich, Objekte minimiert darstellen	465
Die Anzahl der Programmobjekte verringern	466
Weitere Möglichkeiten zur Optimierung von Agilent VEE-Programmen	468
Überblick über kompilierte Funktionen	471
Vorteile der Verwendung kompilierter Funktionen	471
Überlegungen zum Design bei der Verwendung kompilierter Funktionen	472
Richtlinien zur Verwendung kompilierter Funktionen	473
Dynamic Link Libraries verwenden	475
Eine DLL in ein Agilent VEE-Programm integrieren	475
Ein Beispiel zur Verwendung einer DLL	478
Das Objekt „Execute Program“ im Vergleich zu kompilierten Funktionen	481



Agilent VEE-Ausführungsmodi .....	482
Der Agilent VEE-Compiler .....	483
Den Ausführungsmodus ändern .....	483
Die Schaltfläche Default Preferences in der Symbolleiste .....	484
Die Auswirkung der Änderung des Ausführungsmodus .....	485
Der Agilent VEE Profiler .....	489
Kapitel-Checkliste .....	490

### **13 Plattformspezifische Faktoren und Web-Überwachung**

Plattformspezifische Faktoren und Web-Überwachung .....	492
Überblick .....	493
Der aufrufbare VEE ActiveX-Automatisierungs-Server .....	494
Technologien zur Web-Aktivierung .....	495
Überblick über Web-Technologien .....	495
Web-Überwachung mit Agilent VEE .....	499
Allgemeine Richtlinien und Tipps .....	499
Einem entfernten Benutzer Agilent VEE-Daten bereit stellen .....	500
Wie ein entfernter Benutzer auf Agilent VEE auf Ihrem System zugreift .....	504
Anzeigen der Homepage des Agilent VEE-Web-Servers .....	507
Übung 13-1: Übungs-Session mit Agilent VEE-Web-Browser .....	509
Begrenzen des Zugriffs auf Programme über das Web .....	512
Kapitel-Checkliste .....	516

### **Anhang A: Zusätzliche Übungen**

Zusätzliche Übungen .....	518
Allgemeine Programmiertechniken .....	519
Der Äpfelsammler .....	519
Testen von Zahlen .....	522

Zufallszahlen erfassen .....	525
Zufallszahlen-Generator .....	528
Masken verwenden .....	530
Zeichenfolgen und globale Variablen verwenden .....	534
Zeichenfolgen und globale Variablen manipulieren .....	534
Optimierungstechniken .....	536
UserObjects .....	538
UserObject „Random Noise“ .....	538
Agilent VEE-UserFunctions .....	541
UserFunctions verwenden .....	541
UserFunctions-Bibliotheken importieren und löschen .....	546
Benutzerfenster und Einblendfenster erstellen .....	548
Mit Dateien arbeiten .....	553
Daten in und aus Dateien bewegen .....	553
Records .....	555
Records manipulieren .....	555
Tests sequenzieren .....	561

## **Glossar**

# Abbildungsverzeichnis

Abbildung 1. Die VEE-Entwicklungsumgebung	23
Abbildung 2. Das VEE-Begrüßungsdialogfeld	26
Abbildung 3. Das Menü Help verwenden	27
Abbildung 4. Das Register Contents (Inhalt) der VEE-Hilfe	28
Abbildung 5. Objekte auf dem Arbeitsbereich platzieren	31
Abbildung 6. Ein platziertes Function-Generator-Objekt	32
Abbildung 7. Objekt in offener Ansicht und Symbolansicht	33
Abbildung 8. Ein Objektmenü öffnen	34
Abbildung 9. Ein Objekt verschieben	35
Abbildung 10. Ein Objekt klonen	37
Abbildung 11. Die Größe eines Objekts ändern	39
Abbildung 12. Den Titel eines Objekts ändern	40
Abbildung 13. Gewählte und nicht gewählte Objekte	41
Abbildung 14. Mehrere Objekte beim Kopieren	42
Abbildung 15. Datenlinien zwischen Objekten erstellen	44
Abbildung 16. Bildlaufleisten im Arbeitsbereich	45
Abbildung 17. Das Dialogfeld Default Preferences	47
Abbildung 18. Daten- und Sequenz-Pins	49
Abbildung 19. Show Terminals bei einem Objekt	50
Abbildung 20. Die Eigenschaft ShowTerminals setzen	51
Abbildung 21. Einen Anschluss hinzufügen	52
Abbildung 22. Anschlussinformationen abrufen	53
Abbildung 23. Die Dropdown-Liste verwenden	54
Abbildung 24. Das Dialogfeld „Delete Terminal“	55
Abbildung 25. Ein Programm erstellen	57
Abbildung 26. Ausführen eines Programms	58
Abbildung 27. Eine Sinuskurve für das Feld Function wählen	60
Abbildung 28. Eine Zahl im Feld Frequency markieren	61
Abbildung 29. Beispiel: Den Wert Felds Frequency auf 10 Hz ändern	61
Abbildung 30. Die Anzeige drucken	62

Abbildung 31. Das Dialogfeld Save File .....	63
Abbildung 32. Die Schaltfläche Run in der Symbolleiste .....	66
Abbildung 33. Mehrere Fenster im Arbeitsbereich .....	68
Abbildung 34. Typische Anzeige von simple-program.vee .....	70
Abbildung 35. Beispiel: Ein Noise Generator-Objekt hinzufügen .....	71
Abbildung 36. Function and Object Browser .....	72
Abbildung 37. Beispiel: Eingangsanschlüsse hinzufügen .....	73
Abbildung 38. Beispiel: Ein Real64 Slider-Objekt hinzufügen .....	74
Abbildung 39. Den Werts an einen Ausgangs-Pin anzeigen .....	75
Abbildung 40. Das Fenster UserObject .....	81
Abbildung 41. usobj-program.vee ini einer frühen Phase .....	83
Abbildung 42. Ein UserObject erstellen .....	84
Abbildung 43. UserObject, umbenannt in AddNoise .....	85
Abbildung 44. Gestörte Kosinuskurve .....	86
Abbildung 45. Das Konfigurationsfeld „Int32 Input“ .....	88
Abbildung 46. Int32-Eingabe zu usobj-program.vee hinzugefügt .....	88
Abbildung 47. Eingebledetes Eingabefenster bei Programmausführung .....	89
Abbildung 48. Eine Datendatei hinzufügen .....	90
Abbildung 49. Eine E/A-Transaktion wählen .....	91
Abbildung 50. Ein To File-Objekt hinzufügen .....	92
Abbildung 51. Ein From File-Objekt hinzufügen .....	93
Abbildung 52. simple-program.vee .....	94
Abbildung 53. Beispiel: Eine Fensteransicht erstellen .....	95
Abbildung 54. Verschiedene Datentypen .....	97
Abbildung 55. Datenobjekten verbinden .....	99
Abbildung 56. Ein Programm mit Formula-Objekt erstellen .....	100
Abbildung 57. Datenfluss anzeigen .....	105
Abbildung 58. Datenfluss in simple-program.vee .....	106
Abbildung 59. Ausführungsfluss anzeigen .....	107
Abbildung 60. Anzeige des Werts an einem Ausgangs-Pin .....	108
Abbildung 61. Informationen über eine Linie anzeigen .....	109
Abbildung 62. Unterbrechungspunkte setzen .....	110
Abbildung 63. Programm fortsetzen (gleiche Position wie Schaltfläche „Ausführen“) .....	111

Abbildung 64. Unterbrechungspunkt(e) löschen .....	111
Abbildung 65. Programm anhalten oder stoppen .....	112
Abbildung 66. Beispiel einer Laufzeitfehlermeldung mit Go To .....	113
Abbildung 67. Den Aufrufstapel in Wheel.exe einsetzen .....	114
Abbildung 68. Die Reihenfolge der Ereignisse in einem Objekt .....	115
Abbildung 69. Steuerlinie zur Ausführung eines angepassten Titels .....	117
Abbildung 70. Startobjekte zur Ausführung separater Threads .....	118
Abbildung 71. Schaltflächen „Step Into“, „Step Over“ und „Step Out“ in der Symbolleiste .....	119
Abbildung 72. Das Programm „Random“ .....	122
Abbildung 73. Eine globale Variable setzen und abrufen .....	124
Abbildung 74. Das Dialogfeld „Description“ .....	126
Abbildung 75. Der Anfang der Dokumentationsdatei .....	127
Abbildung 76. Die Mitte der Dokumentationsdatei .....	128
Abbildung 77. Der Rest der Dokumentationsdatei .....	129
Abbildung 78. Der Panel-Treiber für das Oszilloskop HP54600A .....	134
Abbildung 79. Ein Objekt „Direct I/O“ für einen Funktionsgenerator .....	134
Abbildung 80. Eine PC-Zusatzbibliothek importieren .....	135
Abbildung 81. Einen VXIPlug&Play-Treiber in VEE aufrufen .....	135
Abbildung 82. Das Dialogfeld „Instrument Manager“ .....	136
Abbildung 83. Dialogfeld „Instrument Properties“ .....	137
Abbildung 84. Das Dialogfeld „Advanced Instrument Properties“ .....	139
Abbildung 85. Die Registerkarte „Panel Driver“ .....	140
Abbildung 86. Oszilloskop zur Liste der Instrumente hinzugefügt .....	142
Abbildung 87. scope(@(NOT LIVE)) .....	143
Abbildung 88. Das Einblendmenü Function zu fgen .....	146
Abbildung 89. Einblendmenü „Sweep Panel“ im Menü „Discrete Component“ .....	147
Abbildung 90. Die Datenein- und -Ausgangsbereiche für einen Treiber .....	148
Abbildung 91. Die Registerkarte „Direct I/O“ .....	150
Abbildung 92. Ein Objekt „Direct I/O“ .....	151
Abbildung 93. Das Dialogfeld „I/O Transaction“ .....	152
Abbildung 94. Eine Transaktion „Direct I/O“ .....	153
Abbildung 95. Konfiguration von „Direct I/O“ mit einer Eingangsvariablen .....	154
Abbildung 96. Eine READ-Transaktion konfigurieren .....	157

Abbildung 97. Direct I/O konfiguriert zum Einlesen einer Messung .....	157
Abbildung 98. Lernzeichenfolge-Konfiguration für HP54100A .....	159
Abbildung 99. Beispiel für Datenerfassung mit Amplicon .....	161
Abbildung 100. Eine 100 KHz-Karte von ComputerBoards mit VEE verwenden .....	162
Abbildung 101. Importieren der ComputerBoards E/A-Bibliothek .....	163
Abbildung 102. Menü ME Board in VEE .....	163
Abbildung 103. Benutzerfenster für die Datenerfassungskarte ME-3000 .....	164
Abbildung 104. Funktionsfenster für ME-DriverSystem .....	165
Abbildung 105. Einen VXIPlug&Play-Treiber wählen .....	167
Abbildung 106. Eine Funktion für einen VXIPlug&Play-Treiber wählen .....	168
Abbildung 107. Das Dialogfeld „hpe1412 Edit Function“ .....	169
Abbildung 108. Funktion „DC Voltage“ in VXIPlug&Play-Objekt .....	169
Abbildung 109. Registerkarte Configuration im Dialogfeld „Edit Function Panel“ .....	170
Abbildung 110. HPE1412-Treiber bereit für Gleichstrommessung .....	170
Abbildung 111. Eine VEE-Funktion im Function & Object Browser .....	180
Abbildung 112. Eine MATLAB-Funktion in Function & Object Browser .....	181
Abbildung 113. Die Schaltfläche „fx“ zum Öffnen des Function and Object Browser .....	182
Abbildung 114. Die Standardabweichung berechnen .....	183
Abbildung 115. Das Objekt Formula .....	184
Abbildung 116. Einen Ausdruck auswerten .....	186
Abbildung 117. Formelbeispiele mit VEE-Funktionen .....	187
Abbildung 118. VEE-Funktionen mit einem Formelobjekt .....	188
Abbildung 119. Lösung zu den Vertiefungs-Übungen: Ramp und SDEV .....	189
Abbildung 120. MATLAB Script-Objekt in einem VEE-Programm .....	192
Abbildung 121. Vom Programm generierte Grafik .....	193
Abbildung 122. Vordefinierte MATLAB-Objekte .....	195
Abbildung 123. Den Datentyps am Eingangsanschluss ändern .....	197
Abbildung 124. Anzeige einer Wellenform .....	201
Abbildung 125. Delta-Markierungen in einer Wellenform-Anzeige .....	203
Abbildung 126. Der Collector erstellt ein Array .....	212
Abbildung 127. Array-Elemente mit Ausdrücken extrahieren .....	213
Abbildung 128. Das Objekt „To File“ .....	215
Abbildung 129. Ein „I/O Transaction“-Dialogfeld .....	215

Abbildung 130. Das E/A-Transaktionsfeld „TIME STAMP“	221
Abbildung 131. Daten mit dem Objekt „To File“ speichern	222
Abbildung 132. Das String-Format wählen	224
Abbildung 133. Daten mit dem Objekt „From File“ abrufen	226
Abbildung 134. Informationen zum Ausgangsanschluss an einem Record	230
Abbildung 135. Feld „AlphaNumeric Properties“	232
Abbildung 136. Das Objekt „Get Field“ verwenden	233
Abbildung 137. Das Objekt „Set Field“ verwenden	235
Abbildung 138. Das Objekt „UnBuild Record“ verwenden	237
Abbildung 139. Ein Array von Records in einem DataSet speichern	240
Abbildung 140. Daten mit DataSets speichern und einlesen	242
Abbildung 141. Eine Suchoperation mit DataSets	244
Abbildung 142. Das Testmenü-Objekt hinzufügen	246
Abbildung 143. Ein Menüs zur Suchoperation hinzufügen	248
Abbildung 144. Die Benutzerschnittstelle für die Datenbank	249
Abbildung 145. Eine Sortieroperation mit einem Record-Feld	251
Abbildung 146. Das Dialogfeld „ActiveX Automation References“	257
Abbildung 147. Der Datentyp „Object“	258
Abbildung 148. Einrichten eines Excel-Arbeitsblatts zur Anzeige von Testergebnissen	259
Abbildung 149. CreateObject und GetObject	260
Abbildung 150. Die UserFunction „Globals“	262
Abbildung 151. Eine MS Excel-Tabelle einrichten	264
Abbildung 152. Titel und Daten in die Tabelle einfügen	267
Abbildung 153. Das Programm „Results Average“	269
Abbildung 154. Excel-Tabelle für das Programm „Results Average“	270
Abbildung 155. Excel-Tabelle aus dem Array von Testdaten	272
Abbildung 156. Programm für ein Array von Testdaten	272
Abbildung 157. Programm zur weiteren Vertiefung	274
Abbildung 158. Ein Beispielprogramm VEE zu MS Excel	274
Abbildung 159. Objektvariablen	277
Abbildung 160. Anfang des Übungsprogramms 6-3	279
Abbildung 161. ActiveX-Anweisungen hinzufügen	280
Abbildung 162. Das fertige Programm für den Bericht in MS Word	282

Abbildung 163. Das mit Übung 6-3 erstellte MS Word-Dokument	283
Abbildung 164. Das Dialogfeld Import Namespaces	291
Abbildung 165. Function & Object Browser – .NET-Objekte	294
Abbildung 166. Erstellen eines .NET-Objekts	295
Abbildung 167. Assemblies, Namespaces, Typen und Members	297
Abbildung 168. .NET Assembly References – Namespaces importieren	298
Abbildung 169. Listenfeld zur Wahl von Namespaces	299
Abbildung 170. Ein .NET-Objekt erstellen und auf sein Instanz-Member zugreifen	312
Abbildung 171. Statische Methode ohne importierten Namespace	312
Abbildung 172. Statische Methode mit importiertem Namespace	312
Abbildung 173. Function & Object Browser – Erstellen einer Instanz	315
Abbildung 174. Das Programm „openFileDialog“	317
Abbildung 175. Schritt 10 von Übung 7-2	319
Abbildung 176. Übung 7-2 abgeschlossen	321
Abbildung 177. Übung 7-3 abgeschlossen	323
Abbildung 178. Das Objekt „Execute Program“ (PC)	336
Abbildung 179. Dateiliste in einem Verzeichnis	340
Abbildung 180. Systeminformationsfunktionen	341
Abbildung 181. Hauptfenster und Fenster „ArrayStats“	351
Abbildung 182. Die Pins für Call myFunction konfigurieren	351
Abbildung 183. Aufruf der User Function ArrayStats	352
Abbildung 184. Die UserFunction ArrayStats ändern	354
Abbildung 185. Ausgabe an einen Datensatz nach dem Ändern von ArrayStats	355
Abbildung 186. Die User Function ArrayStats aufrufen	356
Abbildung 187. Das Menü „Generate“ in einer UserFunction	358
Abbildung 188. Den Aufruf des Objekts ArrayStats(A) von einer UserFunction aus generieren	359
Abbildung 189. Die Schaltfläche „Program Explorer“ in der Symbolleiste	359
Abbildung 190. Den Program Explorer mit UserFunctions verwenden	360
Abbildung 191. Report.vee von der obersten Ebene	362
Abbildung 192. Die UserFunction „BuildRecAry“	363
Abbildung 193. Die UserFunction „ReportHeader“	364
Abbildung 194. Die UserFunction „ReportBody“	365



Abbildung 195. Die ReportDisplay-Detailansicht	366
Abbildung 196. Die Fensteransicht ReportDisplay	367
Abbildung 197. Die UserFunctions-Bibliothek RepGen.vee	368
Abbildung 198. Eine Funktion aus einer importierten Bibliothek wählen	371
Abbildung 199. Aufrufen einer Funktion aus einer Bibliothek	372
Abbildung 200. Das Dialogfeld „Find“	373
Abbildung 201. Das Dialogfeld „Find Results“	373
Abbildung 202. Ein Balkendiagramm-Programm zusammenfügen	376
Abbildung 203. Das Dialogfeld Sequence Transaction	386
Abbildung 204. Einen Test konfigurieren	387
Abbildung 205. Ein einfaches Sequencer-Beispiel	394
Abbildung 206. Ein oder mehrere protokollierte Records	395
Abbildung 207. Auf protokollierte Daten zugreifen	396
Abbildung 208. Die UserFunction Rand	398
Abbildung 209. Daten mit einem Eingangsanschluss weitergeben	400
Abbildung 210. Die UserFunction „Global“ (Detailansicht)	403
Abbildung 211. Die UserFunction „Global“ (Fensteransicht)	403
Abbildung 212. Daten über eine globale Variable weitergeben	404
Abbildung 213. Die UserFunction „noisyWv“ (Detailansicht)	405
Abbildung 214. Das UserObject „noisyWv“ (Panel)	406
Abbildung 215. Eine Wellenform mit einer Maske vergleichen	408
Abbildung 216. Ein protokolliertes Array aus Records von Records	409
Abbildung 217. Mehrerer Durchläufe von Sequencer-Daten analysieren	412
Abbildung 218. Protokollierte Daten mit „To/From File“ speichern	414
Abbildung 219. Speichern von protokollierten Daten mit „To/From DataSet“	415
Abbildung 220. Schaltflächen für die Fenster- und die Detailansicht in der Titelleiste	421
Abbildung 221. Eine Auswahl der VEE-Anzeigerelementen	423
Abbildung 222. Hintergrundbild, nebeneinander angeordnet	425
Abbildung 223. Ein in VEE zurechtgeschnittenes Bild	426
Abbildung 224. Steuerelemente aus verschiedenen Data-Untermenüs	427
Abbildung 225. Das Dialogfeld Properties	428
Abbildung 226. Ein Texteingabefenster	429
Abbildung 227. Ein Beispiel für die automatische Fehlerprüfung	430

Abbildung 228. Ein Meldungsdialogfeld	430
Abbildung 229. Das Dialogfeld zur Listenauswahl	430
Abbildung 230. Ein Dialogfeld zur Dateiauswahl	431
Abbildung 231. Kombinierte Schalter und Alarme	432
Abbildung 232. Fenstereigenschaften konfigurieren	433
Abbildung 233. Ein Softkey zur Ausführung einer UserFunction	434
Abbildung 234. Das Objekt Confirm (OK) als Softkey konfigurieren	434
Abbildung 235. Das Dialogfeld Default Preferences	436
Abbildung 236. Farbauswahl für Bildelemente	436
Abbildung 237. Eine Statusanzeige erstellen	438
Abbildung 238. Eine frühe Phase im Programm Dice	441
Abbildung 239. Das Programm Dice (Detailansicht)	443
Abbildung 240. Das Programm Dice (Fensteransicht)	444
Abbildung 241. Die Bitmap-Funktion	447
Abbildung 242. Die UserFunction Alarm (Detailansicht)	449
Abbildung 243. Die UserFunction „Warning“ (Detailansicht)	451
Abbildung 244. Das Programm Warning	452
Abbildung 245. Das ActiveX-Steuerelement ProgressBar verwenden	454
Abbildung 246. Beispiel eines ActiveX-Steuerelements mit MSChart	455
Abbildung 247. Test1 konfigurieren	456
Abbildung 248. Die UserFunction „LogTest“ (Detail)	457
Abbildung 249. Die UserFunction „LogTest“ (Fensteransicht)	458
Abbildung 250. Statusanzeigeprogramm (vor der Ausführung)	458
Abbildung 251. Das Statusanzeigeprogramm (während der Ausführung)	459
Abbildung 252. Quadratwurzeln pro Messung berechnen	464
Abbildung 253. Quadratwurzeln mit Math-Array berechnen	465
Abbildung 254. Programme durch Verwendung von Symbolen optimieren	466
Abbildung 255. Funktionsaufrufe ohne Optimierung	467
Abbildung 256. Funktionsaufrufe mit Optimierung	468
Abbildung 257. Eine Bibliothek kompilierter Funktionen importieren	476
Abbildung 258. Call-Objekte für kompilierte Funktionen verwenden	477
Abbildung 259. Ein Programm, das eine DLL (MANUAL49) verwendet	478
Abbildung 260. Das UserObject „Shared Library Name“	480

Abbildung 261. Anzeige des Ausführungsmodus in der VEE-Statusleiste	483
Abbildung 262. Den Ausführungsmodus im Dialogfeld Default Preferences ändern	484
Abbildung 263. Chaos.vee im VEE 3-Modus mit geöffneten Anzeigen	485
Abbildung 264. Chaos.vee im VEE 3-Modus mit geschlossenen Anzeigen	486
Abbildung 265. Chaos.vee in VEE 4 oder einem höheren Modus mit ausgeschalteter Fehlerbehebung	487
Abbildung 266. Iteratives mathematisches Beispiel im VEE 3-Modus	488
Abbildung 267. Iteratives mathematisches Beispiel mit VEE 4- oder einem höheren Modus	488
Abbildung 268. Ein Beispiel zum Profiler	489
Abbildung 269. Modell der Web-Messungsanwendung	495
Abbildung 270. Ein Host-Modell zur Script-Sprache	497
Abbildung 271. Das Dialogfeld Default Preferences-Web Server	501
Abbildung 272. Die Standardseite Index.html	508
Abbildung 273. Anzeigen des Hauptprogramms Solitaire.vee im Browser	510
Abbildung 274. Anzeigen einer VEE-Fehlermeldung mit dem Browser	511
Abbildung 275. Die Detailansicht einer UserFunction im Browser	512
Abbildung 276. Beispiel für die Anzeige einer HTML-Meldung statt eines VEE-Programms	514
Abbildung 277. Ein Beispiel eines Kennwortfensters	515
Abbildung 278. Äpfelzähler, Lösung 1	520
Abbildung 279. Äpfelsammler, Lösung 2	521
Abbildung 280. Zahlen testen (mit Einblendfenster)	523
Abbildung 281. Zahlen testen, Schritt 2	524
Abbildung 282. Zahlen testen, Schritt 3	525
Abbildung 283. Zufallszahlen erfassen	527
Abbildung 284. Zufallszahlen-Generator, Schritt 1	528
Abbildung 285. Zufallszahlen-Generator, Schritt 2	529
Abbildung 286. Der Maskentest, Schritt 1	531
Abbildung 287. Der Maskentest, Schritt 2	532
Abbildung 288. Zeichenfolgen und globale Variablen manipulieren	534
Abbildung 289. VEE-Programme optimieren, Schritt 1	536
Abbildung 290. VEE-Programme optimieren, Schritt 2	537
Abbildung 291. Ein UserObject „Random Noise“	539
Abbildung 292. Das UserObject „NoiseGen“	540

Abbildung 293. UserFunctions, Schritt 1	542
Abbildung 294. UserFunctions, Schritt 2	543
Abbildung 295. UserFunctions, Schritt 3	544
Abbildung 296. UserFunctions, Schritt 4	545
Abbildung 297. Bibliotheken importieren und löschen	546
Abbildung 298. UserObject, das den Benutzer zur Eingabe von A und B auffordert	548
Abbildung 299. Fenster zur Eingabe von A und B durch den Benutzer	549
Abbildung 300. UserObject, das den Benutzer fragt, ob A oder B angezeigt werden soll	550
Abbildung 301. Fenster für Benutzer zur Auswahl, ob A oder B angezeigt werden soll	550
Abbildung 302. Fehler generieren, wenn der Benutzer keine Auswahl eingibt	552
Abbildung 303. Bewegen von Daten in und aus Dateien	553
Abbildung 304. Records manipulieren, Schritt 1	555
Abbildung 305. Records manipulieren, Schritt 2	557
Abbildung 306. Records manipulieren, Schritt 3	559
Abbildung 307. Den Test Sequencer verwenden, Schritt 1	562
Abbildung 308. Deaktivieren des ersten Tests in der Sequenz	563
Abbildung 309. Den Test Sequencer verwenden, Schritt 2	564
Abbildung 310. Den Test Sequencer verwenden, Schritt 3	565
Abbildung 311. Einen Zeitstempel zum Aufzeichnungs-Record hinzufügen	567
Abbildung 312. Den Test Sequencer verwenden, Schritt 4	568
Abbildung 313. Einen Record prüfen	569
Abbildung 314. Den Test Sequencer verwenden, Schritt 5	570
Abbildung 315. Den Test Sequencer verwenden, Schritt 6	571
Abbildung 316. Den Test Sequencer verwenden, Schritt 7	572
Abbildung 317. Den Test Sequencer verwenden, Schritt 8	573

# Einführung

Überblick über Agilent VEE	3
Agilent VEE installieren und kennen lernen	12
MATLAB Script – Überblick	15
Support zu Agilent VEE anfordern	17
Zusätzliche Informationsquellen zu MATLAB	18

## **Einführung**

Dieses Kapitel stellt Agilent VEE und seine wichtigsten Funktionen vor. Außerdem erfahren Sie, wie Sie VEE installieren, Informationen dazu erhalten und Support zu VEE anfordern können.

## Überblick über Agilent VEE

Agilent VEE ist eine grafische Programmiersprache, die zum Erstellen von Anwendungen im Bereich Test und Messtechnik sowie von Programmen mit Benutzerschnittstellen optimiert wurde. Diese Version (Version 7.0) von Agilent VEE wurde speziell für Gruppen von Technikern entwickelt, deren Aufgabe es ist, komplexe Test- und Messsysteme zu erstellen.

### Vorteile von Agilent VEE bei der Testentwicklung

VEE bietet eine Vielzahl von Vorteilen bei der Testentwicklung:

- Erhebliche Steigerung der Produktivität. Von Kunden erfahren wir, dass die Zeit für die Programmentwicklung um bis zu 80% reduziert werden konnte.
- Einsatz von VEE in einem breiten Spektrum von Anwendungen einschließlich Funktionstest, Design-Überprüfung, Kalibrierung, Datenbeschaffung und Steuerung.
- Erzielen einer hohen E/A-Flexibilität bei Instrumenten zur Steuerung von GPIB, VXI, seriellen Verbindungen, GPIO, PC-Zusatzkarten sowie LAN-Instrumenten. Verwenden von „Panel“-Treibern, *VXIPlug&Play*-Treibern, ODAS-Treibern, „Direkt-E/A“ über Standardschnittstellen oder importierten Bibliotheken anderer Hersteller.
- Verwenden von ActiveX-Automatisierungs- und Steuerelementen zur Steuerung anderer Anwendungen wie MS Word, Excel und Access zur Unterstützung beim Generieren von Berichten, beim Anzeigen und Analysieren von Daten sowie beim Ablegen von Testergebnissen in einer Datenbank zur späteren Verwendung.
- Verbesserung des Durchsatzes, einfacheres Erstellen größerer Programme und größere Flexibilität bei der Verwaltung von Instrumenten. VEE umfasst einen Compiler, eine professionelle Entwicklungsumgebung für große, komplexe Programme sowie erweiterte Funktionen zur Instrumentenverwaltung.

## Einführung

- VEE unterstützt Visual Studio .NET-Zusammenstellungen, was bedeutet, dass jede Textsprache, die Visual Studio .NET tatsächlich unterstützt, auch von VEE unterstützt wird.
- Sie können Ihre Investitionen in Textsprachen auch künftig nutzen, da VEE auch zusätzliche Textsprachen wie C/C++, Visual Basic, Pascal und Fortran unterstützt.

VEE-Programme werden durch Auswählen von Objekten in Menüs und Verbinden dieser Objekte erstellt. Das Ergebnis in VEE erinnert an ein Datenflussdiagramm; diese Darstellung ist einfacher zu verstehen und zu verwenden als traditionelle Codezeilen. Mit VEE entfällt der aufwändige Zyklus Bearbeiten-Kompilieren-Linken-Ausführen.

Die beiden folgenden Abbildungen vergleichen eine einfache Funktion, die zunächst in einer Textsprache (ANSI C) und anschließend in VEE programmiert wurde. In beiden Fällen erstellt die Funktion ein „Array“ von 10 Zufallszahlen, sucht den größten Wert und zeigt den das Array und den größten Wert an.

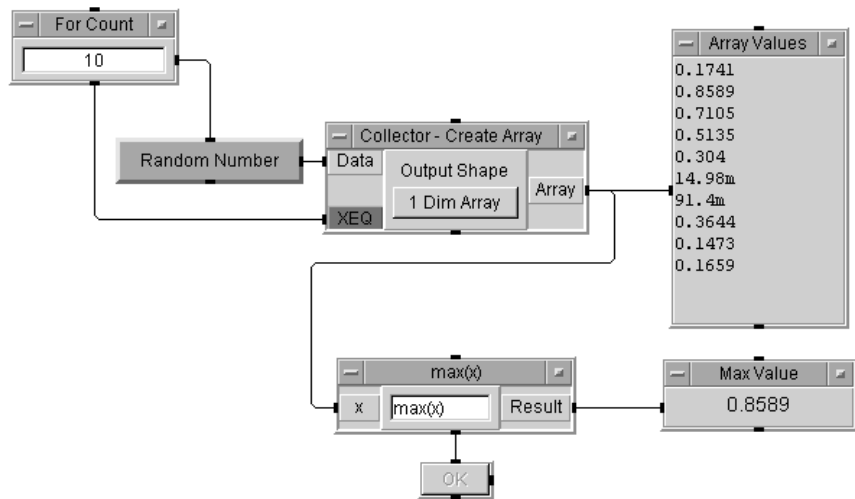
Abbildung I-1 zeigt das Programm „Random“ in der Textsprache ANSI C.



```
/* Programm zum Ermitteln des größten Elements in einem Array */
#include <math.h>
main()
{
double num[10],max;
int i;
for (i=0;i<10;i++){
num[i]=(double) rand()/pow(2.0,15.0);
printf("%f/n",num[i];
}
max=num[0];
for {i=1;i<10;i++){
if (num[i]>max)max=num[i];
}
printf("/nmax; %f/n",max);
}
```

**Abbildung I-1. Das Programm „Random“ in ANSI C**

Abbildung I-2 zeigt das gleiche Programm in VEE.



**Abbildung I-2. Das gleiche Programm „Random“ in VEE**

In VEE wird das Programm mit Programmelementen, den so genannten **Objekten**, erstellt. Objekte sind die Bausteine eines VEE-Programms. Sie führen verschiedene Funktionen aus wie beispielsweise E/A-Operationen, Analyse und Anzeige. Wenn Sie die Objekte mit allen ihren Verbindungen anzeigen, wie in Abbildung I-2 dargestellt, wird dies als **Detailansicht** („detail view“) bezeichnet. Die Detailansicht ist analog zum Quellcode bei einer Textsprache.

In VEE wandern Daten in konsistenter Weise von einem Objekt zum nächsten: Dateneingang links, Datenausgang rechts, operative Pin-Verbindungen oben und unten.

Durch Verbinden der Objekte wird ein Programm gebildet. Verfolgen Sie das Programm von links nach rechts. Im Programm „Random“ (siehe Abbildung I-2) wird dem Objekt **Collector - Create Array** zehn Mal eine Zufallszahl hinzugefügt und somit ein Array erstellt. Das Programm sucht anschließend den größten Wert in dem Array und zeigt den größten Wert (**Max Value**) sowie die Array-Werte (**Array Values**) an.

Mit VEE und seinem modularen Ansatz reduzieren Sie die erforderliche Zeit zum Erstellen von Programmen für die Steuerung von Instrumenten, generieren angepasste Datenanzeigen und entwickeln Benutzerschnittstellen. Diese Methode der Testentwicklung ermöglicht eine deutlich größere Produktivitätssteigerung als herkömmliche Techniken.

### HINWEIS

In Abbildung I-2 werden manche Objekte detailliert angezeigt, andere dagegen nur als Name. Die detailliert dargestellten Objekte werden in einer **offenen Ansicht** („open view“) angezeigt. Die offene Ansicht ermöglicht die Anzeige von Details zu dem Objekt. Sie können Objekte **als Symbol anzeigen**, um Speicherplatz zu sparen und das Programm schneller auszuführen, oder Sie können Objekte reduzieren, sodass nur ihre Namen angezeigt werden.

In Abbildung I-2 wird beispielsweise das Objekt mit dem Namen **Random Number** als Symbol angezeigt. Das Objekt mit dem Namen **Create Array** wird in einer offenen Ansicht dargestellt. Die offene Ansicht ist deutlich größer und ausführlicher.

---

## Benutzerschnittstellen in Agilent VEE erstellen

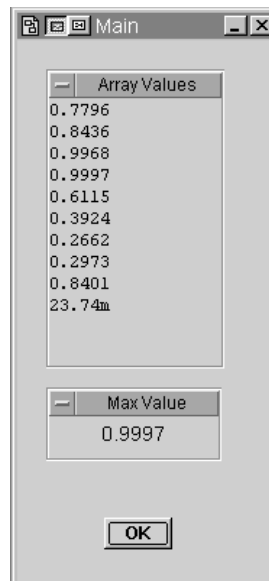
Ein weiterer Vorteil der Programmierung in VEE liegt darin, dass eine Benutzerschnittstelle in wenigen Minuten erstellt werden kann.

Mit dem Programm „Random“ aus Abbildung I-2 werden die Objekte, die der Benutzer sehen muss, ausgewählt und in einer **Fensteransicht** („panel view“) abgelegt. Eine Fensteransicht zeigt nur die Objekte an, die der Benutzer zur Ausführung des Programms und zur Anzeige der resultierenden Daten benötigt. Abbildung I-3 zeigt die Fensteransicht des Programms „Random“ in Abbildung I-2.

### HINWEIS

Das Programm und seine Benutzerschnittstelle sind verschiedene Ansichten des gleichen VEE-Programms. Sie können von einer Ansicht zur anderen wechseln, indem Sie auf die Detailansichts- bzw. die Fensteransichts-Schaltfläche in der Titelleiste des Fensters in VEE klicken. Alle Änderungen oder Aktualisierungen, die Sie an einem Programm (Detailansicht) vornehmen, werden automatisch auch in der Benutzerschnittstelle (Fensteransicht) durchgeführt. Sie können die Benutzerschnittstelle auch gegen unerwünschte Änderungen schützen.

Weitere Informationen zum Erstellen einer Benutzerschnittstelle finden Sie unter „Übung 2-4: Eine Fensteransicht (Benutzerschnittstelle) erstellen“ auf Seite 94.



**Abbildung I-3. Fensteransicht (oder Benutzerschnittstelle) des VEE-Programms**

Mit VEE können Sie manche Aufgaben in Minutenschnelle ausführen, für die Sie in einer Textsprache Tage brauchen würden.

- Erstellen farbenprächtiger, intuitiver Oberflächen für Programme.
- Erstellen von Benutzerschnittstellen, die mit Tastatur und Maus oder nur über die Tastatur bedient werden.
- Auswählen einer breiten Palette von Funktionen für die Benutzereingabe und die Anzeige von Daten.
- Verwenden von Einblendfenstern zum Erstellen eines Fokus und zum Einsparen von Platz auf dem Bildschirm.
- Sichern von Programmen gegen unerwünschte Änderungen.
- Verwenden von Beschriftungen, Auswählen von Farben und Schriftarten sowie Hinzufügen akustischer Signale, Notizblocks, Schaltflächen und Schaltern in den unterschiedlichsten Formaten.
- Verwenden eigener oder standardisierter ActiveX-Steuer-elemente für die Benutzereingabe oder zum Anzeigen von Daten.

### **Vorhandene Testprogramme mit Agilent VEE nutzen**

Für alle unterstützten Betriebssysteme bietet VEE Mechanismen zum Verbinden („Link“) konventioneller Testprogramme sowie kommerzieller Anwendungen. Sie können mit VEE beispielsweise vorhandene Tests in C, C++, Visual Basic, Fortran oder Pascal (oder jeder anderen kompilierten oder interpretierten Sprache auf Ihrem Betriebssystem) verwenden. Zudem können Sie auch jede Sprache verwenden, die Visual Studio .NET tatsächlich unterstützt.

VEE bietet auch eine Reihe von Funktionen zur prozessübergreifenden Kommunikation, um Daten mit kommerziellen Anwendungen wie Datenbanken oder Tabellenkalkulationen gemeinsam zu nutzen. VEE unterstützt Standardverbindungen zur ActiveX-Automatisierung sowie zu Steuerelementen und DLLs.

### Instrumente mit Agilent VEE steuern

VEE bietet eine Reihe von Funktionen zum Steuern von und zur Kommunikation mit Instrumenten.

- Verwendung von Panel-Treibern (Instrumententreibern) für mehr als 450 Geräte verschiedener Hersteller sowie aller von verschiedenen Herstellern verfügbaren Treibern, sofern sie *VXIPlug&Play*-kompatibel mit den Rahmendefinitionen von Windows 98, Windows 2000, Windows NT 4.0 oder Windows XP sind.
- Verwendung der direkten E/A von VEE zum Senden von Befehlszeichenfolgen an Instrumente über Standardschnittstellen wie GPIB (IEEE – 488), GPIO, RS 232, VXI oder LAN-orientierten Instrumenten für ferne Tests.
- Steuern von PC-Zusatzkarten beliebiger Hersteller, sofern die Karten einen standardisierten ODAS-Treiber oder eine Dynamic Link Library unterstützt.
- Verwendung einer direkten VXI-Backplane-Steuerung mit integrierten PCs oder Workstations.
- Steuern einer Vielzahl von Gerätetypen mit einfachen, organisierten Funktionen zur Instrumentenverwaltung.

### Testmöglichkeiten mit Hilfe von Agilent VEE verbessern

Die VEE-Produkte bieten folgende Funktionen und Vorteile:

- Weniger Zeitaufwand für Entwicklung und Wartung durch grafische Programmierung.
- Integration mit konventionellen Sprachen wie C, C++, Visual Basic, Pascal und Fortran.
- Komfortable und flexible Funktionen der Benutzerschnittstelle.
- Unterstützung der meisten gängigen Testplattformen.
- Verwendung von ActiveX-Automatisierung und -Steuerelementen.
- Die hervorragenden Support-Optionen von Agilent Technologies.

- Benutzerfreundliche und leistungsstarke Dokumentationshilfen.
- Einfache Portierung von Testdaten in Standard-Kalkulationstabellen und Textverarbeitungsprogramme für Berichte.
- Tools zur prozessübergreifenden Kommunikation zur Verbindung mit anderen Anwendungen wie relationalen Datenbanken oder Paketen zur Statistikanalyse (Version 6.0 oder neuer ).
- Fehlerbehebungs-Tools zur effizienteren Gestaltung von Entwicklung und Wartung großer, komplexer Programme (Version 6.0 oder neuer).
- Leistungsstarke Testausführungs-Tools im Produkt enthalten (Version 6.0 oder neuer).
- Möglichkeiten zu fernen Tests mit den Web-Monitorfunktionen von VEE (Version 6.0 oder neuer).
- Unbegrenzte Laufzeit für die Weitergabe Ihrer Programme (Version 6.0 oder neuer).
- Preisgünstige Standortlizenzen (Version 6.0 oder neuer).

# Agilent VEE installieren und kennen lernen

Dieser Abschnitt enthält Informationen über die Installation der Software und vermittelt erste Eindrücke über VEE. Darüber hinaus erhalten Sie einen Überblick über die Verwendung von VEE und erfahren, wie Sie Support zu VEE anfordern.

## Agilent VEE und E/A-Bibliotheken installieren

Sie können eine Testversion von VEE bestellen oder aus dem Internet laden. Dabei handelt es sich um eine funktional vollständige Version des Produkts. Bei der Installation werden Sie aufgefordert, einen Produktschlüssel einzugeben. Geben Sie das Wort „eval“ ohne Anführungszeichen ein.

Durch den Erwerb eines Produktschlüssels können Sie diese Version vollständig aktivieren. Sobald Sie den Produktschlüssel per E-Mail erhalten haben, geben Sie ihn ein, indem Sie im Hauptmenü Hilfe ⇒ Product Key wählen. Das Vollprodukt kann ebenfalls bestellt oder aus dem Internet geladen werden. Ein Produktschlüssel-Zertifikat erhalten Sie entweder zusammen mit dem Produkt oder separat, falls Sie das Produkt aus dem World Wide Web geladen haben.

VEE kann auf folgenden Betriebssystemen ausgeführt werden: Windows 98, Windows ME, Windows NT(SP6a), Windows 2000 SP4 und Windows XP SP1.

Stellen Sie sicher, dass sie mit Administratorrechten angemeldet sind, wenn Sie VEE von einer CD-ROM installieren. Legen Sie die CD-ROM in das CD-ROM-Laufwerk ein. Die Installation sollte automatisch beginnen. Falls nicht, wählen Sie im Startmenü den Befehl Ausführen. Geben Sie in das Dialogfenster Ausführen folgendes ein:

```
[CD-ROM-Laufwerkskennbuchstabe]:\cdsetup.exe
```

So installieren Sie eine aus dem WWW geladene Version von VEE: Laden Sie das selbstextrahierende ausführbare Archiv aus dem Internet. Die auf Ihren Rechner geladene selbstextrahierende Datei wird in dem von Ihnen zuvor festgelegten Download-Ordner gespeichert. Packen Sie die VEE-Installations-



dateien aus, indem Sie die selbstextrahierende Archivdatei ausführen. Die Dateien werden in ein Verzeichnis Ihrer Wahl ausgepackt. Starten Sie aus diesem Verzeichnis die Datei cdsetup.exe.

Um Sie bei Ihrem Bedürfnis nach Sicherheit zu unterstützen, werden Sie gefragt, ob Sie das Programm für alle Benutzer des PCs oder nur für einen einzigen Benutzer installieren möchten. Wählen Sie die Option, die Ihren Bedürfnissen entspricht.

Informationen über die E/A-Bibliotheken finden Sie in den Installationsunterlagen, die Sie zusammen mit der Software erhalten haben. (Die E/A-Bibliotheken werden von VEE zur Kommunikation mit Instrumenten verwendet.)

*Installing and Distributing VEE Pro RunTime* (in der Online-Hilfe) beschreibt die Installation und Weitergabe der Laufzeitversion (RunTime version) von VEE Pro. Die Laufzeitversion wird zur Ausführung von VEE-Programmen auf PCs verwendet, auf der keine VEE-Software installiert ist. Weitere Informationen zur Laufzeitumgebung finden Sie in der Online-Hilfe. Wählen Sie dazu **Help, Contents and Index** und **Installing and Distributing Agilent VEE Pro RunTime**. Sie können diese Informationen auch drucken.

## Agilent VEE kennen lernen

Für Informationen zu VEE können Sie die VEE-Multimedia-Lernprogramme ansehen, die Online-Hilfe öffnen, in den Handbüchern (beispielsweise dem vorliegenden) nachschlagen und VEE-Seminare besuchen.

- *VEE-Multimedia-Lernprogramme*: Bei den über **Help** ⇒ **Welcome** abspielbaren VEE-Multimedia-Lernprogrammen handelt es sich um Video-Präsentationen, in denen viele der Konzepte von VEE erläutert werden. Diese Präsentationen zeigen Ihnen, wie Sie die VEE-Menüs verwenden, Objekte bearbeiten und wie Sie Programme ausführen. Jede Präsentation dauert ca. drei bis vier Minuten und kann beliebig oft abgespielt werden. Sie können ein Lernprogramm unterbrechen und VEE ausführen, um das Gelernte auszuprobieren, und das Lernprogramm anschließend fortsetzen.
- *VEE-Online-Hilfe*: Eine Möglichkeit, die neuen Funktionen von VEE kennen zu lernen, ist die Auswahl von **Help** ⇒

**Contents and Index** ⇒ **What's New in Agilent VEE**. Unter ⇒ **Welcome** ⇒ **Introduction** finden Sie einen Überblick über das VEE-Produkt.

Die Online-Hilfe umfasst eine Reihe weiterer Funktionen. Weitere Informationen hierzu finden Sie unter „Die Online-Hilfe“ auf Seite 102.

- VEE Handbücher: Das Handbuchpaket für VEE umfasst das vorliegende *VEE Pro Benutzerhandbuch* sowie das Handbuch *VEE Pro Advanced Techniques*.
- *Agilent VEE-Seminare*: Informationen zu VEE-Seminaren finden Sie im Internet unter <http://www.agilent.com/comms/education>.

### HINWEIS

Die VEE-Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter **Help** ⇒ **Open Example...** ⇒ **Manual** ⇒ **UsersGuide** enthalten.

---

## Eine kostenlose Testversion der Software anfordern

Eine kostenlose Testversion der Software ist auf CD verfügbar oder kann von der VEE-Web-Site aus dem Internet geladen werden. Zum Anfordern der VEE Evaluation Kit CD wenden Sie sich an Ihre lokale Agilent Technologies Niederlassung.

<http://www.agilent.com/find/products>

## MATLAB Script – Überblick

MATLAB<sup>®</sup> Script ist eine Untermenge des Standard-Vollprodukts MATLAB von The MathWorks. Dieses Produkt gibt dem Benutzer direkten Zugriff auf die Kernfunktionen von MATLAB, wie beispielsweise höhere Mathematik, Datenanalyse sowie wissenschaftliche und Engineering-Grafik. Das Objekt MATLAB kann sehr einfach in ein beliebiges Agilent VEE-Programm einbezogen werden.

MATLAB Script umfasst Hunderte von Funktionen für:

- Datenanalyse und -darstellung
- Numerische Berechnung einschließlich:
  - Lineare Algebra und Matrixberechnung
  - Fourier- und statistische Analyse
  - Lösen von Differentialgleichungen
  - Trigonometrische und fundamentale mathematische Operationen
- Engineering- und wissenschaftliche Grafik wie beispielsweise:
  - 2-D- und 3-D-Anzeige einschließlich Dreiecks- und Gitterdaten
  - Volumendarstellung von skalaren oder Vektordaten
  - Quiver-, Band-, Streu-, Balken-, Kreis- und Stamm-Plots

### Toolbox zur Signalverarbeitung

MATLAB Script für VEE umfasst auch eine Untermenge der MATLAB Signalverarbeitungs-Toolbox, die auf einer soliden Basis von Techniken aus den Bereichen Filter-Design und Spektralanalyse aufbaut. Es stehen Funktionen zur Verfügung für:

- Signal- und lineare Systemmodelle
- Analoges Filter-Design
- FIR und IIR digitales Filter-Design, Analyse und Implementierung
- Transformationen wie FFT und DCT

- Spektralschätzung und statistische Signalverarbeitung
- Parametrische Zeitserien-Modellierung
- Waveform-Generierung

### Informationen zur MATLAB-Vollversion

MATLAB ist eine integrierte technische Computing-Umgebung, die numerische Berechnungen und erweiterte Grafik- und Darstellungsfunktionen mit einer höheren Programmiersprache verbindet. MATLAB umfasst Hunderte von Funktionen für:

- Datenanalyse und -darstellung
- Numerische und symbolische Berechnung
- Engineering- und wissenschaftliche Grafik
- Modellierung, Simulation und Prototyp-Verarbeitung
- Programmierung, Anwendungsentwicklung und GUI-Design

MATLAB wird in einer Vielzahl von Anwendungsbereichen eingesetzt, beispielsweise in der Signal- und Bildverarbeitung, im Design von Steuersystemen, im Financial Engineering und in der medizinischen Forschung. Die offene Architektur macht es sehr einfach, MATLAB und ergänzende Produkte bei der Untersuchung von Daten und der Erstellung spezifischer Tools einzusetzen, die zu einem sehr frühen Zeitpunkt wertvolle Informationen liefern und so einen echten Vorsprung im Wettbewerb bieten.

Als VEE-Benutzer können Sie die volle Leistung von MATLAB und der Signalverarbeitungs-Toolbox für Anwendungen mit Datenanalyse, Darstellung und Modellierung implementieren. Durch die Erweiterung auf die Vollversion dieser Produkte können Sie ein breites Spektrum zusätzlicher MATLAB-Funktionen in VEE-Anwendungen einsetzen, beispielsweise zum Erstellen benutzerdefinierter Funktionen (M-Dateien), und haben Zugriff auf das MATLAB-Befehlsfenster, den MATLAB-Editor/Debugger und die grafische Benutzeroberfläche (GUI) der Signalverarbeitung.

**HINWEIS**

Weitere Informationen zur Verwendung der MATLAB Script-Objekte in VEE-Programmen finden Sie unter „MATLAB Script in Agilent VEE verwenden“ auf Seite 190 im Kapitel „Testdaten analysieren und anzeigen“

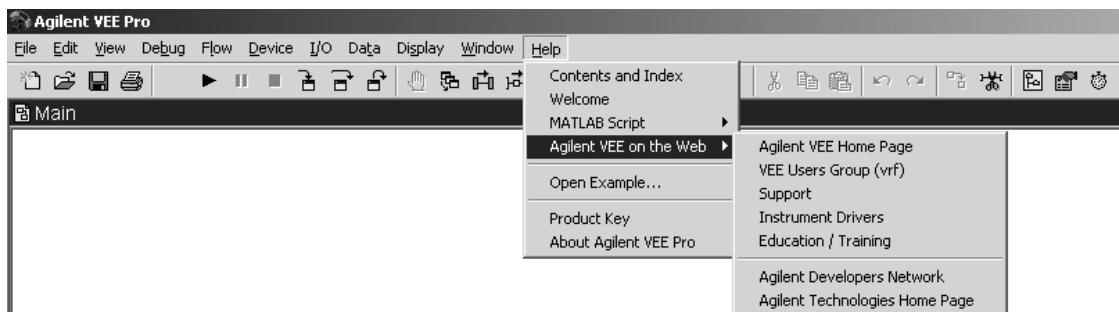
## Support zu Agilent VEE anfordern

Sie können VEE-Support über das Web oder per Telefon (für Hilfestellung bei der Einrichtung) anfordern.

### Abrufen von Informationen über das World Wide Web

Die VEE-Web-Site bietet vielfältige Informationen wie Anwendungshinweise, Benutzertipps, technische Daten und Informationen zu VEE-Partnern wie beispielsweise Herstellern von PC-Zusatzkarten.

- *Haupt-Website für VEE:*  
<http://www.agilent.com/find/vee>.
- *Aktuelle Support-Informationen:* Während die Verbindung zum Netzwerk aktiv ist, wählen Sie in VEE **Help** ⇒ **Agilent VEE on the Web**. Kapitel Abbildung I-4., „Kontaktaufnahme mit Produkt-Support über das VEE-Hilfemenü“ zeigt wie Sie den Support von VEE aus erreichen.  
<http://www.agilent.com/find/vee:de>



**Abbildung I-4. Kontaktaufnahme mit Produkt-Support über das VEE-Hilfemenü**

- *Für ergänzende Hilfestellung bei der Einrichtung:* siehe „Hilfe“, „Phone Support Information“ in der Online-Hilfe.

## Zusätzliche Informationsquellen zu MATLAB

Ausführliche Informationen zur Verwendung des Objekts MATLAB Script erhalten Sie über den MATLAB Script Help Desk. Wählen Sie in VEE **Help** ⇒ **MATLAB Script** ⇒ **Help** Desk. Der **MATLAB Help Desk** wird in einem Web-Browser angezeigt.

Weitere Informationen zu MATLAB, MATLAB-Toolboxen und anderen Produkten von The MathWorks finden Sie im World Wide Web unter [www.mathworks.com](http://www.mathworks.com), oder rufen Sie die Nummer +1 508-647-7000 an.

Weitere Informationsquellen sind:

- **Vollständige MATLAB-Dokumentation:**  
[www.mathworks.com/access/helpdesk/help/helpdesk.shtml](http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml)
- **MATLAB-Produktinformationen:**  
[www.mathworks.com/products](http://www.mathworks.com/products)
- **MATLAB Technische Hilfestellung:**  
[www.mathworks.com/support](http://www.mathworks.com/support)
- **MathWorks Store:** [www.mathworks.com/store](http://www.mathworks.com/store)
- **MathWorks Homepage:** [www.mathworks.com](http://www.mathworks.com)
- **Usenet Newsgroup:** Die Newsgroup `comp.soft-sys.matlab` bietet ein Forum für Experten und Anfänger, die mit MATLAB arbeiten und Fragen oder Kommentare zu MATLAB und den zugehörigen Produkten haben.

# 1

## Die VEE-Entwicklungsumgebung verwenden

Überblick 21

Interaktion mit Agilent VEE 22

Mit Objekten arbeiten 30

Informationen zu Pins und Anschlüssen 49

Objekte zum Erstellen eines Programms verbinden 56

Die Arbeitsweise von Agilent VEE-Programmen 69

Kapitel-Checkliste 76

## **Die VEE-Entwicklungsumgebung verwenden**

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Unterstützte Systeme
- Das Hilfesystem verwenden
- VEE starten
- Das VEE-Fenster
- Arbeiten mit Objekten
- Den Arbeitsbereich verwalten
- Menüelemente wählen
- Pins und Anschlüsse an VEE-Objekten
- Objekte zum Erstellen von Programmen verbinden
- Programme erstellen, ausführen, drucken, speichern und öffnen
- Die Arbeitsweise von VEE-Programmen

*Erforderliche Zeit für dieses Kapitel: 1,5 Stunden*



## Überblick

In diesem Kapitel lernen Sie, wie Sie VEE starten, Menüs verwenden und mit Objekten arbeiten. Sie lernen den Zweck von Pins und Anschlüssen in VEE kennen. Sie verbinden Objekte miteinander, um ein einfaches VEE-Programm zu erstellen, und Sie lernen die Arbeitsweise von VEE-Programmen kennen.

### Interaktion mit Agilent VEE

In diesem Abschnitt wird die Verwendung der grafischen Programmiersprache VEE erläutert. Sie finden hier auch eine Liste der unterstützten Systeme, Informationen zur Arbeitsweise der Maus und der Menüs sowie zum Aufrufen der Hilfe, zum Starten von VEE und zum Arbeiten im VEE-Fenster.

#### Unterstützte Systeme

Diese Version von VEE wird von folgenden Systemen unterstützt:

- Windows 98, Windows 2000, Windows NT 4.0 und Windows XP auf einem PC.

#### Die Maus und die Menüs

Wahrscheinlich sind Sie mit der maus- und menügesteuerten Benutzeroberfläche des Computers bereits vertraut und kennen Menüs, Symbolleisten und Dialogfelder, die Sie über die Maus oder die Tastatur steuern können. VEE verwendet die Benutzerschnittstelle des Computer-Betriebssystems. In den Anleitungen zur Verwendung der Maus beim Arbeiten mit Menüs, Symbolen, Schaltflächen und Objekten werden die folgenden allgemeinen Techniken verwendet:

- Um auf ein Element zu „klicken“ zeigen Sie mit der Maus auf das gewünschte Element, drücken Sie dann die *linke* Maustaste und lassen Sie sie gleich wieder los.
- Zum „Doppelklicken“ auf ein Element zeigen Sie mit der Maus auf das gewünschte Element und klicken anschließend zwei Mal in schneller Folge mit der *linken* Maustaste.
- Zum „Ziehen“ eines Elements zeigen Sie mit der Maus auf das gewünschte Element, *halten Sie die linke Maustaste gedrückt*, und ziehen Sie das Element mit der Maus an die gewünschte Position. Lassen Sie anschließend die Maustaste wieder los.

**HINWEIS**

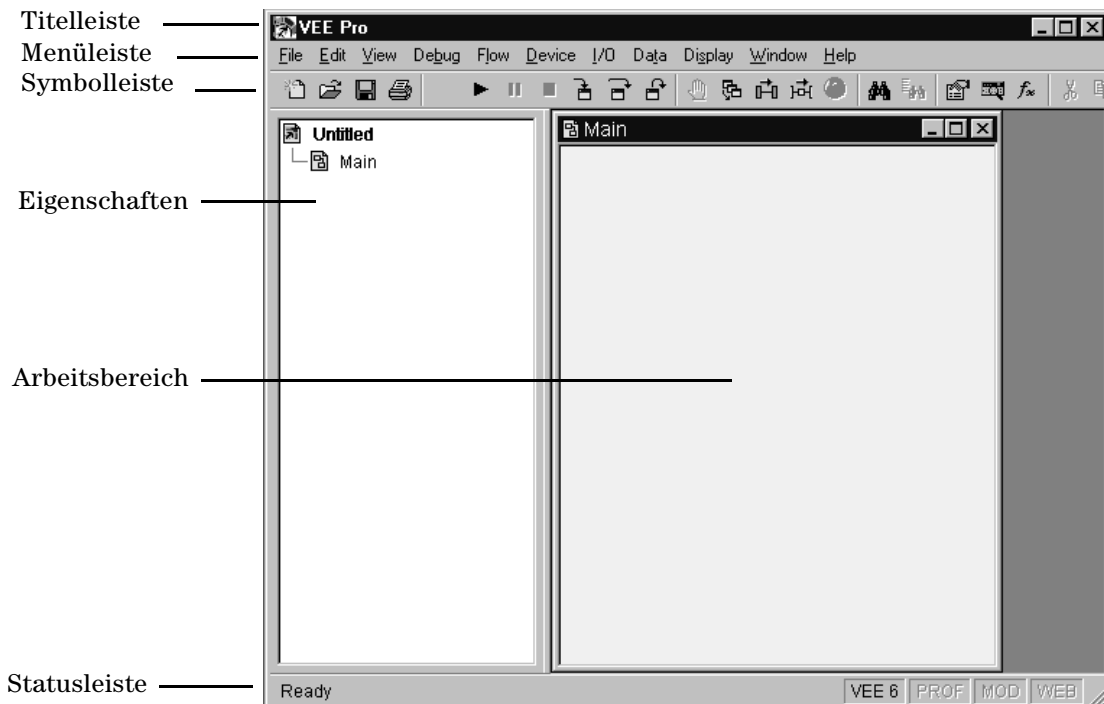
Die rechte Maustaste wird seltener verwendet. Wenn Sie für eine Aktion mit der rechten Maustaste klicken müssen, werden Sie speziell darauf hingewiesen. Ihre Maus hat eventuell auch eine mittlere Taste – in VEE hat diese Taste jedoch keine Funktion.

**Agilent VEE starten**

Klicken Sie auf **Start** ⇒ **Programme** ⇒ **Agilent VEE Pro**.

**Das Agilent VEE-Fenster**

Nach der Installation und dem Start von VEE wird das in Abbildung 1 dargestellte VEE-Fenster angezeigt.



**Abbildung 1** Die VEE-Entwicklungsumgebung

## 1 Die VEE-Entwicklungsumgebung verwenden

Die folgenden Elemente beschreiben die Komponenten des VEE-Fensters.

**Tabelle 1** Beschreibung der Elemente des VEE-Fensters

Bildschirmelement	Beschreibung
Titelleiste	Der oberste Bereich im Fenster enthält das VEE-Symbol, den Fensternamen und die Schaltflächen Minimieren, Maximieren und Schließen. Sie verschieben das Fenster durch Ziehen der Titelleiste. Klicken Sie auf das VEE-Symbol, um das Fenstermenü zu öffnen.
Menüleiste	Der zweite Bereich von oben enthält die Menüleiste, über deren Menüs Sie Zugriff auf die verschiedenen VEE-Befehle und -Objekte haben.
Symboleiste	Unter der Menüleisten befindet sich die Symboleiste, deren Schaltflächen direkten Zugriff („Direktaufrufe“) auf die am häufigsten verwendeten Menübefehle gewähren. (Zeigen Sie mit der Maus auf eine Schaltfläche, um den Namen der jeweiligen Funktion anzuzeigen.)
Arbeitsbereich	Ein Bereich im Programmfenster (Bearbeitungsfenster) wie beispielsweise Main, UserObject oder UserFunction, in denen Sie Objekte positionieren und miteinander verbinden.
Program Explorer	<p>Ein Bereich auf der linken Seite des VEE-Fensters, der die Struktur des VEE-Programms zeigt. Die obere Ecke zeigt den Namen des aktuellen Programms, z. B. <b>myprog.vee</b>, oder sie zeigt den Hinweis <b>Untitled</b> (Ohne Namen) an.</p> <p>Der Program Explorer ermöglicht das Wechseln zwischen den Programmfenstern. Zum Ändern der Größe des Program Explorer zeigen Sie mit der Maus auf die rechte Begrenzung, bis der Mauszeiger die Form eines vertikalen Teilers annimmt, klicken Sie mit der Maustaste und ziehen Sie die rechte Begrenzung an die gewünschte Position.</p>
Hauptfenster	Ein Fenster mit einem Arbeitsbereich, in dem Sie VEE-Programme entwickeln und bearbeiten. Es können auch andere Programm-/Bearbeitungsfenster geöffnet sein, z. B. UserObject.

**Tabelle 1** Beschreibung der Elemente des VEE-Fensters

Bildschirmelement	Beschreibung
Statusleiste	Der Bereich am unteren Ende des Fensters, in dem Meldungen zum Status von VEE sowie vier weitere Statusanzeigen (am rechten Rand) angezeigt werden. Die Anzeigen (von links nach rechts) enthalten folgende Informationen: Den Ausführungsmodus Den Status des Profilers MOD wird angezeigt, wenn das Programm geändert wurde Der Web-Server ist aktiviert

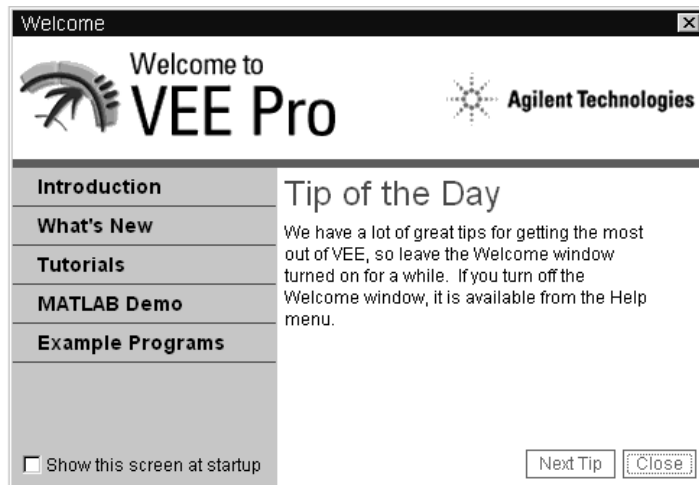
## Hilfe aufrufen

VEE bietet ein Online-Hilfesystem (Help) für die VEE-Umgebung sowie Online-Hilfetexte zu einzelnen Objekten und Themen. Darüber hinaus können Sie weitere Informationen in der Dokumentation nachschlagen, die Sie mit dem Computer bzw. dem Betriebssystem erhalten haben. Die PC-Online-Hilfe enthält Informationen zu Themen wie beispielsweise:

- Befehle in der Menüleiste wählen
- Menüelemente wählen und deren Auswahl aufheben
- Symbolleisten verwenden
- Informationen zu Titel- und Statusleisten
- Auf Symbole und Schaltflächen klicken
- Mit Dialogfeldern arbeiten
- Mit verschiedenen Fenstertypen arbeiten
- Die Online-Hilfe verwenden

Beginnen Sie am besten mit **Help** ⇒ **Welcome**, damit erhalten Sie Zugriff auf die VEE-Multimedia-Lernprogramme. Das Begrüßungsdiaologfeld ist in Abbildung 2 dargestellt.

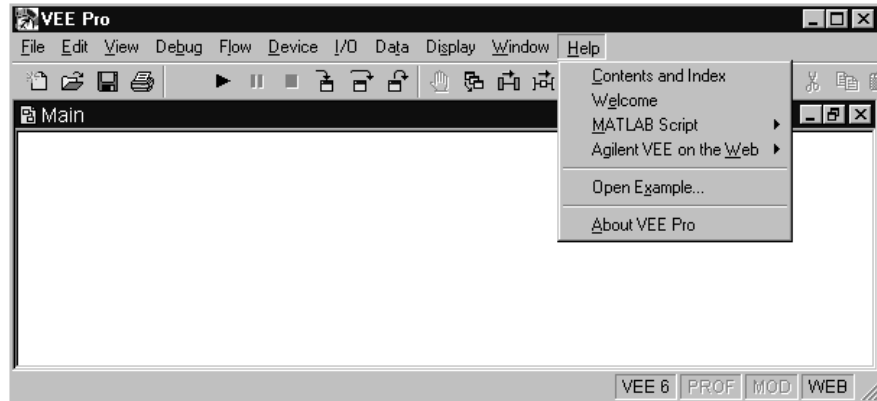
## 1 Die VEE-Entwicklungsumgebung verwenden



**Abbildung 2** Das VEE-Begrüßungsdialogfeld

Die VEE-Online-Hilfe wurde für Ihr Betriebssystem konzipiert. Klicken Sie auf **Help**, um das in Abbildung 3 dargestellte Menü zu öffnen. Das Hilfenü enthält Befehle für den Zugriff auf Inhaltsverzeichnis und Index der Online-Hilfe, das Begrüßungsdialogfeld (in dem die Lernprogramme aufgeführt sind), Gerätetreiber- und Web-Informationen, Beispiele sowie die Versionsnummer.

Zur Durchführung dieses Selbstlernkurses werden Sie die VEE-Dokumentation nicht benötigen; Sie finden in der Produktdokumentation jedoch ausführliche Informationen zu einzelnen Funktionen oder Konzepten. Verwenden Sie das Hilfesystem, um nach VEE-Themen zu suchen, über die Sie mehr erfahren wollen. Die Hyperlinks des Hilfesystems gewähren Ihnen einen bequemen Zugang zu verwandten Themen.



**Abbildung 3** Das Menü Help verwenden

Wählen Sie im Hilfemenü den Befehl Contents and Index, um die VEE-Hilfe wie in Abbildung 4 dargestellt zu öffnen.

# 1 Die VEE-Entwicklungsumgebung verwenden

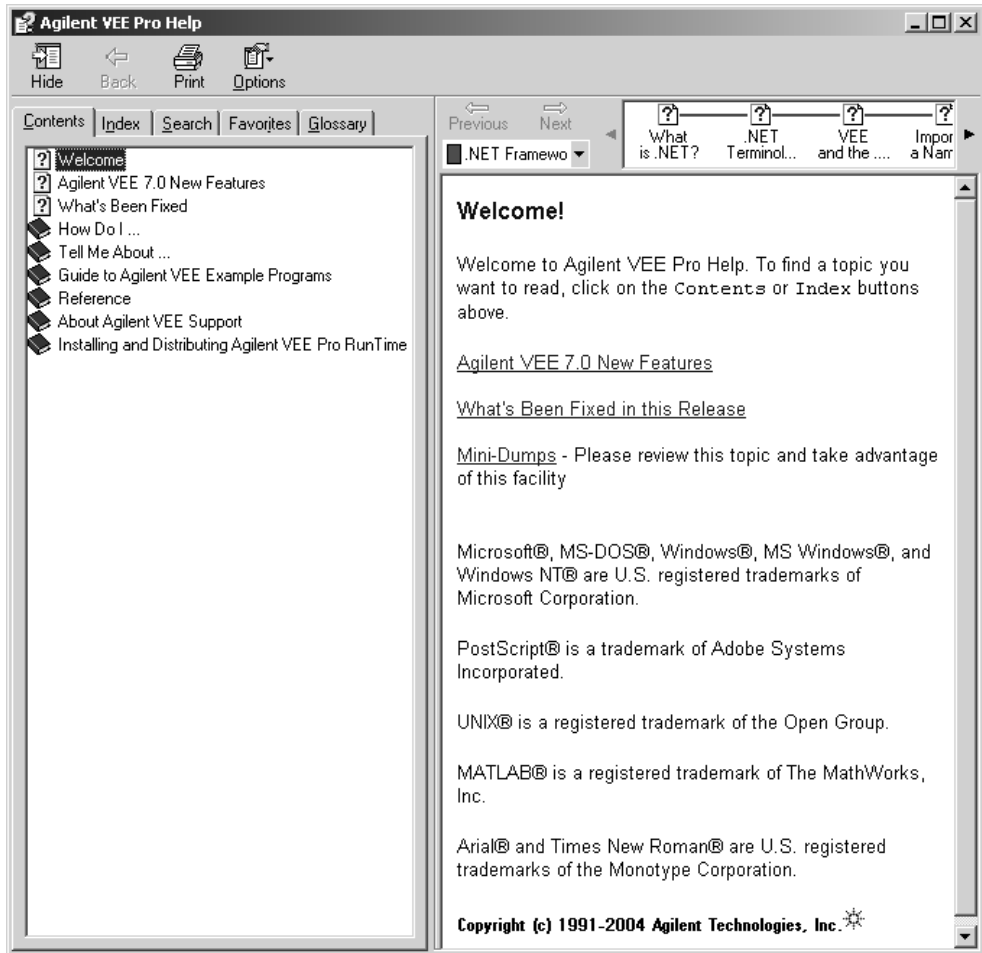


Abbildung 4 Das Register Contents (Inhalt) der VEE-Hilfe



Das Register **Contents** (Inhalt) bietet Informationen zu folgenden Themen.

**Tabelle 2**

<b>What's New in VEE</b>	<b>Definition</b>
How Do I...	Enthält Informationen zur Vorgehensweise bei gängigen Aufgaben.
Tell Me About...	Erläutert die Konzepte von VEE.
Guide to Agilent VEE Example Programs	Fasst die mit VEE bereitgestellten Beispielprogramme zusammen.
Reference	Enthält Referenzinformationen zu allen Funktionen und Objekten.
About Agilent VEE Support	Enthält Informationen zum Abrufen von Support-Leistungen für VEE.
Installing and Distributing Agilent VEE Pro Runtime	Erläutert die Weitergabe der VEE Pro RunTime-Umgebung.

## HINWEIS

Als Tastenkürzel zum Abrufen von Hilfeinformationen zu einem bestimmten Objekt oder einem Dialogfeld drücken Sie die Taste **F1**. Darüber hinaus können Sie in einem Objektmenü auch **Help** wählen, um spezifische Informationen zu dem jeweiligen Objekt abzurufen.

Weitere Informationen zu spezifischen Hilfsfunktionen bei der Entwicklung von Programmen finden Sie im Abschnitt „Verwenden der Hilfsfunktion“ auf Seite 102.

### Mit Objekten arbeiten

Ein VEE-Programm besteht aus verbundenen Objekten. Wählen Sie zum Erstellen eines Programms in den VEE-Menüs geeignete Objekte, beispielsweise in den Menüs Flow, Data und Display. Verbinden Sie die Objekte mit Linien, die an den Objekt-Pins „angeschlossen“ werden. (Weitere Informationen über Pins finden Sie im Abschnitt „Informationen zu Pins und Anschlüssen“ auf Seite 49.) Erstellen Sie ein Programm mit einer Gruppe verbundener Objekte.

In diesem Abschnitt wird beschrieben, wie Sie Objekte in einem Programm markieren (oder wählen) und verwenden.

- 1 Starten Sie VEE. Klicken Sie auf **Start** ⇒ **Programme** ⇒ **Agilent VEE Pro**.
- 2 Verwenden Sie die Anleitungen in diesem Abschnitt, um mit Objekten zu experimentieren.

#### HINWEIS

Bei den späteren Übungen wird davon ausgegangen, dass Sie die VEE-Software gestartet haben. Wenn Sie Informationen zum Starten von VEE benötigen, schlagen Sie auf dieser Seite oder im Abschnitt „Agilent VEE starten“ auf Seite 23 nach.

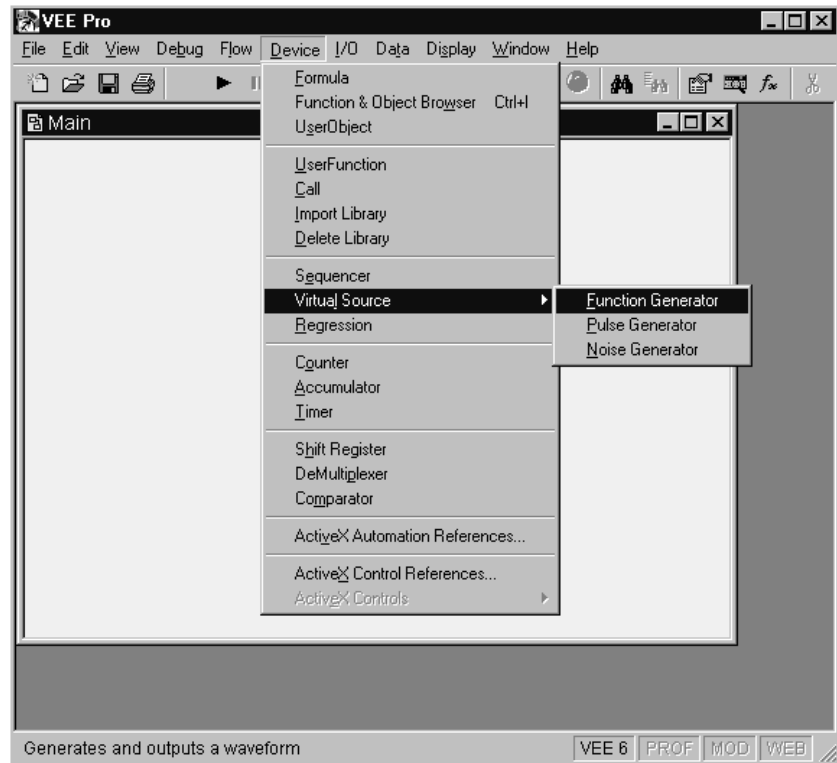
### Objekte auf dem Arbeitsbereich platzieren

Öffnen Sie ein geeignetes Menü, wählen Sie das gewünschte Objekt, ziehen Sie das Objekt an die Zielposition im Arbeitsbereich und klicken Sie mit der Maustaste (der Objekturnriss wird nun durch das Objekt selbst ersetzt).

- 1 Wenn Sie beispielsweise ein Function-Generator-Objekt dem Arbeitsbereich platzieren wollen, wählen Sie in der Menüleiste **Device** ⇒ **Virtual Source** ⇒ **Function Generator**, wie in Abbildung 5 gezeigt.

**HINWEIS**

Der Pfeil rechts von **Virtual Source** weist darauf hin, dass diesem Menüelement ein Untermenü zugeordnet ist. Drei Punkte am Ende eines Menüelements weisen darauf hin, dass der betreffende Befehl ein Dialogfeld öffnet. **File** ⇒ **Save As...** veranschaulicht diese Funktionsweise.

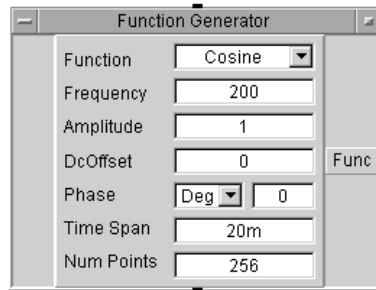


**Abbildung 5** Objekte auf dem Arbeitsbereich platzieren

Im Arbeitsbereich wird der Umriss des Objekts eingeblendet.

- 2 Verschieben Sie den **Function Generator** in die Mitte des Arbeitsbereichs und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Der **Function Generator** erscheint wie in Abbildung 6 dargestellt.

## 1 Die VEE-Entwicklungsumgebung verwenden



**Abbildung 6** Ein platziertes Function-Generator-Objekt

Nachdem Sie ein Objekt im Arbeitsbereich platziert haben, können Sie es durch Ziehen seiner Titelleiste verschieben, genau wie beim Verschieben eines Fensters.

### HINWEIS

Im weiteren Verlauf dieses Handbuchs wird für die Anleitungen eine Kurzschreibweise verwendet. Die Markierung bzw. Auswahl des Objekts Function Generator wird beispielsweise im folgender Weise dargestellt:

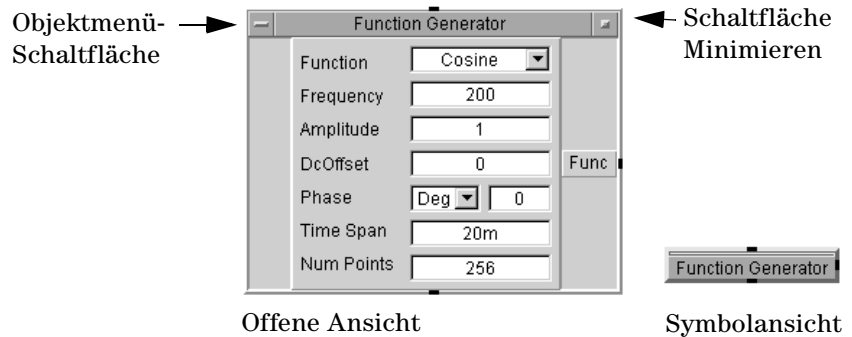
**Device** ⇒ **Virtual Source** ⇒ **Function Generator**

### HINWEIS

Um mehr Platz auf dem Bildschirm zu schaffen, wählen Sie **View** ⇒ **Program Explorer**. Damit heben Sie die Auswahl des Menüelements Program Explorer auf und schließen des entsprechende Fenster. Menüelemente sind „gewählt“, wenn sie mit einem Häkchen markiert sind.

## Die Ansicht von Objekten ändern

VEE zeigt Objekte in der „Symbolansicht“ oder der „offenen Ansicht“ an, wie in Abbildung 7 dargestellt.



**Abbildung 7** Objekt in offener Ansicht und Symbolansicht

Die Symbolansicht spart Platz im Arbeitsbereich und macht Programme besser lesbar. Die offene Ansicht bietet mehr Detailinformationen und ermöglicht das Bearbeiten der Eigenschaften und Einstellungen eines Objekts.

- 1 Klicken Sie zum Wechseln von der offenen zur Symbolansicht die Schaltfläche **Minimieren** (das Feld am rechten Rand der Titelleiste des Objekts) an.
- 2 Doppelklicken auf eine beliebige Stelle in der Symbolansicht des Objekts, um das Objekt wieder in der offenen Ansicht darzustellen.

## HINWEIS

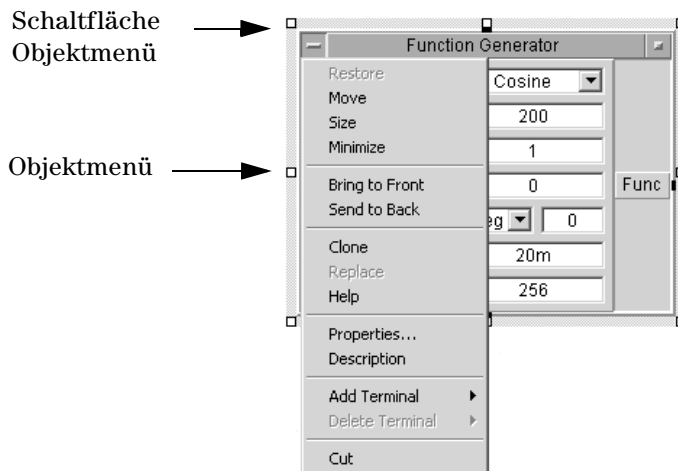
Das Objektmenü enthält ebenfalls Befehle zum Minimieren (Minimize) und Wiederherstellen (Restore). Klicken Sie am linken Rand der Titelleiste auf die Schaltfläche Objektmenü, oder klicken Sie mit der rechten Maustaste auf eine beliebige Stelle des Objekts, um das Objektmenü zu öffnen.

Nicht alle Objekte haben die gleiche Struktur oder die gleichen Elemente; für alle Objekte gilt jedoch: Sie können das Objekt in der offenen Ansicht bearbeiten oder in der platzsparenden Symbolansicht anzeigen.

## Ein Objektmenü öffnen

Jedes VEE-Objekt hat ein *Objektmenü*, über das Sie Aktionen mit dem Objekt ausführen, z. B. Clone, Size, Cut, Move und Minimize. Die meisten Objekte haben ähnliche Attribute, es gibt jedoch je nach der Funktionalität des Objekts gewisse Unterschiede. Ausführliche Informationen zum Objektmenü eines bestimmten Objekts finden Sie in der Online-Hilfe.

- 1 Klicken Sie zum Öffnen des Objektmenüs *ein* Mal auf die Schaltfläche Objektmenü. (Alle Objektmenüs werden auf die gleiche Weise geöffnet.) Das Objektmenü wird wie in Abbildung 8 angezeigt. (Klicken Sie nicht doppelt auf die Schaltfläche „Objektmenü“. Das Doppelklicken des Objektmenüs ist der Direktaufruf zum Löschen des betreffenden Objekts.)
- 2 Sie können jetzt die Befehle des Objektmenüs wählen, um die gewünschte Aktion auszuführen. Zum Schließen des Menüs klicken Sie einfach auf einen leeren Bereich *außerhalb* des Menüs.

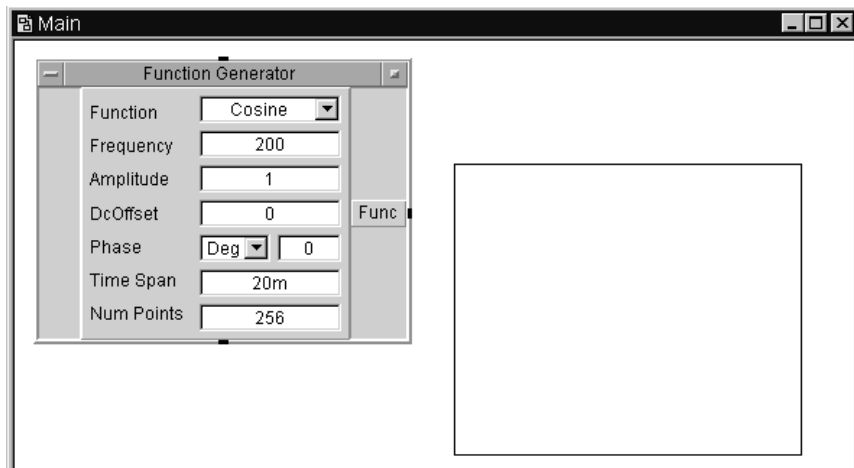


**Abbildung 8** Ein Objektmenü öffnen

Sie können das Objektmenü auch öffnen, indem Sie mit der Maus auf eine beliebige Stelle des Objektkörpers zeigen und mit der *rechten* Maustaste klicken. Dies funktioniert in der offenen und in der Symbolansicht.

## Ein Objekt verschieben

- 1 Wählen Sie zum Verschieben des Objekts Function Generator in dessen Objektmenü den Befehl **Move**, klicken Sie anschließend mit der linken Maustaste und halten Sie diese Taste gedrückt. Daraufhin wird ein Umriss des Objekts angezeigt.
- 2 Ziehen Sie den Umriss an die neue Position und halten Sie die Maustaste weiterhin gedrückt, wie in Abbildung 9 dargestellt. Lassen Sie die Maustaste los, um das Objekt an der neuen Position zu platzieren.



**Abbildung 9** Ein Objekt verschieben

Sie können Objekte auch auf die folgende Weise verschieben:

- Klicken Sie in der offenen Ansicht auf den Titelbereich eines Objekts und ziehen Sie das Objekt an eine neue Position.
- Klicken Sie in der offenen Ansicht auf einen beliebigen Teil eines Objekts (*gilt nicht bei* Schaltflächen, Eingabefeldern, Pins, Anschlüssen oder den vier Ziehmarken zum Ändern der Objektgröße) und ziehen Sie das Objekt an die neue Position.
- Klicken Sie in der Symbolansicht auf einen beliebigen Teil des Objekts *außer* in die Nähe der vier Ecken zum Ändern der Objektgröße und ziehen Sie das Objekt an die neue Position.

### HINWEIS

„Object Location Information“ in der Statusleiste (am unteren Rand des VEE-Fensters) zeigt die X- und Y-Position (in Pixel) der linken oberen Ecke des Umrisses in Relation zur linken oberen Ecke des Arbeitsbereichs an. Zum Anzeigen der genauen Position eines Objekts klicken Sie mit der linken Maustaste auf ein Objekt, um es auszuwählen, und halten Sie die linke Maustaste gedrückt. Die Position wird in der Statusleiste angezeigt. Verwenden Sie diese Informationen beim Platzieren eines Objekts an einer exakten Position.

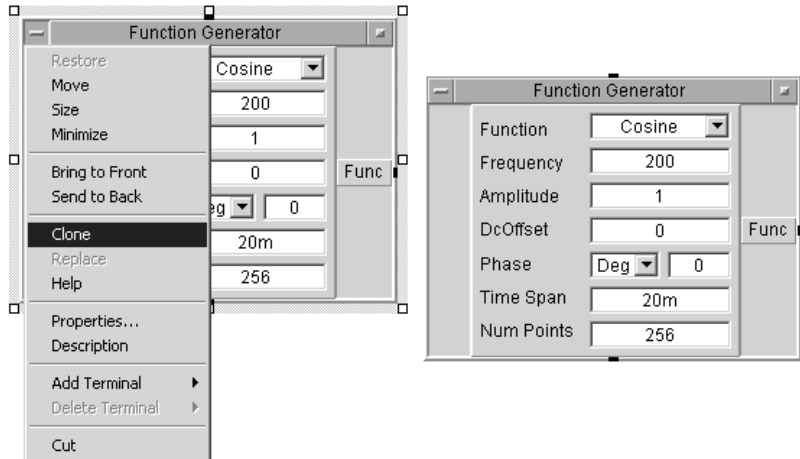
---

### Ein Objekt duplizieren (klonen)

Die Operation **Clone** erstellt das exakte Duplikat eines Objekts einschließlich aller vorgenommenen Änderungen wie beispielsweise einer Größenänderung oder einer neuen Beschriftung. Das Klonen ist eine Kurzform für das Ausschneiden und Einfügen.

- 1 Öffnen Sie das Objektmenü und wählen Sie Clone. Ein Umriss des duplizierten Objekts wird angezeigt.
- 2 Verschieben Sie den Umriss an die gewünschte Position und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Das geklonte Objekt erscheint, während das ursprüngliche Objekt weiterhin erhalten bleibt. In Abbildung 10 wurde der Function Generator bereits ein Mal geklont, und im Objektmenü ist der Befehl für ein erneutes Klonen ausgewählt.





**Abbildung 10** Ein Objekt klonen

## Ein Objekt kopieren

Mit dieser Aktion wird ein Objekt in die Zwischenablage kopiert, sodass Sie es in VEE oder in einer anderen Anwendung wie MS Paint oder MS Word einfügen können.

Klicken Sie ein auf Objekt, um es hervorzuheben, und wählen Sie anschließend **Edit** ⇒ **Copy**.

- ODER -

Klicken Sie auf ein Objekt, um es hervorzuheben, und drücken Sie anschließend **Strg+C**.

## Ein Objekt löschen

Um ein Objekts aus dem Arbeitsbereich zu löschen, öffnen Sie dessen Objektmenü und wählen Sie **Delete**. Öffnen Sie beispielsweise das Objektmenü des Function Generator und wählen Sie **Cut**. Das Objekt wird aus dem Arbeitsbereich entfernt und in die Zwischenablage übertragen.

## 1 Die VEE-Entwicklungsumgebung verwenden

Öffnen Sie das Objektmenü und wählen Sie Delete.

- ODER -

Wählen Sie das Objekt, indem Sie auf das Objekt klicken, und drücken Sie `Strg+X`.

- ODER -

Zeigen Sie mit dem Mauszeiger auf das Objektmenü und klicken Sie doppelt.

### HINWEIS

Seien Sie vorsichtig. Ein Objekt kann durch *Doppelklicken* auf die Schaltfläche Objektmenü leicht versehentlich gelöscht werden. Haben Sie ein Objekt versehentlich gelöscht, klicken Sie in der Symbolleiste auf die Schaltfläche Undo (oder wählen Sie **Edit** ⇒ **Undo**), um das Objekt und alle seine Verbindungen wiederherzustellen.

## Ein Objekt einfügen

Gehen Sie zum Einfügen eines kopierten oder gelöschten (ausgeschnittenen) Objekts in den Arbeitsbereich wie folgt vor:

Nachdem Sie ein Objekt kopiert oder gelöscht haben, wählen Sie **Edit** ⇒ **Paste**. Daraufhin wird ein Umriss des Objekts angezeigt. Platzieren Sie das Objekt und klicken Sie mit der Maustaste, um es freizugeben.

- ODER -

Drücken Sie `Strg+V`.

## Die Größe eines Objekts ändern

Zeigen Sie auf eine der vier Ecken des Objekts, bis der Mauszeiger die Form eines Doppelpfeils annimmt. Klicken jetzt auf das Objekt und ziehen Sie es auf die gewünschte Größe. Lassen Sie die Maustaste los, um die neue Größe zu übernehmen. Abbildung 11 zeigt ein Objekt, dessen Größe auf diese Weise geändert wurde.

- ODER -

Öffnen Sie das Objektmenü und wählen Sie **Size**. Der Mauszeiger nimmt die Form eines Doppelpfeils an. Verschieben Sie den Mauszeiger an die gewünschte Position nach rechts unten und klicken Sie mit der Maustaste, um die neue Größe zu übernehmen.

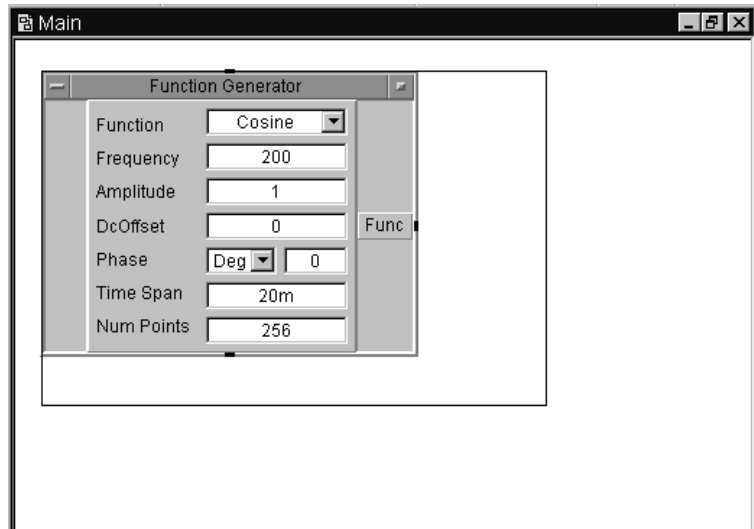


Abbildung 11 Die Größe eines Objekts ändern

## Den des Namen (Titel) eines Objekts ändern

- 1 Öffnen Sie das Objektmenü und wählen Sie **Properties...**  
Auf der linken Seite des Anwendungsfensters wird Fenster Properties angezeigt. Wählen Sie die Eigenschaft Title und ändern Sie den Wert der Eigenschaft wunschgemäß, wie in Abbildung 12 dargestellt.
- 2 Der neue Titel wird im Titelbereich angezeigt. Wenn Sie das Objekt minimieren, erscheint der neue Titel in dem Symbol.

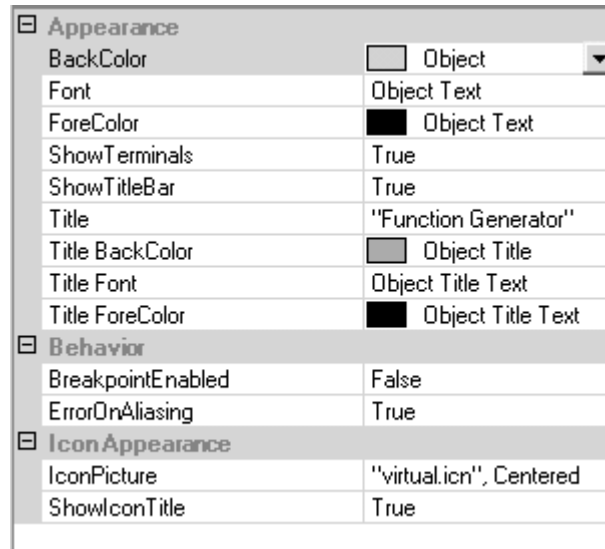
- ODER -

Doppelklicken Sie auf die Titelleiste des Objekts, um das Eigenschaftfenster (Properties) direkt zu öffnen.

## 1 Die VEE-Entwicklungsumgebung verwenden

### HINWEIS

Mit der richtigen Anwendung von Tastatur- und Maustechniken können Sie bei der Bearbeitung viel Zeit sparen. Wenn Sie beispielsweise im Fenster Properties an den linken Rand des Bearbeitungsbereichs des Felds Title klicken, wird dort eine Einfügemarke angezeigt. Sie können jetzt neuen Text eingeben, ohne den vorhandenen Text zu löschen.



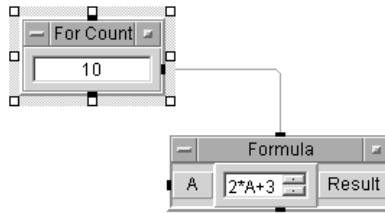
The screenshot shows a 'Properties' window with three expandable sections: 'Appearance', 'Behavior', and 'Icon Appearance'. The 'Title' property in the 'Appearance' section is selected, and a text cursor is visible at the end of the text 'Function Generator'.

<b>Appearance</b>	
BackColor	Object
Font	Object Text
ForeColor	Object Text
ShowTerminals	True
ShowTitleBar	True
Title	"Function Generator"
Title BackColor	Object Title
Title Font	Object Title Text
Title ForeColor	Object Title Text
<b>Behavior</b>	
BreakpointEnabled	False
ErrorOnAliasing	True
<b>Icon Appearance</b>	
IconPicture	"virtual.icn", Centered
ShowIconTitle	True

Abbildung 12 Den Titel eines Objekts ändern

### Objekte wählen oder Auswahl aufheben

- 1 Klicken Sie auf ein Objekt, um das betreffende Objekt zu wählen. Das Objekt wird nun schattiert dargestellt. In Abbildung 13 ist beispielsweise das Objekt For Count gewählt.
- 2 Zeigen Sie zum Aufheben der Auswahl eines Objekts mit der Maus auf eine beliebige leere Stelle im Arbeitsbereich und klicken Sie mit der Maustaste. Der Schatten wird entfernt. Das Formula-Objekt in Abbildung 13 ist beispielsweise nicht gewählt.



**Abbildung 13** Gewählte und nicht gewählte Objekte

## HINWEIS

Das Wort „Wählen“ wird auch im Zusammenhang mit Menübefehlen verwendet; durch den jeweiligen Kontext ist die Bedeutung jedoch klar.

## Mehrere Objekte wählen

Wenn Sie auf ein Objekt klicken, wird nur das betreffende Objekt gewählt. Wenn Sie erneut klicken, um ein weiteres Objekt zu wählen, wird die Auswahl des vorigen Objekts aufgehoben, und der entsprechende Schatten wird entfernt. Wenn Sie mehrere Objekte wählen möchten, um eine Operation (beispielsweise Delete) mit allen diesen Objekten auszuführen, gehen Sie wie folgt vor:

Drücken Sie die Taste `Strg` und halten Sie sie gedrückt, während Sie auf die einzelnen Objekte klicken. Lassen Sie die Taste `Strg` los, wenn alle gewünschten Objekte markiert sind.

- ODER -

Drücken Sie `Strg` und ziehen Sie einen Markierungsrahmen um die Objekte, die sie wählen möchten. Die gewählten Objekte werden mit einem Schatten unterlegt dargestellt.

## Alle Objekte wählen/Auswahl aufheben

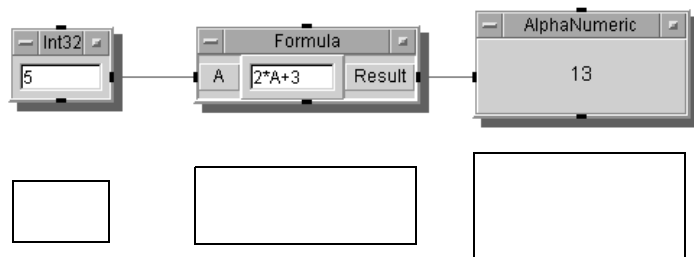
- 1 Wenn Sie alle Objekte markieren möchten, wählen Sie **Edit** ⇒ **Select All**. (Oder drücken Sie `Strg+A`.)
- 2 Klicken Sie auf eine leere Stelle im Arbeitsbereich, um die Auswahl aller Objekte aufzuheben.

### Mehrere Objekte kopieren

Kopieren Sie die gewählten Objekte, indem Sie mit dem Mauszeiger auf eines der Objekte zeigen. Drücken Sie die Taste **Strg** und halten Sie sie gedrückt, während Sie mit der linken Maustaste die Objekte (Umrisse) an die gewünschte Position ziehen. An dieser Position werden Kopien der gewählten Objekte eingefügt.

- ODER -

Verwenden Sie **Edit** ⇒ **Copy**, um die gewählten Objekte in die Zwischenablage zu kopieren. Klicken Sie (im Menü Edit oder in der Symbolleiste) auf **Paste**, verschieben Sie die Objekte (Umrisse) an die gewünschte Position und klicken Sie mit der linken Maustaste. Abbildung 14 zeigt die Objekte beim Kopiervorgang.



**Abbildung 14** Mehrere Objekte beim Kopieren

### HINWEIS

In VEE für Windows werden Objekte, die Sie ausschneiden oder kopieren, in die Zwischenablage übertragen. Sie können diese Objekte in anderen Windows-Anwendungen einfügen, sofern die jeweilige Anwendung die Windows-Zwischenablage unterstützt.

### Objekte bearbeiten

Objekte können in VEE auf verschiedene Weise bearbeitet werden. Verschiedene Bearbeitungsmenüs enthalten unterschiedliche Befehle. Wählen Sie eine Bearbeitungsmethode (in einem Menü oder über ein Symbol) wie folgt:

Klicken Sie in der VEE-Menüleiste auf **Edit**, um das Menü Edit zu öffnen, und wählen Sie dann den gewünschten Befehl. Die Befehle im Menü Edit sind in VEE in jeder Situation gleich.

- ODER -

Klicken Sie in der VEE-Symboleiste auf ein Symbol. Die VEE-Symboleiste enthält Symbole für häufig verwendete Bearbeitungsbefehle wie Cut (Ausschneiden), Copy (Kopieren) und Paste (Einfügen).

- ODER -

Öffnen Sie das Objektmenü eines Objekts, indem Sie auf das Objekt klicken, und wählen Sie den gewünschten Befehl. Objektmenüs enthalten spezifische Bearbeitungsbefehle für ein Objekt, beispielsweise den Befehl Properties, die im Hauptmenü unter Edit nicht enthalten sind. Welche Befehle ein Objektmenü enthält, hängt vom Typ des jeweiligen Objekts ab. Vergleichen Sie beispielsweise die Objektmenüs der Objekte **Device** ⇒ **Formula** und **I/O** ⇒ **To** ⇒ **File**. Die beiden Menüs enthalten unterschiedliche, speziell auf das jeweilige Objekt zugeschnittene Befehle.

- ODER -

Zeigen Sie mit der Maus auf eine beliebige *leere* Stelle im Arbeitsbereich und klicken Sie mit der *rechten* Maustaste. Daraufhin wird ein Kontextmenü zum Bearbeiten (Edit) geöffnet.

## HINWEIS

Inaktive Menüelemente werden in einer anderen Farbe dargestellt als aktive Elemente (sie sind „abgeblendet“). Die Befehle Cut, Copy und Clone im Menü Edit werden beispielsweise so lange anders als aktive Menüelemente dargestellt, bis ein Objekt im Arbeitsbereich markiert ist.

## Datenlinien zwischen Objekten erstellen

- 1 Klicken Sie auf einen Datenausgangs-Pin eines Objekts (oder knapp daneben) und anschließend auf den Dateneingangs-Pin eines anderen Objekts, wie in Abbildung 15 dargestellt.

## 1 Die VEE-Entwicklungsumgebung verwenden

(Hinter dem Mauszeiger wird eine Linie angezeigt, während Sie ihn von einem Pin zum anderen verschieben.)

- 2 Lassen Sie die Maustaste los. VEE zeichnet eine Linie zwischen den beiden Objekten. Beachten Sie, dass VEE beim neu Positionieren der Objekte die Linie zwischen den Objekten beibehält.

### HINWEIS

Weitere Informationen zu Pins finden Sie unter „Informationen zu Pins und Anschlüssen“ auf Seite 49.

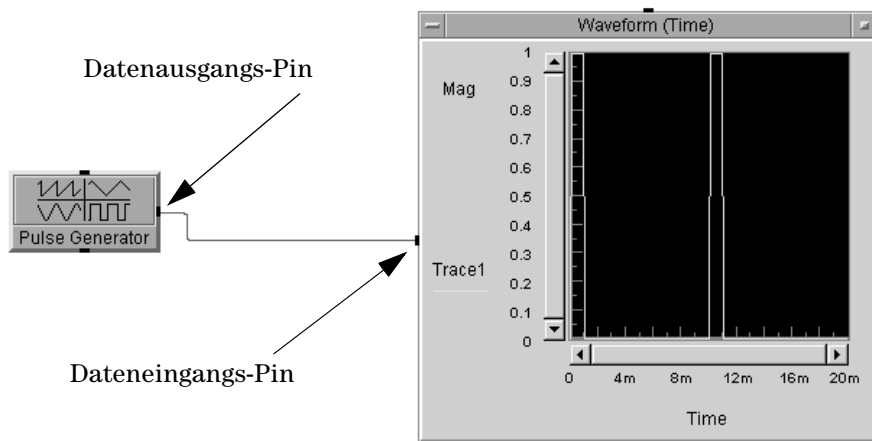


Abbildung 15 Datenlinien zwischen Objekten erstellen

## Datenlinien zwischen Objekten löschen

Drücken Sie die Tastenkombination Umschalt+Strg und klicken Sie auf die zu löschende Linie.

- ODER -

Wählen Sie **Edit** ⇒ **Delete Line** und klicken Sie auf die zu löschende Linie.



## Den sichtbaren Teil des Arbeitsbereichs verschieben

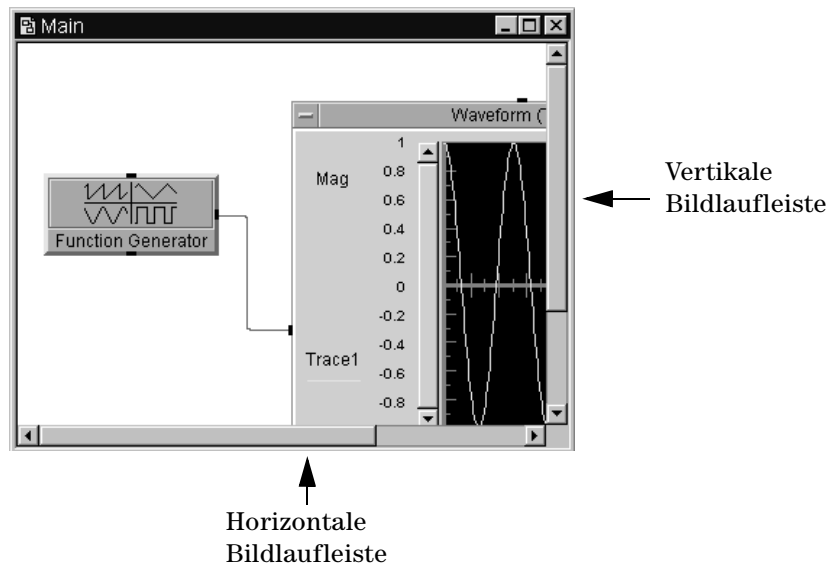
(Vergewissern Sie sich, dass sich mindestens ein Symbol im Arbeitsbereich befindet.) Zeigen Sie mit dem Mauszeiger auf eine beliebige leere Stelle im Arbeitsbereich, halten Sie die linke Maustaste gedrückt und verschieben Sie den Arbeitsbereich mit der Maus in eine beliebige Richtung.

### HINWEIS

Wenn Ihr Programm auf dem Arbeitsbereich mehr Platz einnimmt als dargestellt werden kann, werden Bildlaufleisten eingeblendet (siehe Abbildung 16).

### HINWEIS

Wenn Sie auf einen normalen Anschluss klicken, kann eine Linie angezeigt werden. Zeigen Sie in diesem Fall auf eine leere Stelle im Arbeitsbereich und doppelklicken Sie.



**Abbildung 16** Bildlaufleisten im Arbeitsbereich

### Den Arbeitsbereich löschen

Wählen Sie **Edit** ⇒ **Select All** und klicken Sie anschließend in der Symbolleiste auf die Schaltfläche Cut. Dadurch werden alle Objekte im aktiven Fenster in die Zwischenablage übertragen.

- ODER -

Wählen Sie **File** ⇒ **New** oder klicken Sie in der Symbolleiste auf die Schaltfläche New. VEE fragt, ob Änderungen gespeichert werden sollen.

- ODER -

Wenn Sie ein einzelnes Objekt löschen wollen, klicken Sie auf das Objekt, um es zu markieren, und klicken Sie anschließend in der Symbolleiste auf die Schaltfläche Cut.

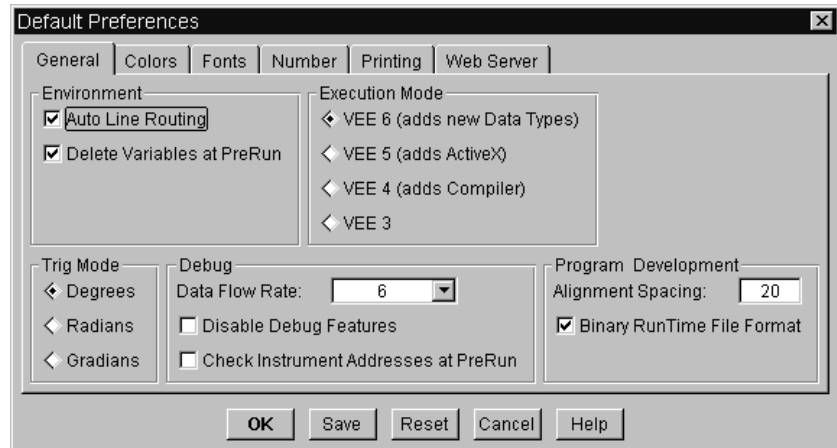
### Standardeinstellungen ändern

Die Standardeinstellungen der VEE-Umgebung können im Dialogfeld Default Preferences (Standardeinstellungen) geändert werden.

Klicken Sie in der Symbolleiste auf die Schaltfläche Default Preferences.

- ODER -

Wählen Sie **File** ⇒ **Default Preferences**. Das Dialogfeld Default Preferences wird geöffnet (siehe Abbildung 17).



**Abbildung 17** Das Dialogfeld Default Preferences

Dieses Dialogfeld enthält *Register*, über die Sie verschiedene Kategorien von Optionen einstellen können.

**Tabelle 3** Beschreibung der Register des Dialogfelds Default Preferences

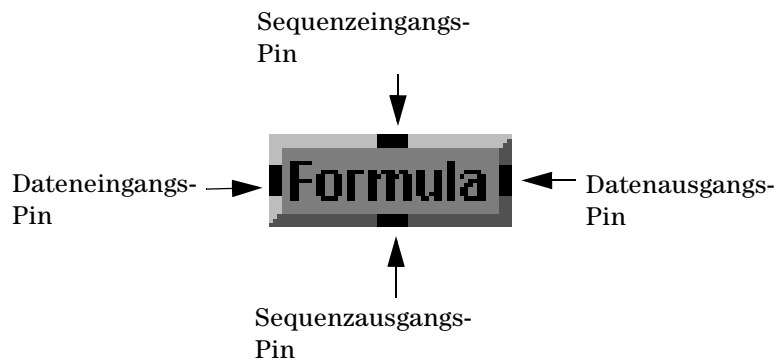
Registername	Beschreibung
General (Allgemein)	Das Standardregister, das beim Öffnen des Dialogfelds Default Preferences aktiv ist (zuvor angezeigt). Sie können die Werte der angezeigten Optionen ändern, beispielsweise unter Environment (Umgebung) und Execution Mode (Ausführungsmodus).
Colors (Farben)	Ermöglicht das Anpassen der Farben in der VEE-Umgebung.
Fonts (Schriftarten)	Ermöglicht das Anpassen der Schriftarten in der VEE-Umgebung.
Number (Zahlen)	Ermöglicht das Ändern des Standardzahlenformats.
Printing (Drucken)	Erlaubt das Ändern die Parameterwerte für einen Drucker.
Web Server	Ermöglicht das Aktivieren des integrierten Web-Servers zur Überwachung und Fehlerbehebung eines Programms mittels eines auf einem anderen Computer ausgeführten Web-Browsers.

## 1 Die VEE-Entwicklungsumgebung verwenden

Weitere Informationen erhalten Sie, wenn Sie in der VEE-Menüleiste **Help** ⇒ **Contents and Index** wählen. Blättern Sie anschließend durch **How Do I...**, **Tell Me About...** oder **Reference**.

## Informationen zu Pins und Anschlüssen

Ein VEE-Programm *besteht aus* den Objekten im Arbeitsbereich *und* den Linien, die diese Objekte verbinden. Die Linien, die VEE-Objekte miteinander verbinden, sind an die *Pins* der Objekte „angeschlossen“. Jedes Objekt hat verschiedene Pins, wie in Abbildung 18 gezeigt. Abbildung 18 verwendet das Formula-Objekt als Beispiel. Sie können jedes beliebige Objekt verwenden .



**Abbildung 18** Daten- und Sequenz-Pins

**Tabelle 4**

Pin-Typ	Beschreibung
Dateneingangs-Pin	Der Pin (bzw. die Pins) auf der linken Seite eines Objekts.
Datenausgangs-Pin	Der Pin (bzw. die Pins) auf der rechten Seite eines Objekts.
Sequenzeingangs-Pin	Der Pin an der Oberseite eines Objekts.
Sequenzausgangs-Pin	Der Pin an der Unterseite eines Objekts.

## 1 Die VEE-Entwicklungsumgebung verwenden

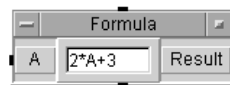
Verbinden Sie die Dateneingangs- und -ausgangs-Pins zum Transport von Daten zwischen Objekten. Standardmäßig werden die Pins von oben nach unten ausgeführt. Die Verbindungen der Sequenz-Pins sind optional. Wenn sie verbunden sind, legen sie die Reihenfolge der Ausführung fest.

### HINWEIS

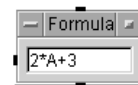
Weitere Informationen hierzu finden Sie unter „Die Ereignisreihenfolge in einem Objekt verfolgen“ auf Seite 115.

In einer offenen Ansicht eines Objekts erscheinen die Dateneingangs- und -ausgangs-Pins als Eingangs- und Ausgangsanschlüsse. (Wenn das Objekt in der Symbolansicht angezeigt wird, klicken Sie doppelt darauf, um zur offenen Ansicht zu wechseln.) Die Anschlüsse enthalten Detailinformationen wie den Namen des Anschlusses, den Typ und den Wert der übertragenen Daten. Die Anschlussbeschriftungen sind nur in der offenen Ansicht sichtbar und nur, wenn die Eigenschaft Show Terminals (Anschlüsse anzeigen) für dieses Objekt auf True (Wahr) gesetzt ist (siehe Properties... im Objektmenü).

Abbildung 19 zeigt beispielsweise zwei Formula-Objekte. Das Formula-Objekt auf der linken Seite zeigt die Terminal-Beschriftungen **A** und **Result**. Für das Formula-Objekt auf der rechten Seite ist die Show Terminals ausgeschaltet, und die Beschriftungen sind nicht sichtbar.



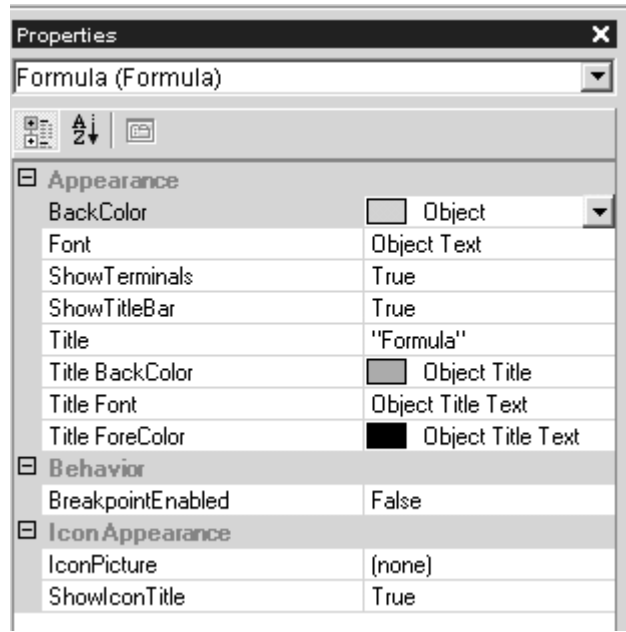
Show terminals  
ist eingeschaltet



Show terminals  
ist ausgeschaltet

**Abbildung 19** Show Terminals bei einem Objekt

Wählen Sie zum Ein- oder Ausschalten von Show Terminals im Objektmenü den Befehl Properties. Auf der linken Seite des Bildschirms wird das Fenster Properties angezeigt. Wählen Sie die Eigenschaft ShowTerminals und setzen Sie sie auf True, (siehe Abbildung 20).



**Abbildung 20** Die Eigenschaft ShowTerminals setzen

Wählen Sie die Eigenschaft **ShowTerminals** und setzen Sie sie auf **False**. Wählen Sie die Eigenschaft **ShowTerminals** und setzen Sie sie auf **True**.

## Einen Anschluss hinzufügen

Sie können einem Objekt Anschlüsse hinzufügen. Beispielsweise können Sie dem Formula-Objekt einen zweiten Dateneingangsanschluss hinzufügen.

Öffnen Sie das Objektmenü und wählen Sie **Add Terminal** ⇒ **Data Input**.

- ODER -

Wenn Show Terminals eingeschaltet ist, können Sie auch mit der Maus auf den „Anschlussbereich“ (den linken Rand des Objekts in der offenen Ansicht) zeigen und **Strg+A** drücken (drücken Sie **Strg** und **A** gleichzeitig).

## 1 Die VEE-Entwicklungsumgebung verwenden

Abbildung 21 zeigt das Objektmenü der Formula-Objekts und die Befehlsfolge zum Hinzufügen eines Dateneingangsanschlusses sowie ein weiteres Formula-Objekt, dem bereits ein zweiter Anschluss hinzugefügt wurde. Der neue Anschluss ist mit **B** beschriftet. Wenn die Dateneingänge an bestimmte Funktionen gebunden sind, wie etwa bei Gerätetreibern, wird ein Menü mit diesen Funktionen angezeigt. Andernfalls haben die Anschlüsse die Namen **A, B, C...**

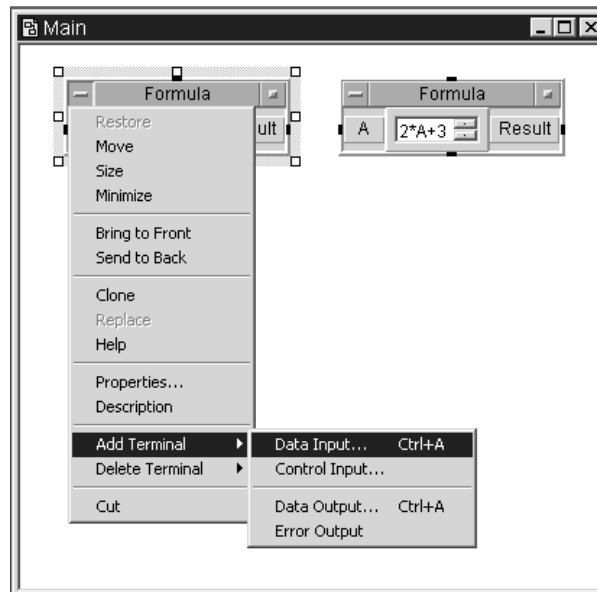
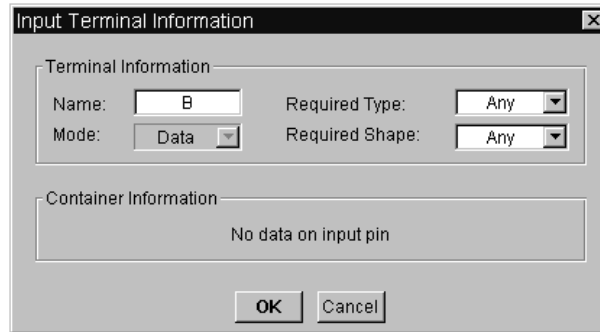


Abbildung 21 Einen Anschluss hinzufügen

### Anschlussinformationen bearbeiten

Zum Abrufen von Informationen zu einem Anschluss klicken Sie doppelt auf den Beschriftungsbereich. Wenn Sie beispielsweise auf **B** doppelklicken, wird das Dialogfeld in Abbildung 22 geöffnet.





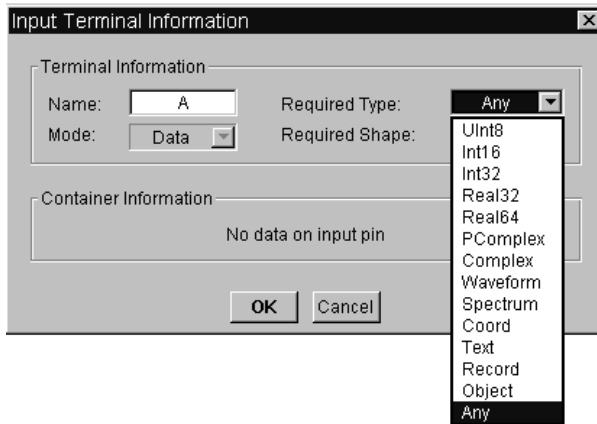
**Abbildung 22** Anschlussinformationen abrufen

Sie können den Anschluss jetzt bearbeiten. Das Dialogfeld enthält drei Arten von Feldern:

**Tabelle 5**

Feldtyp	Beschreibung
Eingabefeld	Ein Feld mit einem weißen Hintergrund, aber ohne Pfeil. Wenn Sie auf dieses Feld klicken, können Sie Daten <i>eingeben</i> . Sie können beispielsweise im Feld <b>Name</b> auf <b>B</b> klicken und den Anschluss dann umbenennen.
Statusfeld	Ein Feld mit einem grauen Hintergrund; es kann nicht geändert werden. Das Feld <b>Mode</b> kann beispielsweise nicht geändert werden.
Dropdown-Liste	Ein Feld mit einem weißen Hintergrund, das auf der rechten Seite eine Schaltfläche mit einem Pfeil hat. Ein Klick auf diese Schaltfläche „klappt“ eine Liste auf. Wenn Sie beispielsweise im Feld <b>Required Type</b> auf <b>Any</b> (oder den Pfeil) klicken, können Sie dem Feld einen anderen Datentyp zuordnen, indem Sie den Typ in der Liste wie in <b>Abbildung 23</b> gezeigt wählen.

## 1 Die VEE-Entwicklungsumgebung verwenden



**Abbildung 23** Die Dropdown-Liste verwenden

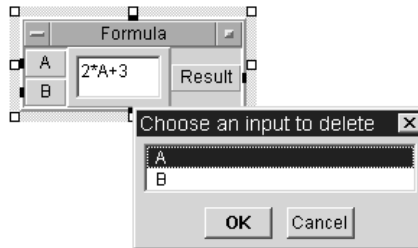
Wenn Sie einen anderen Datentyp als **Any** für einen Dateneingangsanschluss wählen, werden nur diesem Typ entsprechende Daten bzw. Daten, die in diesen Typ umgewandelt werden können, von dem Anschluss akzeptiert. In den meisten Fällen ist es am besten, Required Type und Required Shape auf Any eingestellt zu lassen. Weitere Informationen erhalten Sie über **Help** ⇒ **Contents and Index**. Blättern Sie anschließend durch **How Do I...**, **Tell Me About...** oder **Reference**.

### Einen Anschluss löschen

Öffnen Sie das Objektmenü und wählen Sie **Delete Terminal** ⇒ **Input...** oder **Delete Terminal** ⇒ **Output**, wählen Sie den zu löschenden Ein- oder Ausgang und klicken Sie auf **OK**. Abbildung 24 zeigt beispielsweise das Dialogfeld, das geöffnet wird, wenn Sie **Delete Terminal** ⇒ **Input...** wählen.

- ODER -

Zeigen Sie auf den Anschluss und drücken Sie **Strg+D**.



**Abbildung 24** Das Dialogfeld „Delete Terminal“

Wenn Sie das Löschen des Anschlusses rückgängig machen möchten, dann klicken Sie in der Symbolleiste auf die Schaltfläche Undo.

# Objekte zum Erstellen eines Programms verbinden

In diesem Abschnitt werden VEE-Programme vorgestellt. In Übung 1-1 erstellen Sie ein VEE-Programm, drucken die VEE-Anzeige und speichern das Programm in einer Datei.

## Übung 1-1 Das Programm Display Waveform (Wellenform anzeigen)

Ein VEE-Programm besteht aus VEE-Objekten, die in einem ausführbaren *Objektdiagramm* verbunden wurden. Das folgende Programm zeigt eine Wellenform an.

(Wenn VEE bereits ausgeführt wird, löschen Sie den Inhalt des Arbeitsbereichs, indem Sie in der Symbolleiste auf die Schaltfläche New klicken oder **File** ⇒ **New** wählen. Andernfalls starten Sie VEE und fahren mit dem folgenden Schritt fort.)

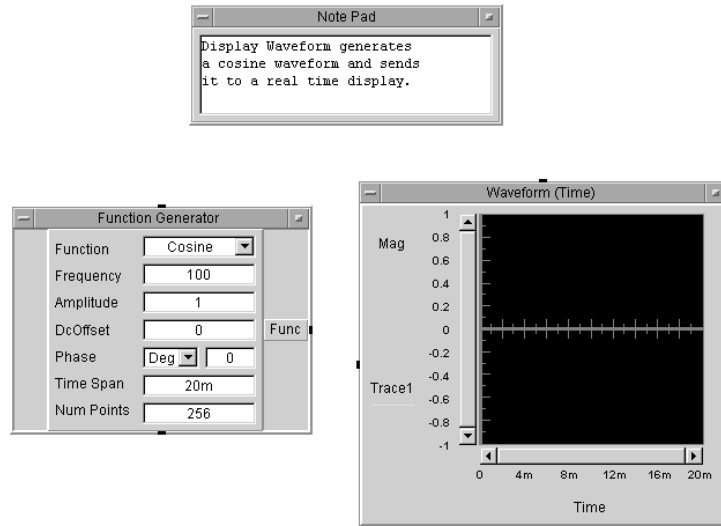
- 1 Dokumentieren Sie das Programm. Wählen Sie **Display** ⇒ **Note Pad** und platzieren Sie diesen Notizblock oben in der Mitte des Arbeitsbereichs. Klicken Sie auf den Bearbeitungsbereich des Notizblocks, um eine Einfügemarke anzuzeigen. Entfernen Sie eventuell vorhandene Vorlageninformationen und geben Sie dann ein:

```
Display Waveform generates a cosine waveform  
and sends it to a real time display.
```

Je nach Bildschirmgröße bzw. Auflösung müssen Sie den Notizblock eventuell verkleinern. Öffnen Sie zum Ändern der Größe das Objektmenü, wählen Sie *Size*, zeigen Sie mit dem Mauszeiger auf eine Ecke des Objekts und ziehen Sie die Ecke an die gewünschte Position. Sie können auch auf eine beliebige Ecke des Objekts klicken und dann ziehen.)

- 2 Fügen Sie das Objekt Function Generator hinzu. Wählen Sie **Device** ⇒ **Virtual Source** ⇒ **Function Generator**, positionieren Sie den Umriss auf der linken Seite des Arbeitsbereichs und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Ändern Sie die Frequenz auf 100, indem Sie auf das Feld Frequency klicken und 100 eingeben.

- 3 Fügen Sie das Objekt Waveform (Time) hinzu. Wählen Sie **Display** ⇒ **Waveform (Time)** und platzieren Sie das Objekt auf der rechten Seite des Arbeitsbereichs, wie in Abbildung 25 gezeigt.



**Abbildung 25** Ein Programm erstellen

In Abbildung 25 weist die Beschriftung Func am Objekt Function Generator auf einen *Datenausgangs-Pin* hin, die Beschriftung Trace1 am Objekt Waveform(Time) dagegen auf einen *Dateneingangs-Pin*. In VEE-Programmen verbinden Sie die Daten-Pins der Objekte und legen dadurch den Ablauf des Programms fest.

- 4 Vervollständigen Sie das Programm, indem Sie den Datenausgangs-Pin am Function Generator (neben Func auf der rechten Seite) mit dem Dateneingangs-Pin an der Anzeige Waveform (Time) (neben Trace1 auf der linken Seite verbinden). Verschieben Sie zum Verbinden den Mauszeiger auf einen der Pins.

Der Mauszeiger nimmt eine andere Form an, wenn er sich in der Nähe eines Pins befindet, an dem eine Verbindung zulässig ist. Klicken Sie mit der linken Maustaste, zeigen Sie mit

## 1 Die VEE-Entwicklungsumgebung verwenden

dem Mauszeiger auf den anderen Pin und klicken Sie erneut. Dadurch wird automatisch eine Linie zwischen den beiden Pins geführt, und das Programm ist fertig.

Versuchen Sie, eines dieser Objekte durch Ziehen seiner Titelleiste zu verschieben. (Ziehen Sie nicht an einem Pin oder einem Anschluss; dadurch wird eine Linie angezeigt.) Die Linie wird automatisch an dem logischen Pfad zwischen den beiden Objekten angelegt.

Wenn die Linien verworren erscheinen, verwenden Sie **Edit** ⇒ **Clean Up Lines** (Linien bereinigen), um die Linien im Programm neu zu ziehen.

### Ein Programm ausführen

- 5 Klicken Sie zur Fortsetzung dieser Übung in der Symbolleiste auf die Schaltfläche Run, um das Programm auszuführen, oder wählen Sie **Debug** ⇒ **Run**. Das Programm zeigt eine 100 Hz Kosinuskurve in der Anzeige Waveform (Time) an, wie in Abbildung 26 gezeigt. (Ihr Objekt hat eventuell eine andere Frequenz; dies ist jedoch für das Beispiel nicht von Bedeutung.)

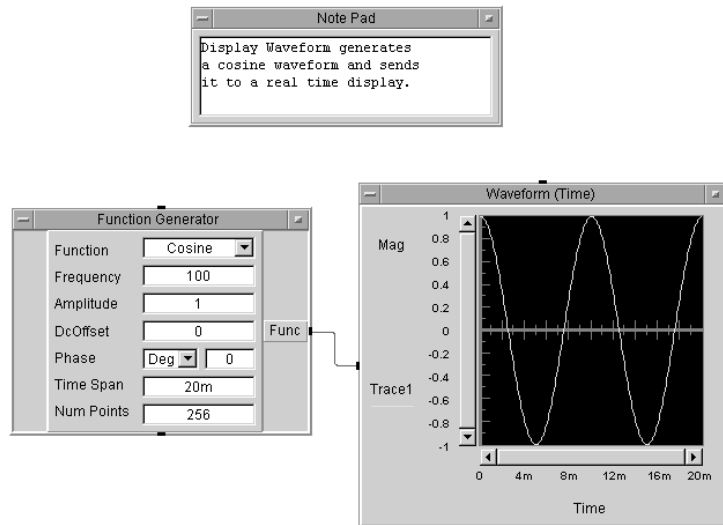


Abbildung 26 Ausführen eines Programms

Zusätzlich zu der Schaltfläche Run in der Symbolleiste können Sie auch die Schaltflächen Stop, Pause und Step (ebenfalls in der Symbolleiste) zur Steuerung des Programms verwenden. Wenn Sie ein laufendes Programm anhalten, verwenden Sie zur Fortsetzung die Schaltfläche Resume (an der gleichen Stelle wie Run). Mit der Schaltfläche Step in der Symbolleiste können Sie ein Programm schrittweise (Objekt für Objekt) ausführen.

Wenn Sie zur Ausführung des Programms aufgefordert werden, klicken Sie in der Symbolleiste auf die Schaltfläche Run oder drücken Sie `Strg+G`. Weitere Tastaturkürzel sind:

**Tabelle 6**

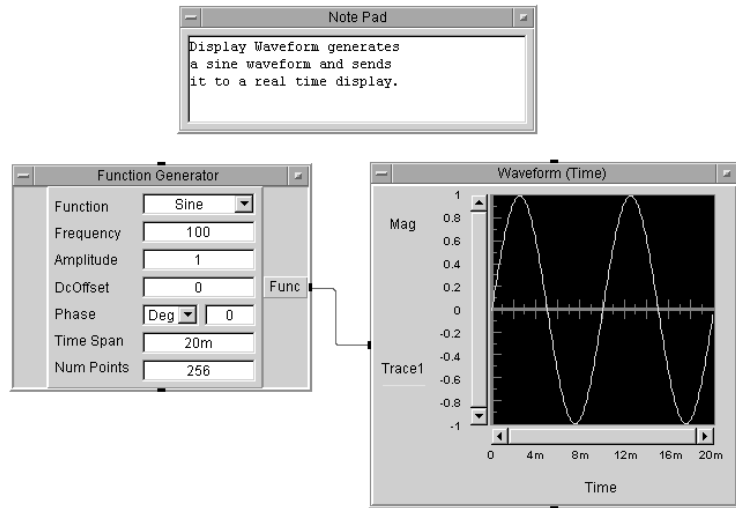
Befehl	Tastenkombination
Pause	<code>Strg+P</code>
Resume (Fortsetzen)	<code>Strg+G</code>
Step Into (Schritt in)	<code>Strg+T</code>

## Objekteigenschaften ändern

Sie haben gesehen, wie einige Eigenschaften eines Objekts im Fenster Properties (Objektmenü  $\Rightarrow$  **Properties**) geändert werden können. Sie können die wichtigsten Eigenschaften eines Objekts auch direkt in seiner offenen Ansicht ändern. Sie haben vielleicht schon bemerkt, dass das Objekt Function Generator zwei Arten von Feldern enthält. Ein Feld mit einem Pfeil auf der rechten Seite ist ein Dropdown-Listefeld.

- Zur Fortsetzung des Beispiels klicken Sie im Feld Function auf Cosine (oder den Pfeil). Es erscheint eine Dropdown-Liste mit Auswahloptionen. Klicken Sie auf Sine, um die Sinusfunktion zu wählen, wie in Abbildung 27 gezeigt. Beachten Sie, dass das Feld Function von Cosine in Sine geändert wurde.

## 1 Die VEE-Entwicklungsumgebung verwenden

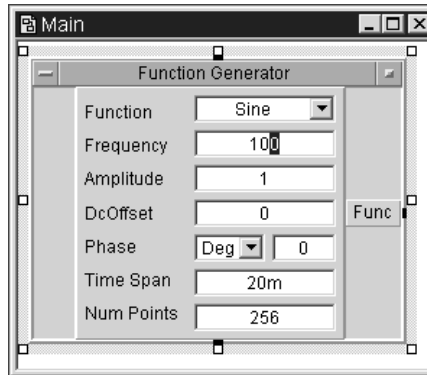


**Abbildung 27** Eine Sinuskurve für das Feld Function wählen

Einige Felder in Dialogfeldern enthalten keine Pfeile. Dies sind Eingabefelder, in die Sie Daten *eintippen* können, nachdem Sie auf die Felder geklickt haben. Klicken Sie einfach auf ein Feld, um eine Einfügemarke anzuzeigen. Sie können die üblichen Tastatur- und Maustechniken zur Bearbeitung verwenden, um die Einfügemarke zu verschieben und den gewünschten Wert einzugeben.

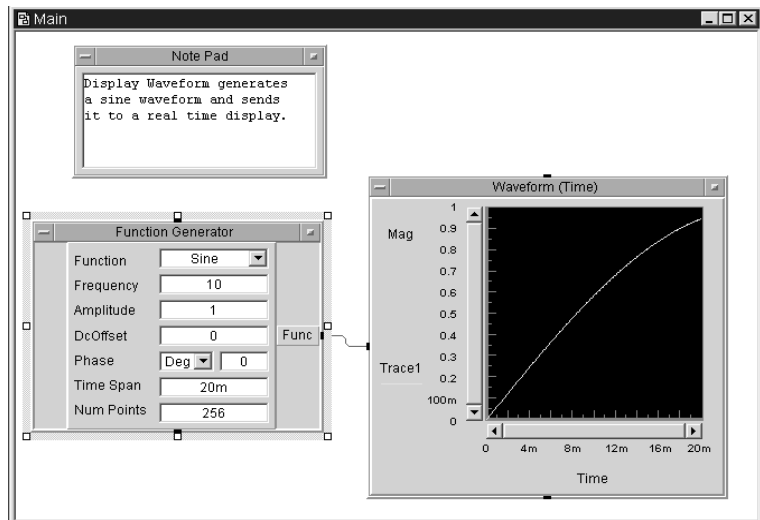
- 7 Klicken Sie das auf Feld Frequency rechts von dem Wert **100**, halten Sie die Maustaste gedrückt und ziehen Sie die Maus nach links, um die letzte **0** zu markieren, wie in Abbildung 28 gezeigt.





**Abbildung 28** Eine Zahl im Feld Frequency markieren

- 8 Drücken Sie die Taste `Ent f`, um die letzte **0** zu löschen. Der Wert für Frequency beträgt jetzt **10**. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 29.



**Abbildung 29** Beispiel: Den Wert Felds Frequency auf 10 Hz ändern

## 1 Die VEE-Entwicklungsumgebung verwenden

Die angezeigte Wellenform ist jetzt eine Sinuskurve mit einer Frequenz von 10 Hz. Sie können einige Objektparameter wie folgt ändern:

- Klicken Sie auf **Deg** (oder den Pfeil) im Objekt Function Generator und ändern Sie die Phaseneinheit in Rad. Klicken Sie als nächstes auf das Feld Phase und geben Sie den Wert  $\pi$  ein. Führen Sie das Programm erneut aus und beachten Sie die Phasenverschiebung in der angezeigten Wellenform. Setzen Sie anschließend den Wert des Felds Phase wieder auf 0 und die Einheit wieder auf Deg.
- Die Y-Achsenbegrenzungen des Objekts Waveform (Time) sind voreingestellt auf die Werte -1 bis 1. Klicken Sie auf Namen der Y-Achse Mag, um ein Dialogfeld zu öffnen, in dem Sie die Werte ändern können. Klicken Sie auf die Felder für das Maximum und Minimum, um die Begrenzungen in 2 und -2 zu ändern. Die Wellenform wird mit den neuen Begrenzungen angezeigt. Klicken Sie zum Ändern vergleichbarer Parameter für die Skala der X-Achse auf Time.

## Die Anzeige drucken

- 9 Setzen Sie nun das Beispiel fort und drucken Sie die Anzeige, indem Sie **File**  $\Rightarrow$  **Print Screen** wählen. Unter Windows wird das in Abbildung 30 dargestellte Dialogfeld geöffnet.

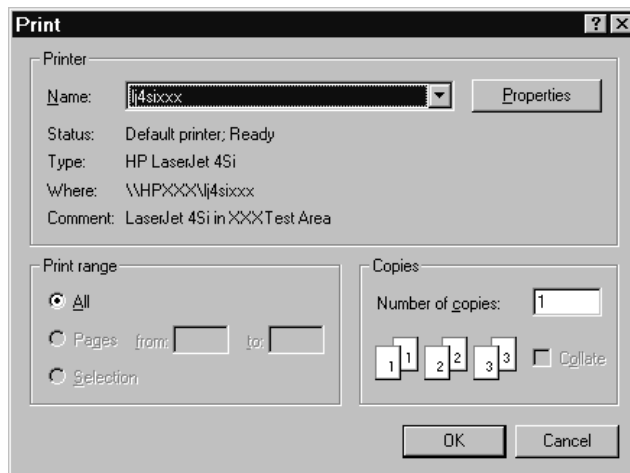


Abbildung 30 Die Anzeige drucken

Wenn Sie auf **OK** klicken, druckt VEE die Anzeige auf dem in dem Dialogfeld angegebenen Standarddrucker aus. Sie können einen anderen Drucker wählen, den Druckbereich ändern und die Anzahl der zu druckenden Exemplare festlegen. Klicken Sie für weitere Auswahloptionen auf die Schaltfläche **Properties** (Eigenschaften). Für verschiedene Druckertreiber können unterschiedliche Dialogfelder angezeigt werden. Weitere Informationen zur Verwendung von Windows-Dialogfeldern finden Sie in der *Microsoft Windows Hilfe*.

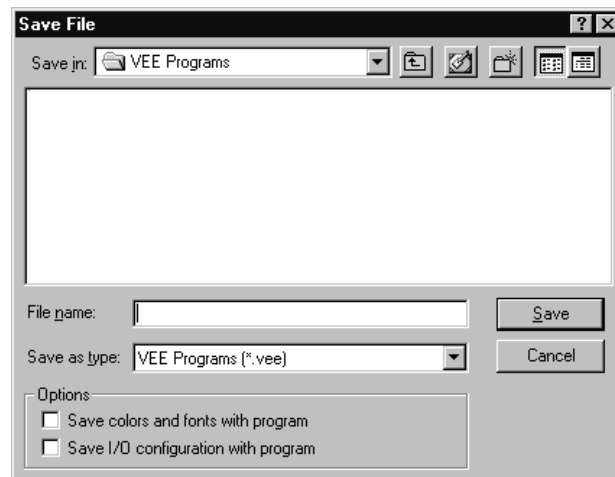
Klicken Sie in der Symbolleiste auf die Schaltfläche **Print Screen**, um die Anzeige direkt zu drucken.

## Ein Programm speichern

Sie können ein Programm jederzeit speichern. (Sie können den Inhalt des Arbeitsbereichs speichern, gleichgültig, ob es sich dabei um ein vollständiges Programm handelt oder nicht.)

**10** Wählen Sie zur Fortsetzung des Beispiels **File** ⇒ **Save As...** und füllen Sie die Felder in dem Dialogfeld aus.

Daraufhin wird das Dialogfeld **Save File** geöffnet. Abbildung 31 stellt dieses Dialogfeld dar.



**Abbildung 31** Das Dialogfeld Save File

## 1 Die VEE-Entwicklungsumgebung verwenden

**11** Standardmäßig speichert VEE für Windows Dateien im Ordner **VEE Programs**, der sich im Ordners Eigene Dateien befindet. Geben Sie zum Speichern des aktuellen Programms im Feld File name (Dateiname) den Namen `simple-program` ein und klicken Sie auf **Save** (Speichern). Wenn Sie die Erweiterung `.vee` nicht angeben, wird sie von VEE automatisch hinzugefügt.

### HINWEIS

Sie können die unter den Systemen Windows 98, Windows 2000, Windows NT 4.0 und Windows XP üblichen langen Dateinamen verwenden.

In den Feldern des Dialogfelds Save File können Sie die folgenden Änderungen vornehmen:

**Tabelle 7** Optionen für das Speichern von Dateien

Optionen für das Speichern von Dateien	Beschreibung
Save in (Speichern in)	Sie können das Verzeichnis oder das Laufwerk wechseln, indem Sie das entsprechende Dropdown-Menü öffnen. Klicken Sie doppelt auf einen Ordner, um ihn zu öffnen.
File name (Dateiname)	Geben Sie einen Dateinamen Ihrer Wahl ein.
Save as type (Dateityp)	VEE-Programme werden normalerweise mit der Erweiterung <code>.vee</code> gespeichert; Sie können den Dateityp jedoch auch ändern. Wenn Sie einen Dateinamen ohne Erweiterung eingeben, wird die Erweiterung <code>.vee</code> automatisch hinzugefügt.

**Tabelle 7** Optionen für das Speichern von Dateien

Optionen für das Speichern von Dateien	Beschreibung
Save colors and fonts with program (Farb- und Schriftinformationen mit dem Programm speichern)	<p>(Optional) Wenn Sie die Farben und Schriftarten in dem Programm über im Dialogfeld Default Preferences geändert haben und diese Farben und Schriftarten auch dann statt der Standardangaben verwendet werden sollen, wenn andere Benutzer das Programm laden, wählen Sie auf dieses Kontrollkästchen.</p> <p>Ist dieses Kontrollkästchen markiert, speichert VEE die vorgenommenen Änderungen an der Standardkonfiguration als Teil des Programms.</p>
Save I/O configuration with program (E/A-Konfiguration mit Programm speichern)	<p>(Optional) Wenn Sie ein Gerät im Instrument Manager konfiguriert haben und Ihre Konfiguration des Geräts auch anderen Benutzern des Programms beim Laden zur Verfügung stehen soll, sollten Sie dieses Kontrollkästchen markieren.</p> <p>Wenn dieses Kontrollkästchen markiert ist, speichert VEE die E/A-Konfiguration als Teil des Programms.</p>

**HINWEIS**

Zum erneuten Speichern des Programms mit dem gleichen Namen klicken Sie auf die Schaltfläche Save, oder drücken Sie zu einem beliebigen Zeitpunkt `Strg+S` (**File** ⇒ **Save**). Es empfiehlt sich, bei der Entwicklung eines Programms die Dateien häufig zu speichern. Wenn Sie ein geändertes Programm unter einem anderen Namen speichern wollen, drücken Sie `Strg+W`, oder wählen Sie **File** ⇒ **Save As**.

**Agilent VEE beenden**

**Wählen Sie File** ⇒ **Exit**, um das VEE-Anwendungsfenster zu schließen. Drücken Sie `Strg+E` zum Beenden von VEE, oder klicken Sie am rechten Rand der Titelleiste auf die Schaltfläche **x**.

## 1 Die VEE-Entwicklungsumgebung verwenden

Sie müssen eventuell die folgenden Techniken verwenden, wenn VEE nicht mehr auf die Eingabe über die Maus oder die Tastatur reagiert:

**Tabelle 8** VEE nach einem Programmabsturz beenden

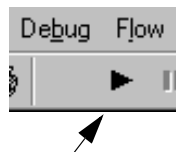
Betriebssystem	Methode
Unter Windows 98	Drücken Sie <b>Strg+Alt+Entf</b> . Daraufhin wird ein Fenster mit verschiedenen Optionen geöffnet. Gehen Sie in MS Windows anhand der Anleitungen in diesem Fenster vor, oder klicken Sie auf <b>Task beenden</b> .
Unter Windows NT4.0, Windows 2000 und Windows XP	Drücken Sie <b>Strg+Alt+Entf</b> und klicken Sie auf die Schaltfläche <b>Task Manager</b> . Wählen Sie in der Liste der Anwendungen <b>VEE</b> und klicken Sie auf <b>Task beenden</b> .

### VEE erneut starten und ein Programm ausführen

- 1 Wählen Sie im Windows **Startmenü** ⇒ **Programme** ⇒ **Agilent VEE Pro**.
- 2 Wählen Sie **File** ⇒ **Open** und füllen Sie das Dialogfeld **Open File** aus.

Dieses Dialogfeld ist dem Dialogfeld **Save File** sehr ähnlich. Das Standardverzeichnis für Benutzerprogramme ist das Verzeichnis **VEE\_USER**, sofern Sie bei der Installation kein anderes Verzeichnis angegeben haben. VEE öffnet das Programm im Hauptfenster.

- 3 Klicken Sie auf die Schaltfläche **Run**. Diese Schaltfläche sieht aus wie eine kleine Pfeilspitze; sie befindet sich in der Symbolleiste unterhalb des Menüs **Debug**, wie in **Abbildung 32** gezeigt.



Die Schaltfläche **Run**

**Abbildung 32** Die Schaltfläche **Run** in der Symbolleiste

**HINWEIS**

Der Befehl **vee.exe -r *Dateiname*** startet VEE und führt automatisch das mit *Dateiname* angegebene Programm aus. Sie können beispielsweise ein Symbol im Windows-Desktop erstellen und **Eigenschaften** ⇒ **Tastenkombination** für dieses Symbol auf die Ausführung eines bestimmten VEE-Programms einstellen. Ein Benutzer kann dann auf das Symbol im Desktop doppelklicken, um VEE zu starten und automatisch ein bestimmtes Programm auszuführen. Weitere Informationen finden Sie in der Windows-Hilfe in den Abschnitten über Befehle und Eingabeaufforderungen.

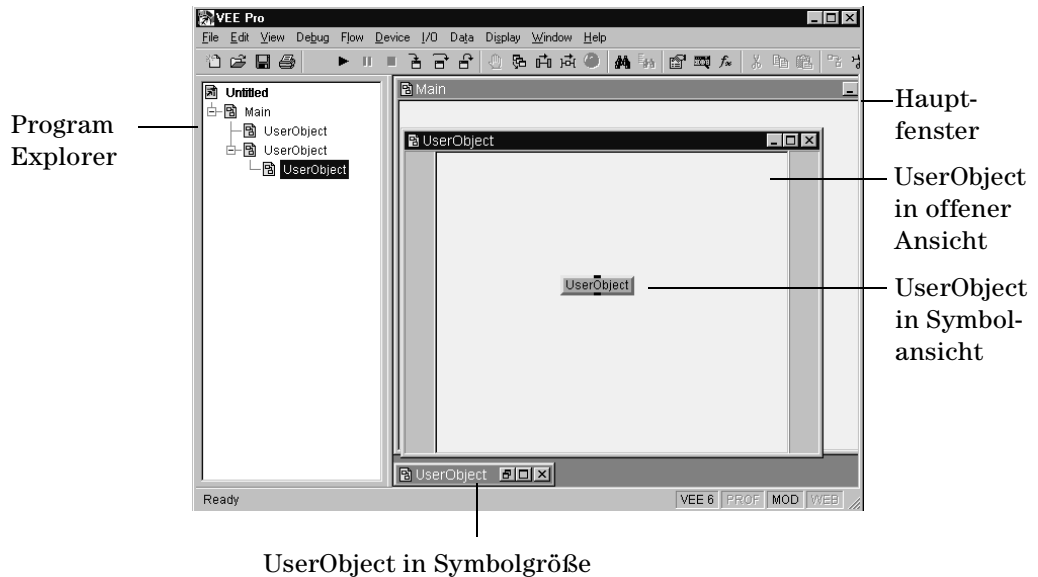
---

## Mehrere Fenster im Arbeitsbereich verwalten

Bis hierher beziehen sich die meisten Erläuterungen auf den Arbeitsbereich im Hauptfenster. Große VEE-Programme können jedoch mehrere Fenster innerhalb des Hauptfensters enthalten. Ein Programm kann beispielsweise von Ihnen definierte Objekte enthalten, etwa UserObjects und UserFunctions. (Sie können sich UserObjects und UserFunctions wie Unterprogramme des Hauptprogramms vorstellen. UserObjects und UserFunctions werden im Abschnitt „Übung 2-1: Erstellen eines UserObject“ auf Seite 80 in Kapitel, „Agilent VEE Programmier-techniken.“ ausführlich beschrieben). Die Begriffe werden hier nur erwähnt, um zu zeigen, wie VEE die Verwaltung von Programmen mit mehreren Fenstern ermöglicht.

Abbildung 33 zeigt ein Programm mit vier Fenstern. Jedes Fenster enthält ein Symbol (mit Menübefehlen), einen Titel und drei Schaltflächen: Minimieren, Maximieren und Schließen. Die Anzeige eines Fensters in maximaler Größe bewirkt, dass das Fenster den gesamten verfügbaren Platz im VEE-Arbeitsbereich belegt. Die Anzeige in minimaler Größe zeigt das Fenster als Symbol am unteren Rand des VEE-Arbeitsbereichs an. Durch das Schließen eines Fensters wird es aus dem Arbeitsbereich entfernt. Die Titelleiste des jeweils aktiven Fensters ist in VEE hervorgehoben.

## 1 Die VEE-Entwicklungsumgebung verwenden



**Abbildung 33** Mehrere Fenster im Arbeitsbereich

Abbildung 33 zeigt, wie der Program Explorer die Hierarchie des Programms auflistet. Diese integrierte modulare Struktur ermöglicht einen einfachen Zugriff auf alle Teile des Programms.

Wenn der Program Explorer nicht geöffnet ist, wählen Sie **View** ⇒ **Program Explorer**. Standardmäßig ist der Program Explorer geöffnet. Wenn Sie das Häkchen neben dem Menübefehl entfernen und unter **File** ⇒ **Default Preferences** auf die Schaltfläche Save klicken, bleibt der Program Explorer beim nächsten Starten von VEE geschlossen.

Sie können das Hauptfenster jederzeit in den Vordergrund bringen, indem Sie auf das Fenster klicken oder auf sein Symbol im Program Explorer doppelklicken.

### HINWEIS

Wenn Sie das Hauptfenster in VEE schließen, können Sie es über **View** ⇒ **Main** erneut öffnen.



## Die Arbeitsweise von Agilent VEE-Programmen

In VEE wird der allgemeine Ausführungsablauf in einem Programm als *Weitergabe* (propagation) bezeichnet. Die Weitergabe in einem Programm ist *nicht* durch die geografische Position der Objekte im Programm bestimmt, sondern durch die Verbindungen zwischen den Objekten. Die Weitergabe wird in erster Linie bestimmt durch den *Datenfluss*, der wiederum davon abhängt, wie die Datenein- und -ausgangs-Pins der Objekte miteinander verbunden sind.

### HINWEIS

In anderen Programmiersprachen wie C, BASIC oder Pascal wird die Reihenfolge, in der Programmanweisungen ausgeführt werden, von einer Reihe von Sequenz- und Auswahlregeln bestimmt. Im Allgemeinen werden die Anweisungen in der Reihenfolge ausgeführt, in der sie im Programm aufgeführt sind, es sei denn, bestimmte Anweisungen bewirken eine Verzweigung zu einer anderen Anweisung oder in einen Code-Thread.

---

Die Regeln des Datenflusses in einem VEE-Programm lauten:

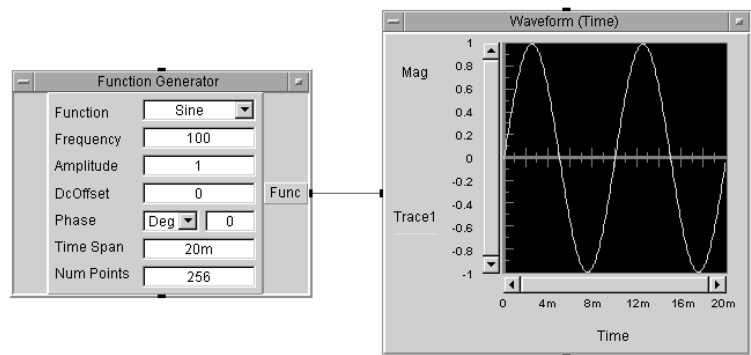
- *Daten fließen von links nach rechts durch ein Objekt.* Dies bedeutet, dass bei allen Objekten mit Daten-Pins die linken Daten-Pins dem Eingang entsprechen und die rechten Daten-Pins dem Ausgang.
- *Alle Dateneingangs-Pins eines Objekts müssen angeschlossen sein.* Anderenfalls tritt bei der Programmausführung ein Fehler auf.
- *Ein Objekt wird erst ausgeführt, wenn alle seine Dateneingangs-Pins neue Daten empfangen haben.*
- *Die Ausführung eines Objekts wird erst beendet, wenn alle angeschlossenen und relevanten Datenausgangs-Pins aktiviert wurden.*

## 1 Die VEE-Entwicklungsumgebung verwenden

In VEE können Sie die Reihenfolge der Ausführung mit den Sequenzein- und -ausgangs-Pins ändern. Sie brauchen die Sequenz-Pins normalerweise nicht zu verwenden; diese sind nur in Sonderfällen erforderlich. *Im Allgemeinen ist es das Beste, auf die Verwendung der Sequenz-Pins ganz zu verzichten. Wenn möglich, lassen Sie den Datenfluss die Ausführung des Programms steuern.*

### Übung 1-2: Datenfluss und Weitergabe anzeigen

Wenn Sie sich den Datenfluss veranschaulichen möchten, öffnen Sie das zuvor erstellte Programm. Öffnen Sie das Programm **simple-program.vee**, indem Sie in der Symbolleiste auf die Schaltfläche **Öffnen** klicken. (Das Programm **simple-program.vee** ist im Abschnitt „Übung 1-1 Das Programm Display Waveform (Wellenform anzeigen)“ auf Seite 56 beschrieben.) Führen Sie jetzt das Programm aus. Es sollte wie in Abbildung 34 dargestellt werden; allerdings können für die Parameter andere Werte eingetragen sein.

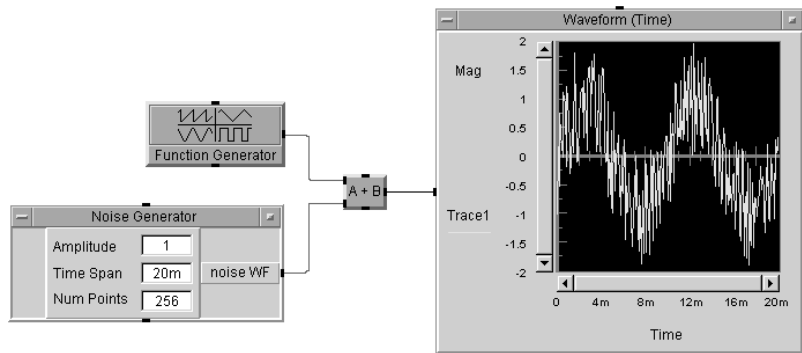


**Abbildung 34** Typische Anzeige von simple-program.vee

Der Datenausgangs-Pin des Objekts Function Generator ist mit dem Dateneingangs-Pin des Objekts Waveform (Time) verbunden. Bei der Ausführung des Programms wird das Objekt Waveform (Time) erst ausgeführt, wenn es Daten aus dem Objekt Function Generator empfängt. Dies ist ein einfaches Beispiel des Datenflusses.

## Übung 1-3: Einen Noise Generator hinzufügen

Erzeugen Sie eine „gestörte Sinuskurve“, indem Sie dem Programm **simple-program.vee** ein Objekt Noise Generator hinzufügen, wie in Abbildung 35 gezeigt.



**Abbildung 35** Beispiel: Ein Noise Generator-Objekt hinzufügen

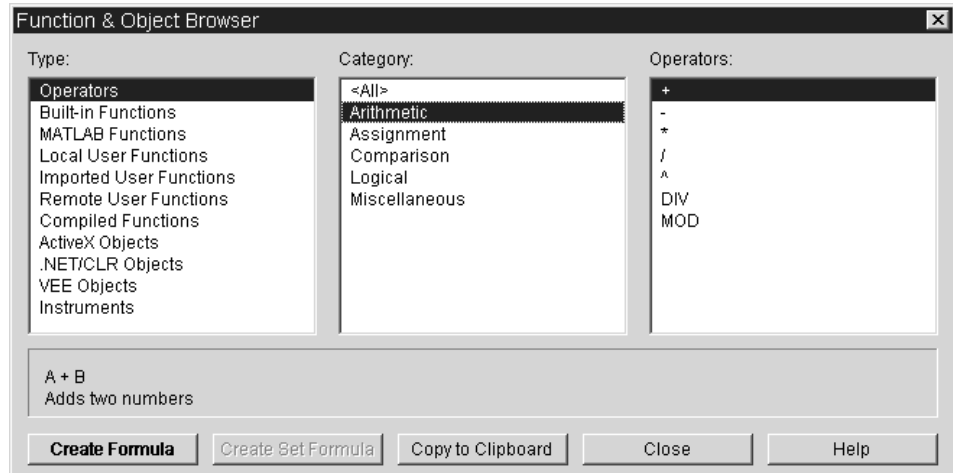
### HINWEIS

Die VEE-Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind im VEE-Paket enthalten und können über **Help** ⇒ **Open Example...** ⇒ **Manual** ⇒ **UsersGuide** geöffnet werden.

- 1 Löschen Sie die Linie, die die Objekte Function Generator und Waveform (Time) im ursprünglichen Programm verbindet. Klicken Sie in der Symbolleiste auf die Schaltfläche Delete Line und anschließend auf die Linie. Oder drücken Sie die Tastenkombination Umschalt+Strg, halten Sie diese Tasten gedrückt und klicken Sie auf die Linie.
- 2 Minimieren Sie die Darstellung des Function Generator.
- 3 Fügen Sie das Objekt Noise Generator hinzu (**Device** ⇒ **Virtual Source** ⇒ **Noise Generator**).
- 4 Fügen Sie das Objekt **A + B** mit **Device** ⇒ **Function & Object Browser** hinzu.

## 1 Die VEE-Entwicklungsumgebung verwenden

Der Function & Object Browser ist in Abbildung 36 dargestellt. Wählen Sie für Type die Option Operators. Für Category wählen Sie Arithmetic. Für Operators wählen Sie +) Klicken Sie auf **Create Formula** und platzieren Sie das Objekt im Arbeitsbereich zwischen den Objekten Function Generator und Waveform Time). Minimieren Sie das Objekt **A + B**.



**Abbildung 36** Function and Object Browser

- 1 Verbinden Sie die Ein- und Ausgangs-Pins wie in Abbildung 37 dargestellt.
- 2 Führen Sie das Programm aus.

Beachten Sie, dass das Objekt **A + B** erst ausgeführt wird, wenn die Objekte Function Generator und Noise Generator ausgeführt wurden. Es spielt jedoch keine Rolle, ob der Function Generator oder der Noise Generator zuerst ausgeführt wird; das Ergebnis ist das Gleiche.

Sobald beide Eingangs-Daten-Pins von **A + B** Daten empfangen, wird das Objekt **A + B** ausgeführt; dabei werden die beiden Signale summiert und das Ergebnis an das Objekt Waveform (Time) weitergeleitet.

**HINWEIS**

Der Datenfluss in einem VEE-Programm bestimmt dessen Ausführung.

Wählen Sie zur Anzeige der Ausführungsreihenfolge die Debug-Befehle Show Execution Flow (Ausführungsfluss anzeigen) und Show Data Flow (Datenfluss anzeigen), oder klicken Sie in der Symbolleiste auf die entsprechenden Schaltflächen. Führen Sie das Programm erneut aus. Jedes Objekt wird bei seiner Ausführung hervorgehoben, wobei zur Veranschaulichung des Datenflusses eine kleine quadratische Markierung entlang der Linien bewegt wird.

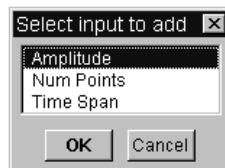
**HINWEIS**

Show Execution Flow und Show Data Flow können gleichzeitig oder einzeln gewählt sein; klicken Sie hierzu in der Symbolleiste auf die entsprechenden Schaltflächen oder wählen Sie im Debug-Menü die entsprechenden Befehle. Normalerweise sollten diese Befehle nicht gewählt sein, da sie die Ausführung des Programms deutlich verlangsamen.

## Übung 1-4: Einen Amplituden-Eingang und einen Real64 Slider hinzufügen

Erweitern Sie **simple-program.vee** um einen Amplituden-Eingang und einen Real64 Slider.

- 1 Klicken Sie auf das Objektmenü oder drücken Sie `Strg+A`, während sich der Mauszeiger im „Anschlussbereich“ auf der linken Seite des Noise Generator befindet. Ein Dialogfeld wird geöffnet, in dem Sie einen Eingang hinzufügen können (siehe Abbildung 37).



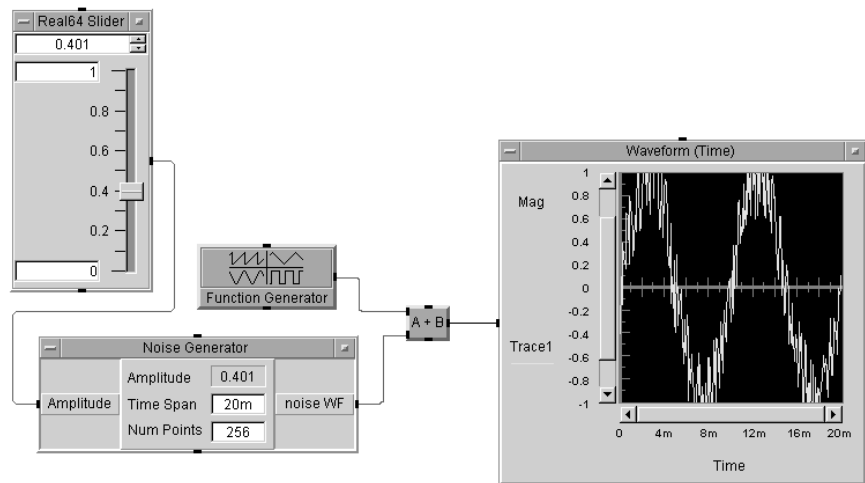
**Abbildung 37** Beispiel: Eingangsanschlüsse hinzufügen

## 1 Die VEE-Entwicklungsumgebung verwenden

- 2 Wählen Sie Amplitude, indem Sie auf **OK** klicken – ein Eingangsanschluss Amplitude wird angezeigt.

Da das Objekt Noise Generator jetzt einen Amplituden-Eingangs-Pin enthält, können Sie diese Daten als reale Zahl eingeben. VEE bietet ein Objekt, das diese Aufgabe erleichtert: den Real64 Slider (Real64-Schieberegler) im Menü Daten. (Sie können auch das Objekt Real64 Constant oder einen Real64 Knob verwenden.)

- 3 Fügen Sie ein Objekt Real64 Slider hinzu (**Data**  $\Rightarrow$  **Continuous**  $\Rightarrow$  **Real64 Slider**) und verbinden Sie seinen Datenausgangs-Pin mit dem Anschluss Amplitude, wie in Abbildung 38 gezeigt. Führen Sie das Programm aus.



**Abbildung 38** Beispiel: Ein Real64 Slider-Objekt hinzufügen

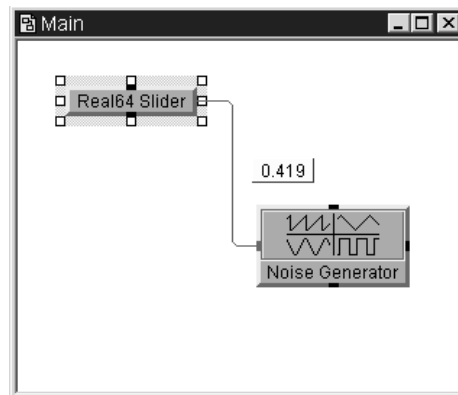
Versuchen Sie, die Amplitude der Störung zu ändern, indem Sie den Schieberegler auf dem Objekt Real64 Slider ziehen. Die Amplitude der Störung ändert sich nicht, bis Sie das Programm ausführen. Der Störfaktor der angezeigten Wellenform hängt vom Ausgabewert des Real64 Slider ab.

Auch hier bestimmt der Datenfluss die Reihenfolge der Ausführung. Der Noise Generator kann erst ausgeführt werden, wenn der Real64 Slider ausgeführt wird. Das Objekt **A + B** kann erst

ausgeführt werden, wenn sowohl der Function Generator als auch der Noise Generator ausgeführt werden; es spielt jedoch keine Rolle, welches dieser Objekte zuerst ausgeführt wird. Das Objekt Waveform (Time) wird erst ausgeführt, nachdem das Objekt **A + B** ausgeführt wurde.

## HINWEIS

Sie können den Wert eines Ausgangs anzeigen, indem Sie mit der Maus auf die Linie zeigen. Wenn Sie beispielsweise auf die Linie zwischen den Objekten Real64 Slider und Noise Generator zeigen, wird der Wert 0,401 angezeigt. Beachten Sie, dass der Wert der Linie (0,401) dem im Real64 Slider angezeigten Wert entspricht, wie in Abbildung 39 gezeigt. (Beachten Sie, dass die Objekte in der Symbolansicht angezeigt werden.)



**Abbildung 39** Den Werts an einen Ausgangs-Pin anzeigen

- 4 Speichern Sie das Programm erneut in **simple-program.vee**. Im nächsten Kapitel werden Sie dieses Programm um einige weitere Funktionen erweitern.

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie zum nächsten Kapitel übergehen.

- Die Online-Hilfedokumentation über die Haupt-Menüleiste und die Objektmenüs anzeigen.
- VEE starten.
- Die Haupt-Menüleiste, die Schaltflächen in der Symbolleiste, den Arbeitsbereich und die Statusleiste erkennen.
- Den Program Explorer und seine Funktion erläutern.
- Menübefehle im Hauptmenü und in den Objektmenüs wählen.
- Folgende Operationen mit einem Objekt ausführen: Verschieben, Umbenennen, Minimieren, Erweitern, Größe ändern, Wählen, Auswahl aufheben, Löschen, Klonen etc.
- Den Arbeitsbereich verschieben, löschen und mehrere Fenster verwalten.
- Daten- und Sequenz-Pins an einem Objekt erkennen und ihre Aufgaben erläutern.
- Terminals untersuchen und ihre Namen ändern.
- Objekte zum Erstellen eines Programms für die Simulation von Wellenform-Daten verbinden.
- Ein Programm erstellen, ausführen, drucken und speichern.
- VEE beenden und ein Programm erneut öffnen.
- Erläutern, wie Daten durch ein VEE-Programm fließen.



## 2 Agilent VEE Programmier Techniken

Übersicht	79
Allgemeine Techniken	80
Die Online-Hilfe	102
Programme in Agilent VEE debuggen	105
Übungen mit Programmen	121
Agilent VEE-Programme dokumentieren	125
Kapitel-Checkliste	130

# Agilent VEE Programmier Techniken

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Ein UserObject erstellen
- Ein Dialogfeld für die Benutzereingabe hinzufügen
- Datendateien verwenden
- Fensteransichten erstellen (Benutzerschnittstelle)
- Mathematisches Verarbeiten von Daten
- Kommunikation mit Instrumenten
- Ein Programm dokumentieren
- Die Tools zur Fehlerbehebung („Debugging“)

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*

## Übersicht

In diesem Kapitel erlernen Sie ausgewählte VEE-Programmier-techniken zum Erstellen eigener Programme. Mit VEE können Sie beispielsweise angepasste Objekte, die so genannten User-Objects, erstellen. Sie können auch Schnittstellen für die Benutzer (Operator) erstellen, in denen nur die erforderlichen Teile des Programms angezeigt werden. Diese Teile werden in der Fensteransicht des Programms angezeigt.

Sie können Daten von VEE aus in eine Datei schreiben und Daten aus einer Datei in VEE einlesen. Datendateien und die ihnen zugeordneten E/A-Transaktionen können für viele Bereiche angewendet werden einschließlich der Kommunikation mit Instrumenten, Dateien, Zeichenfolgen, dem Betriebssystem, den Schnittstellen, anderen Programmen, und Druckern.

VEE unterstützt eine Vielzahl von Datentypen und bietet umfangreiche Funktionen zur mathematischen Verarbeitung von Daten. Sie haben verschiedene Möglichkeiten, VEE zur Kommunikation mit Instrumenten zu verwenden. VEE bietet darüber hinaus leistungsstarke Debugging-Tools zum Suchen und Beheben von Problemen in Programmen.

### Allgemeine Techniken

Im VEE-Hauptprogramm können Sie logische Gruppen von Objekten, so genannte UserObjects, erstellen. Ein UserObject (Benutzerobjekt) wird erstellt, indem Sie eine logische Gruppe in einem UserObject-Fenster platzieren. In diesem UserObject-Fenster verbinden Sie die Ein- und Ausgänge auf die gleiche Weise wie im Hauptprogramm. Das UserObject selbst ist mit anderen Objekten im Hauptprogramm mit Ein- und Ausgängen verbunden wie jedes andere Objekt auch.

Ein UserObject dient dazu, einen eindeutigen Kontext zur Ausführung einer bestimmten Aufgabe innerhalb des Hauptprogramms zu erstellen. Auf diese Weise sparen Sie nicht nur Platz im Hauptarbeitsbereich, sondern strukturieren das Programm und gestalten es dadurch übersichtlicher.

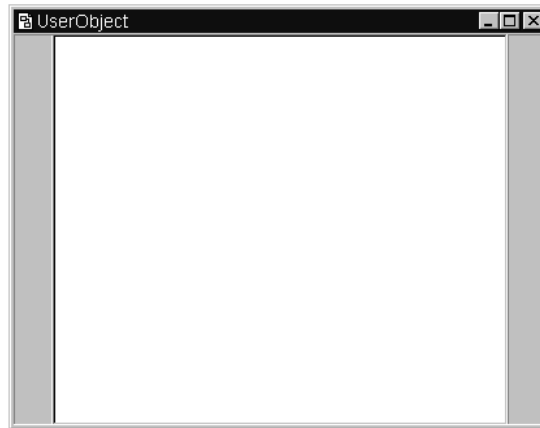
Ein VEE-Programm kann viele im Hauptprogramm verschachtelte UserObjects enthalten. Jedes UserObject enthält eine Symbolansicht im Hauptfenster. Wenn Sie die Symbolansicht der UserObjects im Hauptprogramm den entsprechenden UserObject-Fenstern zuordnen wollen, geben Sie den UserObjects in ihren Bearbeitungsfenstern einen Namen; dadurch werden sie auch in der entsprechenden Symbolansicht benannt. Wenn Sie beispielsweise ein UserObject AddNoise nennen, haben sowohl das Symbolfenster im Hauptprogramm als auch die Titelleiste im UserObject den Inhalt AddNoise. In der folgenden Übung lernen Sie, wie Sie ein UserObject erstellen.

#### Übung 2-1: Erstellen eines UserObject

Es gibt verschiedene Möglichkeiten, ein UserObject in einem VEE-Programm zu erstellen:

- Wählen Sie in der Menüleiste **Device** ⇒ **UserObject**, um ein leeres UserObject-Symbol im Hauptfenster (Main) anzuzeigen, und fügen Sie diesem UserObject die gewünschten Objekte hinzu. Durch Doppelklicken auf das Symbol UserObject wird es in der offenen Ansicht angezeigt (siehe Abbildung 40).

- Wählen Sie Objekte in einem Programm und erstellen Sie anschließend aus diesen Objekten ein **UserObject**, indem Sie die Objekte markieren und **Edit** ⇒ **Create UserObject** wählen.



**Abbildung 40** Das Fenster UserObject

Nachdem Sie ein UserObject erstellt haben, ist es Teil des Hauptprogramms. Das UserObject-Fenster kann als Symbol angezeigt werden, in der offenen Ansicht oder in Symbolgröße am unteren Rand der Anzeige:

- Schließen Sie das Fenster durch Klicken auf die Schaltfläche „Schließen“, um das UserObject als Symbol im Hauptfenster anzuzeigen.
- Zur Anzeige des Fenster mit maximaler Größe klicken Sie auf die Schaltfläche „Maximieren“; das UserObject-Fenster belegt jetzt den gesamten verfügbaren Bereich im VEE-Arbeitsbereich.
- Stellen Sie das Fenster in Symbolgröße dar, indem Sie auf die Schaltfläche „Minimieren“ klicken. Das UserObject wird jetzt in Symbolgröße am unteren Rand des VEE-Arbeitsbereichs angezeigt.

### HINWEIS

Die Symbolansicht des UserObject befindet sich immer im Hauptfenster; in dieser Ansicht können Sie die Pins des Objekts mit anderen Objekten im Hauptfenster verbinden.

---

### HINWEIS

Stellen Sie zunächst sicher, dass im Menü View Program Explorer deaktiviert ist; Sie haben so mehr Platz im Hauptfenster.

---

Als nächstes erstellen Sie ein UserObject für ein Programm.

- 1 Öffnen Sie das Programm (**simple-program.vee**), das Sie in „Übung 1-4: Einen Amplituden-Eingang und einen Real64 Slider hinzufügen“ auf Seite 73 erstellt haben. Das Programm sollte im Hauptarbeitsbereich angezeigt werden.
- 2 Entfernen Sie das Objekt Real64 Slider aus dem Programm. (Es wird in dieser Übung nicht verwendet.) Wählen Sie dazu im Objektmenü des Objekts **Real64 Slider** den Befehl Delete, oder doppelklicken Sie auf die Schaltfläche „Objektmenü“.

### HINWEIS

Das Objekt Real64 Slider ist nun entfernt, der Eingangs-Pin ist noch immer an dem Objekt Noise Generator angeschlossen. Wenn Sie das Programm jetzt erneut ausführen, wird die VEE-Fehlermeldung eingeblendet, dass der Eingangs-Pin Amplitude am Noise Generator nicht angeschlossen ist. *Denken Sie daran, dass alle Eingangs-Pins angeschlossen sein müssen, damit ein VEE-Programm ausgeführt werden kann.*

---

- 3 Klicken Sie im Objekt Noise Generator auf die Schaltfläche Objektmenü oder mit der rechten Maustaste in das Objekt, um das Objektmenü zu öffnen. Wählen Sie **Delete Terminal** ⇒ **Input** und klicken Sie im Dialogfeld unter **Choose an input to delete** auf die Schaltfläche **OK**, während **Amplitude** hervorgehoben ist.
- 4 Benennen Sie das Programm um, indem Sie **File** ⇒ **Save As...** wählen und den neuen Namen `usrobj-program1.vee` eingeben.

- 5 Zeigen Sie anschließend das Objekt Noise Generator in Symbolgröße an, und ordnen Sie die Objekte wie in Abbildung 41 gezeigt neu an.

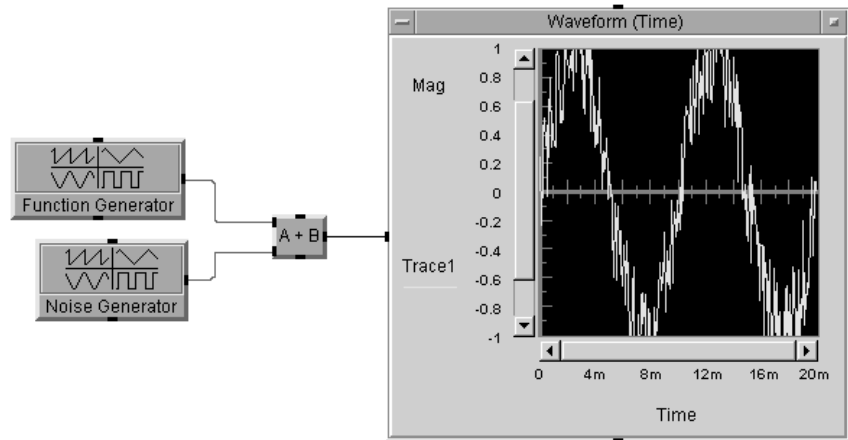


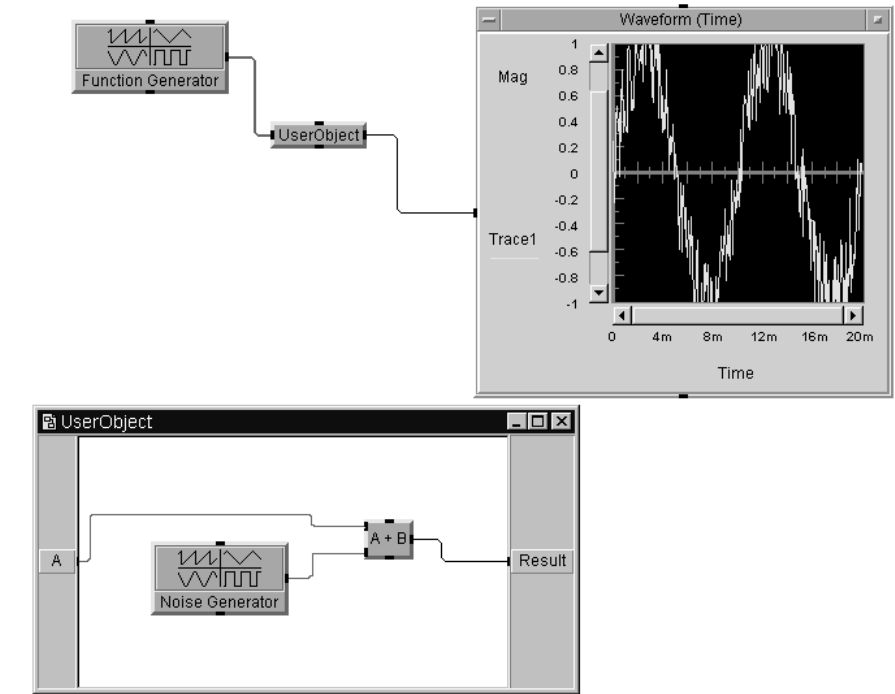
Abbildung 41 usrobj-program.vee ini einer frühen Phase

- 6 Wählen Sie die Objekte **Noise Generator** und **A + B** mit dem Direktauftrag `Strg + linke Maustaste`. Klicken Sie auf **Edit** ⇒ **Create UserObject**. Es wird ein Dialogfeld mit der Bezeichnung **Create UserObject** eingeblendet. (Sie können das Objekt umbenennen, indem Sie einen neuen Namen eingeben. Klicken Sie einstweilen auf **OK**, um das UserObject zu erstellen.)

Das UserObject enthält die Objekte **Noise Generator** und **A + B** im UserObject-Bearbeitungsfenster und wird automatisch im Hauptfenster mit den entsprechenden Ein- und Ausgangspins erstellt, wie in Abbildung 42 dargestellt.

Sie können die Symbole bequem in die linke obere Ecke des UserObject positionieren, wenn Sie auf der Tastatur die Taste `Pos1` drücken.

## 2 Agilent VEE Programmieretechniken



**Abbildung 42** Ein UserObject erstellen

### HINWEIS

Die Neuordnung der Objekte vor der Ausführung von Create UserObject dient lediglich der besseren Übersichtlichkeit. Wenn Sie die in einen Bereich einzubeziehenden Objekte nicht erfassen, legt das UserObject seine Größe selbst fest, so dass es alle ausgewählten Objekte enthalten kann. Sie können den Arbeitsbereich des UserObject anschließend neu anordnen und seine Größe ändern sowie das UserObject an eine geeignete Stelle im Arbeitsbereich verschieben. Das Bereinigen ist jedoch einfacher, wenn Sie die Objekte bereits vorher logisch anordnen.



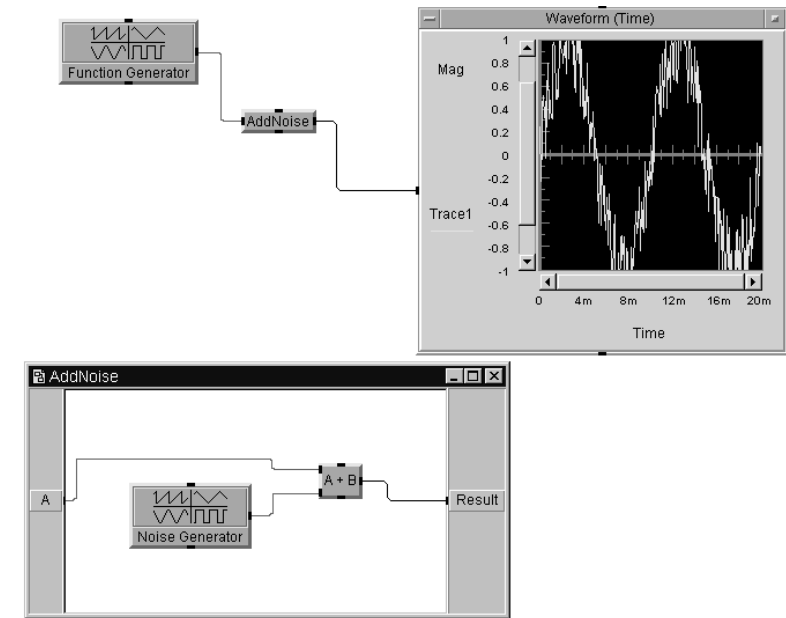
**HINWEIS**

Sie können mit **Edit**  $\Rightarrow$  **Clean Up Lines** die Linienführung in einem Programm bereinigen. Dieser Befehl ist kontextabhängig. Zum Bereinigen der Linien für das UserObject muss dieses Fenster aktiv sein. Klicken Sie auf das Objekt **UserObject** an und wählen Sie **Edit**  $\Rightarrow$  **Clean Up Lines**.

Durch das Erstellen eines UserObject in seinem Bearbeitungsfenster und die anschließende Verwendung der Symbolansicht des UserObject können Sie auf dem Bildschirm Platz sparen.

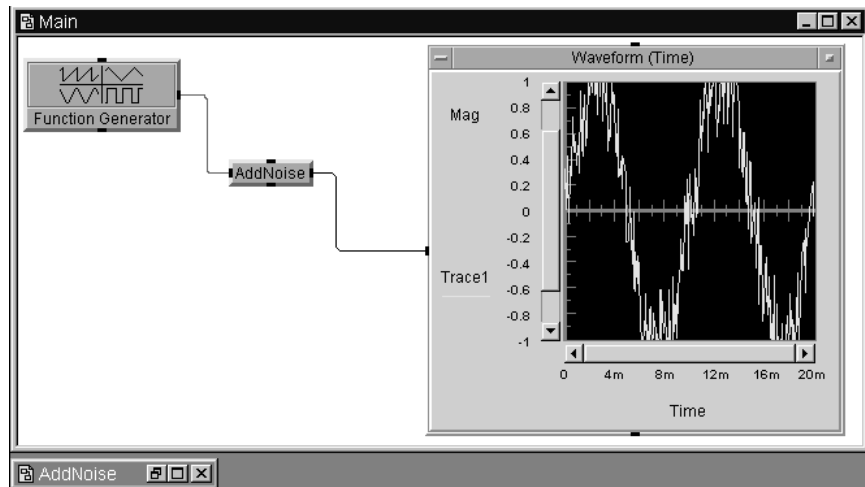
- 7 Ändern Sie den Titel des UserObject von UserObject in AddNoise, um so eine Verwechslung auszuschließen. Doppelklicken Sie auf die Titelleiste und geben Sie den neuen Titel im Dialogfeld ‚Properties‘ ein. Abbildung 43 zeigt, wie das Programm dadurch übersichtlicher wird.

Sie können das Dialogfeld ‚Properties‘ eines Objekts rasch aufrufen, indem Sie auf seine Titelleiste doppelklicken.



**Abbildung 43** UserObject, umbenannt in AddNoise

- 8 Klicken Sie auf die Schaltfläche **Ausführen**, um wie in Abbildung 44 dargestellt die gestörte Kosinuskurve anzuzeigen. Beachten Sie, dass AddNoise in Symbolgröße am unteren Rand des Arbeitsbereichs angezeigt wird. Zum Anzeigen von AddNoise in Symbolgröße klicken Sie in der Titelleiste auf die Schaltfläche Minimieren; diese Schaltfläche wird als Unterstreichungszeichen (   ) dargestellt.



**Abbildung 44** Gestörte Kosinuskurve

UserObjects sind dann effektiv, wenn sie innerhalb des Programms einen logischen Zweck erfüllen. Dieses eindeutige Objekt spart nicht nur Platz, sondern ermöglicht auch ein effizientes Strukturieren eines Programms. UserObjects erleichtern die Verwendung eines „Top-Down“-Design in VEE-Programmen. VEE enthält zudem ein Objekt *UserFunction*, das ein mehrfach verwendbares Code-Modul ist. Weitere Informationen zu UserObjects und UserFunctions finden Sie in Kapitel „Agilent VEE Programmierstechniken“ auf Seite 77.

Weitere Informationen zu UserObjects können Sie durch Auswählen von **Help** ⇒ **Contents and Index** in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch **How Do I...**, **Tell Me About...** oder **Reference**.

Dieses Beispiel wird im folgenden Abschnitt fortgesetzt. Wenn Sie diese Übung jedoch hier beenden möchten, speichern Sie das Programm unter dem Namen **usrobj-program3.vee**.

## Übung 2-2: Ein Dialogfeld für die Benutzereingabe erstellen

Öffnen Sie das Programm **usrobj-program3.vee**, falls dies nicht bereits geschehen ist.

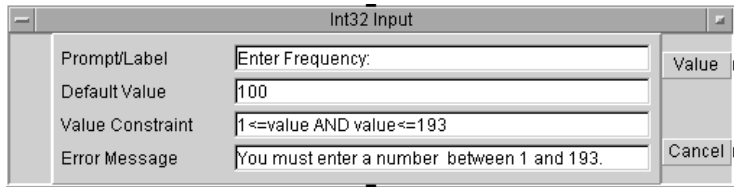
Im Untermenü **Data** ⇒ **Dialog Box** können Sie unter sechs Typen von Dialogfeldern wählen: Text Input (Texteingabe), Int32 Input (Int32-Eingabe) und Real64 Input (Real64 -Eingabe) sowie Message Box (Meldungsfeld), List Box (Listenfeld) und File Name Selection (Dateinamen- Auswahl). Bei der Eingabe von Text, Ganzzahlen (Integer) und realen Zahlen erleichtert ein entsprechendes Dialogfeld die Konfiguration der Eingabeaufforderung bzw. Beschriftung, der Standardwerte, der Wertebegrenzungen und Fehlermeldungen. Wenn eines dieser Dialogfelder hinzugefügt wurde, wird bei der Ausführung des Programms das entsprechende Eingabefenster eingeblendet.

- 1 Wählen Sie **Data** ⇒ **Dialog Box** ⇒ **Int32 Input** aus und platzieren Sie das Feld links von dem Objekt **Function Generator**. Ändern Sie den Eintrag im Feld Prompt/Label in Enter Frequency. (Sie müssen den vorhandenen Text erst markieren, indem Sie in das Textfeld klicken und den Mauszeiger über den Text ziehen.) Ändern Sie den Default Value (Standardwert) in 100.

Sie können einen Eintrag auch durch Doppelklicken in das Textfeld markieren.

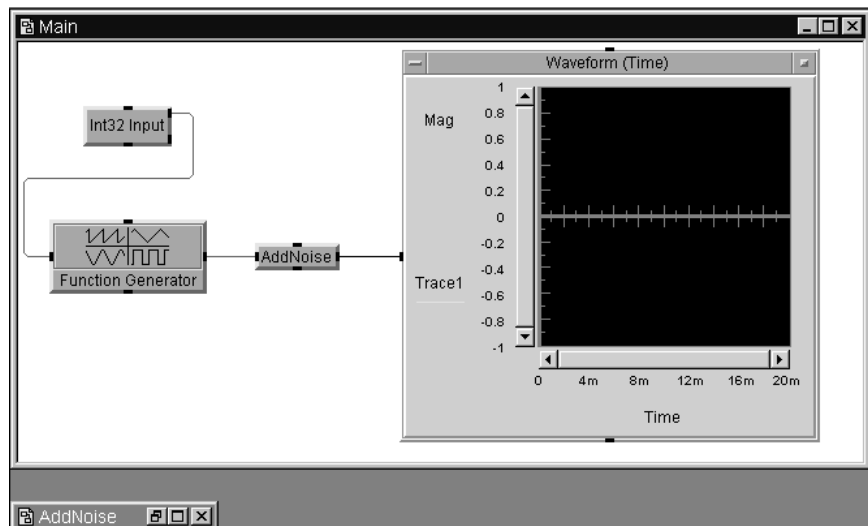
- 2 Ändern Sie die Angaben unter **Value Constraints** (Wertebegrenzungen) in **1** als unteren Wert und **193** als oberen Wert. Ändern Sie die Fehlermeldung entsprechend diesen neuen Werten, wie in Abbildung 45 dargestellt. Stellen Sie jetzt das Objekt **Int32 Input** als Symbol dar.

## 2 Agilent VEE Programmieretechniken



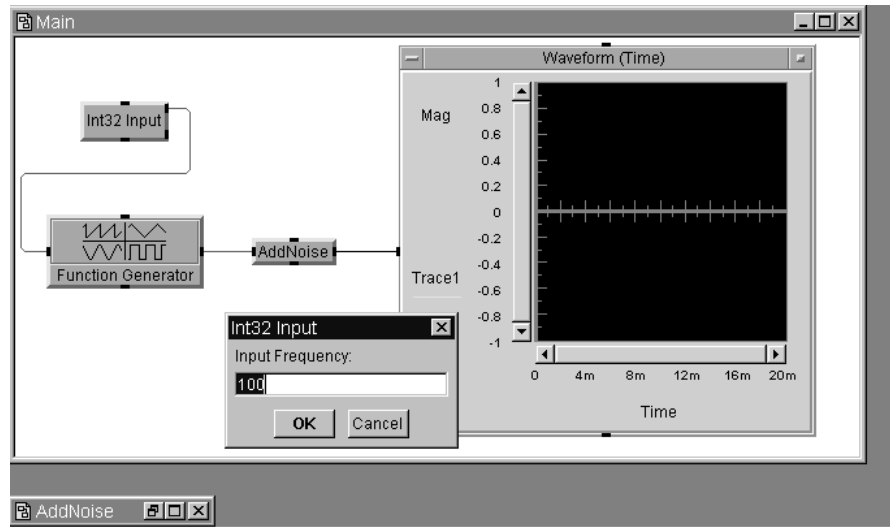
**Abbildung 45** Das Konfigurationsfeld „Int32 Input“

- 3 Öffnen Sie das Objektmenü des Objekts **Function Generator** und wählen Sie den Befehl **Add Terminal** ⇒ **Data Input**. Wählen Sie im Dialogfeld für **Select input to add** die Option **Frequency** und klicken Sie auf **OK**.
- 4 Verbinden Sie den oberen Ausgangs-Pin des Objekts Int32 Input mit dem Eingangs-Pin des Function Generator. Beachten Sie, dass Frequency jetzt nur über den Eingangs-Pin geändert werden kann; Sie können das Eingabefeld Frequency nicht mehr ändern. Das Programm sollte jetzt aussehen wie in Abbildung 46.



**Abbildung 46** Int32-Eingabe zu usrobj-program.vee hinzugefügt

- 5 Führen Sie das Programm aus. Das Eingabefenster für Int32 Input wird angezeigt; unter Input Frequency kann ein Wert eingegeben werden. Führen Sie das Programm versuchsweise mit verschiedenen Frequenzen im Eingabefenster aus. Abbildung 47 zeigt das während der Ausführung des Programms eingeblendete Eingabefenster. Sie können auf das Eingabefenster klicken und es ziehen, um seine Position auf dem Bildschirm zu ändern.



**Abbildung 47** Eingeblendetes Eingabefenster bei Programmausführung

Wenn Sie eine Frequenz über 193 eingeben, wird eine Fehlermeldung eingeblendet. Diese Meldung enthält genau den von Ihnen eingegebenen Text.

Dieses Beispiel wird im folgenden Abschnitt fortgesetzt. Wenn Sie die Übung jedoch hier beenden möchten, speichern Sie das Programm unter dem Namen **usrobj1-program4.vee**.

## HINWEIS

Die VEE-Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter **Help** ⇒ **Open Example...** ⇒ **Manual** ⇒ **UsersGuide** enthalten.

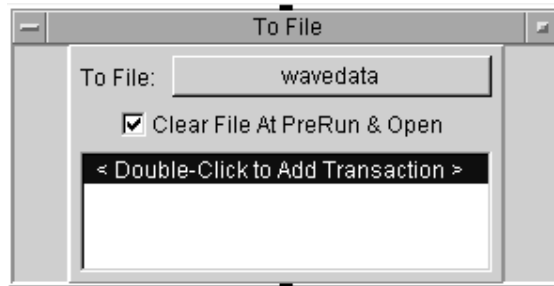
### Übung 2-3: Datendateien verwenden

Sie können Daten aus VEE in eine Datendatei schreiben und Daten aus einer Datei in VEE einlesen. Hierzu verwenden Sie im Programm die Objekte To File und From File. Fügen Sie beispielsweise ein Objekt To File der Detailansicht des von Ihnen erstellten Programms hinzu.

Öffnen Sie das Programm **usrobj-program4.vee**, falls dies nicht bereits geschehen ist.

- 1 Wählen Sie **I/O** ⇒ **To** ⇒ **File** aus und platzieren Sie dieses Objekt im Hauptarbeitsbereich.
- 2 Ändern Sie den standardmäßigen Dateinamen **myFile** in **wavedata**.

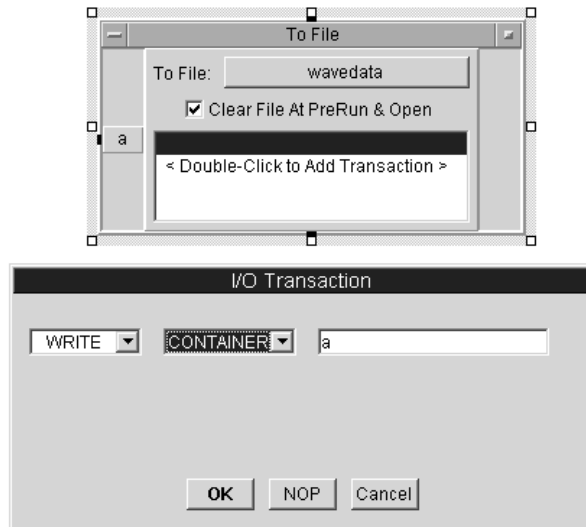
Das Kontrollkästchen **Clear File At PreRun & Open** muss aktiviert sein. To File hängt nämlich standardmäßig Daten an die vorhandene Datei an. In diesem Fall wollen Sie jedoch die Datei bei jeder Ausführung des Programms löschen. Das Objekt To File sollte jetzt aussehen wie in Abbildung 48.



**Abbildung 48** Eine Datendatei hinzufügen

- 3 Doppelklicken Sie auf den Bereich mit der Beschriftung **Double-Click to Add Transaction**. Das in Abbildung 49 dargestellte Dialogfeld wird angezeigt. Klicken Sie in das Feld **TEXT** (oder auf den Pfeil), um die Dropdown-Liste der Datentypen anzuzeigen, und wählen Sie **CONTAINER**. Klicken Sie auf **OK**. Wenn Sie im Dialogfeld **I/O Transaction** auf **OK** klicken, wird dem Objekt To File automatisch ein Eingangs-Pin **a** hinzugefügt.

Die weiteren Optionen für Transaktionen neben **WRITE CONTAINER** sind in der Hilfe erläutert, auf die Sie über das To File-Objektmenü mit dem Befehl Help zugreifen. Transaktionen werden ausführlich beschrieben im Anhang des Handbuchs *VEE Pro Advanced Techniques* und in Kapitel, „Testergebnisse speichern und abrufen“



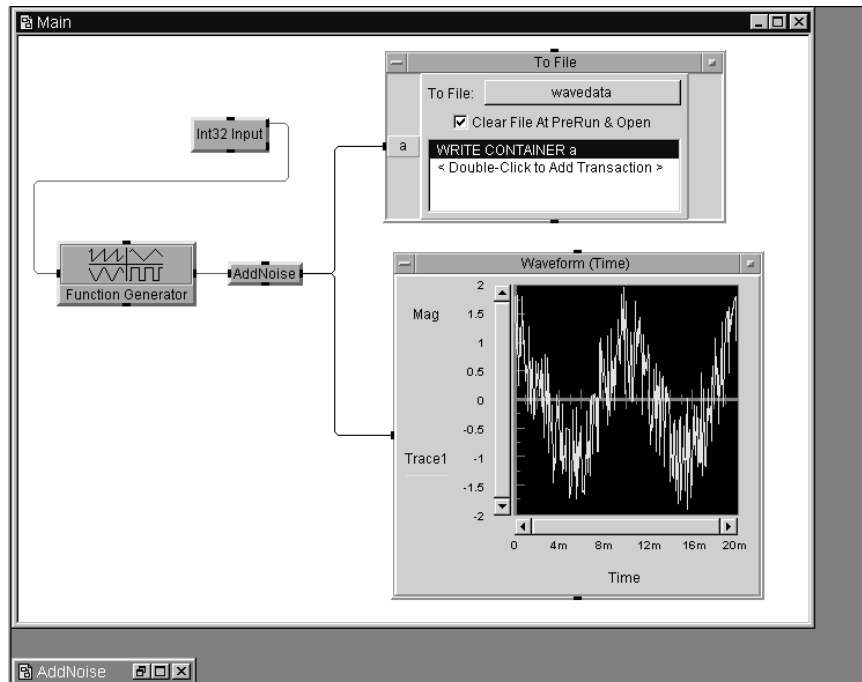
**Abbildung 49** Eine E/A-Transaktion wählen

- 4 Verbinden Sie den Datenausgangs-Pin von AddNoise mit dem Dateneingangs-Pin von To File. Das Programm sollte jetzt aussehen wie in Abbildung 50.

### HINWEIS

Sie können einen Datenausgangs-Pin mit mehreren Dateneingangs-Pins verbinden.

## 2 Agilent VEE Programmiertechniken



**Abbildung 50** Ein To File-Objekt hinzufügen

- 5 Klicken Sie in der Symbolleiste erneut auf die Schaltfläche **Run**, um das Programm zu testen. Das Programm zeigt jetzt die durch das UserObject AddNoise gestörte Kosinuskurve an und schreibt einen Container mit Wellenformdaten in die Datei wavedata.

Doppelklicken Sie auf das Objekt To File, um es für die Bearbeitung zu aktivieren; mit einem Doppelklick auf den Eingangsanschluss **a** können Sie anschließend dessen Eigenschaften anzeigen. Es wird ein Array mit 256 Punkten angezeigt.

Fügen Sie dem Programm ein Objekt From File hinzu, um die Daten wieder einzulesen.

- 6 Wählen Sie **I/O** ⇒ **From** ⇒ **File** aus und platzieren Sie dieses Objekt im Hauptarbeitsbereich. Fügen Sie dem **READ CONTAINER x** eine Lese-Transaktion hinzu und ändern Sie den Datei-



namen in wavedata (das Vorgehen ist dasselbe wie für **To File**). Löschen Sie anschließend die Linie zwischen AddNoise und dem Objekt Waveform (Time), und verbinden Sie die Objekte wie in Abbildung 51 dargestellt. Die Sequenzlinie zwischen To File und From File sorgt dafür, dass die Daten in die Datei geschrieben werden, bevor sie gelesen werden.

- 7 Führen Sie das Programm aus. Es sollte jetzt aussehen wie in Abbildung 51. Speichern Sie das Programm als **usrobj-program.vee**.

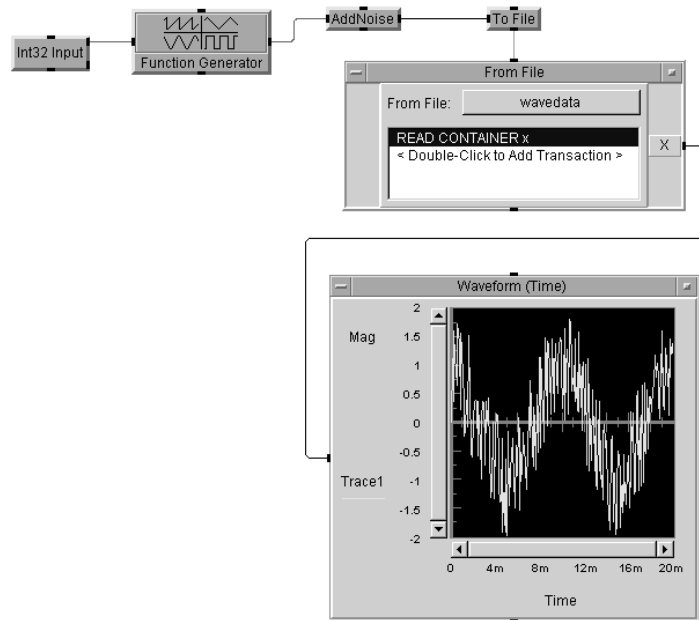


Abbildung 51 Ein From File-Objekt hinzufügen

### Übung 2-4: Eine Fensteransicht (Benutzerschnittstelle) erstellen

Nach dem Entwickeln eines Programms können Sie eine Benutzerschnittstelle erstellen. Hierzu erstellen Sie eine Fensteransicht des Programms. In dieser Übung wird das Programm verwendet, das Sie „Übung 1-2: Datenfluss und Weitergabe anzeigen“ auf Seite 70 erstellt haben.

- 1 Öffnen Sie das Programm **simple-program.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 52.

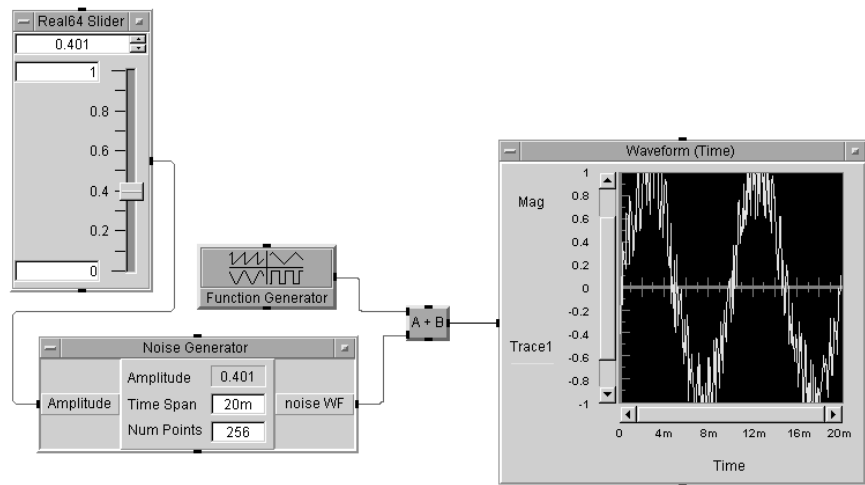
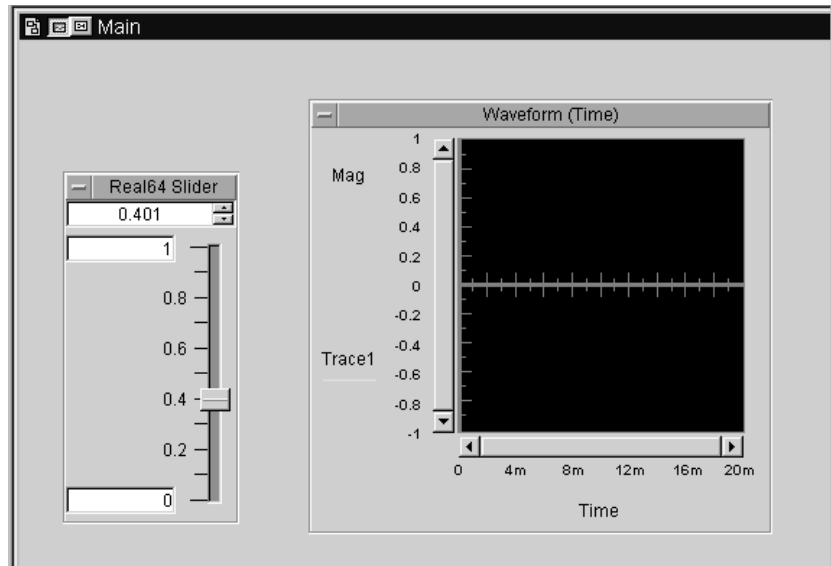


Abbildung 52 simple-program.vee

- 2 Wählen Sie die Objekte, die in der Fensteransicht angezeigt werden sollen. Die Fensteransicht dient als Benutzerschnittstelle. Halten Sie die Taste **Strg** gedrückt, während Sie nacheinander auf die gewünschten Objekte klicken. (Vergewissern Sie sich, dass kein Objekt versehentlich gewählt wird.) Wählen Sie in unserem Fall die Objekte **Real64 Slider** und **Waveform (Time)**. Diese beiden Objekte haben nun jeweils einen Schatten zur Kennzeichnung, dass sie markiert wurden.
- 3 Klicken Sie in der Symbolleiste auf die Schaltfläche **Add to Panel**, um die ausgewählten Objekte dem Fenster hinzuzufügen (oder wählen Sie **Edit** ⇒ **Add To Panel**). Es wird eine Fensteransicht mit den beiden hinzugefügten Objekten angezeigt.

Sie können die Objekte in der Fensteransicht an die gewünschte Position verschieben und ihre Größe ändern, um eine Anzeige wie die in Abbildung 53 zu erhalten.



**Abbildung 53** Beispiel: Eine Fensteransicht erstellen

- 4 Klicken Sie links oben in der Titelleiste des Hauptfensters auf die Schaltfläche To Detail, um zur Detailansicht zu wechseln. Klicken Sie auf die Schaltfläche To Panel, um zur Fensteransicht zurückzukehren.

Die Detailansicht entspricht der normalen Anzeige, in der Sie ein Programm bearbeiten. Sie können Objekte in der Fensteransicht unabhängig von der Detailansicht verschieben, löschen oder ihre Größe ändern. Die Detailansicht wird zum Entwickeln eines Programms verwendet; die Fensteransicht dagegen für die Gestaltung der Benutzerschnittstelle.

- 5 Speichern Sie das Programm als **simple-program\_with\_panel.vee**.

Machen Sie sich mit der Fensteransicht vertraut, indem Sie die folgenden Änderungen vornehmen:

- Wählen Sie in der Fensteransicht im Objektmenü des Hauptfensters den Eintrag Properties, um die Farben des Fensters zu ändern. Klicken Sie unter Appearance auf **BackColor** und wählen Sie die gewünschte Farbe aus der Dropdown-Liste.
- Sie können die Farben oder Schriftarten aller Objekte in der Anzeige Properties ändern, die Sie durch Doppelklicken in die Titelleiste öffnen. Klicken Sie dazu in der linken Spalte auf den gewünschten Typ der Farbe oder Schriftart, und wählen Sie in der rechten Spalte einen Wert.
- Genauso können Sie in der Ansicht Properties für die Eigenschaft Border den Wert Raised wählen, wenn die Objekte in der Fensteransicht leicht abgehoben dargestellt werden sollen.
- Zum Ändern des Namens der Fensteransicht öffnen Sie die Ansicht Properties, und ändern Sie die Eigenschaft Title wünschgemäß. Der eingegebene Name wird bei der Ausführung des Programms angezeigt.

### Übung 2-5: Mathematische Verarbeitung von Daten

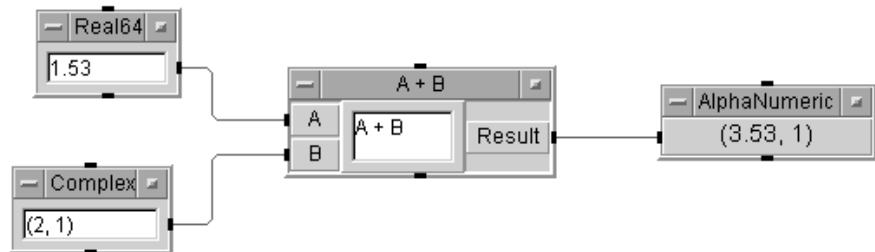
VEE bietet umfangreiche integrierte mathematische Funktionen und unterstützt eine Vielzahl mathematischer Datentypen wie auch die Daten- und Signalverarbeitung von MATLAB. Ausführliche Hinweise hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

#### Verwenden von Datentypen

VEE unterstützt verschiedene Datentypen einschließlich Text, Integer (Ganzzahlen) und Real (reale Zahlen) sowie verschiedene Arten von komplexen und Koordinatenzahlen. Sie haben bereits in früheren Beispielen gesehen, wie das Objekt **A + B** zwei Wellenformen addieren kann. Mathematische Operatoren wie Addition (+) können mit verschiedenen Datentypen und sogar mit Mischtypen verwendet werden.

Löschen Sie zum Erstellen des folgenden Beispielprogramms zunächst den Inhalt des Hauptfensters. Platzieren Sie jetzt die folgenden Objekte im Hauptfenster und verbinden Sie sie wie gezeigt; notieren Sie dabei die folgenden Informationen.

- 1 Wählen Sie **File**  $\Rightarrow$  **New**, um den Arbeitsbereich zu löschen.
- 2 Fügen Sie ein Objekt **Real64 Constant** hinzu, indem Sie **Data**  $\Rightarrow$  **Constant**  $\Rightarrow$  **Real64** wählen.
- 3 Fügen Sie ein Objekt **Complex Constant** hinzu, indem Sie **Data**  $\Rightarrow$  **Constant**  $\Rightarrow$  **Complex** wählen.
- 4 Fügen Sie ein Objekt **A + B** hinzu. Wählen Sie **Device**  $\Rightarrow$  **Function & Object Browser**, um den **Function & Object Browser** aufzurufen. Wählen Sie anschließend Type: Operators; Category: Arithmetic; Operators: +. Klicken Sie auf die Schaltfläche **Create Formula**, um das Objekt zu erstellen.
- 5 Fügen Sie ein Objekt AlphaNumeric hinzu, indem Sie **Display**  $\Rightarrow$  **AlphaNumeric** auswählen. Verbinden Sie die Objekte wie in Abbildung 54 gezeigt. Geben Sie den Wert 1.53 im Dateneingabefeld des Objekts Real64 Constant und den komplexen Wert (2, 1) im Objekt Complex ein. Führen Sie das Programm aus. Das Ergebnis sollte wie das in Abbildung 54 aussehen.



**Abbildung 54** Verschiedene Datentypen

VEE wandelt die Daten automatisch nach Bedarf um und führt die Addition in dem Objekt **A + B** durch. Der reale Wert **1.53** wird in den komplexen Wert **(1.53,0)** umgewandelt und anschließend zu dem komplexen Wert **(2,1)** addiert. Das Ergebnis **(3.53,1)** (eine komplexe Zahl) wird im Objekt AlphaNumeric angezeigt.

### HINWEIS

Normalerweise verarbeitet VEE alle Umwandlungen von Datentypen automatisch. Weitere Informationen können Sie über **Help** ⇒ **Contents and Index** in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch **How Do I...**, **Tell Me About...** oder **Reference**.

---

### Verschiedene Datenformen

VEE unterstützt eine Vielzahl von Datenformen wie beispielsweise Skalare und Arrays. Anders als bei den meisten anderen Programmiersprachen können VEE-Objekte mit einem vollständigen Array verwendet werden und nicht nur mit einzelnen Elementen.

Das folgende Programm erstellt ein eindimensionales Array aus zehn Elementen, berechnet den Mittelwert der 10 Werte und zeigt diesen Mittelwert an.

- 1 Wählen Sie **File** ⇒ **New**, um den Arbeitsbereich zu löschen.
- 2 Fügen Sie ein Objekt For Range hinzu, indem Sie **Flow** ⇒ **Repeat** ⇒ **For Range** wählen.
- 3 Fügen Sie ein Objekt Sliding Collector hinzu, indem Sie **Data** ⇒ **Sliding Collector** wählen.
- 4 Fügen Sie ein Objekt **median(x)** hinzu. Wählen Sie **Device** ⇒ **Function & Object Browser**. Wählen Sie anschließend Type: Built-in Functions; Category: Probability & Statistics; Functions: median, und klicken Sie auf die Schaltfläche **Create Formula**.

Sie können den Function & Object Browser anzeigen, indem Sie in der Symbolleiste auf die Schaltfläche **fx** klicken.

- 5 Fügen Sie ein Objekt AlphaNumeric hinzu, indem Sie **Display** ⇒ **AlphaNumeric** auswählen. Verbinden Sie die Objekte wie in Abbildung 55 gezeigt. Führen Sie das Programm aus. Wenn Sie keine der Eingänge an den Objekten geändert haben, sollte das Ergebnis Abbildung 55 gleichen.

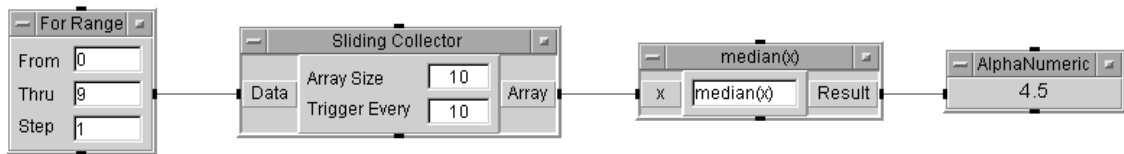


Abbildung 55 Datenobjekten verbinden

### Das Formula-Objekt

VEE bietet mathematische Operatoren und Funktionen, die in der Online-Hilfe im Abschnitt Reference beschrieben sind. Wählen Sie **Help** ⇒ **Contents and Index**. Sie können sich nun in der Online-Hilfe über die gewünschten Themen informieren.

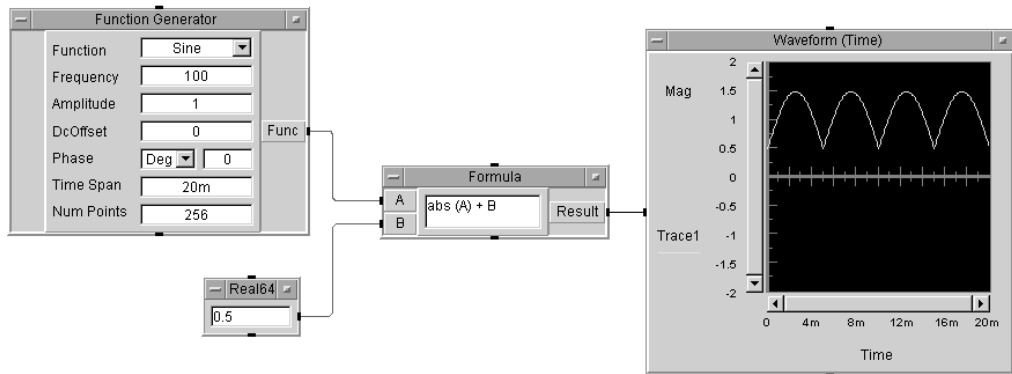
Die vordefinierten Operator- und Funktionsobjekte sind über **Device** ⇒ **Function & Object Browser** (oder **fx** in der Symbolleiste) verfügbar. Sie wählen sie im Function & Object Browser, indem Sie unter Type, Category und Functions jeweils auf den gewünschten Eintrag klicken. Klicken Sie auf die Schaltfläche **Create Formula**, um das Objekt zu erstellen.

Neben den vordefinierten Operatoren und Funktionen können Sie jeden beliebigen gültigen mathematischen VEE-Ausdruck mit dem Formula-Objekt erstellen, auf das Sie über das Menü Device zugreifen. In diesem Abschnitt erstellen Sie ein Programm mit einem Formula-Objekt. Löschen Sie zunächst den Inhalt des Hauptfensters, und gehen Sie anhand der folgenden Schritte vor:

- 1 Fügen Sie das Objekt Function Generator dem Hauptfenster hinzu und ändern Sie es, so dass sich eine **100 Hz**-Sinuskurve ergibt. Wählen Sie **Device** ⇒ **Virtual Source** ⇒ **Function Generator**.
- 2 Wählen Sie **Device** ⇒ **Formula**, um das **Formula**-Objekt dem Hauptfenster hinzuzufügen. Fügen Sie dem Objekt einen zweiten Eingang (**B**) hinzu, indem Sie mit dem Mauszeiger in den Eingangsanschlussbereich zeigen und **Strg+A** drücken.
- 3 Geben Sie den mathematischen Ausdruck **abs(A)+B** im Eingabefeld ein.

## 2 Agilent VEE Programmieretechniken

- 4 Wählen Sie **Data** ⇒ **Constant** ⇒ **Real64**, um ein Objekt Real64 Constant dem Hauptfenster hinzuzufügen. Geben Sie den Wert 0.5 ein.
- 5 Wählen Sie **Display** ⇒ **Waveform (Time)** und legen Sie die Skala der y-Achse (**y-axis**) auf **-2 bis 2** fest. Stellen Sie Automatic Scaling (Automatische Skalierung) auf Off (Aus). Klicken Sie zum Aufrufen des Dialogfeldes für diese Parameter auf **Mag**.
- 6 Verbinden Sie die Objekte wie in Abbildung 56 gezeigt. Führen Sie das Programm aus.



**Abbildung 56** Ein Programm mit Formula-Objekt erstellen

Bei der Ausführung des Programms verwendet das Formula-Objekt den Wellenformeingang **A** sowie den realen Wert **B** und addiert **B** zu dem absoluten Wert von **A**. Der Ausdruck **abs(A)+B** führt ein „Gleichrichten“ der Sinuskurve durch und addiert einen „Gleichstromversatz“. Sie können das gleiche Ergebnis auch über die Objekte **A + B** und **abs(x)** erzielen, es ist jedoch einfacher, einen Ausdruck in ein **Formula**-Objekt einzulesen. (Außerdem sparen Sie dadurch Platz.)

Doppelklicken Sie auf die Ein- und Ausgangsanschlüsse des Formula-Objekts. Beachten Sie, dass das reale Skalar an Eingang **B** jedem Element der Wellenformdaten (einem eindimensionalen Array) an Eingabe **A** hinzugefügt wird; die resultierende Wellenform ist der Ausgang am Anschluss **Result**.



**HINWEIS**

Zur Ergänzung der umfangreichen mathematischen Funktionen von VEE stehen über die MATLAB Script-Integration Hunderte weiterer mathematischer Funktionen zur Verfügung. Sie können im Function & Object Browser durch diese Funktionen blättern. Weitere Informationen zu MATLAB-Funktionen finden Sie im Abschnitt „MATLAB Script in Agilent VEE verwenden“ auf Seite 190 in Kapitel, „Testdaten analysieren und anzeigen“.

---

### Die Online-Hilfe

Sie haben bereits einfache Programme erstellt – jetzt werden Sie einige Möglichkeiten kennen lernen, wie Sie mehr über VEE erfahren können.

- 1** Führen Sie zunächst die Multimedia-Lernprogramme auf die Sie über **Help** ⇒ **Welcome** zugreifen können. Sie können so viele der wichtigsten Funktionen von VEE kennen lernen. Sie werden ihnen helfen, in kurzer Zeit effizient arbeiten zu können. Die Lernprogramme bestehen aus Bildschirmdemonstrationen zum Erstellen und Ausführen von VEE-Programmen und enthalten Erläuterungen zu der jeweiligen Anzeige. In den Lernprogrammen werden auch Konzepte für einen effizienten Einsatz von VEE vorgestellt.
- 2** Wenn Sie mit VEE vertraut sind, schlagen Sie weitere Informationen in den Help-Einträgen der Objektmenüs nach. Sie können mit den Objekten experimentieren, bis Sie deren Funktionsweise vollständig verstanden haben. Wenn Sie mehr über ein Objekt erfahren wollen, finden Sie spezifische Informationen in den jeweiligen Objektmenüs. Schlagen Sie zunächst in diesen Menüs nach.
- 3** Wenn Sie mit den Hilfethemen, dem Index oder den Suchfunktionen arbeiten wollen, rufen Sie in der VEE-Hauptmenüleiste die Hilfe auf.

#### HINWEIS

Eine Übersicht zum Öffnen der Hilfe-Hauptfunktion und zum Öffnen einer Liste der Hilfe-Inhalte finden Sie im Abschnitt in „Hilfe aufrufen“ auf Seite 25.

---

### Verwenden der Hilfefunktion

Die Online-Hilfe bietet Informationen zu folgenden Themen:

- Alle Menübefehle mit den meisten Direktaufrufen
- Informationen zu Gerätetreibern
- Häufig ausgeführte Aufgaben und viele Beispielprogramme

- Definition der VEE-Begriffe
- Verwenden der Hilfefunktion
- VEE-Version

Sie können durch die Themen blättern, den Schlüsselwort-Index verwenden, Hyperlinks zu verwandten Themen aufrufen oder die Suchfunktion verwenden. VEE umfasst viele Hilfefunktionen, die Sie bei der Entwicklung von Programmen einsetzen können.

## HINWEIS

Außerdem enthält VEE weitere hilfreiche Funktionen für die Entwicklung und Fehlerbehebung von Programmen wie beispielsweise „Line Probe“. Weitere Informationen hierzu finden Sie unter „Programme in Agilent VEE debuggen“ auf Seite 105.

## Hilfethemen zu einem Objekt anzeigen

Wenn Sie Informationen zu einem Objekt aufrufen möchten, klicken Sie auf die Schaltfläche „Objektmenü“ und wählen Sie den Eintrag Help.

- Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Count**, um ein Objekt **For Count** zu erstellen. Klicken Sie auf das Objektmenü und wählen Sie Help. Das Hilfethema mit der Beschreibung zu For Count wird angezeigt.
- Wählen Sie **Device** ⇒ **Formula**, um ein Formula-Objekt zu erstellen. Klicken Sie das Objektmenü und wählen Sie Help. Das Hilfethema wird mit einer Beschreibung der im Formula-Objekt gewählten Formel angezeigt.
- Wählen Sie **Device** ⇒ **Function & Object Browser**. Wählen Sie eine Kombination der Auswahloptionen und klicken Sie auf **Help**. Das Hilfethema zu dem ausgewählten Objekt wird angezeigt.

### Die Menüposition eines Objekts suchen

Wenn Sie herausfinden möchten, über welches Menü Sie auf ein bestimmtes Objekt zugreifen können, oder wenn Sie Informationen zu diesem Objekt anzeigen möchten, wählen Sie **Help** ⇒ **Contents and Index**, klicken Sie auf die Registerkarte **Index**, geben den Namen des Objekts ein und klicken auf **Anzeigen**.

Wählen Sie beispielsweise **Help** ⇒ **Contents and Index**, klicken Sie auf die Registerkarte **Index** und geben Sie **Collector** ein. Klicken Sie auf **Anzeigen**, um das Hilfethema zu dem Objekt **Collector** anzuzeigen.

### Weitere praktische Übungen mit der Hilfefunktion

- Nachschlagen des Direktaufrufs zum Löschen eines Objekts.
- Wählen Sie **Help** ⇒ **Contents and Index** ⇒ **How Do I...** ⇒ **Use the Keyboard Shortcuts** ⇒ **Editing Programs** ⇒ **To Cut an Object or Text**.
- Nachschlagen des Stichworts „terminal“ (Anschluss).

Wählen Sie **Help** ⇒ **Contents** ⇒ **Reference** ⇒ **Glossary** ⇒ **Terminal**.

- Nachschlagen der VEE-Versionsnummer.

Wählen Sie **Help** ⇒ **About VEE Pro**.

- Feststellen, was in dieser Version von VEE neu ist.

Wählen Sie **Help** ⇒ **Contents and Index** ⇒ **What's New in Agilent VEE Pro**.

## Programme in Agilent VEE debuggen

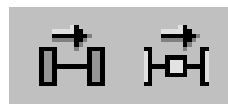
In dieser Übung arbeiten Sie mit dem Programm verwendet, das Sie in „Übung 2-4: Eine Fensteransicht (Benutzerschnittstelle) erstellen“ auf Seite 94 erstellt haben. Wählen Sie **File** ⇒ **Open**, markieren Sie **simple-program\_with\_panel.vee** und klicken Sie auf **OK**.

VEE zeigt während der Entwicklung und bei der Ausführung eines Programms Fehlermeldungen an; die Meldungen „Caution“ (Warnung), „Error“ (Fehler) oder „Information“ können wie folgt angezeigt werden:

- Während der Ausführung eines Programms kann eine Meldung mit der gelben Überschrift Caution angezeigt werden.
- Während der Ausführung eines Programms kann eine Meldung mit der roten Überschrift Error angezeigt werden.
- Wenn Sie beim Erstellen eines Programms einen Fehler gemacht haben, beispielsweise einen Werts außerhalb des zulässigen Bereichs (**33000** in einem **Int16 Constant**-Feld) eingegeben haben, zeigt VEE eine **Error**-Meldung mit einer dunkelblauen Titelleiste an.
- Zudem zeigt VEE Informationen zu Fehlern und Warnungen in der Statusleiste an. Die Statusleiste befindet sich am unteren Rand des VEE-Fensters.

### Den Datenfluss anzeigen

- 1 Klicken Sie in der Mitte der Symbolleiste auf die Schaltfläche Show Data Flow; diese ist in Abbildung 57 abgebildet. (Sie können statt dessen auch **Debug** ⇒ **Show Data Flow** wählen.)

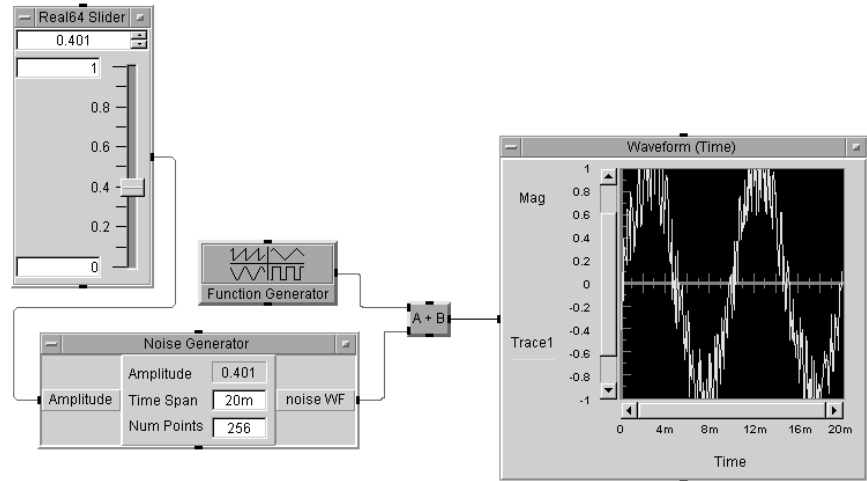


Schaltfläche „Show Data Flow“  
in der Symbolleiste

**Abbildung 57** Datenfluss anzeigen

## 2 Agilent VEE Programmierstechniken

(Klicken Sie erneut auf die Schaltfläche, um die Funktion zu deaktivieren.) Bei der Ausführung des Programms werden kleine Quadrate entlang der Datenlinien verschoben, die den Datenfluss kennzeichnen.

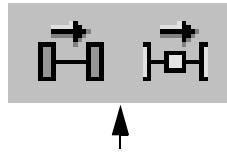


**Abbildung 58** Datenfluss in simple-program.vee

In Abbildung 58 beispielsweise werden Daten vom Objekt Real64 Slider zum Objekt Noise Generator geleitet. Die von Noise Generator und Function Generator ausgegebenen Daten werden in das Objekt **A + B** eingegeben; die Ergebnisse werden in der Anzeige Waveform (Time) dargestellt.

### Den Ausführungsfluss anzeigen

- 1 Klicken Sie in der Symbolleiste auf die Schaltfläche Show Execution Flow; diese ist in Abbildung 59 dargestellt. (Sie können statt dessen auch **Debug** ⇒ **Show Execution Flow** wählen.)



Schaltfläche „Show Execution Flow“ in der Symbolleiste

**Abbildung 59** Ausführungsfluss anzeigen

Bei der Ausführung eines Programms wird eine farbige Umrisslinie um die gerade ausgeführten Objekte angezeigt.

Setzen Sie die Anzeige von Datenfluss und Ausführungsfluss ein, wenn Sie die Arbeitsweise eines Programms nachvollziehen möchten. Für eine höhere Leistung sollten Sie diese Funktionen jedoch ausschalten. Durch Kombinieren dieser Funktionen mit Fehlerbehebungs-Tools wie Unterbrechungspunkten können Sie leichter nachvollziehen, wie ein VEE-Programm arbeitet und wo mögliche Fehlerquellen liegen.

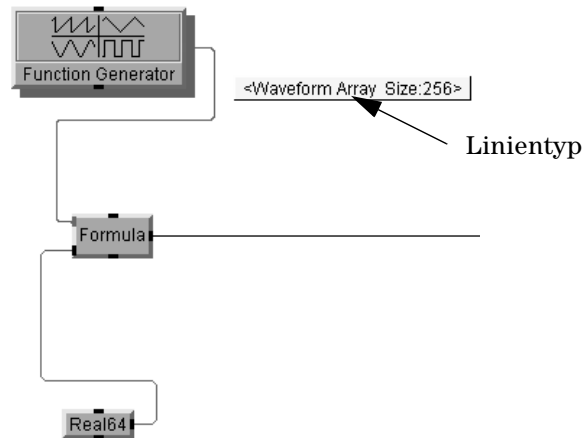
## Daten an einer Linie untersuchen

Das Überprüfen von Daten an verschiedenen Stellen in Ihrem Programm ist eine schnelle und hilfreiche Methode, Fehler in Ihrem Programm zu finden und zu beheben. Die Linienprüfung (Line Probe) ist eine Möglichkeit, die Daten an einer gegebenen Linie zu überprüfen.

Positionieren Sie den Mauszeiger auf einer Datenlinie in der Detailansicht. Der Cursor wird als grafische Darstellung eines Vergrößerungsglases angezeigt. Die Linie und ihre Verbindungen werden markiert, ein Feld mit dem Datenwert an der Linie wird eingeblendet. Klicken Sie mit dem Cursor (Vergrößerungsglas), um ein Dialogfeld mit weiteren Informationen zu der Datenlinie anzuzeigen. (Sie können auch **Debug** ⇒ **Line Probe** wählen und anschließend auf eine Linie klicken.)

Abbildung 60 beispielsweise zeigt ein VEE-Programm, in dem die Ausgabe des als Symbol dargestellten **Function Generator** angezeigt wird. Sie können sehen, dass der **Function Generator** ein Wellenform-Array mit 256 Punkten generiert.

## 2 Agilent VEE Programmier Techniken



**Abbildung 60** Anzeige des Werts an einem Ausgangs-Pin

Wenn Sie auf eine Datenlinie klicken, wird ein Dialogfeld mit allen Informationen zu den Daten an der Linie eingeblendet. Abbildung 61 zeigt beispielsweise das Dialogfeld, das durch Klicken auf den Ausgang des Function Generator eingeblendet wird.



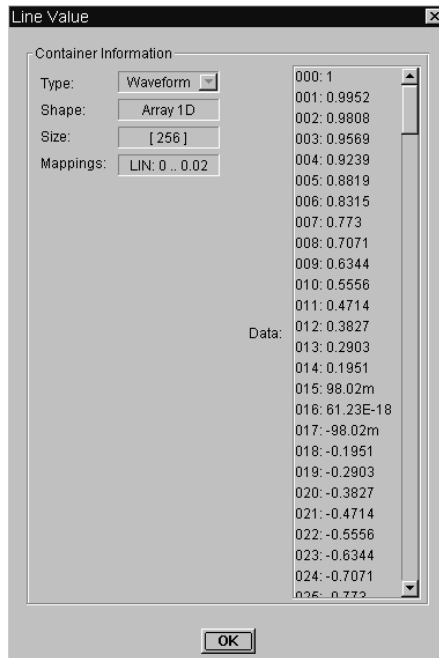


Abbildung 61 Informationen über eine Linie anzeigen

## Anschlüsse überprüfen

Sie können einen Anschluss überprüfen, indem Sie ihn in der offenen Ansicht doppelklicken, wie im Abschnitt „Informationen zu Pins und Anschlüssen“ auf Seite 49 beschrieben. Wenn ein Objekt als Symbol angezeigt wird, positionieren Sie den Mauszeiger auf den Anschluss. VEE blendet daraufhin automatisch den Namen des Anschlusses ein.

## Alphanumerischen Anzeigen für die Fehlerbehebung

Sie können die Anzeigen Alphanumeric oder Logging Alphanumeric an verschiedenen Punkten in einem Programm hinzufügen, um den Datenfluss zu verfolgen. Wenn das Programm richtig ausgeführt wird, löschen Sie diese Anzeigen wieder. AlphaNumeric zeigt einen einzelnen Daten-Container an (einen Scalar-Wert, ein **Array 1D** oder ein **Array 2D**), während Logging

AlphaNumeric (ein **Scalar** oder ein **Array 1D**) die fortlaufende Eingabe als Historie früherer Werte anzeigt. Sie können auch einen Counter (Zähler) verwenden, um anzuzeigen, wie oft ein Objekt ausgeführt wurde.

### Unterbrechungspunkte einsetzen

Ein Unterbrechungspunkt bewirkt, dass ein Programm vor der Ausführung eines bestimmten Objekts angehalten wird. Sie können Unterbrechungspunkte in einem Programm setzen, um die Daten zu überprüfen. Wenn ein Unterbrechungspunkt auf ein Objekt gesetzt wird, wird das Objekt mit einer orangefarbenen Umrisslinie hervorgehoben. Das Programm wird vor der Ausführung dieses Objekts angehalten.

- 1 Setzen Sie einen Unterbrechungspunkt auf ein einzelnes Objekt. Doppelklicken Sie auf die Titelleiste eines Objekts, um das Dialogfeld Properties aufzurufen, wählen Sie Breakpoint Enabled (Unterbrechungspunkt aktiviert) und klicken Sie auf **OK**. Wählen Sie anschließend **Debug** ⇒ **Activate Breakpoints**. Führen Sie das Programm aus. Es wird an dem gesetzten Unterbrechungspunkt angehalten.
- 2 Setzen Sie weitere Unterbrechungspunkte an verschiedenen weiteren Objekten. Sie können mehrere Objekte wählen, indem Sie die **Strg**-Taste gedrückt halten und nacheinander auf die Objekte klicken. Klicken Sie in der Symbolleiste auf die Schaltfläche **Toggle Breakpoint(s)** (siehe Abbildung 62). (Sie können statt dessen auch **Strg+B** drücken.) Führen Sie das Programm erneut aus. Das Programm wird an dem ersten Objekt mit einem gesetzten Unterbrechungspunkt angehalten.



↑  
Schaltfläche  
„Toggle Breakpoint“

**Abbildung 62** Unterbrechungspunkte setzen

- 3 Setzen Sie das Programm fort; es wird am nächsten Objekt mit einem Unterbrechungspunkt angehalten. Klicken Sie in der Symbolleiste auf die Schaltfläche **Fortsetzen** (siehe Abbildung 63). Sie können auch die entsprechende Funktion im Menü Debug verwenden.



Die Schaltfläche „Resume“  
(gleiche Position wie „Run“)

**Abbildung 63** Programm fortsetzen (gleiche Position wie Schaltfläche „Ausführen“)

- 4 Löschen Sie jetzt die Unterbrechungspunkte aus dem Programm. Markieren Sie dazu die Objekte mit Unterbrechungspunkten. Klicken Sie in der Symbolleiste auf die Schaltfläche **Toggle Breakpoint(s)** (siehe Abbildung 64). Sie können statt dessen auch **Debug** ⇒ **Clear All Breakpoints** wählen.



Schaltfläche  
„Toggle Breakpoint“

**Abbildung 64** Unterbrechungspunkt(e) löschen

- 5 Klicken Sie zum Anhalten oder Stoppen des Programms in der Symbolleiste auf die Schaltfläche **Pause** bzw. **Stop** (siehe Abbildung 65). Sie können auch die entsprechende Funktion im Menü Debug verwenden.

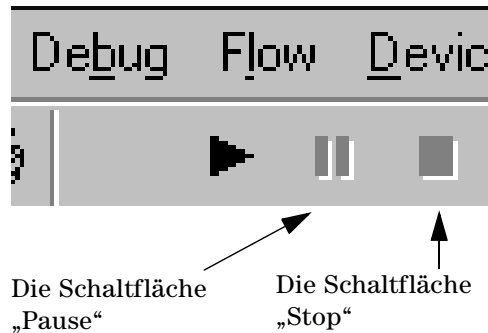


Abbildung 65 Programm anhalten oder stoppen

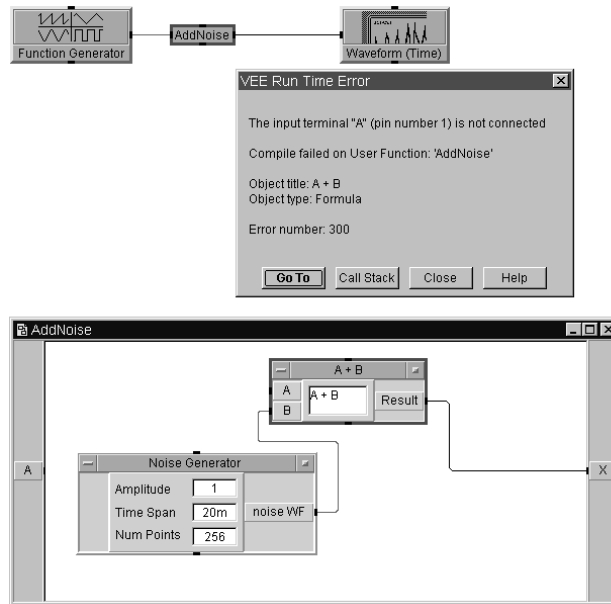
### Fehlerbehebung

Wenn bei der Ausführung eines Programms eine Fehlermeldung angezeigt wird, zieht VEE automatisch eine rote Umrisslinie um das Objekt, bei dem der Fehler festgestellt wurde.

Sie können entweder sofort den Fehler korrigieren, wodurch die Umrisslinie wieder ausgeblendet wird, oder Sie können auf die Schaltfläche **Stop** klicken; in diesem Fall wird die rote Umrisslinie ausgeblendet, und Sie können den Fehler beheben. Wenn Sie auf **Stop** klicken, können Sie den Fehler vor der Fortsetzung mit **View** ⇒ **Last Error** noch einmal anzeigen.

### Die Schaltfläche „Go To“ zum Suchen eines Fehlers

Abbildung 66 zeigt ein Beispiel für eine Laufzeit-Fehlermeldung. Bei der Ausführung dieses Programms zeigt VEE einen Laufzeitfehler an und zieht eine rote Umrisslinie um das UserObject AddNoise. Wenn Sie auf die Schaltfläche Go To klicken, öffnet VEE das UserObject AddNoise und zeigt eine rote Umrisslinie um das Objekt **A + B** an; an diesem Objekt fehlt eine Verbindung am Eingangs-Pin **A**. In einem großen Programm kann die Funktion Go To die schnelle Suche nach einem Fehler erheblich erleichtern.



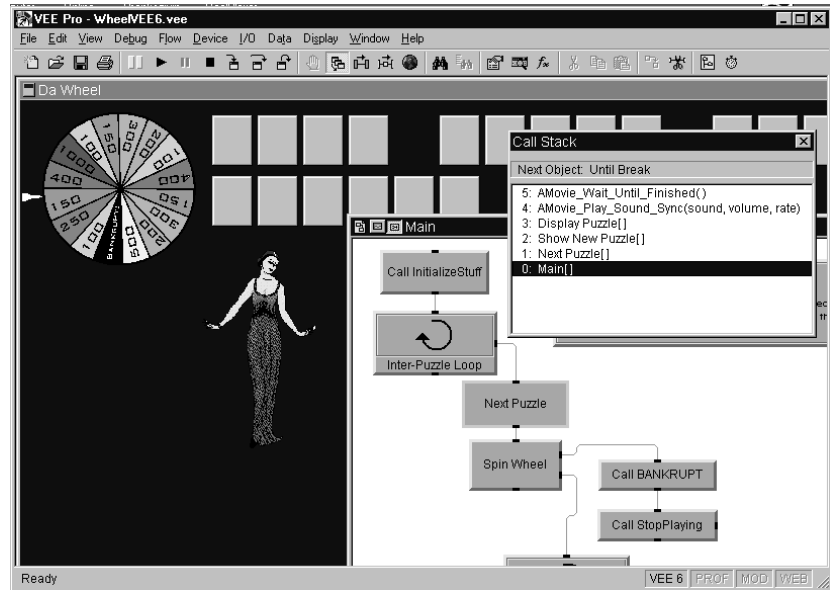
**Abbildung 66** Beispiel einer Laufzeitfehlermeldung mit Go To

## Den Aufrufstapel zur Fehlersuche nutzen

Wenn ein Fehler im Hauptprogramm vorliegt, ist er meist recht leicht zu sehen. In einem großen Programm erleichtert der Aufrufstapel (Call Stack) das Finden von Fehlern, die in mehreren Ebenen verschachtelt sind.

- 1 Klicken Sie auf die Schaltfläche **Pause** in der Symbolleiste (neben der Schaltfläche **Run**).
- 2 Klicken Sie im Dialogfeld des Fehlers auf die Schaltfläche **Call Stack**, oder wählen Sie **View** ⇒ **Call Stack**. Call Stack listet die Hierarchie der Programmausführung auf.

## 2 Agilent VEE Programmiertechniken

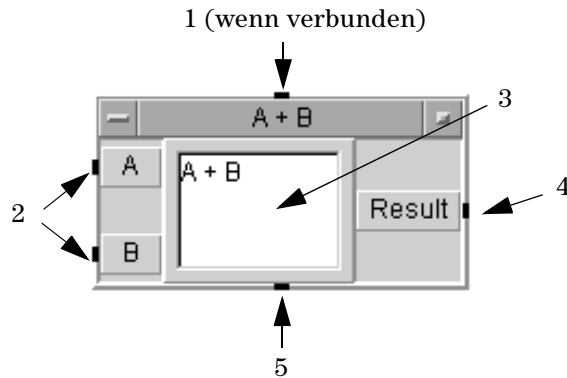


**Abbildung 67** Den Aufrufstapel in Wheel.exe einsetzen

Der Aufrufstapel zeigt die Hierarchie der Programmausführung an. Abbildung 67 zeigt ein Beispielprogramm, das zusammen mit VEE ausgeliefert wird: das Programm **Wheel.exe** in **Examples/Games**. In Abbildung 67 führt das Programm gerade die Benutzerfunktion `AMovie_Wait_Until_Finished()` aus, die von `AMovie_Play_Sound_Sync` aufgerufen wurde, die wiederum von `...Next_Puzzle` im Hauptprogramm aufgerufen wurde. Sie können auf jedes beliebige Element im Aufrufstapel doppelklicken, damit VEE die entsprechende Funktion sucht und anzeigt.

## Die Ereignisreihenfolge in einem Objekt verfolgen

Abbildung 68 zeigt die Reihenfolge von Ereignissen in einem Objekt.



**Abbildung 68** Die Reihenfolge der Ereignisse in einem Objekt

In Abbildung 68 arbeiten die Pins wie folgt:

**Tabelle 9** Sequenz der Pin-Operationen

1	Wenn der Sequenzeingangs-Pin verbunden ist, wird das Objekt erst ausgeführt, wenn es eine Meldung erhält, die es zur Ausführung auffordert (in VEE-Terminologie ein „ping“). Der Sequenzeingangs-Pin muss jedoch <i>nicht</i> angeschlossen zu sein.
2	Alle Dateneingangs-Pins müssen Daten enthalten, bevor das Objekt ausgeführt werden kann. (Sie können den meisten Objekten Datenein-/ausgangs-Pins hinzufügen. Sie können in jedem Objektmenü über das Untermenü <b>Add/Delete Terminal</b> feststellen, welche Pins hinzugefügt werden können.)
3	Das Objekt führt seine Aufgabe aus. In diesem Fall wird <b>A</b> dem Objekt <b>B</b> hinzugefügt, und das Ergebnis wird am Ausgangs-Pin angelegt.
4	Der Datenausgangs-Pin löst aus. Das Objekt wartet auf ein Signal des nächsten Objekts, dass die Daten empfangen wurden, und schließt dann die Operation ab. Ein gegebenes Objekt löst daher seinen Sequenzausgangs-Pin erst aus, wenn alle an seinem Datenausgangs-Pin angeschlossenen Objekte Daten empfangen haben.
5	Der Sequenzausgangs-Pin löst aus.

Es gibt einige Ausnahmen zu dieser Reihenfolge von Ereignissen:

- Sie können Fehler-Ausgangs-Pins hinzufügen, um Fehler innerhalb eines Objekts zu erfassen. Die Fehler-Ausgangs-Pins überschreiben das Verhalten des Standardobjekts. Wenn bei der Ausführung des Objekts ein Fehler auftritt, sendet der Fehler-Pin eine Meldung, und die Datenausgabe-Pins werden nicht ausgelöst.
- Sie können manchen Objekten Steuereingangs-Pins hinzufügen; diese können bewirken, dass das Objekt eine sofortige Aktion ausführt. Eine Unterfunktion eines Objekts wie Title oder Autoscale in der Anzeige Waveform (Time) kann beispielsweise mit Steuer-Pins ausgeführt werden. Steuerlinien zu einem Objekt werden in VEE-Programmen als gestrichelte Linien angezeigt.

Abbildung 69 beispielsweise zeigt eine Steuerlinie, die einen geänderten Titel für die Wellenformanzeige festlegt. Beachten Sie, dass das Objekt zur Ausführung dieser Aktion keine Daten an einem Steuer-Pin benötigt. Das Objekt wird nicht ausgeführt; die einzige ausgeführte Aktion ist das Festlegen des Titels. Durch Klicken auf **Show Data Flow** können Sie anzeigen, wie zuerst Daten über die Steuerlinie zum Steuereingang Title übertragen werden.



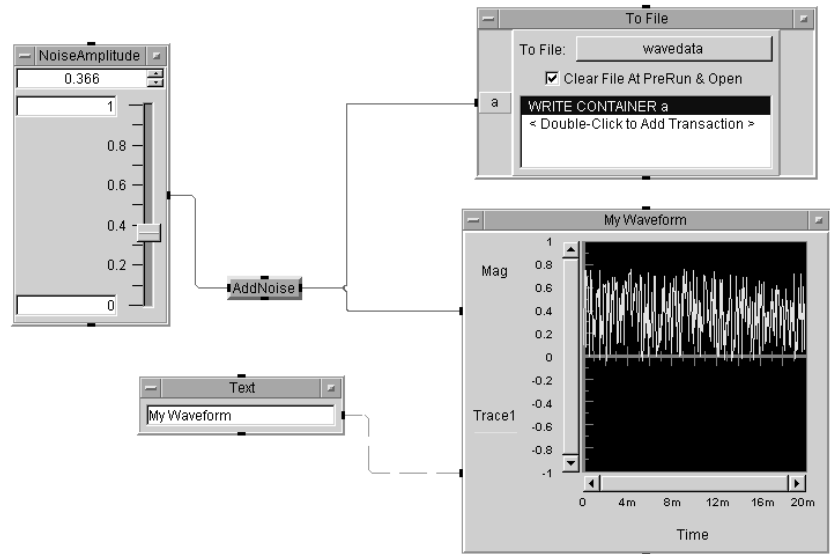


Abbildung 69 Steuerlinie zur Ausführung eines angepassten Titels

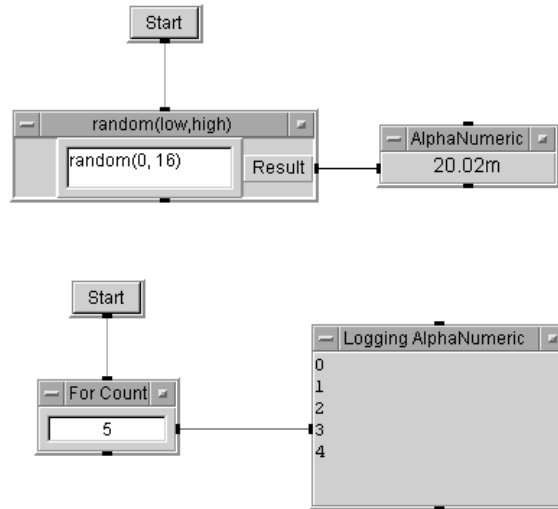
## Die Ausführungsreihenfolge von Objekten in einem Programm verfolgen

Bei der Ausführung eines VEE-Programms werden die Objekte in der folgenden Reihenfolge ausgeführt:

- 1 Startobjekte werden als Erste ausgeführt.

Abbildung 70 zeigt ein VEE-Programm mit zwei Threads; hierbei handelt es sich um Gruppen von Objekten, die mit durchgezogenen Linien in einem VEE-Programm verbunden sind. Die Startobjekte unter **Flow** ⇒ **Start** werden zur Ausführung der einzelnen Threads in einem Programm verwendet. Wenn ein Programm Startobjekte enthält, werden diese zuerst ausgeführt.

## 2 Agilent VEE Programmiertechniken



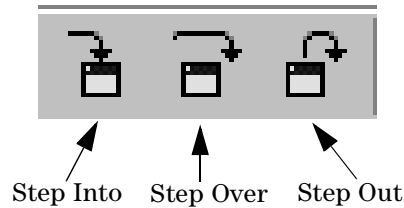
**Abbildung 70** Startobjekte zur Ausführung separater Threads

- 2 Objekte ohne Dateneingangs-Pins werden als nächste ausgeführt. **Data**  $\Rightarrow$  **Constant**-Objekte gehören häufig zu dieser Kategorie.
- 3 Objekte mit Eingangs-Pins werden nur ausgeführt, wenn an allen angeschlossenen Eingängen Daten anliegen. (Denken Sie daran, dass das Anschließen von Sequenzeingängen optional ist.)

### Schrittweise Ausführung eines Programms

Die schrittweise Ausführung eines Programms ist eine sehr effektive Methode zur Fehlerbehebung. VEE enthält die Funktionen Step Into, Step Over und Step Out, mit denen Sie in, über und aus einem Objekt springen können.

Zum Aktivieren der Schrittfunktion klicken Sie in der Symbolleiste auf eine der Schaltflächen Step Into, Step Over oder Step Out (siehe Abbildung 71).



**Abbildung 71** Schaltflächen „Step Into“, „Step Over“ und „Step Out“ in der Symbolleiste

- Mit Step Into führt ein Programm ein Objekt nach dem anderen aus. Wenn das Programm ein UserObject oder eine UserFunction erreicht, öffnet VEE das UserObject bzw. die UserFunction in die Detailansicht und führt die darin enthaltenen Objekte aus.
- Mit Step Over und Step Out führt ein Programm ein Objekt nach dem anderen aus, ohne UserObjects oder UserFunctions zu öffnen. Wenn das Programm ein UserObject oder eine UserFunction erreicht, führt VEE das UserObject bzw. die UserFunction vollständig aus.

Beispiel zur schrittweisen Ausführung eines Programms:

- 1 Öffnen Sie das Programm **simple-program\_with\_panel.vee**.
- 2 Klicken Sie in der Symbolleiste auf die Schaltfläche.
- 3 Mit jedem neuen Klicken auf Step Into wird das nächste Objekt ausgeführt und mit farbiger Umrisslinie angezeigt; Sie durchlaufen so das Programm Schritt für Schritt.

Bei der schrittweisen Ausführung zeigt VEE die Fensteransicht hinter der Detailansicht an; Sie sehen so die Reihenfolge, in der die Objekte ausgeführt werden. Innerhalb des Hauptprogramms sind an den Eingangsfeldern keine Eingangs-Pins angeschlossen, sie werden daher ohne bestimmte Reihenfolge ausgeführt. Wenn diese Felder in einer bestimmten Reihenfolge ausgeführt werden sollen, können Sie hierzu ihre Sequenz-Pins anschließen.

Der Datenfluss erfolgt von links nach rechts, die Datengeneratoren werden daher als nächstes ohne eine bestimmte Reihenfolge ausgeführt. Das Additionsobjekt (**A + B**) kann nicht ausge-

führt werden, bis beide Eingänge Daten empfangen haben. Anschließend wird das Objekt Waveform (Time) ausgeführt. Auch hierbei können Sie die Ausführung an einer bestimmten Stelle im Programm über die Sequenz-Pins oder das Objekt **Flow** ⇒ **Do** festlegen. (Weitere Informationen über das Objekt Do finden Sie in der Online-Hilfe.)

### HINWEIS

Weitere Informationen zu den Schrittfunktionen entnehmen Sie bitte der Online-Hilfe. Weitere Informationen zu UserFunctions finden Sie in Kapitel, „Agilent VEE-Funktionen verwenden“ auf Seite 345.

---

### Suchen eines Objekts in einem komplexen Programm

Wählen Sie **Edit** ⇒ **Find**, wenn Sie ein bestimmtes Objekt suchen möchten, insbesondere wenn es sich in einem großen Programm befindet. Geben Sie den Objekt- oder Funktionsnamen in dem eingblendeten Dialogfeld der Suchfunktion ein; VEE zeigt alle Instanzen und Positionen dieses Objekts bzw. dieser Funktion in dem Programm an. (Weitere Informationen siehe „Funktionen in großen Programmen finden“ auf Seite 373.)

## Übungen mit Programmen

Die Übungsprogramme in diesem Abschnitt verdeutlichen weitere VEE-Funktionen.

### 2-6: Generieren einer Zufallszahl

- 1 Dokumentieren Sie das Programm:
  - a Wählen Sie **Display** ⇒ **Note Pad** und platzieren Sie diesen Texteditor oben in der Mitte des Arbeitsbereichs. Klicken Sie in den Bearbeitungsbereich, um einen Cursor zu erhalten, und entfernen sie möglicherweise vorhandene Vorlagen. Geben Sie dann ein:

```
This program, Random, generates a real number
between 0 and 1, then displays the results.
```

- 2 Wählen Sie **Device** ⇒ **Function & Object Browser**. Wählen Sie dort **Type** ⇒ **Built-in functions**, **Category** ⇒ **All** und **Functions** ⇒ **random**. Klicken Sie auf die Schaltfläche **Create Formula Formula**. Positionieren Sie das Objekt im Arbeitsbereich und klicken Sie mit der Maustaste, um das Objekt zu platzieren.
- 3 Wählen Sie **Data** ⇒ **Constant** ⇒ **Int32** und platzieren Sie das Objekt links von random. Öffnen Sie das Int32-Objektmenü, klicken Sie auf **Clone** und setzen Sie dieses Objekt unter das andere Int32-Objekts. Doppelklicken Sie auf **0**, um einen Cursor anzuzeigen, und geben Sie 1 ein. Verbinden Sie die Konstantenobjekte mit random: das Objekt mit dem Wert **0** mit dem Eingangs-Pin „Low“, das Objekt mit dem Wert **1** mit dem Eingangs-Pin „High“.
- 4 Wählen Sie **Display** ⇒ **AlphaNumeric** und platzieren Sie das Objekt rechts von dem Objekt random. Öffnen Sie die Objektmenüs, und wählen Sie Help, um mehr über die Objekte zu erfahren.
- 5 Verbinden Sie den Ausgangs-Pin des Objekts random mit dem Eingangs-Pin von AlphaNumeric. Es wird eine Datenlinie angezeigt, die die beiden Objekte verbindet.

### HINWEIS

Wenn Sie die Linie mit dem Mauszeiger zum Ziel-Pin ziehen, wird dieser durch einen Rahmen hervorgehoben. Klicken Sie auf den Rahmen, um die Verbindung vollständig herzustellen.

### HINWEIS

Wenn Sie aus irgendeinem Grund die Operation zum Verbinden der Linie abbrechen möchten, bevor die Verbindung geschlossen wurde, doppelklicken Sie mit der Maustaste – die Linie wird ausgeblendet.

- 6 Klicken Sie in der Symbolleiste auf die Schaltfläche Run, um eine Zufallszahl anzuzeigen (siehe Abbildung 72).

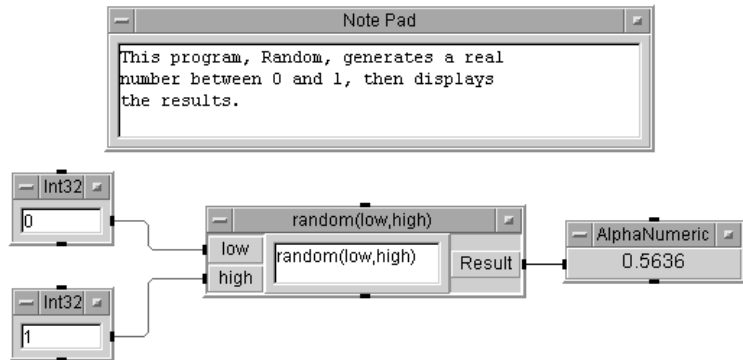


Abbildung 72 Das Programm „Random“

- 7 Wählen Sie **File** ⇒ **Save As...**, geben Sie `Random.VEE` ein und klicken Sie auf **Speichern**. Dieser Name wird beim nächsten Öffnen neben VEE in der Titelleiste angezeigt.

## 2-7: Eine globale Variable setzen und abrufen

Dieses Programm bietet Ihnen zusätzliche Übung mit den Grundfunktionen beim Erstellen eines VEE-Programms und erläutert das Prinzip der globalen Variablen. Sie können das Objekt Set Variable zum Erstellen einer Variablen verwenden, die Sie später in dem Programm mit einem Objekt Get Variable abrufen können. Sie können einen beliebigen VEE-Datentyp verwenden. In diesem Beispiel wird eine Zahl des Typs **Real64** ver-

wendet. (Weitere Informationen zu VEE-Datentypen finden Sie in Kapitel, „Testdaten analysieren und anzeigen“)

- 1 Wählen Sie **Display** ⇒ **Note Pad** und platzieren Sie diesen Texteditor oben in der Mitte des Arbeitsbereichs. Klicken Sie in den Bearbeitungsbereich, um einen Cursor zu erhalten, entfernen sie möglicherweise vorhandene Vorlagen und geben Sie den folgenden Text ein:

```
Set and Get a Global Variable prompts the user
to enter a real number. The variable, num, is
set to this real number. Then num is recalled
and displayed.
```

- 2 Wählen Sie **Data** ⇒ **Constant** ⇒ **Real64** und platzieren Sie das Objekt auf der linken Seite des Arbeitsbereichs. Öffnen Sie das Objektmenü und rufen Sie unter Help den Hilfe-Eintrag auf.
- 3 Öffnen Sie das Real64-Objektmenü und wählen Sie Properties. Geben Sie als neuen Titel die Eingabeaufforderung Enter a Real Number: ein und drücken Sie die Eingabetaste.

## HINWEIS

In dieser Übung wird der Titel eines der Konstantenobjekte einfach in eine Eingabeaufforderung geändert; es dient so im Programm als Eingabedialogfeld. Dies ist ein übliches Vorgehen, um Benutzereingaben zu ermöglichen. Sie können auch **Data** ⇒ **Dialog Box** ⇒ **Real64 Input** verwenden. Das Fenster Properties des Konstantenobjekts können Sie auch durch Doppelklicken auf die Titelleiste aufrufen.

- 4 Wählen Sie **Data** ⇒ **Variable** ⇒ **Set Variable** und platzieren Sie das Objekt rechts von dem Objekt Real64. Doppelklicken Sie auf globalA, um es zu markieren, und geben Sie anschließend num ein. Beachten Sie, dass sich der Name des Objekts in Set num ändert.

Dies bedeutet, dass der Benutzer eine reale Zahl im Objekt Real64 eingibt. Wenn der Benutzer auf die Schaltfläche Run klickt, wird die Zahl auf die globale Variable num gesetzt.

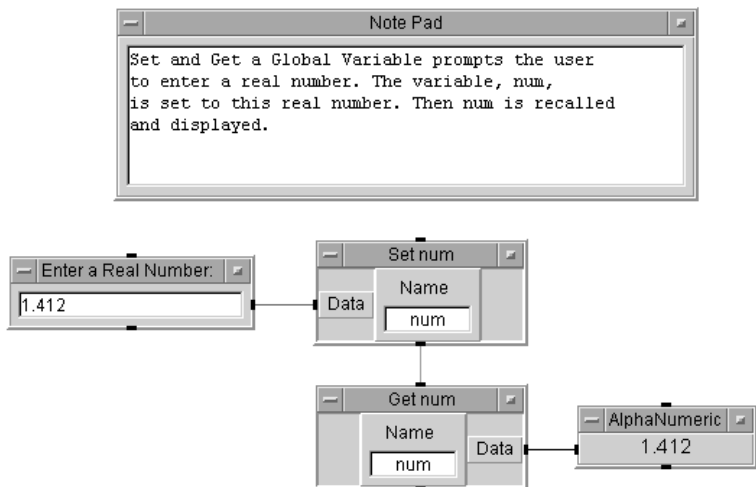
- 5 Verbinden Sie den Datenausgangs-Pin des Objekts Real mit dem Dateneingangs-Pin des Objekts Set num.

- 6 Wählen Sie **Data** ⇒ **Variable** ⇒ **Get Variable** und platzieren Sie das Objekt rechts von dem Objekt Real64. Ändern Sie den Variablennamen in num. Beachten Sie, dass sich der Name des Objekts in Get num ändert.
- 7 Verbinden Sie den Sequenzausgangs-Pin von Set num mit dem Sequenzeingangs-Pin von Get num.

### HINWEIS

Eine globale Variable muss gesetzt werden, bevor Sie sie verwenden können. Sie müssen daher in diesem Fall die Sequenz-Pins verwenden, um sicherzustellen, dass die Variable „num“ gesetzt wurde, bevor Sie sie mit **Get num** abrufen.

- 8 Wählen Sie **Display** ⇒ **AlphaNumeric** und platzieren Sie das Objekt rechts von dem Objekt Get num.
- 9 Verbinden Sie den Datenausgangs-Pin des Objekts Get num mit dem Dateneingangs-Pin von AlphaNumeric.
- 10 Geben Sie eine reale Zahl ein und klicken Sie in der Symbolleiste auf die Schaltfläche „Run“. Das Programm sollte jetzt aussehen wie in Abbildung 73.
- 11 Wählen Sie **Select File** ⇒ **Save As...**, und nennen Sie das Programm **global.vee**.



**Abbildung 73** Eine globale Variable setzen und abrufen



## Agilent VEE-Programme dokumentieren

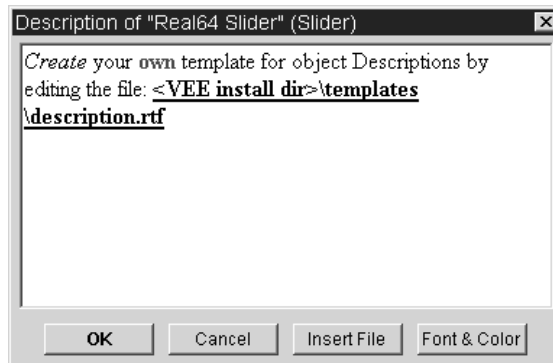
Durch die Verwendung des Befehls **File** ⇒ **Save Documentation...** können Sie automatisch die Dokumentation zu Ihrem Programm erstellen. VEE listet alle Objekte mit ihren wichtigsten Einstellungen, den Standard- und Benutzernamen, den Beschreibungseinträgen und allen Verschachtelungen auf. Objekt innerhalb eines UserObject sind beispielsweise eine Ebene von der VEE-Hauptumgebung aus verschachtelt; diese Ebenen werden mit Nummern gekennzeichnet.

Sie können auch einzelne Objekte mit einer Beschreibung dokumentieren. Zunächst wird in dieser Übung erläutert, wie ein einzelnes Objekt dokumentiert wird; anschließend wird das Generieren der Programmdokumentation beschrieben.

### Objekten mit dem Dialogfeld **Description** dokumentieren

Alle Objekte enthalten in ihrem Objektmenü den Befehl **Description** (Beschreibung). Mit diesem Befehl wird ein Dialogfeld aufgerufen, in das eine Beschreibung des entsprechenden Objekts eingegeben werden kann. Diese Dokumentationsdatei bietet außerdem die Möglichkeit, die Dokumentation mit Screenshots zu verknüpfen. In diesem Abschnitt erstellen Sie einen Eintrag im Dialogfeld **Description**.

- 1 Öffnen Sie das Programm **Random.vee**.
- 2 Löschen Sie den in Abbildung 74 gezeigten Vorlagen-Text aus dem Dialogfeld **Description**.



**Abbildung 74** Das Dialogfeld „Description“

- 3 Beachten Sie, dass das Feld Description ein „Rich Text Format“-Feld (RTF-Feld) ist. Sie können also jede beliebige RTF-Datei einfügen. Das RTF-Format bietet Ihnen enorme Flexibilität in der Präsentation Ihrer Textdaten.
- 4 Klicken Sie im Objektmenü des Fensters Main auf den Befehl **Description**. Geben Sie den Text in dem Dialogfeld ein: Das Programm Random.vee erzeugt Zufallszahlen. Klicken Sie auf **OK**, sobald die Angaben vollständig sind.

### HINWEIS

Die Einträge in dem Dialogfeld **Description** sind für die Benutzer nicht sichtbar, es sei denn, sie rufen sie über das Objektmenü auf. Beachten Sie auch, dass Sie in diesem Dialogfeld eine Datei einfügen können.

## Die Dokumentation automatisch generieren

Gehen Sie anhand der folgenden Schritte vor, um eine Datei mit der Programmdokumentation zu generieren:

- 1 Öffnen Sie das Programm **Random.vee**. Klicken Sie auf **File** ⇒ **Save Documentation....** Geben Sie den Dateinamen mit der Endung \*.txt (z. B. **Random.txt**) ein und klicken Sie auf **Speichern**. Standardmäßig wird die Datei auf dem PC in dem Ordner **C:\Eigene Dokumente\VEE Programs** gespeichert.
- 2 Öffnen Sie die Datei in einem beliebigen Texteditor. In Abbildung 75, Abbildung 76 und Abbildung 77 wird die Dokumentationsdatei mit Notepad in MS Windows 98 angezeigt.

Abbildung 75 zeigt den Anfang der Datei mit Informationen zu der Datei, dem Datum der Änderungen und der System-E/A-Konfiguration.

Source file: „C:\My Documents\VEE Programs\Random.vee“

File last revised: Mon Jan 03 15:29:02 2003

Date documented: Mon Feb 28 14:43:27 2003

VEE revision: 7.0

Execution mode: VEE 6

Convert Infinity on Binary Read: no

**Abbildung 75** Der Anfang der Dokumentationsdatei

## 2 Agilent VEE Programmier Techniken

M: Main

Device Type : Main

Description :

1. The program, Random, generates a real number between 0 and 1
2. and then displays the results.

Context is secured : off

Trig mode : Degrees

Popup Panel Title Text : Untitled

Show Popup Panel Title : on

Show Popup Panel Border : on

Popup Moveable : on

Popup Panel Title Text Color : Object Title Text

Popup Panel Title Background Color : Object Title

Popup Panel Title Text Font : Object Title Text

Delete Globals at Prerun : on

M.0: Main/Note Pad

Device Type : Note Pad

Note Contents :

1. This program, Random, generates a real
2. number between 0 and 1, then displays

### Abbildung 76 Die Mitte der Dokumentationsdatei

In Abbildung 76 sind die VEE-Objekte zusammen mit ihren Einstellungen beschrieben. Die Nummer vor den einzelnen Objekten gibt an, wo sich das Objekt befindet. Das erste Objekt im Hauptprogramm ist beispielsweise als **M1** aufgelistet. Abbildung 77 zeigt den Rest dieser Dokumentationsdatei als Referenz.

## M.2: Main/Int32

Device Type : Constant

Output pin 1 : Int32

Wait For Event : off

Auto execute : off

Initialize At Prerun : off

Initialize at Activate : off

Constant size fixed : off

Password masking : off

Indices Enabled : on

Int32 Value : 0

## M.4: Main/Int32

Device Type : Constant

Output pin 1 : Int32

Wait For Event : off

Auto execute : off

Initialize At Prerun : off

Initialize at Activate : off

Constant size fixed : off

**Abbildung 77** Der Rest der Dokumentationsdatei**HINWEIS**

Führen Sie nach der Ausführung des Befehls **Save Documentation** den Befehl **File** ⇒ **Print Program** aus, um die Objekte mit Identifikationsnummern zu versehen, damit Sie die Textdokumentation mit der Druckausgabe abstimmen können.

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Wiederholen Sie die Themen nötigenfalls, bevor Sie mit dem nächsten Kapitel fortfahren.

- Ein UserObject erstellen und erläutern, wie UserObjects Programmen eine Struktur verleihen und Platz auf dem Bildschirm sparen.
- Dialogfelder und Regler (oder Knöpfe) für die Benutzereingabe erstellen.
- Mit Datendateien Daten in einer Datei speichern und Daten aus einer Datei laden.
- Eine Benutzerschnittstelle in einer Fensteransicht des Programms erstellen.
- Verschiedene Datentypen und Datenformen verwenden.
- Mathematische Operatoren und Funktionen einsetzen.
- Die Online-Hilfe benutzen.
- Datenfluss und Ausführungsfluss in einem Programm anzeigen.
- Fehler in einem Programm durch Überprüfen von Daten an einer Linie, Anschlüssen und alphanumerischen Anzeigen beheben.
- Unterbrechungspunkte verwenden.
- Fehlern beheben mit dem Befehl GoTo.
- Fehler beheben mit Hilfe des Aufrufstapels.
- Mit Use Step Into, Step Over und Step Out die Fehler in einem Programm verfolgen und beheben.
- Die Suchfunktion verwenden.
- Objekte über die Beschreibungs-Dialogfelder dokumentieren.
- Eine Dokumentationsdatei generieren.

# 3

## Einfache Methoden zum Steuern von Instrumenten

Überblick	133
Instrumente konfigurieren	136
Einen Panel-Treiber verwenden	145
Das Objekt Direct I/O verwenden	150
PC-Zusatzkarten	160
Einen VXIPlug&Play-Treiber verwenden	166
Weitere E/A-Funktionen	171
Kapitel-Checkliste	172

## Einfache Methoden zum Steuern von Instrumenten

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Instrumente konfigurieren
- Einen Panel-Treiber verwenden
- Das Objekt „Direct I/O“ verwenden
- PC-Zusatzkarten steuern
- Einen *VXIPlug&Play*-Treiber verwenden

*Erforderliche Zeit für dieses Kapitel: 1 Stunde*



## Überblick

In diesem Kapitel lernen Sie, wie Sie mit VEE Instrumente steuern können. Dies ist auf verschiedene Arten möglich:

- **„Panel“-Treiber** bieten eine einfache Benutzerschnittstelle (ein „Bedienfeld“) zum Steuern eines Instruments von Ihrem Computerbildschirm aus. Wenn Sie Parameter im VEE-Panel-Treiber ändern, wird der entsprechende Status des Instruments geändert. Panel-Treiber werden von Agilent Technologies zusammen mit VEE zur Verfügung gestellt und decken mehr als 450 Instrumente von verschiedenen Herstellern ab.
- **Das Objekt „Direct I/O“** ermöglicht das Übertragen von Befehlen und das Empfangen von Daten über eine Reihe unterstützter Schnittstellen.
- **I/O libraries** (E/A-Bibliotheken) können zum Steuern von PC-Zusatzkarten importiert werden, um über das Objekt **Call** Funktionen aus dieser Bibliothek aufzurufen. Diese Bibliotheken werden meist als Dynamic Link Libraries (DLLs) ausgeliefert.
- **VXIPlug&Play-Treiber** können zum Aufruf von C-Funktionen zur Steuerung von Instrumenten verwendet werden. Diese Treiber werden von Agilent Technologies und anderen Herstellern mit ihren unterstützten Instrumenten zur Verfügung gestellt.

Dieses Kapitel vermittelt Ihnen Grundkenntnisse für die Steuerung von Instrumenten in den meisten Situationen. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

### Panel-Treiber

Agilent VEE enthält über 450 Panel-Treiber für Instrumente von verschiedenen Herstellern. Ein Panel-Treiber arbeitet mit einer Anzeige im VEE-Programm, über die die Einstellung des entsprechenden physischen Instrument gesteuert wird. Panel-Treiber bieten maximale Benutzerfreundlichkeit und sparen viel Zeit bei der Entwicklung. Abbildung 78 zeigt ein Beispiel eines Panel-Treibers.

### 3 Einfache Methoden zum Steuern von Instrumenten

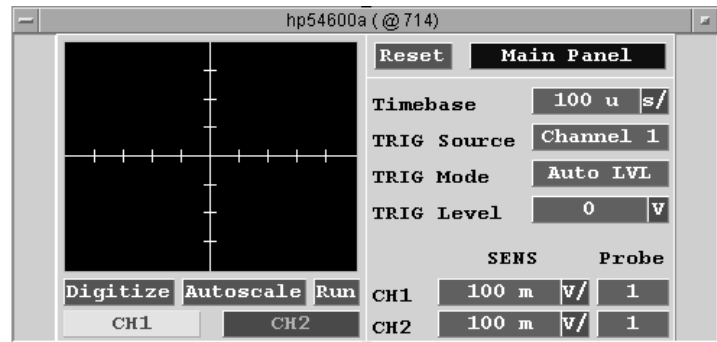


Abbildung 78 Der Panel-Treiber für das Oszilloskop HP54600A

#### Das Objekt „Direct I/O“

Das VEE-Objekt „Direct I/O“ ermöglicht die Kommunikation mit allen Instrumenten aller Hersteller über Standardschnittstellen, unabhängig davon, ob ein Treiber für das Instrument vorhanden ist oder nicht. Das Objekt „Direct I/O“ überträgt Befehle an das Instrument und empfängt Daten von dem Instrument. Die Verwendung von „Direct I/O“ ermöglicht im Allgemeinen eine schnellere Ausführung. Die Auswahl der besten Methode zur Steuerung von Instrumenten hängt von der Verfügbarkeit der Treiber, der Notwendigkeit für eine schnelle Testentwicklung und den Leistungsanforderungen ab. Abbildung 79 zeigt ein Beispiel für die Verwendung von „Direct I/O“ zur Steuerung eines Funktionsgenerators.

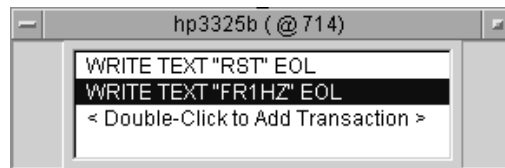


Abbildung 79 Ein Objekt „Direct I/O“ für einen Funktionsgenerator

## PC-Zusatzkarten mit E/A-Bibliothek

E/A-Bibliotheken werden normalerweise vom Hersteller der PC-Zusatzkarte in Form von Dynamically Linked Libraries (DLLs) für PC-Zusatzkarten geliefert. VEE ermöglicht die Steuerung von PC-Zusatzkarten durch den Aufruf von Bibliotheksfunktionen über das Objekt Call. Abbildung 80 zeigt ein Beispiel des Objekts Import Library, das die Funktionen in VEE verfügbar macht.

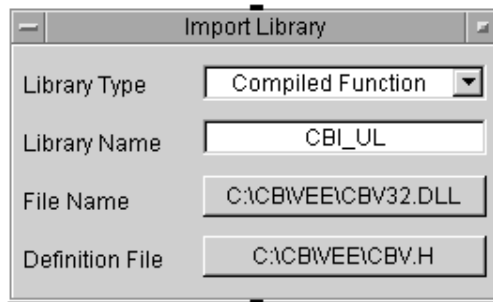


Abbildung 80 Eine PC-Zusatzbibliothek importieren

## VXIPlug&Play-Treiber

VXIPlug&Play-Treiber werden vom Hersteller des Instruments oder von Agilent Technologies geliefert. (Eine Liste der VXIPlug&Play-Treiber von Agilent Technologies finden Sie in der Literatur zu VEE oder im Handbuch *VEE Pro Advanced Techniques*. Wenden Sie sich bezüglich weiterer VXIPlug&Play-Treiber an den Hersteller Ihres Instruments.) VEE ermöglicht die Steuerung eines Instruments mit einem VXIPlug&Play-Treiber durch Aufrufen des Treibers. Abbildung 81 zeigt ein Beispiel für Aufrufe eines VXIPlug&Play-Treibers in VEE.

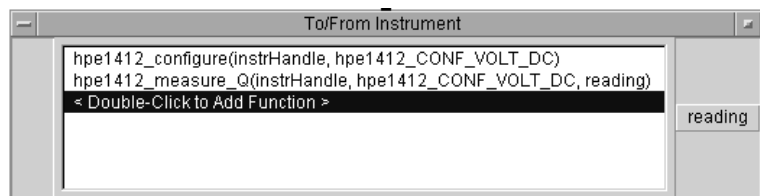


Abbildung 81 Einen VXIPlug&Play-Treiber in VEE aufrufen

## Instrumente konfigurieren

Mit VEE können Sie Programme entwickeln, ohne dass die Instrumente tatsächlich vorhanden sein müssen. In dieser Übung konfigurieren Sie ein Oszilloskop zur Verwendung mit einem Panel-Treiber. Anschließend fügen Sie das physische Instrument der Konfiguration hinzu.

### Übung 3-1: Ein aktuell nicht vorhandenes Instrument konfigurieren

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...** Verschieben Sie das Dialogfeld nach links oben in den Arbeitsbereich, indem Sie auf seine Titelleiste klicken und es ziehen (siehe Abbildung 82).



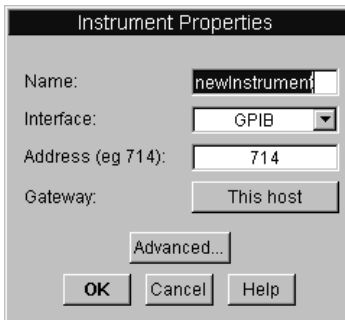
**Abbildung 82** Das Dialogfeld „Instrument Manager“

**HINWEIS**

Wenn Sie Instrumente angeschlossen und eingeschaltet haben, kann VEE diese Instrumente und die passenden Treiber dazu automatisch erkennen. Weitere Informationen zum automatischen Suchen und Konfigurieren von Instrumenten finden Sie in den Online-Lernprogrammen, auf die Sie in der in der VEE-Menüleiste unter **Help** ⇒ **Welcome** ⇒ **Tutorials** zugreifen können.

Standardmäßig sind keine Instrumente konfiguriert; in diesem Beispiel wird davon ausgegangen, dass keine Instrumente in der Liste des Instrument Manager angezeigt werden.

- 2 Vergewissern sie sich, dass im Dialogfeld Instrument Manager der Eintrag My Configuration gewählt ist, und klicken Sie unter Instrument auf **Add...** Das Dialogfeld Instrument Properties wird angezeigt (siehe Abbildung 83).



**Abbildung 83** Dialogfeld „Instrument Properties“

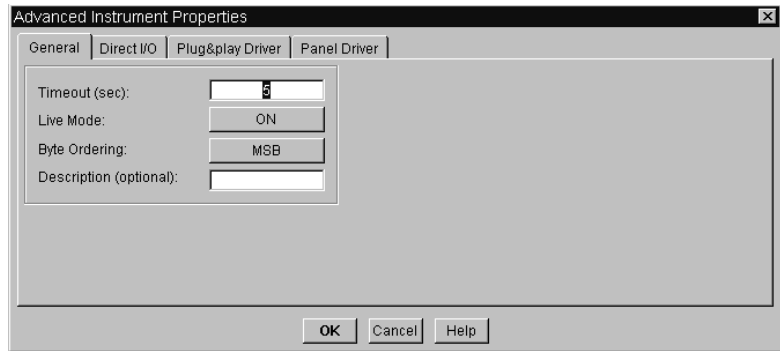
### 3 Einfache Methoden zum Steuern von Instrumenten

Das Dialogfeld Instrument Properties enthält die folgenden Elemente:

**Tabelle 10** Elemente des Dialogfelds Instrument Properties

Element	Beschreibung
Name	Der Name des Instruments wird im Programm angezeigt. Wählen Sie einen Namen entsprechend den folgenden Syntaxvorgaben: <ul style="list-style-type: none"><li>• Namen von Instrumenten müssen mit einem alphabetischen Zeichen beginnen, gefolgt von alphanumerischen Zeichen oder Unterstreichungszeichen.</li><li>• Sie können keine integrierten Leerzeichen in den Namen von Instrumenten verwenden.</li></ul>
Interface	Typ der Schnittstelle. Wählen Sie GPIB, Serial, GPIO, USB, LAN oder VXI.
Address	Die logische Einheit der Schnittstelle (GPIB verwendet normalerweise 7) plus der lokalen Busadresse des Instruments (eine Zahl von 0 bis 31). Wenn Sie die Adresse 0 beibehalten, bedeutet dies, dass bei der Entwicklung kein Instrument vorhanden ist.
Gateway	Gibt an, ob Instrumente lokal oder fern gesteuert werden. Übernehmen Sie den Standardeintrag This host für die lokale Steuerung von Instrumenten, oder geben Sie einen Gateway für Remote-Steuerung ein. (Weitere Informationen hierzu finden Sie im Handbuch <i>VEE Pro Advanced Techniques</i> .)

- 3 Ändern Sie den Namen in `scope`, behalten Sie alle weiteren Standardwerte bei, und klicken Sie auf **Advanced**. Das Dialogfeld Advanced Instrument Properties wird angezeigt (siehe Abbildung 84).



**Abbildung 84** Das Dialogfeld „Advanced Instrument Properties“

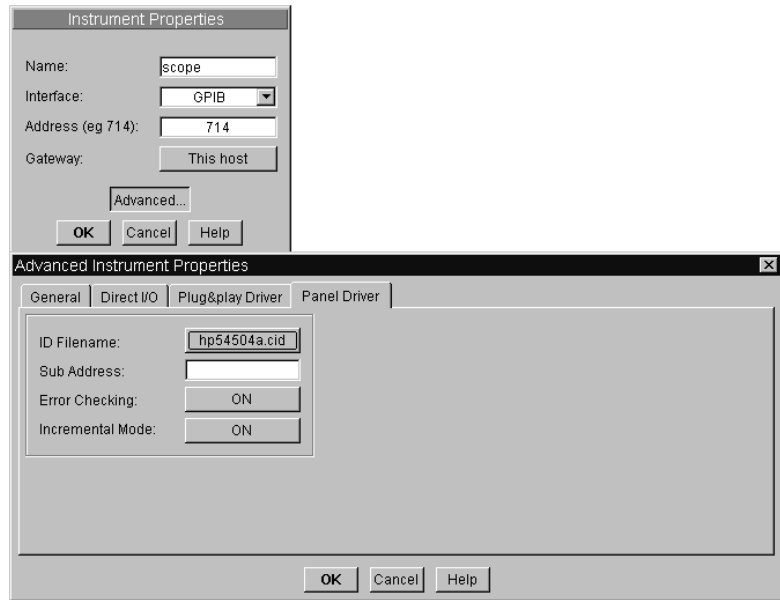
Die Registerkarte **General** enthält die folgenden Elemente:

**Tabelle 11** Einträge im Ordner General

Element	Beschreibung
Timeout	Die maximale Anzahl von Sekunden, die für die Ausführung einer E/A-Transaktion zulässig ist, bis eine Fehlermeldung angezeigt wird.
Live Mode	Gibt an, ob eine Echtzeit-Kommunikation mit dem Instrument erfolgt. Setzen Sie diesen Wert auf <b>OFF</b> ein, wenn kein Instrument angeschlossen ist. VEE verwendet den Standardwert <b>ON</b> .
Byte Ordering	Gibt an, welche Reihenfolge das Gerät zum Lesen und Schreiben von Binärdaten verwendet. Die Schaltfläche zeigt entweder „Most Significant Byte First“ (MSB) oder „Least Significant Byte First“ an. Alle IEEE488.2-kompatiblen Geräte müssen standardmäßig die MSB-Reihenfolge verwenden.
Beschreibung	Geben Sie hier eine Beschreibung ein. Wenn beispielsweise die Nummer des Instruments in der Titelleiste angezeigt werden soll, geben Sie sie hier ein.

- 4 Setzen Sie das Element Live Mode auf OFF. Klicken Sie anschließend auf die Registerkarte Panel Driver; das Dialogfeld entspricht nun Abbildung 85.

### 3 Einfache Methoden zum Steuern von Instrumenten



**Abbildung 85** Die Registerkarte „Panel Driver“

- 5 Klicken Sie auf die Schaltfläche rechts von ID Filename, um das Listenfeld Read from what Instrument Driver? zu öffnen. Die Liste enthält alle Panel-Treiberdateien, die mit Ihrer Version von VEE in dem angegebenen Verzeichnis gespeichert wurden.

#### HINWEIS

Sie müssen die Panel-Treiber von der VEE-CD-ROM installieren, um das Beispiel ausführen zu können. Die \*.cid-Dateien kennzeichnen die kompilierten Treiberdateien für das Instrument.

- 6 Navigieren Sie in der Liste zu dem Treiber **hp54504a.cid**, markieren Sie ihn und klicken Sie anschließend auf **Öffnen**. In Abbildung 85 ist diese Datei bereits gewählt. Sie können auch eine markierte Datei doppelklicken, um sie zu wählen.



Die weiteren Elemente der Registerkarte „Panel Driver“ sind:

**Tabelle 12** Panel-Treiberkonfiguration

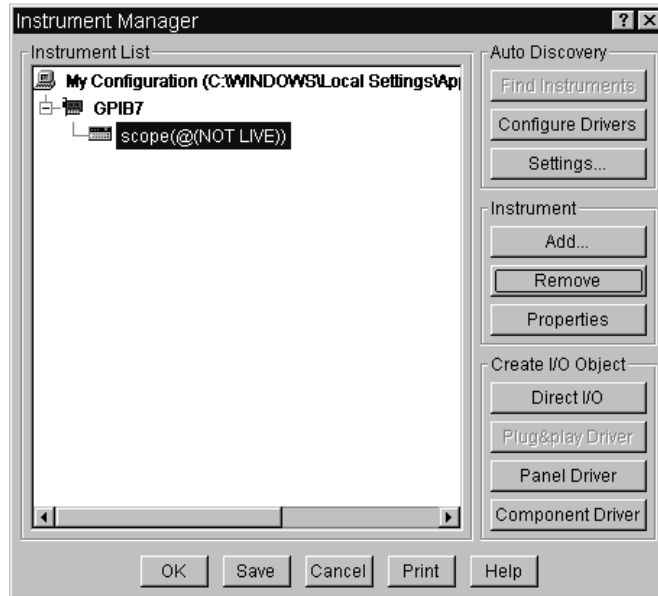
Panel-Treiber Element	Beschreibung
Sub Address	Lassen Sie dieses Feld leer. Die <b>Unteradresse</b> wird nur von Nicht-VXI-Gehäuseinstrumenten zur Kennzeichnung von Zusatzmodulen verwendet.
Error Checking	Behalten Sie die Standardeinstellung ON bei. Die Fehlerprüfung kann für einen höheren Durchsatz deaktiviert werden; in diesem Fall wird jedoch nicht auf E/A-Fehler geprüft.
Incremental Mode	Behalten Sie die Standardeinstellung <b>ON</b> bei. <b>Incremental Mode</b> kann auch deaktiviert werden; in diesem Fall wird die gesamte Befehlsfolge für den Status des Instruments bei jeder Änderung einer Einstellung gesendet.

- 7** Klicken Sie auf **OK**, um zum Dialogfeld Instrument Properties zurückzukehren. Klicken Sie auf **OK**.

Die Liste der verfügbaren Instrumente sollte jetzt eine Instrumentenkonfiguration mit dem Namen scope enthalten, die die Treiberdatei **hp54504a.cid** verwendet, wie in Abbildung 86 gezeigt. Für das Instrument ist keine Busadresse angegeben, da es nicht tatsächlich vorhanden ist. Sie können das Programm in diesem Modus entwickeln und die Adresse später hinzufügen, wenn das Instrument bereit ist zum Anschließen an den Computer.

Wenn Sie ein Feld ausgefüllt haben, drücken Sie Tabulator, um zum nächsten Feld zu springen. Mit der Tastenkombination Umschalt-Tabulator können Sie zum vorigen Feld wechseln. Das Drücken der Eingabetaste hat die gleiche Wirkung wie das Klicken auf **OK**: VEE schließt das Dialogfeld.

### 3 Einfache Methoden zum Steuern von Instrumenten



**Abbildung 86** Oszilloskop zur Liste der Instrumente hinzugefügt

- 8 Klicken Sie auf **Save**, um das Dialogfeld Instrument Manager zu schließen. (Sie können auch unter Create I/O Object auf Panel Driver klicken, um das Objekt sofort in das Programm einzufügen; VEE speichert in diesem Fall die Konfiguration automatisch.)

Sie haben jetzt das HP 54504A Oszilloskop mit dem Namen `scope` der Liste der Instrumente hinzugefügt. Sie können diesen Treiber bei der Programmierung verwenden, obwohl das eigentliche Instrument gar nicht vorhanden ist.

#### Ein Instrument für die Verwendung in einem Programm wählen

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...**
- 2 Wählen Sie **scope(@ (NOT LIVE))** und klicken Sie unter Create I/O Object auf **Panel Driver**.

**HINWEIS**

In Instrument Manager können Sie häufig verschiedene Arten von Objekten unter Create I/O Object erstellen, je nachdem, welche Art von Instrument konfiguriert ist. Wenn Sie für diese Übung beispielsweise Direct I/O statt Panel Driver ausgewählt hatten, erhalten Sie ein Objekt Direct I/O mit dem Namen **scope(@(NOT LIVE))**. VEE bietet außerdem einen Component Driver, der eine Teilmenge der Funktionen eines Panel Driver verwendet. (Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

- 3 Positionieren Sie den Umriss des Objekts scope, und klicken Sie mit der Maustaste, um es zu positionieren. Die Anzeige sollte jetzt aussehen wie in Abbildung 87.

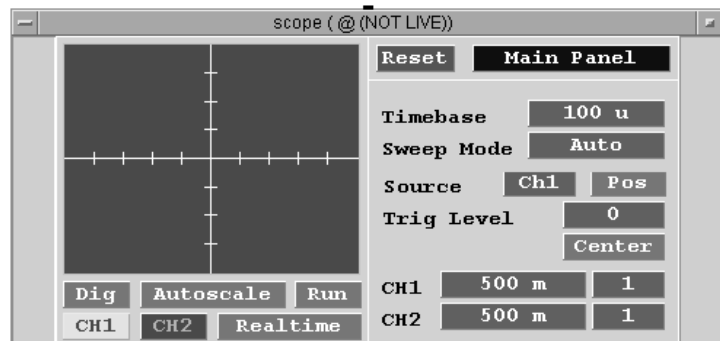


Abbildung 87 scope(@(NOT LIVE))

Sie können den Panel-Treiber jetzt wie jedes beliebige VEE-Objekt verwenden.

## Das physische Instrument zur Konfiguration hinzufügen

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...** und wählen Sie scope. Klicken Sie unter **Instrument...** auf **Properties**.
- 2 Doppelklicken Sie auf das Feld **Address**, um den aktuellen Eintrag markieren, und geben Sie 709 ein. Die 7 in 709 gibt die logische Einheit an. (Wenn die logische Einheit GPIB (HP-IB) nicht 7 ist, ersetzen Sie die 7 durch die tatsächliche

### 3 Einfache Methoden zum Steuern von Instrumenten

logische Nummer.) Die 9 in 709 ist die Standardadresse für Oszilloskope.

- 3 Klicken Sie auf **Advanced** und setzen Sie das Element Live Mode auf ON. Klicken Sie anschließend auf **OK**. Klicken Sie auf **OK**, um das Dialogfeld Instrument Properties zu schließen.
- 4 Klicken Sie auf **Save**, um die Änderungen zu speichern.

## Einen Panel-Treiber verwenden

In dieser Übung dient der HP 3325B Funktionsgenerator als Beispiel. Das gleiche Prinzip gilt jedoch für alle VEE-Panel-Treiber. Wenn Sie an Stelle der direkten Programmierung des Instruments einen Panel-Treiber verwenden, sparen Sie viel Zeit bei der Entwicklung und der Änderung von Programmen. Änderungen an den Instrumenteinstellungen werden über eine Menüauswahl durchgeführt oder durch eine Bearbeitung von Dialogfeldern. Wenn das Instrument angeschlossen und das Element Live Mode auf ON gesetzt ist, werden die von Ihnen vorgenommenen Änderungen im Instrument übernommen.

Wenn Sie einen Panel-Treiber in einem Programm verwenden möchten, müssen Sie die Ein- und Ausgänge nach Bedarf hinzufügen und den Panel-Treiber mit anderen Objekten verbinden. Sie können mehrere Instanzen eines Treibers in einem Programm verwenden, um das Instrument für verschiedene Statusvarianten zu konfigurieren. In VEE können Sie einen Panel-Treiber als Symbol anzeigen, um Platz zu sparen, oder Sie können die Einstellungen des Instruments in der offenen Ansicht anzeigen. Die Einstellungen können auch während der Ausführung des Programms geändert werden.

### Übung 3-2: Die Einstellungen eines Panel-Treibers ändern

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...** Wählen Sie My Configuration und klicken Sie unter Instrument auf **Add...**, um das Dialogfeld Instrument Properties anzuzeigen. Bearbeiten Sie die Informationen wie folgt:

**Tabelle 13** Panel-Treibereinstellungen

Einstellung	Beschreibung
Name	Ändern Sie den Namen in <b>fgn</b> und drücken Sie zweimal die Tabulatortaste, um zum Feld Address zu wechseln.
Address	Ändern Sie diesen Wert in 713 bzw. in die gewünschte Busadresse.

### 3 Einfache Methoden zum Steuern von Instrumenten

- 2 Klicken Sie auf die Schaltfläche **Advanced**. Setzen Sie auf der Registerkarte General das Element Live Mode auf OFF.
- 3 Klicken Sie auf die Registerkarte **Panel Driver** und wählen Sie für ID Filename: die Datei **hp3325b.cid**. Doppelklicken Sie auf **OK**, um zum Dialogfeld Instrument Manager zurückzukehren.
- 4 Klicken Sie unter Create I/O Object auf **Panel Driver**. Platzieren Sie das Objekt auf der linken Seite des Arbeitsbereichs. (Dieses Verfahren ist für alle Instrumente gleich, falls das Instrument konfiguriert und in der Liste enthalten ist.)

#### HINWEIS

Sie programmieren, ohne dass das Instrument angeschlossen ist. Wäre das Instrument angeschlossen, würden Sie in der Konfiguration die richtige Adresse eintragen.

- 5 Klicken Sie im Feld Function auf **Sine**, um das entsprechende Dialogfeld einzublenden, und wählen Sie aus dieser Liste den Eintrag Triangle (siehe Abbildung 88).

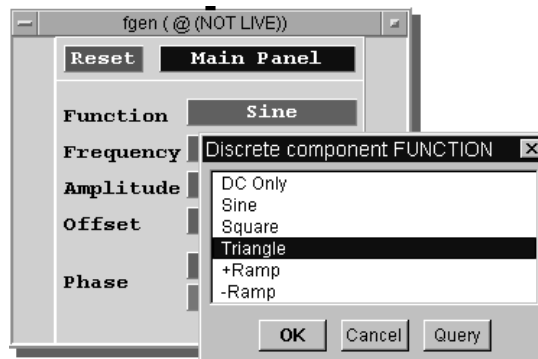


Abbildung 88 Das Einblendmenü Function zu fgen

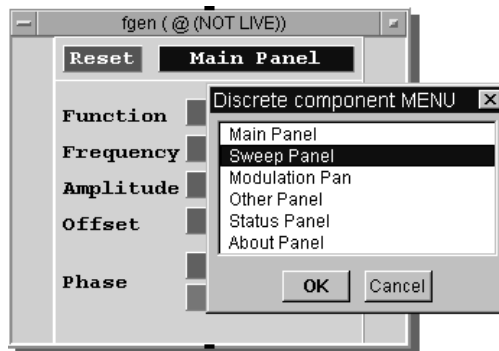
- 6 Klicken Sie auf das Feld rechts von **Frequency**.
- 7 Geben Sie im Dialogfeld **Continuous component FREQUENCY** den Wert 100 ein und klicken Sie auf **OK**. Beachten Sie, dass die Einstellung für Frequency geändert wurde.

Sie können bei allen Treibern die Einstellungen von Instrumenten grundsätzlich auf diese Weise ändern. Wenn das Instrument mit einer Adresse konfiguriert und das Element Live Mode auf ON gesetzt ist, werden alle im Treiberfenster vorgenommenen Änderungen von dem Instrument übernommen.

## Wechseln zu anderen Fenstern des gleichen Treibers

Die meisten Treiber enthalten mehrere Fenster, um die Benutzeroberfläche einfacher zu gestalten. Klicken Sie im Objekt auf **Main Panel**, wenn Sie zu einem anderen Fenster wechseln möchten. Sie rufen so ein Menü mit den verfügbaren Fenstern auf.

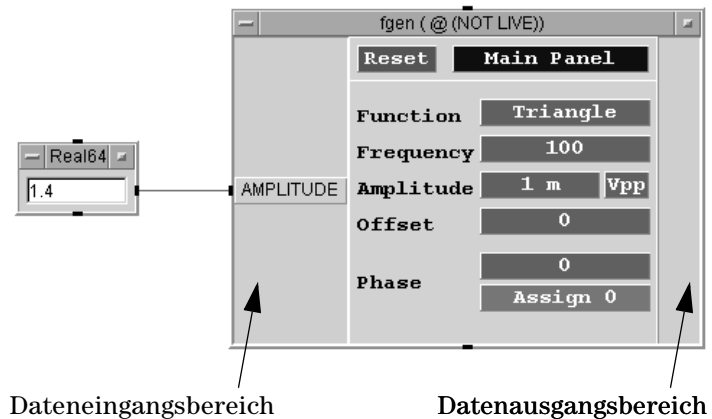
- 1 Klicken Sie im Objekt Panel Driver auf die Schaltfläche **Main Panel** und wählen Sie im Dialogfeld Discrete Component MENU das Element Sweep Panel (siehe Abbildung 89).
- 2 Klicken Sie auf **OK**, um Sweep Panel anzuzeigen. Sie können auch die übrigen Fenster aufrufen, um sich einen Überblick über die verfügbaren Funktionen zu verschaffen.
- 3 Klicken Sie auf **OK**, um zum Main Panel zurückzukehren.



**Abbildung 89** Einblendmenü „Sweep Panel“ im Menü „Discrete Component“

## Einem Panel-Treiber Eingänge und/oder Ausgänge hinzufügen

Zusätzlich zu der direkten Interaktion über das Fenster können Sie die Einstellungen steuern oder Daten von einem Instrument in ein Programm einlesen, indem Sie dem Treiber Datenein- oder -ausgänge hinzufügen. Die Ein- und Ausgangsbereiche sind in Abbildung 90 dargestellt.



**Abbildung 90** Die Datenein- und -Ausgangsbereiche für einen Treiber

- 1 Deuten Sie mit dem Mauszeiger in den Dateneingangsbe-  
reich des Fensters für das Instrument „Funktionsgenerator“,  
und drücken Sie `Strg+A`, um einen Dateneingabeanschluss  
hinzuzufügen. Daraufhin wird eine Liste der Instrumentkom-  
ponenten angezeigt.
- 2 Wählen Sie die gewünschte Komponente in dem angezeigten  
Menü.

### HINWEIS

Sie können auch im Objektmenü **Add Terminal** ⇒ **Data Input** wählen und unter **Select Input Component** das gewünschte Komponentenfeld für den Treiber wählen.

Zum Hinzufügen eines Datenausgangs deuten Sie mit dem Mauszeiger in den Datenausgangsbereich und gehen ansonsten auf die gleiche Weise vor.



## Dateneingangs- oder -ausgangsanschlüsse löschen

Deuten Sie mit dem Mauszeiger auf den Anschluss und drücken Sie **Strg+D**.

### HINWEIS

Sie können auch das Objektmenü öffnen, **Delete Terminal** ⇒ **Input...** wählen und den gewünschten Eingang in dem angezeigten Menü wählen.

---

## Vertiefung

Geben Sie einen Status für den Funktionsgenerator HP 3325B oder einen beliebigen anderen Funktionsgenerator an. Ändern Sie die Einstellung **Function** in eine **Square-Kurve**. Fügen Sie Eingangskomponenten für **Amplitude** und **Frequency** hinzu. Erstellen Sie Eingabe-Dialogfelder für die Amplitude und die Frequenz, und ändern Sie deren Titel in eine Eingabeaufforderung für den Benutzer. Geben Sie verschiedene Werte für die Amplitude und die Frequenz ein, und führen Sie das Programm aus, um festzustellen, ob die Einstellungen durch die Benutzereingaben geändert wurden. (Wenn ein Instrument angeschlossen ist, werden die Einstellungen geändert, sobald **Live Mode** auf **ON** eingesetzt ist.)

## Das Objekt Direct I/O verwenden

Wenn für ein bestimmtes Instrument kein Treiber verfügbar ist oder Sie einen höheren Durchsatz erzielen wollen, verwenden Sie das Objekt „Direct I/O“.

### Übung 3-3: Direct I/O verwenden

In dieser Übung konfigurieren Sie den Funktionsgenerator HP 3325B mit Hilfe des Objekts „Direct I/O“.

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...**
- 2 Wählen Sie den Eintrag **fgn(@(NOT LIVE))** und wählen Sie **Instrument** ⇒ **Properties.**
- 3 Klicken Sie auf die Schaltfläche **Advanced.** Wählen Sie die Registerkarte **Direct I/O**, wie in Abbildung 91 gezeigt. Gewinnen Sie einen Überblick über die verfügbaren Optionen, und klicken Sie auf **OK**, um zu **Instrument Properties** zurückzukehren. Klicken Sie jetzt erneut auf **OK**, um zu **Instrument Manager** zurückzukehren.

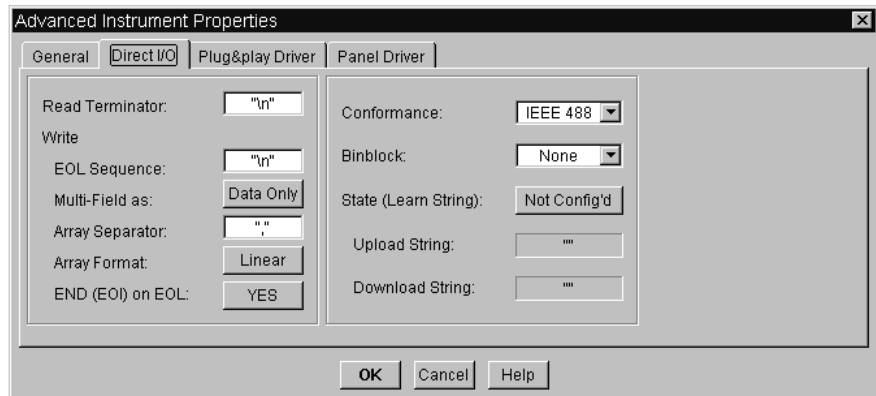
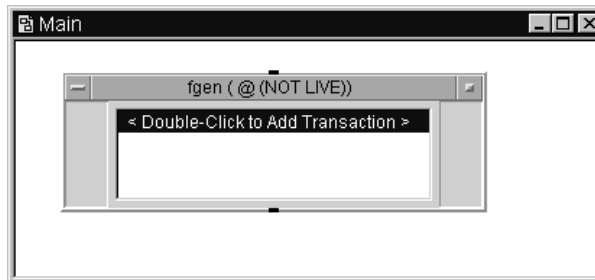


Abbildung 91 Die Registerkarte „Direct I/O“

**HINWEIS**

In diesem Beispiel wird die GPIB-Schnittstelle (IEEE488) verwendet. Informationen zur Konfiguration von seriellen, GPIO- oder VXI-Instrumenten finden Sie im Handbuch *VEE Pro Advanced Techniques*.

- 4 Wenn Sie das Objekt in das Hauptfenster platzieren möchten, vergewissern Sie sich, dass **fgen(@ (NOT LIVE))** noch immer gewählt ist, und klicken Sie auf **Create I/O Object** ⇒ **Direct I/O**. Abbildung 92 zeigt das Objekt „Direct I/O“.



**Abbildung 92** Ein Objekt „Direct I/O“

Zur Verwendung eines Objekts Direct I/O in einem Programm müssen Sie E/A-Transaktionen konfigurieren. Im nächsten Abschnitt werden das Schreiben von Textbefehlen, das Lesen von Daten sowie Upload und Download des Instrumentenstatus beschrieben.

### Einen einzelnen Textbefehl an ein Instrument senden

Geben Sie zum Senden eines einzelnen Textbefehls an ein Instrument die entsprechende Zeichenfolge ein. Die meisten GPIB-Instrumente verwenden alphanumerische Zeichenfolgen für Befehle, die an das Instrument gesendet werden. Wenn Sie beispielsweise an den Funktionsgenerator HP3325B einen Befehl senden wollen, der die Amplitude auf 5 Volt setzt, geben Sie die Befehlszeichenfolge "AM 5 VO" ein.

### 3 Einfache Methoden zum Steuern von Instrumenten

In dieser Übung wird der Funktionsgenerator HP 3325B verwendet, der im vorigen Abschnitt konfiguriert wurde. Gehen Sie ggf. zurück zum Abschnitt „Das Objekt Direct I/O verwenden“ auf Seite 150 und konfigurieren Sie das Instrument, bevor Sie mit dieser Übung fortfahren.

- 1 Doppelklicken Sie im Objekt **fgen(@(NOT LIVE))** auf die Transaktionszeile, um das Dialogfeld **I/O Transaction** aufzurufen (siehe Abbildung 93).

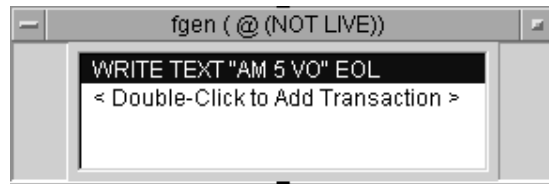


**Abbildung 93** Das Dialogfeld „I/O Transaction“

Mit dem Abwärtspfeil neben WRITE zeigen Sie eine Drop-down-Liste mit Transaktionen an: READ, WRITE, EXECUTE und WAIT. Verwenden Sie zum Schreiben von Daten auf ein Instrument die Standardauswahl WRITE. Informationen zu den einzelnen Aktionen können Sie im Objektmenü unter Help nachschlagen.

- 2 Verwenden Sie die Standardauswahl WRITE, TEXT, DEFAULT FORMAT und EOL ON. Klicken Sie in das Eingabefeld a, geben Sie „**AM 5 VO**“ (einschließlich der Anführungszeichen) ein und klicken Sie auf **OK**.

Jetzt sollte die Transaktion **WRITE TEXT „AM 5 VO“ EOL** wie in Abbildung 94 dargestellt angezeigt werden. Der Text in Anführungszeichen ist der Befehl, der bei der Ausführung des Programms an das Instrument HP3325B gesendet wird.



**Abbildung 94** Eine Transaktion „Direct I/O“

In den meisten Fällen ist dieses Verfahren zum Senden von Befehlen an Instrumente gleich. Es gibt jedoch auch Instrumente, für die am Ende jedes gesendeten Befehls oder jeder gesendeten Befehlsgruppe bestimmte Zeichen angegeben werden müssen. Sie finden diese Informationen in der Dokumentation zu dem jeweiligen Instrument. Geben Sie die entsprechenden Zeichen im Dialogfeld Direct I/O Configuration ein.

## Eine Ausdrucksliste an ein Instrument senden

In manchen Fällen ist es erforderlich, eine Ausdrucksliste an ein Instrument zu senden. Es könnte beispielsweise sein, dass im Funktionsgenerator eine Reihe von Frequenzen durchlaufen werden soll. Dies kann mit einer „Direct I/O“-Transaktion erreicht werden; Sie verwenden dabei in der Ausdrucksliste eine „einen Dateneingang für diese Variable hinzu. In den folgenden Schritten wird beschrieben, wie Sie eine Ausdrucksliste an ein Instrument senden.

- 1 Platzieren Sie ein zweites Objekt Direct I/O für den HP3325B im Hauptfenster. Doppelklicken Sie auf die Transaktionszeile, um das Dialogfeld I/O Transaction aufzurufen.

Sie können alle Standardwerte mit Ausnahme der Befehlszeichenfolge verwenden. Verwenden Sie in diesem Fall das Format „FR“, *<frequency>*, „HZ“. Dies ist eine Ausdrucksliste; die einzelnen Ausdrücke sind durch Kommas abgegrenzt. Die Frequenz wird durch die Variable **A** dargestellt; diese Variable wird als Dateneingang für das Objekt **Direct I/O** verwendet.

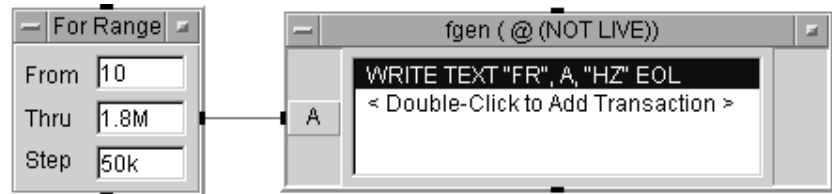
- 2 Klicken Sie in das Eingabefeld für Befehlszeichenfolgen („Command string“) und geben Sie ein: "FR" , A, "HZ" .

### 3 Einfache Methoden zum Steuern von Instrumenten

(Wenn **A** beispielsweise 100 ist, sendet VEE die Zeichenfolge „**FR100HZ**“.) Klicken Sie auf **OK**. Beachten Sie, dass VEE automatisch einen Dateneingangs-Pin mit der Bezeichnung **A** hinzufügt.

- 3 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Range** und platzieren Sie das Objekt links von dem Objekt Direct I/O.
- 4 Verbinden Sie den Datenausgangs-Pin des Objekts For Range mit dem Dateneingangs-Pin von Direct I/O.
- 5 Ändern Sie die Felder in **For Range** wie folgt: From 10, Thru 1.8M und Step 50k.

**For Range** sendet jetzt Zahlen von 10 bis 1,8 Millionen in Intervallschritten zu 50.000. Wenn die Zahlen vom Objekt „Direct I/O“ empfangen werden, bewirkt die Befehlszeichenfolge, dass der Funktionsgenerator die entsprechenden Frequenzen ausgibt. Die Konfiguration des Objekts „Direct I/O“ sollte aussehen wie in Abbildung 95.



**Abbildung 95** Konfiguration von „Direct I/O“ mit einer Eingangsvariablen

- 6 (Optional) Verbinden Sie – falls verfügbar – einen HP3325B mit Ihrem Computer und geben Sie die Adresse des Instruments in die Konfiguration des Objekts „Direct I/O“ ein. Wenn Sie das Programm ausführen, können Sie sehen, wie das Instrument diese Frequenzen generiert.

### Daten von einem Instrument einlesen

Instrumente senden Daten in vielen verschiedenen Formaten an Ihren Computer. Zum Einlesen von Daten aus einem Instrument müssen Sie wissen, um welchen Datentyp es sich handelt

und ob die Daten als Einzelwert (Skalar) oder Array zurückgegeben werden. Außerdem müssen Sie wissen, ob das Instrument Daten als Text (ASCII) oder als Binärdaten sendet.

Sie finden diese Informationen in der Dokumentation des jeweiligen Instruments, oder Sie können mit dem VEE Bus I/O Monitor im Menü I/O die Daten prüfen, die zurückgegeben werden. Diese Informationen legen fest, wie die E/A-Transaktion konfiguriert werden muss.

In diesem Beispiel wird ein HP3478A Multimeter an den in der vorigen Übung beschriebenen Funktionsgenerator HP3325B angeschlossen. Wenn der Generator eine bestimmte Frequenz aussendet, löst der Multimeter ein Lesen aus und sendet das Ergebnis zurück an VEE. Die folgenden Schritte beschreiben, wie die Transaktion für den Multimeter konfiguriert wird.

## HINWEIS

In diesem Beispiel wird eine Transaktion READ TEXT beschrieben. Weitere Auswahloptionen für READ sind BINARY, BINBLOCK und CONTAINER; diese werden im Handbuch *VEE Pro Advanced Techniques* ausführlich beschrieben.

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...** Klicken Sie **Add...** an. Ändern Sie den Namen in `dvm`. Klicken Sie auf **Advanced...** und setzen Sie das Element Live Mode auf OFF. Klicken Sie auf **OK**, um zum Instrument Manager zurückzukehren, falls kein HP3478A angeschlossen ist. (Falls *doch* ein HP3478A angeschlossen ist, ändern Sie die Adresse entsprechend, damit das Instrument die Befehle protokollieren kann.)
- 2 Wählen Sie **dvm(@(NOT LIVE))** und klicken Sie unter Create I/O Object auf **Direct I/O**.
- 3 Doppelklicken Sie auf die Leiste **<Double-Click to Add Transaction>**, um das Dialogfeld I/O Transaction anzuzeigen.
- 4 Markieren Sie das Eingabefeld, geben Sie "T5" ein und klicken Sie auf **OK**. Dadurch wird der Befehl „T“ an das Instrument geschrieben. T5 ist der Befehl für ein einzelnes Auslösen an den Multimeter.

- 5 Öffnen Sie das Objektmenü und klicken Sie auf **Add Trans...** oder doppelklicken Sie auf **<Double-Click to Add Transaction>**, um das Dialogfeld I/O Transaction aufzurufen und eine weitere Transaktionszeile hinzuzufügen.
- 6 Klicken Sie auf den Abwärtspfeil neben WRITE, um die Dropdown-Liste aufzurufen, und wählen Sie READ. Wenn Sie READ wählen, werden neue Schaltflächen im Dialogfeld I/O Transaction angezeigt.
- 7 Überprüfen Sie das Eingabefeld ExpressionList, um sicherzustellen, dass es ein **x** enthält. Drücken Sie Tabulator, um zum nächsten Feld zu springen. Die von einem Instrument zurückgegebenen Daten werden an die Datenausgangs-Pins gesendet. In diesem Fall werden Daten von dem Instrument eingelesen und an einen Datenausgang mit dem Namen **x** gesendet.

#### HINWEIS

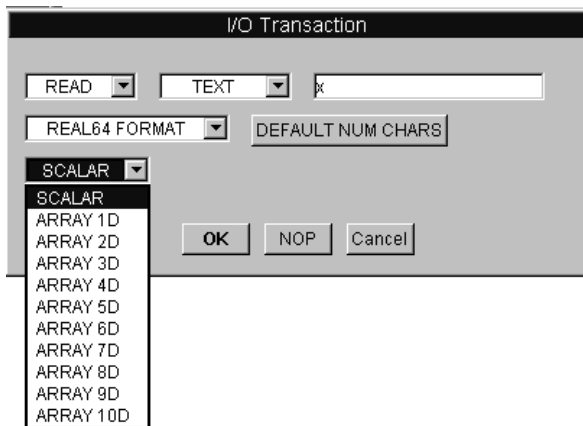
Bei den Namen wird *nicht* zwischen Groß- und Kleinbuchstaben unterschieden.

- 
- 8 Behalten Sie den Standardwert REAL64 FORMAT bei. Der Multimeter gibt einzelne Lesevorgänge als reale Zahlen zurück.
  - 9 Behalten Sie DEFAULT NUM CHARS bei.  
Der Standardwert für die Anzahl der Zeichen beträgt 20. Wenn Sie diese Anzahl ändern möchten, klicken Sie auf DEFAULT NUM CHARS, um auf MAX NUM CHARS umzuschalten, und ändern Sie die Anzahl **20** in den gewünschten Wert.
  - 10 Behalten Sie die Angabe SCALAR bei und klicken Sie auf **OK**.  
Die Transaktion wird in der Leiste als READ TEXT X REAL64 angezeigt. Beachten Sie, dass VEE automatisch einen Datenausgang mit dem Namen **x** hinzufügt.



**HINWEIS**

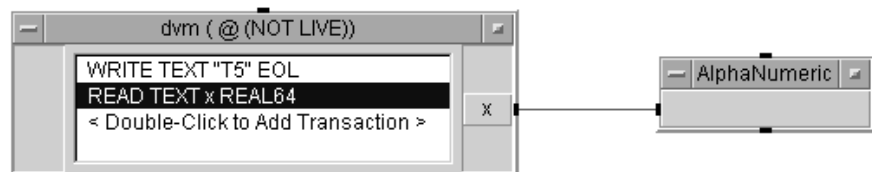
Wenn das Instrument ein Array von Werten zurückgibt, wählen Sie im Dialogfeld I/O Transaction das Menü SCALAR, das Befehle für verschiedene Abmessungen enthält (siehe Abbildung 96). Wenn Sie die Array-Abmessung ausgewählt haben, müssen Sie auch eine Größe für das Array angeben.



**Abbildung 96** Eine READ-Transaktion konfigurieren

**11** Fügen Sie ein Objekt **Display** ⇒ **AlphaNumeric** auf der rechten Seite hinzu und verbinden Sie seinen Eingang mit dem Direct I/O-Ausgang mit der Beschriftung **x**.

Die beiden Transaktionen „Direct I/O“ sollten aussehen wie in Abbildung 97.



**Abbildung 97** Direct I/O konfiguriert zum Einlesen einer Messung

Die Konfiguration einer Transaktion verläuft sehr ähnlich, unabhängig von dem Datenformat für die Transaktion READ TEXT. Sie können auch die weiteren verfügbaren Formate anzeigen. Ausführlichere Informationen zu den einzelnen Elementen finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Zum Erstellen eines vollständigen Testprogramms können dieses Multimeter-Objekt und ein Funktionsgenerator-Objekt mit VEE-Daten- und -Anzeigeobjekten kombiniert werden. Voll funktionsfähige Testprogramme sind in VEE sehr einfach zu erstellen. Es würde jedoch den Rahmen dieses Einführungskapitels sprengen, ausführliche Details zu allen verschiedenen verfügbaren Instrumenten aufzuzeigen. (Weitere komplexere Beispiele hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

## Upload und Download von Statusangaben zu Instrumenten

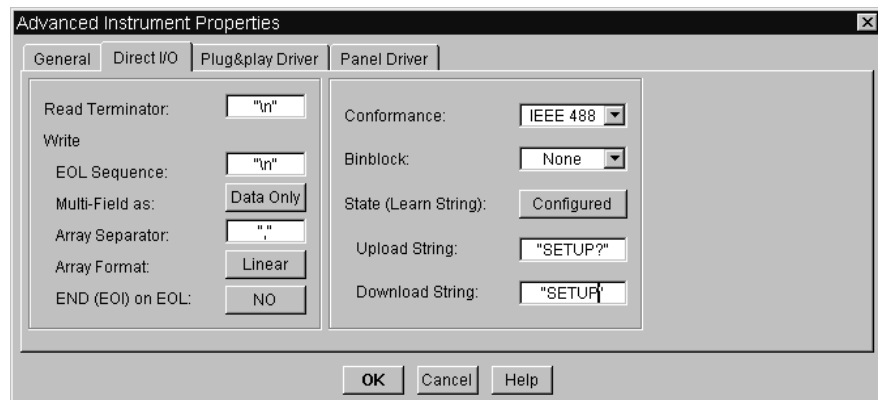
Manche Instrumente bieten eine „Learn-String“-Funktion. Diese Lernzeichenfolge umfasst alle Funktionseinstellungen, aus denen sich der Status eines Instruments zusammensetzt. Die Lernzeichenfolge wird nach einem Upload durch das Objekt „Direct I/O“ mit diesem speziellen „Direct I/O“-Objekt gespeichert; später kann ein Download in das Instrument in dem Programm erfolgen. Führen Sie für den Upload eines Instrumentenstatus die folgenden Schritte aus:

- 1 Setzen Sie manuell den gewünschten Status des Instruments.
- 2 Öffnen Sie das Direct I/O-Objektmenü und klicken Sie auf **Upload State**.

Dieser Status ist jetzt dieser speziellen Instanz des Objekts „Direct I/O“ zugeordnet.

- 3 Öffnen Sie das Dialogfeld I/O Transaction durch Doppelklicken im Transaktionsbereich.
- 4 Klicken Sie auf **TEXT**, wählen Sie STATE (LEARN STRING) und klicken Sie anschließend auf **OK**, um das Dialogfeld I/O Transaction zu schließen. Der zuvor erfasste Status wird an das Instrument gesendet, wenn diese WRITE-Transaktion ausgeführt wird.

Upload und Download werden über die Einstellungen im Dialogfeld „Direct I/O Configuration“ gesteuert. Sofern die Kompatibilität zu IEEE 488.2 gegeben ist, verarbeitet VEE automatisch die Lernzeichenfolgen über die Definition „488.2 \*LRN?“ (ohne Anführungszeichen). Bei einer Kompatibilität zu IEEE 488 gibt Upload String den Befehl zum Abfragen des Status an, und Download String gibt den Befehl an, der beim Herunterladen der Statuszeichenfolge vorangestellt wird. Abbildung 98 zeigt hierzu ein Beispiel.



**Abbildung 98** Lernzeichenfolge-Konfiguration für HP54100A

Die Kompatibilität kann definiert sein für IEEE 488 oder IEEE 488.2. In diesem Beispiel wird das HP 54100A Digitalisierungs-Oszilloskop verwendet, das IEEE 488 entspricht und den Befehl "SETUP?" erfordert, um die Lernzeichenfolge abzufragen; beim Herunterladen muss der Lernzeichenfolge "SETUP" vorangestellt werden. Bei Auswahl von Configured for State (Learn String) werden zwei weitere Felder mit den Bezeichnungen Upload String und Download String eingeblendet. Die entsprechenden Zeichenfolgen wurden in den Eingabefeldern eingetragen.

## PC-Zusatzkarten

VEE bietet drei verschiedene Möglichkeiten zum Steuern von PC-Zusatzkarten:

- 1 Data Translation Visual Programming Interface. (Bestellen Sie die VPI-Anwendung direkt bei Data Translation.)
- 2 Dynamic Link Libraries vom Hersteller der PC-Zusatzkarte, z. B. finden Sie Informationen zur Verwendung von Dynamic Link Libraries.)

### **Data Translation Visual Programming Interface (VPI).**

Data Translation VPI kann zusammen mit VEE für ein reibungsloses Erfassen von Daten über PC-Zusatzkarten verwendet werden. Durch die Flexibilität der Data Translation Open Layers Standards haben Sie Zugriff auf mehr als 50 Datenerfassungskarten.

VPI arbeitet direkt mit Datenerfassungskarten nach ISA-, PCI- und USB-Standard; diese Karten erfordern nur eine geringe Channel-Kapazität. VPI fügt VEE eine Menüauswahl und ein Symbole für die jeweilige PC-Zusatzkarte zur Datenerfassung hinzu. Diese Elemente steuern die Data Translation Hardware-Funktionalität.

### **Amplicon**

Amplicon bietet eine breite Palette von analogen und digitalen E/A-PC-Zusatzkarten der Serie 200; alle diese Karten unterstützen VEE.

Die Software-Schnittstelle ist Teil des Amplicon AmpDIO-Treiberpakets, einer 32-Bit-API mit einer Multithread-DLL für Windows und Unterstützung einer interrupt-gesteuerten Erfassung. Die API umfasst mehr als 100 Aufrufe für eine effiziente und flexible Programmierung als kompilierte Funktion über eine VEE-spezifische Definitionsdatei und bietet die Möglichkeit, bis zu acht Karten in einem Programm zu verwenden.

Zusätzlich zu der Amplicon Produktfamilie von Zusatzkarten, zu der auch serielle Kommunikationsgeräte gehören, kann Amplicon auch Karten von einer Vielzahl anderer Hersteller für die Datenerfassung, die serielle Kommunikation oder GPIB-Anwendungen liefern.

Abbildung 99 zeigt die VEE-Laufzeit-Software (im Lieferumfang der analogen Ausgangskarten PCI224 und PCI234 sowie der analogen Eingangskarten PCI230 und PCI260 von Amplicon enthalten) für gleichzeitige Ein- und Ausgangssignale an einem PC.

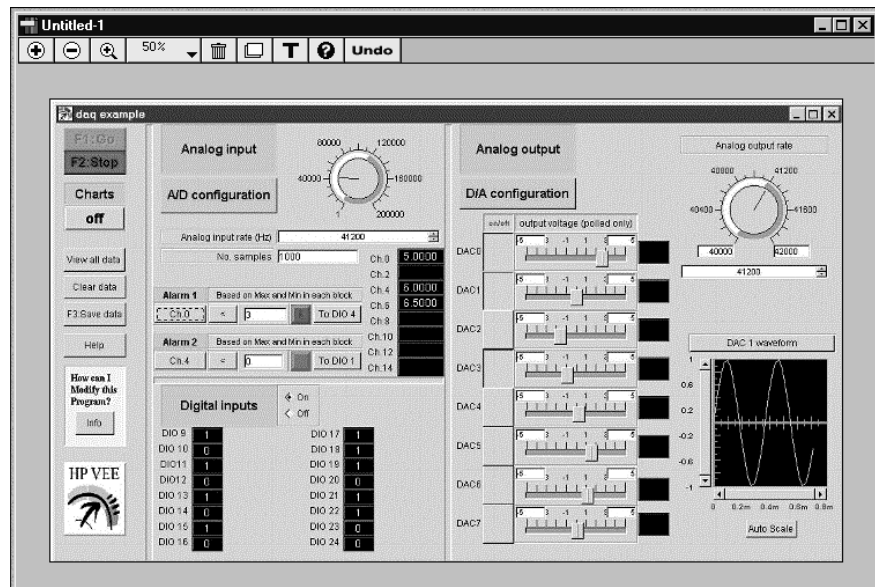


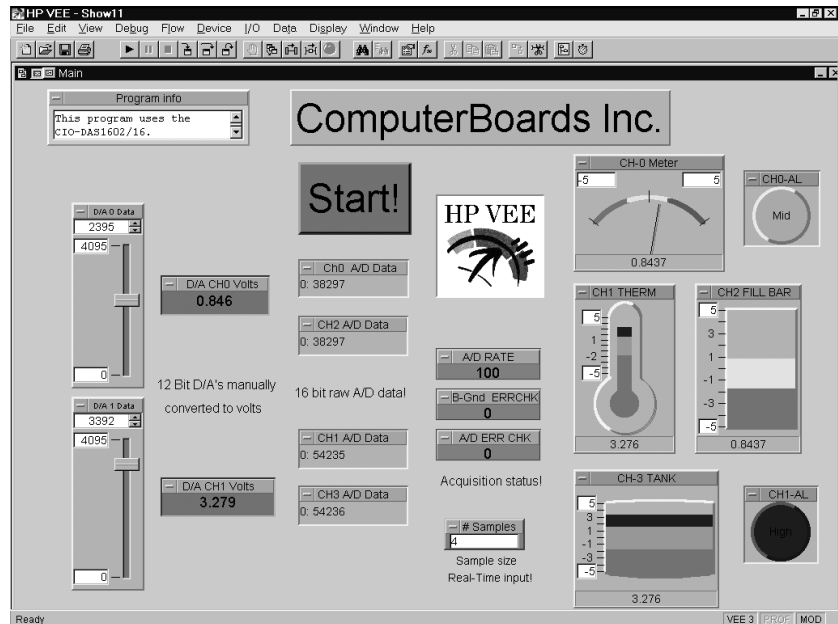
Abbildung 99 Beispiel für Datenerfassung mit Amplicon

## PC-Zusatzkarten von ComputerBoards

ComputerBoards bietet preisgünstige und leistungsstarke PC-Zusatzkarten, die mit VEE kompatibel sind. (Eine vollständige Liste der Hersteller unterstützter PC-Zusatzkarten finden Sie in der VEE-Literatur oder in *VEE Pro Advanced Techniques*.)

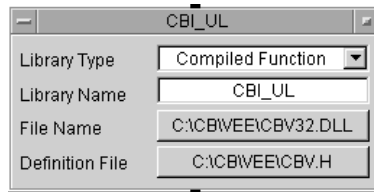
### 3 Einfache Methoden zum Steuern von Instrumenten

Sie müssen nur die Karte und die dazugehörige E/A-Bibliothek installieren und die Karte mit einem vom Hersteller bereitgestellten Programm konfigurieren. Schließen Sie die Karte nach den Anleitungen des Herstellers an das Gerät an. Importieren Sie dann die Bibliothek in VEE. Sie können nun die Messfunktionen aus der E/A-Bibliothek von ComputerBoards aufrufen. Die nachfolgenden Abbildungen zeigen ein vom Hersteller geliefertes Tutorial.



**Abbildung 100** Eine 100 KHz-Karte von ComputerBoards mit VEE verwenden

Abbildung 100 zeigt die Fensteransicht des Tutorialprogramms, das die 100 KHz A/D-Karte verwendet. Abbildung 101 zeigt VEE beim Importieren der ComputerBoards E/A-Bibliothek, die diese Funktionsaufrufe zur Datenerfassung möglich macht.

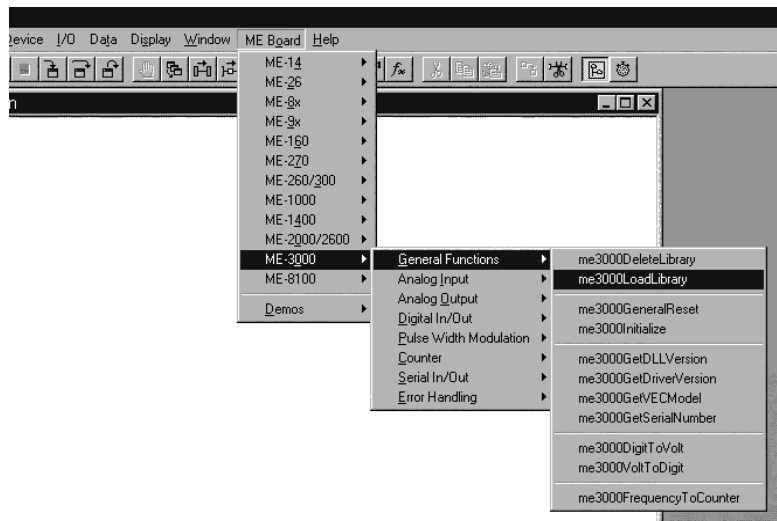


**Abbildung 101** Importieren der ComputerBoards E/A-Bibliothek

## ME-DriverSystem von Meilhaus Electronic

Meilhaus Electronic ist eines der europaweit in Entwicklung, Fertigung und Vertrieb von PC-Datenerfassungs- und -Schnittstellenlösungen führenden Unternehmen. Das ME-DriverSystem für Windows auf CD-ROM ist im Lieferumfang aller von Meilhaus Electronic hergestellten Datenerfassungskarten (d.h. der ME Serie) enthalten. Das ME-DriverSystem ist ebenfalls in die VEE-Menüstruktur integriert.

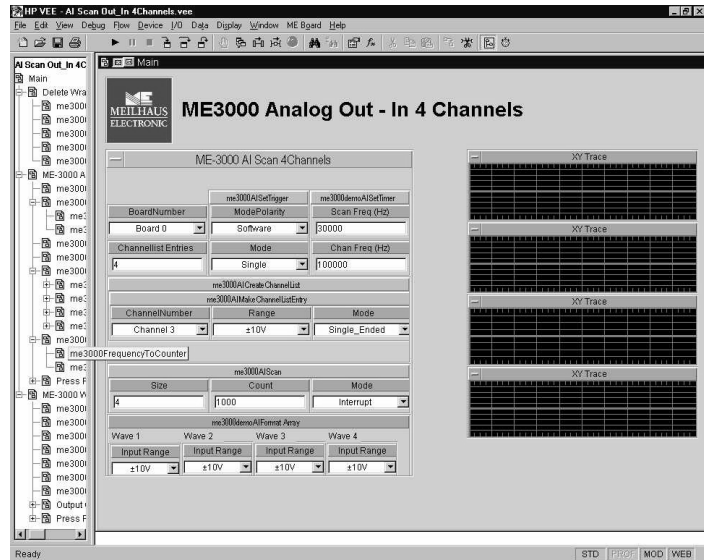
Nach der Installation des ME-DriverSystem für VEE werden die Treiberfunktionen in einem Menü in VEE angezeigt. Abbildung 102 zeigt das Menü „ME Board“ in VEE.



**Abbildung 102** Menü ME Board in VEE

### 3 Einfache Methoden zum Steuern von Instrumenten

Die zweite Menüebene bietet funktionale Gruppen wie den analogen Eingang und Ausgang, digitale E/A und spezielle Funktionen bestimmter Karten. Abbildung 103 zeigt das Benutzerfenster für die Datenerfassungskarte ME-3000.



**Abbildung 103** Benutzerfenster für die Datenerfassungskarte ME-3000

In der dritten Menüebene schließlich befinden sich die eigentlichen Funktionen, z. B. me3000AISingle. Abbildung 104 zeigt das Funktionsfenster.



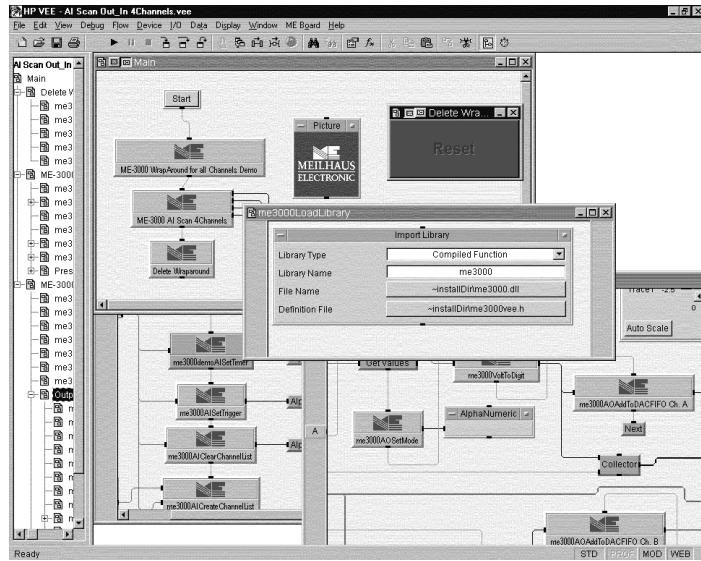


Abbildung 104 Funktionsfenster für ME-DriverSystem

## Einen *VXIPlug&Play*-Treiber verwenden

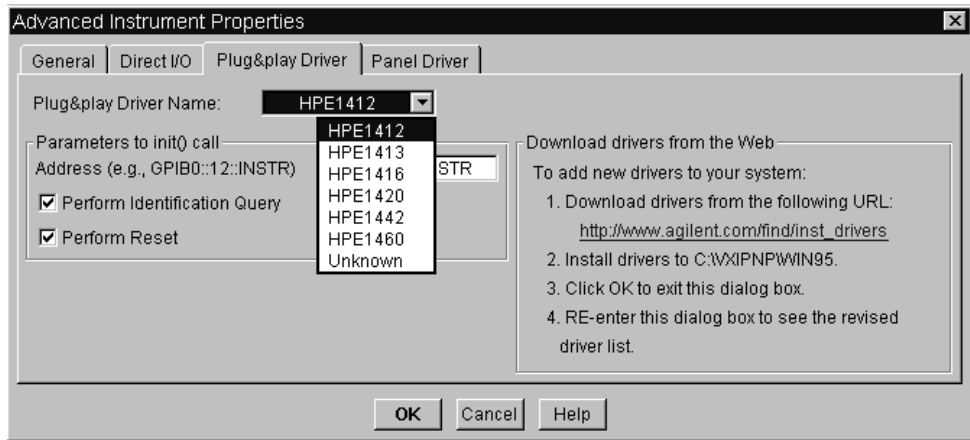
*VXIPlug&Play*-Treiber werden von verschiedenen Instrumentenherstellern geliefert und unterstützt. Hierbei handelt es sich um C-basierende Treiber, die für maximale Leistung und Benutzerfreundlichkeit konzipiert wurden.

Agilent VEE ist vollständig *VXIPlug&Play*-kompatibel. Alle verfügbaren *VXIPlug&Play*-Treiber von Agilent Technologies werden als separate Produkte geliefert und sind auch im Internet unter [http://www.agilent.com/find/inst\\_drivers](http://www.agilent.com/find/inst_drivers) sowie im Agilent Developers Network (ADN) unter <http://www.agilent.com/find/adn> verfügbar. Die gleichen Treiber sind auch in VEE zusammen mit allen Agilent Technologies Panel-Treibern enthalten. *VXIPlug&Play*-Treiber für weitere Instrumente erhalten Sie vom jeweiligen Hersteller des Instruments.

### Übung 3-4: Einen *VXIPlug&Play*-Treiber konfigurieren

Dieses Beispiel beschreibt die Konfiguration eines hpe1412-Treibers.

- 1 Wählen Sie **I/O** ⇒ **Instrument Manager...**
- 2 Wählen Sie **My configuration** und klicken Sie unter **Instrument** auf **Add...**, um das Dialogfeld **Instrument Properties** aufzurufen. Geben Sie einen Namen ein, z. B. **Instrument**, und klicken Sie auf **Advanced...**, um das Dialogfeld **Advanced Instrument Properties** anzuzeigen.
- 3 Setzen Sie im Dialogfeld **Advanced Instrument Properties** das Element **Live Mode** auf **OFF** und wählen Sie die Registerkarte **Plug&play Driver**. Klicken Sie in das Feld **Plug&play Driver Name**, und zeigen Sie so eine Dropdown-Liste an, die alle auf dem Computer installierten Treiber enthält. Dieses Beispiel verwendet den Treiber **HPE1412** (siehe Abbildung 105).



**Abbildung 105** Einen VXIPlug&Play-Treiber wählen

Wählen Sie den Treiber hpe1412, klicken Sie auf **OK**, um zum Dialogfeld Instrument Properties, und dort erneut auf **OK**, um zum Dialogfeld Instrument Manager zurückzukehren. Es sollte jetzt ein Eintrag angezeigt werden für Instrument(@ (NOT LIVE)).

- 4 Wählen Sie Instrument(@ (NOT LIVE)) und unter Create I/O Object den Eintrag Plug&play Driver. Klicken Sie mit der Maustaste, um das Objekt zu platzieren.

## HINWEIS

In VEE ähnelt ein VXIPlug&Play-Treiber einem „Direct I/O“-Objekt.

Wenn Sie mit dem Instrument Messungen durchführen wollen, müssen Sie E/A-Transaktionen konfigurieren, die im VXI-Plug&Play-Treiber C-Funktionen verwenden. Der Treiber bietet Fenster, in denen Sie die gewünschten Funktionen wählen können.

- 5 Doppelklicken Sie auf die Transaktionszeile <Double-click to Add Function> und wählen Sie Select a Function Panel, wie in Abbildung 106 gezeigt. Abbildung 107 zeigt die Hierarchie der Funktionen im Funktionsfenster. Beachten Sie, dass das

### 3 Einfache Methoden zum Steuern von Instrumenten

Hilfethema zu dem ausgewählten Element direkt im Dialogfeld angezeigt wird.

#### HINWEIS

VEE initialisiert das Instrument automatisch. Eine init- Funktion ist nicht wie in anderen Sprachen nötig.

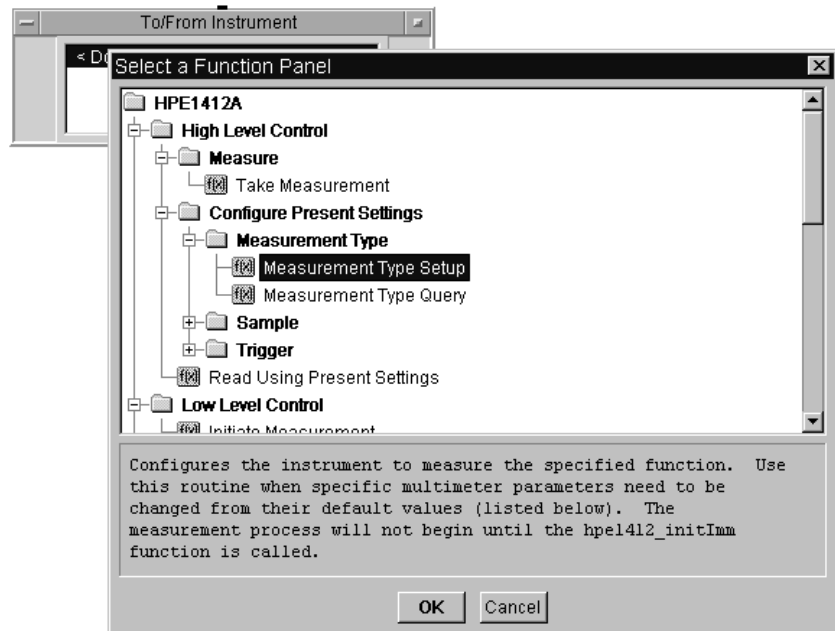
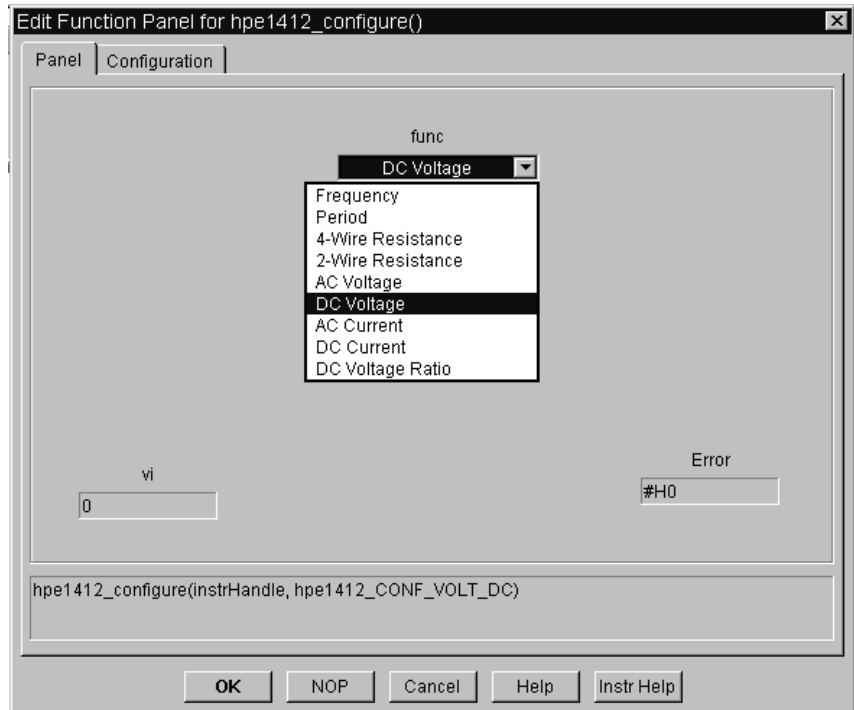


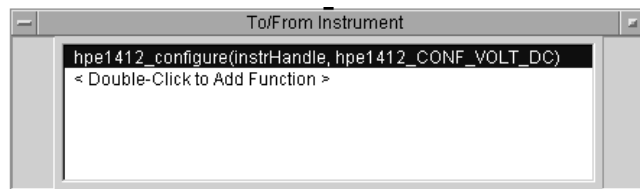
Abbildung 106 Eine Funktion für einen VXIPlug&Play-Treiber wählen

- Wählen Sie **Configure Present Settings** ⇒ **Measurement Type** ⇒ **Measurement Type Setup**. Das Dialogfeld **Edit Function** wird angezeigt. Klicken Sie in das Eingabefeld unter **func**, um eine Dropdown-Liste anzuzeigen. Wählen Sie **DC Voltage** (Gleichspannung) als Standardwert (siehe Abbildung 107).



**Abbildung 107** Das Dialogfeld „hpe1412 Edit Function“

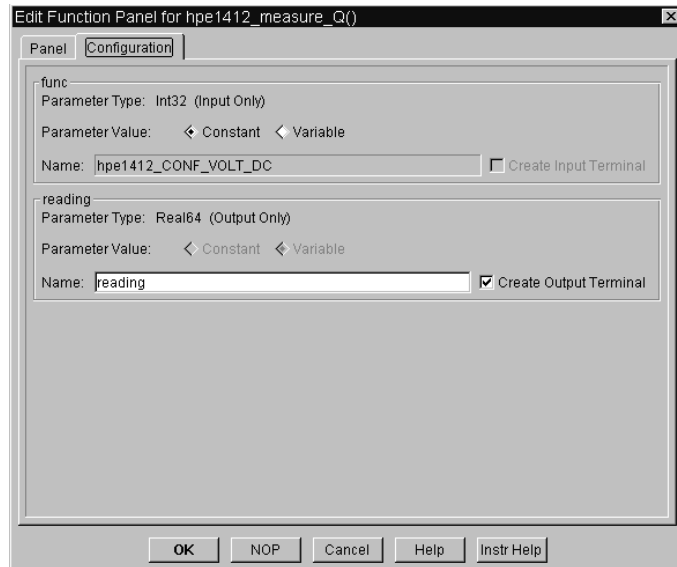
- 7** Klicken Sie auf **OK**. Das Objekt To/From Instrument enthält jetzt den Eintrag **hpe1412\_configure(instrHandle,hpe1412\_CONF\_VOLT\_DC)**, wie in Abbildung 108 gezeigt.



**Abbildung 108** Funktion „DC Voltage“ in VXIPlug&Play-Objekt

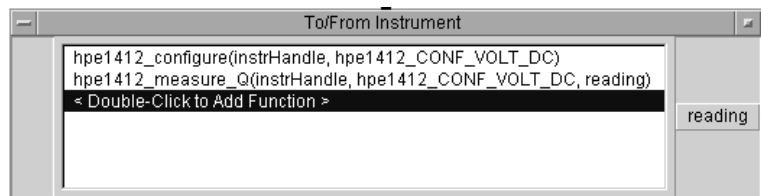
### 3 Einfache Methoden zum Steuern von Instrumenten

- 8 Doppelklicken Sie in das Objekt To/From Instrument, um eine Funktion hinzuzufügen, und wählen Sie unter Measure den Eintrag Take Measurement. Klicken Sie auf die Registerkarte Configuration; das Dialogfeld entspricht nun Abbildung 109.



**Abbildung 109** Registerkarte Configuration im Dialogfeld „Edit Function Panel“

- 9 Klicken Sie auf **OK**. Im Objekt To/From Instrument wird ein zweiter Funktionsaufruf aufgelistet (siehe Abbildung 110).



**Abbildung 110** HPE1412-Treiber bereit für Gleichstrommessung

## Weitere E/A-Funktionen

- 1 Lernen Sie die ganze Leistung der VEE-E/A-Funktionen im Untermenü **I/O** ⇒ **Advanced I/O** kennen: Interface Operations, Instrument Event, Interface Event und MultiInstrument Direct I/O.
- 2 Sie können die Busaktivität über das Menü I/O mit „Bus I/O Monitor“ anzeigen, drucken oder zur Fehlerbehebung speichern.
- 3 VEE enthält einen ActiveX-Automatisierungs-Server zur programmgesteuerten Suche von Instrumenten. Zusätzliche Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)
- 4 Sie können die E/A-Konfiguration auch während der Ausführung programmgesteuert ändern. Zusätzliche Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

## Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Die Vorteile von Instrumententreibern und des Objekts „Direct I/O“ erläutern.
- Den Prozess zur Steuerung von Instrumenten erläutern.
- Ein Instrument für einen Treiber konfigurieren.
- Ein Instrument für das Objekt „Direct I/O“ konfigurieren.
- Die Einstellungen an einem Instrumententreiber ändern.
- Komponentenein- und -ausgänge hinzufügen und löschen.
- Zwischen den verschiedenen Fenstern eines Treibers wechseln.
- Befehle an ein Instrument mit dem Objekt „Direct I/O“ schreiben.
- Daten über das Objekt "Direct I/O" aus einem Instrument lesen.
- Upload und Download des Instrumentenstatus über Lernzeichenfolgen.
- *VXIPlug&Play*-Treiber zur Kommunikation mit einem Instrument verwenden.
- Die zwei Methoden zum Steuern von PC-Zusatzkarten erläutern.



# 4

## Testdaten analysieren und anzeigen

Überblick	175
Agilent VEE: Datenformen und Datentypen	176
Agilent VEE-Analysefunktionen	179
Integrierte mathematische Objekte verwenden	180
Ausdrücke mit dem Objekt Formula erstellen	184
MATLAB Script in Agilent VEE verwenden	190
Anzeigen von Testdaten	198
Testdatenanzeigen anpassen	200
Kapitel-Checkliste	205

### Testdaten analysieren und anzeigen

*Dieses Kapitel enthält Informationen zu folgenden Themen:*

- VEE-Datentypen
- VEE-Analysefunktionen
- Mathematische Objekte verwenden
- Formelobjekte verwenden
- MATLAB Script-Objekte verwenden
- VEE-Anzeigefunktionen
- Anzeigen anpassen

*Erforderliche Zeit für dieses Kapitel: 1,5 Stunden*

## Überblick

In diesem Kapitel erfahren Sie mehr über die Analyse- und Anzeigefunktionen von VEE. Sie lernen, wie Sie die richtigen mathematischen Objekte für Ihre Anwendungen finden und wie Sie die Testergebnisse anzeigen, damit Sie den Daten rasch und bequem die relevanten Informationen entnehmen können.

Mithilfe der ActiveX-Automatisierung können Sie auch andere gängige Anwendungen wie MS Excel zum Analysieren der Daten einsetzen. (Weitere Informationen hierzu finden Sie in Kapitel, „Berichte mit ActiveX bequem erstellen“ auf Seite 253.) Mit ActiveX-Steuer-elementen können Sie auch VEE-externe Anzeigefunktionen verwenden. (Weitere Informationen hierzu finden Sie im Abschnitt „Übung 11-4: Ein ActiveX-Steuer-element verwenden“ auf Seite 453). Der Schwerpunkt dieses Kapitels liegt auf den VEE-spezifischen Tools und dem Objekt „MATLAB Script“, das im Lieferumfang von VEE enthalten ist.

### Agilent VEE: Datenformen und Datentypen

In einem VEE-Programm werden Daten entlang der Linien zwischen Objekten übertragen und von den nachfolgenden Objekten verarbeitet. Ein Container enthält in VEE zusammengehörige Datensätze; ein Container hat eine *Datenform* (*scalar* oder *array*) und einen *Datentyp* (z. B. Int32, Real64 oder Text).

**Datenform:** Ein Skalar ist eine einzelne Zahl einschließlich komplexer Zahlen, die aus zwei oder mehr Komponenten bestehen. Ein Array enthält eine Gruppe von Datenelementen, die als eindimensionales Array (Array 1D), zweidimensionales Array (Array 2D) etc. angegeben werden können.

**Datentypen:** Die VEE-Datentypen sind in Tabelle 14 beschrieben.

Im Allgemeinen müssen Sie sich mit Datentypen oder -formen nicht befassen, da die meisten Objekte mit allen VEE-Datentypen arbeiten und Daten automatisch in den für das Objekt erforderlichen Typ konvertieren. Wenn beispielsweise die Anzeige eines Größenspektrums einen Wellenform-Datentyp empfängt, führt VEE automatisch eine Fast Fourier Transformation durch, um die Daten aus dem Zeitbereich in den Frequenzbereich zu konvertieren.

Gelegentlich benötigt ein Objekt jedoch einen bestimmten Datentyp; sie sollten daher problemlos mit verschiedenen Datentypen umgehen können. Außerdem wichtig sind die Unterschiede zwischen den von VEE und MATLAB unterstützten Datentypen. Weitere Informationen hierzu finden Sie im Abschnitt „Mit Datentypen arbeiten“ auf Seite 195.

Die folgende Tabelle enthält einen kurzen Überblick über die VEE-Datentypen. Detaillierte Angaben zu diesen Datentypen finden Sie in den nachfolgenden Kapiteln.

**Tabelle 14** Agilent VEE-Datentypen

<b>Datentyp</b>	<b>Beschreibung</b>
UInt8	Unsigned Byte 0 bis 255.
Int16	Ein ganzzahliges 16-Bit-Zweierkomplement (-32768 bis 32767).
Int32	Ein ganzzahliges 32-Bit-Zweierkomplement (-2147483648 bis 2147483647).
Real32	Eine 32-Bit-Gleitkommazahl entsprechend dem Standard IEEE 754 (+/-3.40282347E+/-38).
Real64	Eine 64-Bit-Gleitkommazahl entsprechend dem Standard IEEE 754 (+/- 1.797693138623157 E308).
PComplex	Eine Größen- und Phasenkomponente in der Form (mag, @phase). Die Phase ist standardmäßig auf degrees (Grad) gesetzt, kann jedoch mit der Einstellung <b>File</b> ⇒ <b>Default Preferences</b> ⇒ <b>Trig Mode</b> auf radians (Radiant) oder gradians (Gradient) gesetzt werden.
Complex	Eine rechtwinklige oder kartesische komplexe Zahl mit einer realen und einer imaginären Komponente in der Form (real, imag). Jede Komponente entspricht einem Real64. Die komplexe Zahl $1 + 2i$ wird beispielsweise als (1,2) dargestellt.
Waveform	Ein zusammengesetzter Datentyp aus Zeitbereichswerten, der die Real64-Werte gleichmäßig verteilter, linearer Punkte und die gesamte Zeitspanne der Wellenform enthält. Die Datenform einer Waveform muss ein eindimensionales Array (Array 1D) sein.
Spectrum	Ein zusammengesetzter Datentyp aus Frequenzbereichswerten, der die PComplex-Werte von Punkten sowie die Minimum- und Maximumwerte der Frequenz enthält. Die Bereichsdaten können als Aufzeichnung oder linear zugeordnet werden. Die Datenform eines Spectrum muss ein eindimensionales Array (Array 1D) sein.

**Tabelle 14** Agilent VEE-Datentypen

Datentyp	Beschreibung
Coord	Ein zusammengesetzter Datentyp, der mindestens zwei Komponenten in der Form (x,y,...) enthält. Jede Komponente entspricht einem Real64. Die Datenform des Datentyps Coord muss ein Scalar oder ein Array 1D sein.
Enum	Eine Textzeichenfolge mit einem zugeordneten ganzzahligen Wert. Sie können auf die Ganzzahl mit der Funktion ordinal(x) zugreifen.
Text	Eine Folge aus alphanumerischen Zeichen.
Record	Ein zusammengesetzter Datentyp mit einem Feld für jeden Datentyp. Jedes Feld hat einen Namen und einen Container, der einen beliebigen Typ und eine beliebige Form (einschließlich Record) haben kann.
Object	Nur für ActiveX-Automatisierungs- und -Steuerelemente, eine Referenz auf ein ActiveX-Steuerelement oder eine von einem Automatisierungs-Aufruf zurückgegebene Referenz. Genau genommen ist dies eine Referenz auf eine IDispatch- oder eine IUnknown-Schnittstelle.
Variant	Nur für ActiveX-Automatisierung und -Steuerelemente. Ein Datentyp, der für manche ActiveX-Methodenaufrufe als Parametertyp „By Ref“ erforderlich ist.

### HINWEIS

Informationen zur gemeinsamen Nutzung von Daten in Mischumgebungen finden Sie unter **I/O** ⇒ **To/From Socket**.

## Agilent VEE-Analysefunktionen

VEE unterstützt allgemeine mathematische Operationen und Hunderte von weiteren Funktionen. Darüber hinaus enthält VEE auch die Funktion „MATLAB Script“. Die Funktion „MATLAB Script“ ist eine Untermenge des Standard-Vollprodukts MATLAB von The MathWorks. Sie bietet mathematische Funktionen in VEE einschließlich Signalverarbeitung, höherer Mathematik, Datenanalyse sowie wissenschaftlicher und Engineering-Grafik. Die MATLAB Script-Funktion ist mit VEE vollständig integriert; MATLAB Script-Objekte können in jedes VEE-Programm eingefügt werden.

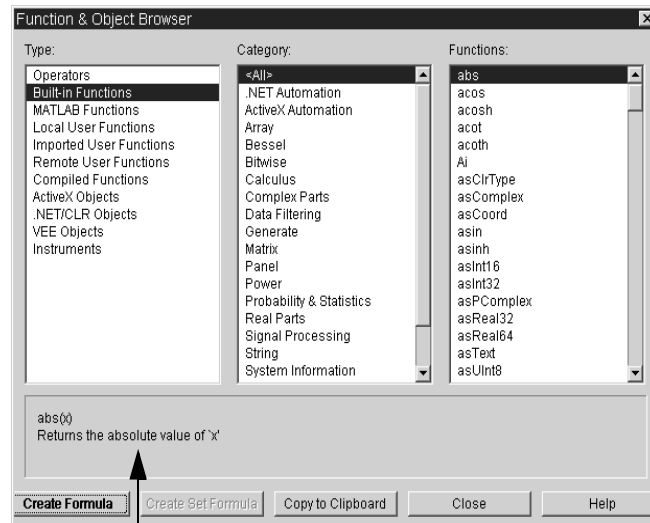
Wenn die erforderliche mathematische Funktion weder in VEE noch in MATLAB enthalten ist, stehen Ihnen verschiedene Möglichkeiten zur Verfügung. Sie können die Funktion mit dem Objekt Formula erstellen; dieses Objekt wird weiter hinten in diesem Kapitel beschrieben. Die Funktion kann auch in einer Compilersprache wie beispielsweise C geschrieben und mit VEE verknüpft werden, oder Sie können von VEE aus mit einer anderen Software-Anwendung kommunizieren.

# Integrierte mathematische Objekte verwenden

Unter VEE **Device** ⇒ **Function & Object Browser** können Sie auf integrierte (vorprogrammierte) mathematische Ausdrücke für VEE und MATLAB zugreifen.

## Integrierte Operatoren oder Funktionen aufrufen

Wählen Sie zum Aufrufen mathematischer VEE-Operatoren und Funktionen **Device** ⇒ **Function & Object Browser**. Wenn Sie beispielsweise eine Formel erstellen möchten, die eine Zufallszahl in einem angegebenen Bereich zurückgibt, wählen Sie **Type: Built-in Functions**, **Category: Probability & Statistics** und **Functions: random** (siehe Abbildung 111).



Beschreibung  
der Funktion

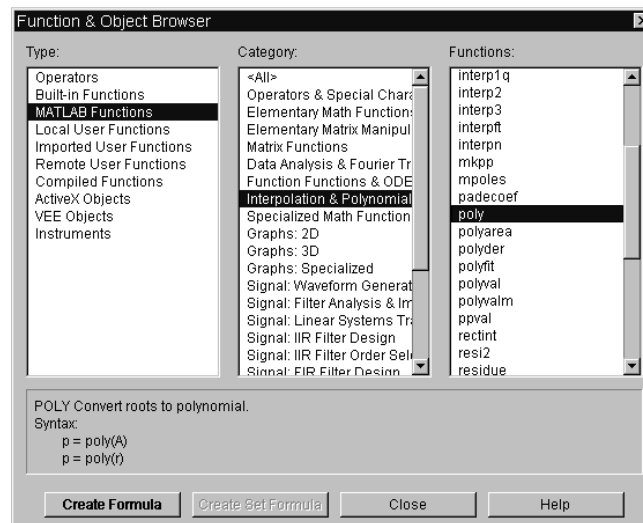
Hilfethema zur  
aktuellen Auswahl

**Abbildung 111** Eine VEE-Funktion im Function & Object Browser



In Abbildung 111 sehen Sie, dass im Dialogfeld Function & Object Browser eine kurze Beschreibung der aktuellen Auswahl angezeigt wird. Sie können auch auf die Schaltfläche Help klicken, um eine ausführlichere Beschreibung der aktuellen Auswahl und Informationen über Definition, Verwendung und Syntax sowie Beispiele aufzurufen.

Wählen Sie zum Aufrufen von MATLAB-Operatoren und -Funktionen **Device** ⇒ **Function & Object Browser** und unter Type die Option MATLAB Functions. Um beispielsweise Wurzeln in Polynome umzuwandeln, wählen Sie Type: MATLAB Functions, Category: Interpolation & Polynomials, und Functions: poly (siehe Abbildung 112).



**Abbildung 112** Eine MATLAB-Funktion in Function & Object Browser

Auch hier wird im Dialogfeld Function & Object Browser eine kurze Beschreibung der aktuellen Auswahl angezeigt; durch Klicken auf Help können Sie weitere Informationen aufrufen. Die „MATLAB Runtime Engine and Script“ wird im Abschnitt „MATLAB Script in Agilent VEE verwenden“ auf Seite 190 ausführlich beschrieben.

## Übung 4-1: Die Standardabweichung berechnen

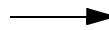
Generieren Sie eine Kosinus-Wellenform mit einer Frequenz von 1 kHz, einer Amplitude von 1 V und einem Zeitbereich von 20 ms, dargestellt durch 256 Punkte. Berechnen Sie die Standardabweichung und zeigen Sie diese an.

- 5 Wählen Sie **Device**  $\Rightarrow$  **Virtual Source**  $\Rightarrow$  **Function Generator**. Geben Sie für Frequency einen entsprechenden Wert ein und minimieren Sie die offene Ansicht.
- 6 Wählen Sie **Device**  $\Rightarrow$  **Function & Object Browser** und in diesem Dialogfeld Built-in Functions, Probability & Statistics und sdev. Klicken Sie auf die Schaltfläche **Create Formula**.

### HINWEIS

Sie können direkt zum Dialogfeld „Function & Object Browser“ wechseln, indem Sie in der Symbolleiste auf die Schaltfläche **fx** klicken (siehe Abbildung 113) oder die Tastenkombination `Strg+I` drücken.

Die Schaltfläche  
„Function and Object Browser“



**Abbildung 113** Die Schaltfläche „fx“ zum Öffnen des Function and Object Browser

- 7 Öffnen Sie das Objektmenü des Objekts `sdev()`, um mit dem Befehl `Help` weitere Informationen anzuzeigen.

### HINWEIS

Das Objekt `sdev(x)` ist definiert als die Quadratwurzel der Varianz von `x`, wobei `x` den Datentyp `UInt8`, `Int16`, `Int32`, `Real32`, `Real64`, `Coord` oder `Waveform` haben kann. Der Funktionsgenerator gibt den Datentyp `Waveform` aus.

- 8 Verbinden Sie das Objekt `Function Generator` mit `sdev(x)`.
- 9 Wählen Sie **Display**  $\Rightarrow$  **AlphaNumeric** und verbinden Sie das Objekt mit dem Datenausgangs-Pin von `sdev(x)`.

10 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 114.



Abbildung 114 Die Standardabweichung berechnen

### Ausdrücke mit dem Objekt Formula erstellen

Mit dem Objekt Formula können mathematische Ausdrücke in VEE geschrieben werden. Die Variablen eines Ausdrucks sind die Namen der Dateneingangs-Pins oder globale Variablen. Das Ergebnis der Auswertung eines Ausdrucks wird am Datenausgangs-Pin angelegt.

Abbildung 115 zeigt das Objekt Formula. Das Feld für die Eingabe von Ausdrücken befindet sich in der Mitte des Objekts. Standardmäßig enthält dieses Feld den Ausdruck  $(2*A+3)$ . Doppelklicken Sie in das Feld, wenn Sie einen anderen Ausdruck eingeben möchten.

#### HINWEIS

Sie können den Ausdruck einer Formel auch über mehrere Zeilen eingeben. Wenn ein Ausdruck einen Zeilenumbruch enthält, wird er als mehrzeiliger Einzelausdruck interpretiert. Wenn eine Formel durch Semikolons (;) getrennte Anweisungen enthält, werden diese Anweisungen als mehrere Ausdrücke in der Formel interpretiert.

Das Dialogfeld Formula unterstützt Rich Text Format. Sie können also in einer Formel die standardmäßigen Textbearbeitungsbefehle anwenden. Sie können beispielsweise die Maus ziehen, um Zeichen zu markieren, diese Zeichen mit `Strg+C` kopieren, mit `Strg+V` einfügen oder mit `Strg+X` löschen und die Tasten Ende, Einfg, Entf, Rücktaste etc. in gewohnter Weise verwenden.

Eingabefeld

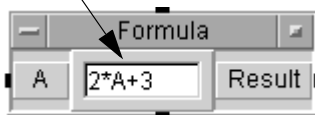


Abbildung 115 Das Objekt Formula

**HINWEIS**

Alle Funktionen, die über **Devices**  $\Rightarrow$  **Function & Object Browser** unter Type:Built-in Functions erstellt wurden, sind einfach Formeln, deren Ausdrücke bereits richtig gesetzt wurden. Sie können auch bearbeitet werden, um Funktionen zu kombinieren und Eingänge hinzuzufügen oder zu löschen. Sie können in einem Formelobjekt auch mehrzeilige Einträge vornehmen und den Ausgangsanschlüssen Werte zuweisen.

**Einen Ausdruck mit dem Objekt Formula auswerten**

In diesem Beispiel werten Sie den Ausdruck  $2 * A^6 - B$  aus, wobei  $A=2$  und  $B=1$ . (Beachten Sie das Zeichen  $^$  für die Potenzierung.)

**HINWEIS**

Bei den Variablennamen wird *nicht* zwischen Groß- und Kleinschreibung unterschieden.

- 1 Wählen Sie **Device**  $\Rightarrow$  **Formula**. Klicken Sie in das Eingabefeld des Objekts Formula und geben Sie  $2 * A^6 - B$  ein.
- 2 Positionieren Sie den Mauszeiger auf den Dateieingangsbereich (aber nicht direkt auf den Eingang **A** ) und drücken Sie **Strg+A**. Sie fügen so ein Eingangs-Pin hinzu.

**HINWEIS**

Dieses Pin erhält standardmäßig die Bezeichnung **B**, kann jedoch auch umbenannt werden.

- 3 Wählen Sie **Data**  $\Rightarrow$  **Constant**  $\Rightarrow$  **Int32**. Klonen Sie das Objekt mit dem Befehl Clone (im Objektmenü) und verbinden Sie die beiden Int32-Objekte mit den Eingängen **A** und **B** des Objekts Formula.
- 4 Geben Sie im Objekt **A Int32** den Wert 2 und im Objekt **B Int32** den Wert 1 ein.
- 5 Wählen Sie **Display**  $\Rightarrow$  **AlphaNumeric**, verbinden Sie dieses Objekt mit dem Ausgang des Objekts Formula und führen Sie das Programm aus. Das Ergebnis sollte, wie in Abbildung 116 abgebildet, 127 betragen.

## 4 Testdaten analysieren und anzeigen

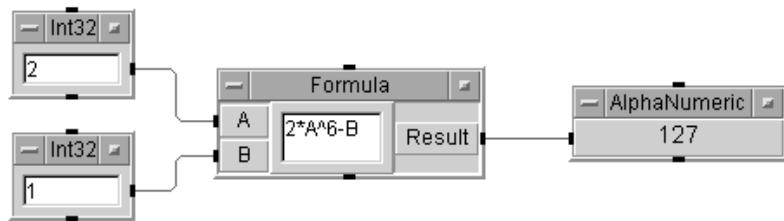


Abbildung 116 Einen Ausdruck auswerten

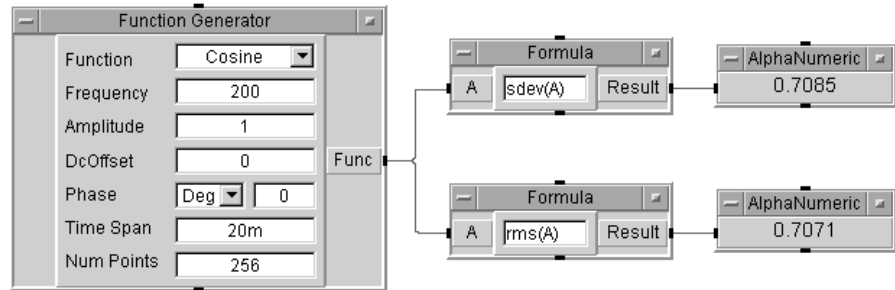
### Eine Agilent VEE-Funktion im Objekt Formula verwenden

In dieser Übung wird eine Kosinuskurve generiert; mit einem Formelobjekt werden die Standardabweichung und der quadratische Mittelwert berechnet.

- 1 Erstellen Sie im Arbeitsbereich je ein Objekt Function Generator, Formula und AlphaNumeric und verbinden Sie sie mit ihren Daten-Pins.
- 2 Klonen Sie das Objekt Formula mit dem Befehl Clone (im Objektmenü) und positionieren Sie dieses zweite Objekt direkt unter dem ersten. Verbinden Sie das Datenausgangsp-Pin des Function Generator auch mit dem zweiten Objekt Formula.
- 3 Klonen Sie eine weitere AlphaNumeric-Anzeige und verbinden Sie sie mit dem zweiten Formula-Objekt.
- 4 Geben Sie im ersten Formula-Objekt `sdev(A)` und im zweiten Formula-Objekt `rms(A)` ein.

**sdev(A)** und **rms(A)** sind mathematische Funktionen aus dem Dialogfeld **Device** ⇒ **Function & Object Browser**. Beachten Sie, dass sie als Funktionen oder unabhängige Objekte aufgerufen werden können; sie werden in beiden Fällen auf die gleiche Weise ausgeführt.

- 5 Führen Sie das Programm aus. Das Ergebnis wird nicht dadurch beeinflusst, ob die Funktionen in Formelobjekte eingefügt oder als unabhängige Objekte hinzugefügt werden (siehe Abbildung 117).



**Abbildung 117** Formelbeispiele mit VEE-Funktionen

Berechnen Sie nun die Standardabweichung und den quadratischen Mittelwert mit nur einem Formelobjekt. Einem Formelobjekt können mehrere Ausgangsanschlüsse mit Werten zugeordnet sein.

- 6 Doppelklicken Sie auf das Objektmenü eines der Formula-Objekte, um es zu löschen.
- 7 Ändern Sie in dem verbleibenden Formula-Objekt den Ausdruck in
 
$$B=sdev(A) ;$$

$$C=rms(A)$$

## HINWEIS

Wenn ein Formelobjekt mehrere Ausdrücke enthält, müssen Sie ein Semikolon am Ende eines Ausdrucks einfügen, um den Ausdruck von dem nächsten abzugrenzen. In der Formel **B=sdev(A);** kennzeichnet das Semikolon beispielsweise das Ende des Ausdrucks.

## 4 Testdaten analysieren und anzeigen

### HINWEIS

Sie können an jeder beliebigen Stelle in einem Formelobjekt *Zeilenumbrüche* einfügen. Die Formel wird als ein einziger Ausdruck gelesen, wenn keine Semikolons darin enthalten sind. Sie können beispielsweise einen einzelnen Ausdruck eingeben als

```
B=sdev  
(A)
```

Sie können auch Leerzeichen in eine Formel einfügen, um die Lesbarkeit zu erhöhen.

- 8 Fügen Sie in dem Formula-Objekt einen Ausgangsanschluss hinzu. Benennen Sie die Ausgangsanschlüsse um in **B** und **C**. Verbinden Sie den Ausgangsanschluss **B** mit dem einen und den Ausgangsanschluss **C** mit dem anderen Alphanumeric-Objekt.
- 9 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 118.

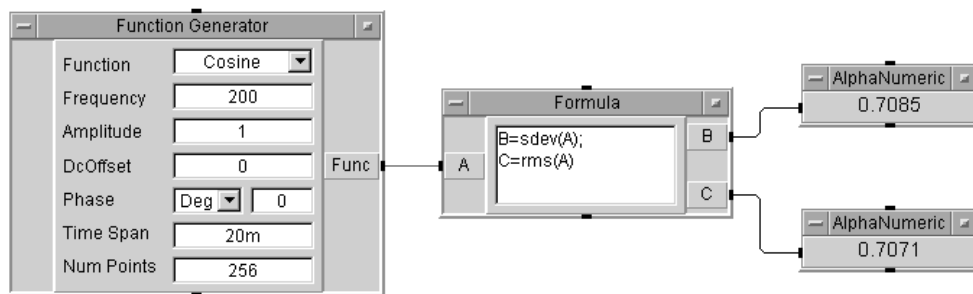


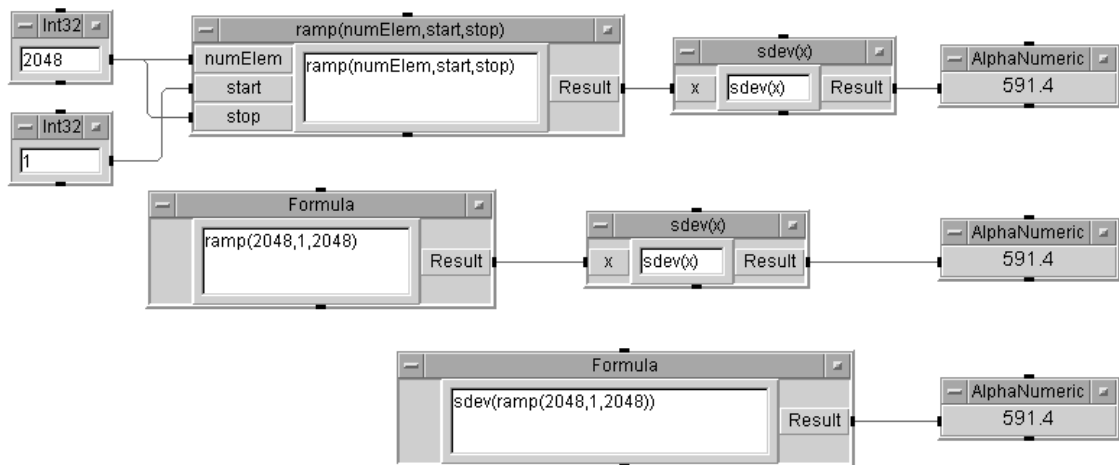
Abbildung 118 VEE-Funktionen mit einem Formelobjekt



## Vertiefung

Führen Sie die folgende Übung durch und überprüfen Sie die Ergebnisse (siehe Abbildung 119).

- 1 Erstellen Sie mit dem Objekt ramp (Type: Built-in Functions, Category: Generate) ein Array mit Zahlen von 1 bis 2048. Berechnen Sie die Standardabweichung dieses Arrays und zeigen Sie sie an.
- 2 Führen Sie die gleiche Übung durch wie im vorigen Schritt; verwenden Sie dabei nicht das Objekt ramp, sondern die Funktion ramp() in einem Formelobjekt.
- 3 Führen Sie nun die Übung mit verschachtelten Funktionen durch. Verwenden Sie nur zwei Objekte.



**Abbildung 119** Lösung zu den Vertiefungs-Übungen: Ramp und SDEV

Für die zweite und dritte Übung müssen Sie den Eingangsanschluss **A** am **Formula**-Objekt löschen, um eine Fehlermeldung zu vermeiden, da alle Dateneingangs-Pins angeschlossen sein und Daten aufweisen müssen, bevor ein Objekt ausgeführt werden kann.

### MATLAB Script in Agilent VEE verwenden

VEE umfasst das Objekt „MATLAB Script“, mit dem Sie Zugriff auf die Funktionalität von MATLAB haben. VEE kann Daten an die MATLAB Script-Engine übergeben und umgekehrt Daten von dieser Engine empfangen. Somit können Sie mathematische MATLAB-Funktionen in VEE-Programme einbeziehen.

#### HINWEIS

Wenn Sie MATLAB bereits installiert haben, verwendet VEE das installierte MATLAB-Produkt zur Verarbeitung des MATLAB Script. Falls Sie jedoch nicht über die Signalverarbeitungs-Toolbox verfügen, können Sie diese Funktionen von VEE aus nur nutzen, wenn die mit VEE ausgelieferte MATLAB Script Engine registriert wurde. Zum Registrieren von MATLAB wechseln Sie zum Verzeichnis `<VEE_installation_dir>\MATLAB\bin` und starten das Programm **MATLAB.exe /regserver**.

---

**HINWEIS**

Einige Anmerkungen zur Fehlerbehandlung bei VEE-MATLAB-Installationen. VEE fügt den Verzeichnispfad von MATLAB Script (~\installDir\matlab\bin\win32) immer am Ende der PATH-Umgebungsvariablen für die aktive VEE-Sitzung ein. Das bedeutet, dass eine bereits installierte Version von MATLAB Vorrang vor der mit VEE installierten Version von MATLAB Script hat. VEE wird daher bei der Ausführung von LoadLibrary für die DLLs libeng.dll und libmx.dll immer diejenigen Versionen dieser DLLs laden, die über den früheren Eintrag in der PATH-Anweisung zuerst gefunden werden. Eine Installation von MATLAB erweitert normalerweise die PATH-Anweisung um die Information über den Speicherort der Dateien matlab.exe, libeng.dll und libmx.dll. Achtung: Die verwendeten Versionen von libeng.dll und libmx.dll müssen zu der MATLAB-Version passen, die über COM registriert wurde.

Bei der MATLAB-Version, die geladen wird, handelt es sich um die zuletzt registrierte Version, was üblicherweise die zuletzt ausgeführte MATLAB-Version ist (MATLAB erneuert die eigene Registrierung bei jeder Ausführung). Es können also dann Probleme auftreten, wenn Sie gleichzeitig eine 6.1 Vollversion und eine 6.5 Vollversion auf Ihrem System installiert haben. Diejenige Version von MATLAB, die Sie mit VEE ausführen möchten, muss also als letzte registriert sein und, was noch wichtiger ist, der Pfad zu ihrem Verzeichnis [matlabroot\bin\win32] muss in der PATH-Anweisung vor den Einträgen zu anderen MATLAB-Version stehen.

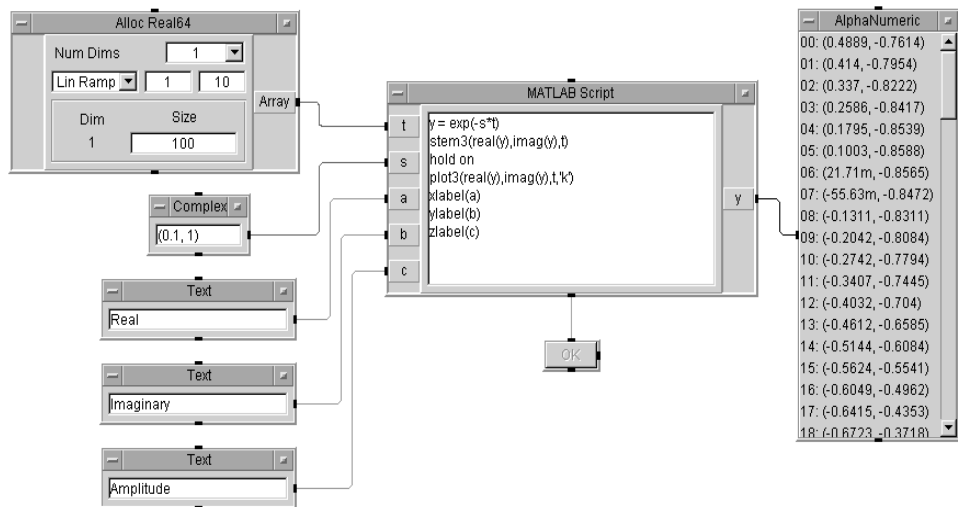
---

Einige der Anwendungen für das MATLAB Script-Objekt sind:

- Mit VEE generierte Daten mit MATLAB verarbeiten.
- Ergebnisse vom Objekt MATLAB Script zurückgeben und diese Ergebnisse in anderen Teilen des VEE-Programms verwenden.
- Komplexe Filter-Designs und Implementierungen im Objekt MATLAB Script durch Verwenden der Funktionalität der MATLAB Signalverarbeitungs-Toolbox ausführen.
- Daten als 2-D- oder 3D-Grafiken darstellen.

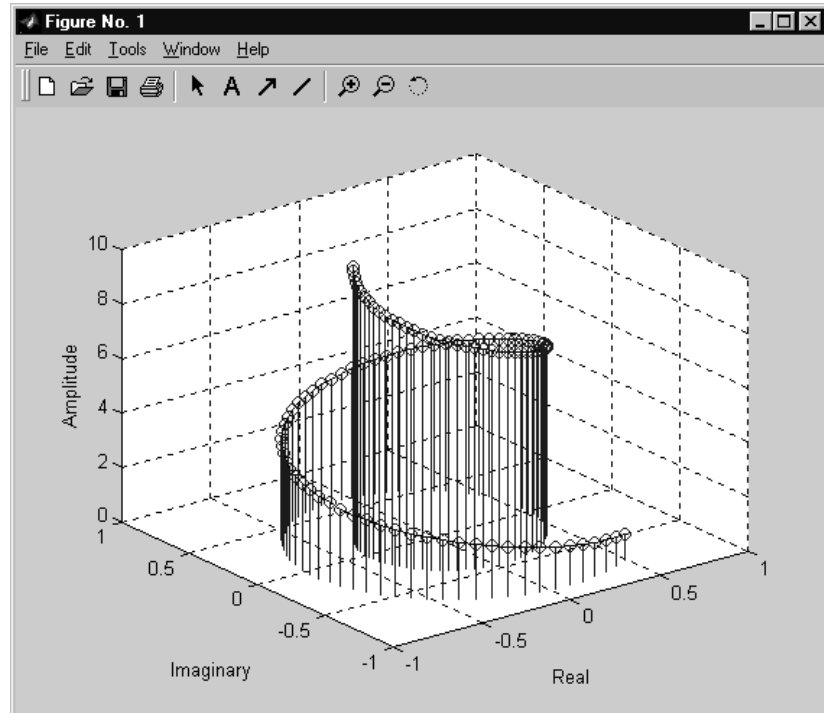
Abbildung 120 zeigt das Objekt MATLAB Script in einem VEE-Programm. Bei der Ausführung des MATLAB Script-Programms werden die dann im Objekt Alphanumeric angezeigten Daten generiert.

## 4 Testdaten analysieren und anzeigen



**Abbildung 120** MATLAB Script-Objekt in einem VEE-Programm

Abbildung 121 zeigt die bei der Ausführung des Programms generierte Grafik an.



**Abbildung 121** Vom Programm generierte Grafik

Wenn Sie MATLAB Script-Objekte in einem VEE-Programm verwenden, ruft VEE die MATLAB Script-Engine zur Ausführung von Operationen in den MATLAB Script-Objekten auf. Es werden Daten von VEE an MATLAB übertragen und wieder zurück. Einige Hinweise zu MATLAB:

- Das erste in einem Programm ausgeführte MATLAB Script-Objekt öffnet eine einzelne MATLAB-Session. Alle weiteren Instanzen des Objekts MATLAB Script nutzen diese Session gemeinsam. MATLAB Script-Objekte können daher globale Variablen im MATLAB-Arbeitsbereich gemeinsam nutzen.

- VEE führt *keine* Syntaxprüfung der MATLAB-Befehle vor dem Aufruf der MATLAB Script-Engine durch. Von MATLAB generierte Fehler und Warnungen werden in den normalen VEE-Dialogfeldern angezeigt wie alle anderen VEE-Fehler- und Warnungsmeldungen.
- Im Gegensatz zu VEE wird bei MATLAB zwischen Groß- und Kleinbuchstaben unterschieden. Vergewissern Sie sich also, dass Sie in MATLAB wirklich den Großbuchstaben **X** und nicht etwa den Kleinbuchstaben **x** verwendet haben, wenn Sie einen Ein- oder Ausgangsanschluss eines MATLAB Script-Objekts mit dem Großbuchstaben **X** benennen.
- Als Eingang für MATLAB Scripts sind nur bestimmte VEE-Datentypen zulässig. Dies wird auf Seite 195 ausführlich erläutert.

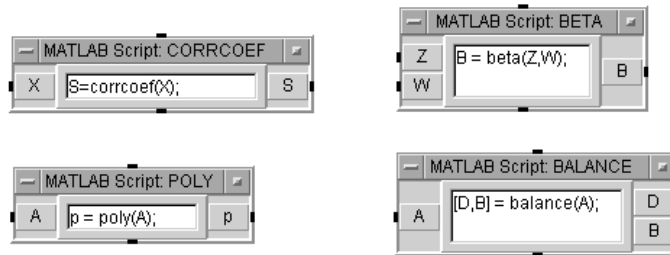
### Das Objekt MATLAB Script in Agilent VEE verwenden

Wenn Sie ein MATLAB-Objekt in einem VEE-Programm verwenden, gleicht es einem Formelobjekt. Es gibt zwei Möglichkeiten, ein MATLAB Script-Objekt einem Programm hinzuzufügen:

- 1 Wählen Sie **Device** ⇒ **MATLAB Script** und klicken Sie mit der Maustaste, um das Objekt im Programm zu platzieren. Dadurch wird ein standardmäßiges MATLAB Script-Objekt erstellt, das Sie Ihren nach Ihren Anforderungen anpassen können.

- ODER -

Wählen Sie **Device** ⇒ **Function & Object Browser** und anschließend Type: MATLAB Functions. Wählen Sie eine vordefinierte MATLAB-Funktion und klicken Sie auf die Schaltfläche Create Formula. Klicken Sie mit der Maustaste, um das Objekt im Programm zu platzieren. Abbildung 122 zeigt einige vordefinierte MATLAB-Funktionen, die Sie einem VEE-Programm hinzufügen können.



**Abbildung 122** Vordefinierte MATLAB-Objekte

Beachten Sie, dass jedes Objekt den Namen MATLAB Script<function name> erhält, damit Sie es von anderen VEE-Formelobjekten unterscheiden können. Jedes Objekt enthält bereits die Funktion, die es ausführen wird, sowie die voraussichtlich erforderlichen Eingangs- und Ausgangs-Pins, genau wie bei integrierten VEE-Formelobjekten. Sie können MATLAB Script-Objekte auch bearbeiten, genau wie andere VEE-Objekte.

## HINWEIS

Weitere Informationen zu MATLAB-Funktionen können Sie über die VEE-Menüleiste durch Klicken auf **Help** ⇒ **MATLAB Script** ⇒ **Help Desk** aufrufen.

## Mit Datentypen arbeiten

MATLAB-Objekte können nur einen Teil der VEE-Datentypen als Ein- und Ausgang empfangen bzw. senden.

VEE konvertiert einige der eindimensionalen Arrays automatisch, um so die Verarbeitung durch Programme zu ermöglichen, die sowohl VEE- als auch MATLAB-Funktionen enthalten. Ein eindimensionales VEE-Text-Array wird z. B. automatisch in ein zweidimensionales Zeichen-Array umgewandelt, wenn es in ein MATLAB Script-Objekt eingeht; ein eindimensionales Zeichen-Array aus einem MATLAB Script-Objekt wird automatisch in ein Text-Skalar umgewandelt, wenn es aus einem MATLAB Script-Objekt ausgegeben wird.

### HINWEIS

Eine vollständige Liste und Beschreibung der automatischen Konvertierungen zwischen VEE- und MATLAB-Datentypen finden Sie in der VEE-Online-Hilfe.

---

Sie können auch Begrenzungen für die Datentypen am Eingangsanschluss verwenden, um sicherzustellen, dass die von einem anderen Objekt eingehenden Daten in einen unterstützten Typ umgewandelt werden, wie im folgenden Beispiel gezeigt.

- 1 Wählen Sie **Data** ⇒ **Constant** ⇒ **Int32** und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Ändern Sie den Wert in 7. Klonen Sie das Objekt mit Clone und platzieren Sie das zweite Int32-Objekt unter dem ersten. Ändern Sie seinen Wert in 20.
- 2 Wählen Sie **Device** ⇒ **MATLAB Script** und platzieren Sie das Objekt rechts von den Konstantenobjekten.
- 3 Wählen Sie Display Alphanumeric und platzieren Sie das Objekt rechts von dem MATLAB Script-Objekt.
- 4 Verbinden Sie das Ausgangs-Pin des oberen Int32-Objekts mit dem Eingangs-Pin **A** des MATLAB Script-Objekts. Verbinden Sie das Ausgangs-Pin des unteren Int32-Objekts mit dem Eingangs-Pin **B** des MATLAB Script-Objekts. Verbinden Sie das Ausgangs-Pin des MATLAB Script-Objekts mit dem Eingangs-Pin des Alphanumeric-Objekts.

Führen Sie das Programm aus. Es generiert einen VEE-Laufzeitfehler mit der Angabe, dass für die eingehenden Daten die Datentypen Real64, Complex, Waveform oder Text möglich sind, nicht aber Int32.

Sie können Fehler wie diesen vermeiden, indem Sie den Datentyp am Eingangsanschluss des MATLAB Script-Objekts ändern.

- 5 Doppelklicken Sie auf den Anschluss **A**, um das Dialogfeld Input Terminal Information zu öffnen. Klicken Sie auf **Required Type**, um eine Dropdown-Liste anzuzeigen, wählen Sie Real64 und klicken Sie auf **OK**. Doppelklicken Sie auf den Anschluss **B** und ändern Sie ihn ebenfalls in Real64, wie in Abbildung 123 gezeigt.



- 6 Führen Sie das Programm aus. Die Int32-Daten werden jetzt am Eingangs-Pin automatisch in Real64 umgewandelt und dann an MATLAB weitergegeben.

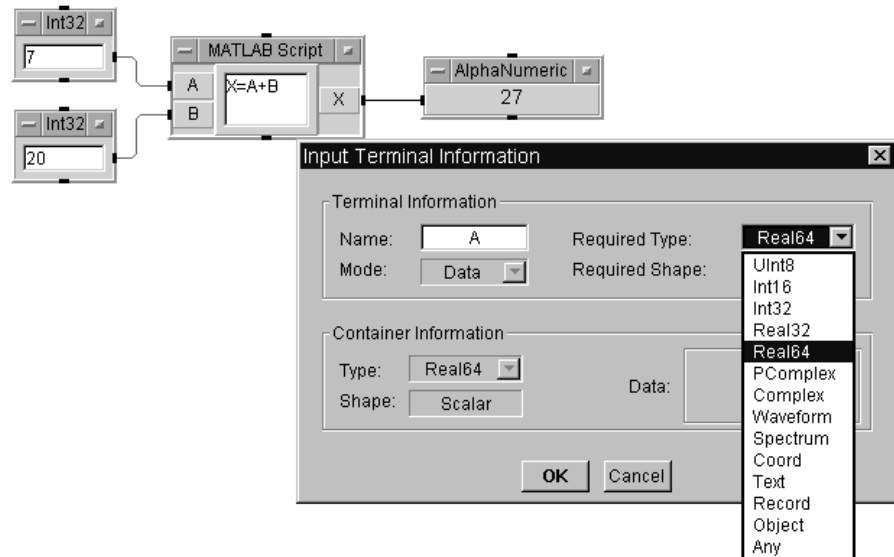


Abbildung 123 Den Datentyps am Eingangsanschluss ändern

## Anzeigen von Testdaten

Tabelle 15 beschreibt die Anzeigefunktionen für die verschiedenen VEE-Objekte.

**Tabelle 15** Anzeigen

Anzeige	Beschreibung
Alphanumeric	Werte als Text oder als Zahlen anzeigen. Erfordert SCALAR, ARRAY 1D oder ARRAY 2D.
Beep	Gibt einen hörbaren Ton aus, um eine Stelle in Ihrem Programm zu kennzeichnen.
Complex Plane	Zeigt die Datenwerte Complex, Polar Complex (PComplex) oder Coord an einer realen gg. einer imaginären Achse an.
Indicator=>> Meter, Thermometer, Fill Bar, Tank, Color Alarm	Alle diese Indikatoren zeigen Zahlen mit einer grafischen Darstellung an, die jeweils ihren Namen entspricht. Alle diese Indikatoren haben farbig codierte Bereiche – normalerweise drei, „Meter“ sogar fünf. Der „Color Alarm“ kann eine bei einem Alarm in den verschiedenen Bereichen blinkende LED über eine blinkende Textmeldung simulieren.
Label	Ein Objekt, mit dem eine Textbeschriftung in der Fensteransicht platziert wird. Die Farben und Schriften können in der Ansicht „Properties...“ leicht angepasst werden, die Sie im Objektmenü aufrufen.
Logging Alphanumeric	Zeigt bei einer wiederholten Aufzeichnung Werte als Text oder Zahlen an. Erfordert SCALAR oder ARRAY 1D.
Note Pad	Nutzt eine Textnotiz zur Beschreibung eines Programms.
Picture (PC)	Ein Objekt, mit dem eine grafische Beschriftung in der Fensteransicht platziert wird. Die unterstützten Formate sind: *.BMP (Bitmap), *.GIF (GIF87a und GIF89), *.JPEG, *.PNG und *.WMF (Windows Meta File)
Polar Plot	Zeigt Daten grafisch auf einer polaren Skala an, wenn separate Informationen für Radius- und Winkeldaten verfügbar sind.

**Tabelle 15** Anzeigen

Anzeige	Beschreibung
Spectrum (Freq)	Ein Menü, das Frequenzbereichsanzeigen enthält: Magnitude Spectrum, Phase Spectrum, Magnitude gg. Phase (Polar) und Magnitude gg. Phase (Smith). Der Eingang muss Waveform, Spectrum oder ein Koordinaten-Array sein. Waveform-Eingänge werden über eine a Fast Fourier Transformation (fft) automatisch in Frequenzbereiche konvertiert.
Strip Chart	Zeigt die jüngste Historie der bei der Ausführung des Programms laufend generierten Daten grafisch an. Für jeden y-Eingangswert wird der x-Wert um eine angegebene Schrittgröße hochgezählt. Wenn neue Daten über die rechte Seite der Anzeige hinausragen, blättert die Anzeige automatisch nach rechts, um die neuesten Daten anzuzeigen.
Waveform (Time)	Zeigt Wellenformen (Waveform) oder Spektren im Echtzeitbereich grafisch an. Spektren werden über eine invertierte Fast Fourier Transformation (ifft) automatisch in den Zeitbereich umgewandelt. Die x-Achse entspricht den Sampling-Einheiten der Eingangs-Wellenform.
X gg. Y Plot	Zeigt Werte grafisch an, wenn separate Dateninformationen für X- und Y-Daten verfügbar sind.
XY Trace	Zeigt zugeordnete Arrays oder eine Gruppe von Werten grafisch an, wenn y-Daten mit gleichmäßig verteilten x-Werten generiert werden. Der automatisch generierte x-Wert hängt von Datentyp der Trace-Daten ab. Ein Real-Trace generiert beispielsweise gleichmäßig verteilte reale x-Werte; ein Waveform-Trace generiert dagegen x-Werte für die Zeitachse.

### Testdatenanzeigen anpassen

Die Anzeigen können auf verschiedene Arten angepasst werden. Sie können die Anzeigen nicht nur beschriften, verschieben und ihre Größe ändern wie bei anderen VEE-Objekten auch, sondern Sie können auch die x/y-Skala und die Traces ändern, Markierungen hinzufügen oder Teile der Anzeige in der grafischen Anzeigen zoomen.

Die folgende Übung verdeutlicht einige dieser Funktionen. Das Objekt Noise Generator wird zum Generieren einer Wellenform verwendet, die anschließend im Objekt Waveform (Time) angezeigt wird. Die Übung zeigt auch, wie die X-Skala geändert, ein Segment der Wellenform gezoomt und die Markierungen zum Messen der Abstände zwischen Punkten der Wellenform verwendet werden können. Dasselbe Prinzip kann auf alle grafischen Anzeigen angewendet werden.

#### Eine Wellenform anzeigen

- 1 Wählen Sie **Device** ⇒ **Virtual Source** ⇒ **Noise Generator**.
- 2 Wählen Sie **Display** ⇒ **Waveform (Time)**.
- 3 Verbinden Sie den Datenausgang des Noise Generator mit dem Dateneingang von Waveform (Time) und führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 124.

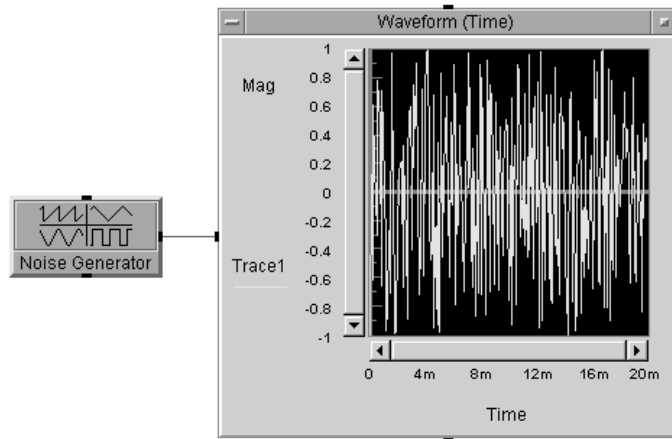


Abbildung 124 Anzeige einer Wellenform

## Die X- und Y-Skala ändern

- 1 Doppelklicken Sie auf die Titelleiste Waveform (Time), um die Ansicht Properties Y Plot zu öffnen. Doppelklicken Sie unter Design auf Scales; mit der daraufhin angezeigten Schaltfläche rufen Sie das Dialogfeld auf, in dem Sie für **X Maximum** den Wert **20m** durch 1m ersetzen.

Das Zeitfenster der Anzeige wird dadurch von 20 Millisekunden auf 1 Millisekunde geändert.

- 2 Doppelklicken Sie unter **Y** in Feld Minimum, und ersetzen Sie den Wert **-1** durch den Wert **- .5**. Klicken Sie auf **OK**.

## Einen Teil der Wellenform zoomen

- 1 Öffnen Sie das Objektmenü Waveform (Time) und klicken Sie auf **Zoom** ⇒ **In**.

Der Mauszeiger wird als kleiner rechter Winkel angezeigt. Durch Klicken und Ziehen können Sie ein Quadrat in der Grafik zeichnen, das den zu zoomenden Bereich angibt.

- 2 Ziehen Sie einen Umriss um einen Bereich der Wellenform, der mehrere Spitzen enthält, und lassen Sie die Maustaste los.

Die Anzeige zoomt in den ausgewählten Bereich der Wellenform. Beachten Sie, dass die **x** und **y**-Skala automatisch geändert werden.

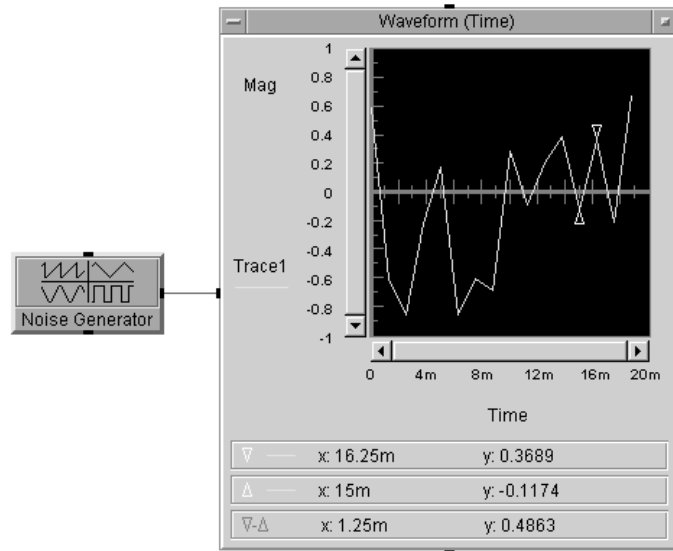
### Delta-Markierungen in der Anzeige hinzufügen

- 1 Wechseln Sie zur offenen Ansicht des Noise Generator.
  - a Ändern Sie die Einstellung von Num Points in 16. Führen Sie das Programm erneut aus.
  - b Öffnen Sie das Objektmenü Waveform (Time) und wählen Sie Properties (oder doppelklicken Sie auf die Titelleiste). Wählen Sie anschließend unter Markers für NumMarkers den Eintrag **Delta**. Klicken Sie anschließend auf **OK**.

#### HINWEIS

Sie können die Werte der Markierungen während der Laufzeit abrufen und ändern. Weitere Informationen können Sie in der Online-Hilfe unter **Contents and Index** ⇒ **How Do I...** ⇒ **Display Data** anzeigen.

Sie sehen nun zwei weiße Pfeile an einem der Datenpunkte der Wellenform, die nach oben und unten weisen. Beachten Sie auch, dass die **x**- und **y**-Koordinaten dieser Markierungen in der unteren Hälfte der Anzeige angezeigt werden. Wenn Sie den **x**- oder **y**-Abstand zwischen zwei Spitzen messen möchten, können Sie die Pfeile zu den entsprechenden Spitzen ziehen. Sie sehen, wie sich eine der Markierungen zu diesen neuen Spitzen verschiebt; die neuen Koordinaten werden am unteren Rand der Anzeige eingeblendet (siehe Abbildung 125).



**Abbildung 125** Delta-Markierungen in einer Wellenform-Anzeige

VEE interpoliert automatisch zwischen den Datenpunkten der Wellenform. Öffnen Sie das Objektmenü, wählen Sie Properties und klicken Sie unter Markers auf die Option **MarkerInterpolate**.

## Die Farbe des Trace ändern

- 1 Doppelklicken Sie auf die Titelleiste, um die Ansicht Properties zu öffnen, und doppelklicken Sie anschließend erneut auf die Eigenschaft **Trace**.

Sie können anschließend die Farbe, den Linientyp, den Namen sowie den Punkttyp für den ausgewählten Trace in diesem Ordner ändern.

### HINWEIS

Sie können diese Werte während der Laufzeit mit den Steuereingängen Traces oder Scales ändern. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

## 4 Testdaten analysieren und anzeigen

- 2 Der Trace wird jetzt in der neuen Farbe angezeigt. Weitere Anzeigemerkmale wie Gittertyp können auf ähnliche Weise angepasst werden.

### **HINWEIS**

VEE enthält auch die Funktion Plot in den Anzeigen-Objektmenüs. Mit dieser Funktion können Sie die Testergebnisse auf dem Bildschirm plotten, ohne den Rest des Programms auszudrucken.

---

### **Zur weiteren Übung**

Informationen zu anderen VEE-Objekten und zusätzliche Sicherheit beim Umgang mit Agilent VEE erhalten Sie, wenn Sie die Übungen in Anhang, „Tests sequenzieren“ auf Seite 561 durchführen. Die Lösungen mit einer Erläuterung der wichtigen Punkte sind ebenfalls enthalten.



## Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie die entsprechenden Themen bei Bedarf noch einmal an und gehen Sie dann zum nächsten Kapitel über.

- Die wichtigsten Datentypen in VEE beschreiben.
- Einige der Hauptanwendungen für die Analysefunktionen in VEE beschreiben.
- Das Hilfethema zu jedem Objekt im Dialogfeld Function & Browser anzeigen.
- Die Beziehung zwischen Eingangs-Pins und Variablen in einem mathematischen VEE-Objekt beschreiben.
- Einen mathematischen Ausdruck mit dem Formelobjekt auswerten, anschließend zwei Ausdrücke mit dem Formelobjekt auswerten. (Denken Sie daran, nach der ersten Zeile ein Semikolon zu verwenden.)
- Eine VEE-Funktion in einem mathematischen Ausdruck im Formelobjekt verwenden.
- Das Objekt MATLAB Script verwenden.
- Die wichtigsten Anzeigefunktionen in VEE beschreiben.
- Die grafische Anzeige in Bezug auf die verwendeten Skalen, den angezeigten Teil der Wellenform, die verwendeten Markierungen und die Farbe des Trace anpassen.

## **4 Testdaten analysieren und anzeigen**

# 5

## Testergebnisse speichern und abrufen

Überblick 209

Testergebnisse in Arrays speichern 210

Die Objekte To/From File verwenden 214

Records (Datensätze) zum Speichern gemischter Datentypen  
verwenden 227

DataSets zum Speichern und Abrufen von Records verwenden 238

Eine einfache Testdatenbank anpassen 243

Kapitel-Checkliste 252

### Testergebnisse speichern und abrufen

*In diesem Kapitel erhalten Sie Informationen zu folgenden Themen:*

- Testdaten in Arrays ablegen
- Das Objekt `Collector` verwenden
- Die Objekte `To/From File` verwenden
- Gemischte Datentypen mit `Records` erstellen
- Such- und Sortieroperationen mit `DataSets` ausführen
- Einfache Testdatenbanken mit den `Dataset`-Objekten erstellen

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*

## Überblick

In diesem Kapitel lernen Sie die Grundfunktionen zum Speichern und Abrufen von Testdaten kennen. Sie erstellen Arrays des richtigen Datentyps und der richtigen Größe für Ihre Testergebnisse und rufen diese Daten oder Teile davon zur Analyse oder Anzeige ab.

Darüber hinaus werden in diesem Kapitel die Objekte To/From File, der Datentyp „Record“ (Datensatz) sowie Dataset-Dateien beschrieben. Die Objekte To File und From File schreiben über E/A-Transaktionen Daten in Dateien und lesen Daten aus Dateien. Der Datentyp Record kann zum Speichern von Daten verschiedenen Typs, die jedoch in einer einzigen Struktur zusammengefasst sind, verwendet werden. Sie können Datasets verwenden, um einen oder mehrere Datensätze in einer Datei zu speichern sowie Such- und Sortieroperationen mit Datasets ausführen.

### HINWEIS

Das Objekt **To File** ist auch im Abschnitt „Übung 2-3: Datendateien verwenden“ auf Seite 90 in Kapitel „Agilent VEE Programmier Techniken“ beschrieben.

---

### Testergebnisse in Arrays speichern

Datentypen können auf zwei Arten gespeichert werden:

- Als Skalarwerte (einzelne Zahlen wie beispielsweise 9 oder (32, @10))
- ODER -
- Als ein- bis zehndimensionale Arrays.

#### HINWEIS

Eine Übersicht über die VEE-Datentypen finden Sie in Kapitel „Testergebnisse speichern und abrufen“.

Die Indizierung für Arrays erfolgt in VEE mit der Basis Null, wobei die Position der Array-Elemente in Klammern angegeben wird. Wenn das Array **A** beispielsweise die Elemente [4, 5, 6] enthält, gilt:

$A[0] = 4$ ,  $A[1] = 5$  und  $A[2] = 6$

Die Syntax für Arrays lautet wie folgt:

**Tabelle 16** Syntax für Arrays

Syntax-Element	Beschreibung
Doppelpunkt	Kennzeichnet einen Bereich von Elementen. Im obigen Beispiel-Array etwa: <b>A[0:1] = [4, 5]</b>
Stern (*)	Ein Platzhalterzeichen zur Angabe aller Elemente einer bestimmten Array-Dimension. <b>A[*]</b> gibt alle Elemente eines Array <b>A</b> zurück.
Kommata	In der Syntax für Unter-Arrays werden die Array-Dimensionen durch Kommata getrennt. Wenn <b>B</b> ein zweidimensionales Array mit drei Elementen in jeder Dimension ist, gibt <b>B[1,0]</b> das erste Element in der zweiten Reihe von <b>B</b> zurück.

Die Syntax für den Zugriff auf Elemente eines Array kann im Formula-Objekt oder einem Ausdrucksfeld, beispielsweise im Objekt To/From File, verwendet werden.

### Übung 5-1: Ein Array für Testergebnisse erstellen

Die einfachste Möglichkeit zum Erstellen eines Array ist die Verwendung des Objekts Collector.

In dieser Übung wird das Objekt For Count zum Simulieren von vier Einlesevorgängen von einem Instrument verwendet. Die „eingelassenen“ Daten werden in einem Array abgelegt, und die Ergebnisse werden gedruckt. Das Prinzip ist unabhängig vom Datentyp oder der Größe des Array immer gleich, da der Collector beliebige Datentypen verwendet und die Größe des Array entsprechend der Anzahl der gesendeten Elemente erstellt.

- 1 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Count, Data** ⇒ **Collector** und **Display** ⇒ **AlphaNumeric**.

Tabelle 17

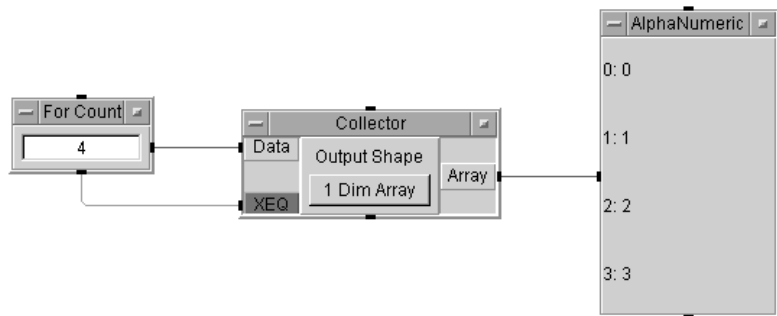
Objektname	Beschreibung
Informationen zum Objekt For Count	For Count gibt aufsteigende ganzzahlige Werte aus, beginnend mit 0, entsprechend der im Eingabefeld angegebenen Iterationen. Heben Sie den vorgegebenen Wert 10 durch Doppelklicken hervor, und geben Sie anschließend 4 ein. For Count gibt dann die Werte <b>0, 1, 2</b> und <b>3</b> aus.
Informationen zum Objekt Collector	Das Objekt Collector empfängt Datenwerte über seinen Dateneingangsanschluss. Wenn Sie mit der Datenerfassung fertig sind, führen Sie einen „ping“ auf den Anschluss XEQ aus, um dem Collector mitzuteilen, dass das Array erstellt und ausgegeben werden soll. Sie können den Sequenzausgangspin des Objekts For Count verwenden, um den „ping“ auf den Collector XEQ auszuführen. Der Collector zeigt eine Schaltfläche zum Umschalten zwischen einem <b>1 Dim Array</b> und einem $n+1$ <b>Dim Array an</b> .  Doppelklicken Sie auf den Collector, um in die offene Ansicht wechseln, und lesen Sie unter Help im Objektmenü nach, um die Arbeitsweise des Objekts zu verstehen.

## 5 Testergebnisse speichern und abrufen

- 2 Klicken Sie im Collector auf **n+1 Dim**, um die Auswahl in **1 Dim Array** zu ändern.
- 3 Verbinden Sie den Datenausgangs-Pin des Objekts For Count mit dem Dateneingangs-Pin am Collector.
- 4 Verbinden Sie den Sequenzausgangs-Pin des Objekts For Count mit dem Eingangs-Pin XEQ am Collector.

Der XEQ-Pin, ein spezieller Auslöser-Pin, den es bei verschiedenen Objekten gibt, legt fest, wann das Objekt ausgeführt wird. In diesem Fall soll das Objekt ausgelöst werden, wenn alle Daten für das Array erfasst wurden.

- 5 Verbinden Sie den Datenausgangs-Pin des Collector mit dem Dateneingangs-Pin des AlphaNumeric-Objekts.
- 6 Vergrößern Sie das AlphaNumeric-Objekt, um genügend Platz für das Array zu schaffen. Klicken und ziehen Sie hierzu eine der Ecken des Objekts. (Sie könnten AlphaNumeric auch schon beim Platzieren vergrößern, indem Sie auf seine Umrisslinie klicken und ziehen.)
- 7 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 126.



**Abbildung 126** Der Collector erstellt ein Array



## Übung 5-2: Werte aus einem Array extrahieren

Zum Extrahieren von Werten aus einem Array können Sie die Klammersyntax in einem Ausdruck oder das Objekt **Access Array** ⇒ **Get Values** verwenden. Im folgenden Beispiel werden Ausdrücke im Formula-Objekt verwendet. In dieser Übung erweitern Sie das Programm um mehrere Objekte.

- 1 Löschen Sie die Datenlinie zwischen dem Collector und dem AlphaNumeric-Objekt, indem Sie auf die Linie zeigen, Umschalt+Strg drücken und mit der linken Maustaste klicken. Minimieren Sie den Collector anschließend.
- 2 Wählen Sie **Device** ⇒ **Formula** und klonen Sie das Objekt. Verschieben Sie AlphaNumeric nach rechts, und platzieren Sie beide Formula-Objekte rechts vom Collector.
- 3 Verbinden Sie den Datenausgang des Collector mit dem Dateneingang der Formula-Objekte. Geben Sie im oberen Formeleingabefeld den Wert `A[2]` ein und im unteren Formeleingabefeld den Wert `A[1:3]`.
- 4 `A[2]` extrahiert das dritte Element des Array als Skalar; `A[1:3]` gibt ein Unter-Array aus drei Elementen mit dem zweiten, dritten und vierten Element von **A** (das Array des **A**-Eingangsanschlusses) zurück.
- 5 Klonen Sie AlphaNumeric und verbinden Sie eine Anzeige mit jedem Formula-Objekt.
- 6 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 127.

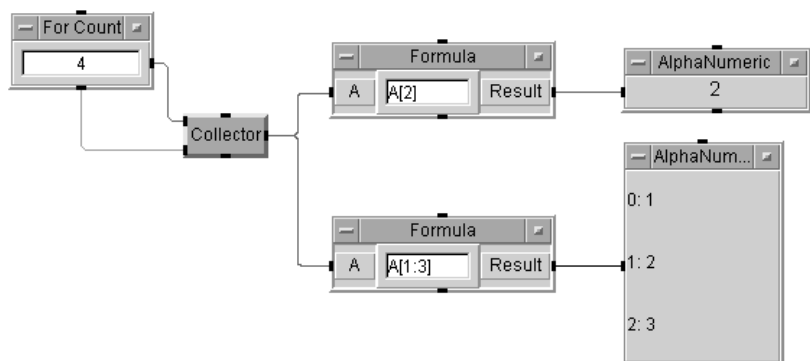


Abbildung 127 Array-Elemente mit Ausdrücken extrahieren

### Die Objekte To/From File verwenden

Die Objekte To File und From File schreiben über E/A-Transaktionen Daten in Dateien und lesen Daten aus Dateien. Sie haben folgende Merkmale:

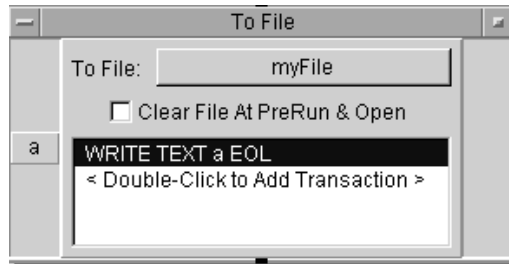
- Bei der ersten READ- oder WRITE-Transaktion wird eine Datendatei geöffnet. Beim Beenden des Programms schließt VEE alle geöffneten Dateien automatisch.
- VEE verwaltet einen Lesezeiger und einen Schreibzeiger pro Datei, unabhängig davon, wie viele Objekte auf die Datei zugreifen. Der Lesezeiger kennzeichnet die Daten, die als nächste gelesen werden, und der Schreibzeiger kennzeichnet die Stelle, an der das nächste Datenelement geschrieben wird.
- Die Objekte To/From File können Daten an vorhandene Dateien anhängen oder Dateien überschreiben. Wenn in der offenen Ansicht des Objekts To File die Einstellung Clear File at PreRun & Open markiert ist, verweist der Schreibzeiger auf den Anfang der Datei. Ist diese Option nicht markiert, verweist der Zeiger auf das Ende der bestehenden Datei. Jede WRITE-Transaktion hängt Daten an der Position des Schreibzeigers an die Datei an. Beim Ausführen einer EXECUTE CLEAR-Transaktion wird der Schreibzeiger an den Anfang der Daten gesetzt und löscht damit den bisherigen Inhalt der Datei.
- Ein Lesezeiger beginnt am Anfang der Datei und wandert entsprechend den READ-Transaktionen vorwärts durch die Datei. Sie können im Objekt From File eine EXECUTE REWIND-Transaktion ausführen, um den Zeiger wieder an den Anfang der Datei zu setzen; dies hat keine Auswirkung auf die Daten.

#### HINWEIS

Das Objekt To File ist auch im Abschnitt „Übung 2-3: Datendateien verwenden“ auf Seite 90 in Kapitel „Agilent VEE Programmieretechniken“ beschrieben.

## Die Arbeitsweise der E/A-Transaktionen

E/A-Transaktionen werden von VEE zur Kommunikation mit Instrumenten, Dateien, Zeichenfolgen, dem Betriebssystem, Schnittstellen, anderen Programmen und Druckern verwendet. Sehen Sie sich als Beispiel das Objekt To File in Abbildung 128 an.



**Abbildung 128** Das Objekt „To File“

Das in Abbildung 128 dargestellte Objekt To File sendet Daten an das angegebene Objekt myFile. Es kann Eingänge, die so genannten Transaktionen, enthalten, die Daten von einem Programm empfangen. Dieses Objekt To File enthält beispielsweise die Transaktion WRITE TEXT a EOL. Wenn Sie auf die Transaktion doppelklicken, wird ein „I/O Transaction“-Dialogfeld geöffnet (siehe Abbildung 129); über dieses Dialogfeld wird die spezifische Transaktionsanweisung konfiguriert.



**Abbildung 129** Ein „I/O Transaction“-Dialogfeld

Je nach Objekt gibt es verschiedene Varianten dieses Dialogfelds, alle diese Dialogfelder enthalten jedoch einige gemein-

same Elemente wie die „Aktionen“, die „Codierung“, die „Ausdruckliste“, das „Format“ und die „Zeilenende-Sequenz“ (EOL).

### E/A-Transaktionsformat

Eine E/A-Transaktion zum Schreiben von Daten hat normalerweise das folgende Format:

```
<Aktion> <Codierung> <Ausdruckliste> <Format>  
<EOL>
```

Tabelle 18 beschreibt die gängigsten Aktionen: READ, WRITE, EXECUTE und WAIT.

**Tabelle 18** Typen von E/A-Transaktionen

Aktion	Beschreibung
READ	Liest Daten aus der angegebenen Quelle mit der angegebenen Codierung und dem angegebenen Format ein.
WRITE	Schreibt Daten in das angegebene Ziel mit der angegebenen Codierung und dem angegebenen Format.
EXECUTE	Führt einen angegebenen Befehl aus. EXECUTE REWIND setzt beispielsweise den Schreib- oder Lesezeiger einer Datei auf den Anfang der Datei, ohne den Inhalt der Datei zu löschen. EXECUTE CLOSE schließt eine geöffnete Datei.
WAIT	Wartet die angegebene Anzahl von Sekunden, bevor die nächste Transaktion ausgeführt wird.

#### HINWEIS

Sie können auch eine Reihe weiterer Aktionen für **I/O** ⇒ **Advanced I/O Operations** untersuchen. Sehen Sie sich hierzu die Elemente in dem Menü an.

Codierungen und Formate beziehen sich auf die Art, in der Daten verpackt und gesendet werden. Eine TEXT-Codierung sendet Daten beispielsweise als ASCII-Zeichen. Die TEXT-Codierung kann auf unterschiedliche Arten formatiert werden.

Wenn Sie beispielsweise eine Folge von Buchstaben und Ziffern an eine Datei senden wollen, wird diese Folge mit einer WRITE TEXT STRING-Transaktion als ASCII-Zeichen gesendet. Eine WRITE TEXT REAL-Transaktion dagegen extrahiert nur die Realen Zahlen aus der Zeichenfolge und sendet ASCII-Zeichen für die einzelnen Stellen der Zahlen. Tabelle 19 bietet eine kurze Erläuterung der Codierungen.

**Tabelle 19** E/A-Transaktionscodierung

Codierung	Erläuterung
<b>TEXT</b>	Liest oder schreibt alle Datentypen in einer für den Benutzer lesbaren Form (ASCII); diese Form kann leicht bearbeitet oder in andere Software-Anwendungen portiert werden. Numerische VEE-Daten werden automatisch in Text umgewandelt.
<b>BYTE</b>	Wandelt numerische Daten in binäre Ganzzahlen um und sendet oder empfängt das letzte signifikante Byte.
<b>CASE</b>	Ordnet einen Aufzählungswert oder eine Ganzzahl einer Zeichenfolge zu und liest/schreibt diese Zeichenfolge. Sie können CASE beispielsweise verwenden, um Fehlernummern zu empfangen und Fehlermeldungen zu schreiben.
<b>BINARY</b>	Verarbeitet alle Datentypen in einem maschinenspezifischen Binärformat.
<b>BINBLOCK</b>	Verwendet IEEE488.2 Block-Headers mit definierter Länge mit allen VEE-Datentypen in binären Dateien.
<b>CONTAINER</b>	Verwendet ein VEE-spezifisches Textformat mit allen Datentypen.

In einer WRITE-Transaktion ist eine „Ausdrucksliste“ einfach eine durch Kommas begrenzte Liste von Ausdrücken, die ausgewertet werden müssen, um die gesendeten Daten zu liefern. Der Ausdruck kann aus einem mathematischen Ausdruck, dem Namen eines Dateneingangsanschlusses, einer Zeichenfolgekonstanten, einer VEE-Funktion, einer UserFunction oder einer globalen Variablen bestehen. In einer READ-Transaktion muss die Ausdrucksliste aus einer durch Kommas begrenzten Liste der Namen von Ausgangsanschlüssen bestehen, die angeben, wo die Daten nach dem Einlesen gespeichert werden sollen.

## 5 Testergebnisse speichern und abrufen

Die Datenformate werden in Verbindung mit dem Lesen von Daten aus Instrumenten in Kapitel, „Einfache Methoden zum Steuern von Instrumenten“ auf Seite 131 beschrieben. Die meisten dieser Formate gelten für alle E/A-Transaktionen.

EOL (Zeilenende-Zeichenfolge) kann ein- oder ausgeschaltet werden. Sie können die EOL-Sequenz definieren, indem Sie das Objektmenü der meisten Objekte des Typs **I/O** ⇒ **To** öffnen, Properties... und anschließend Data Format wählen, und die gewünschten Änderungen unter Separator Sequence vornehmen.

### Übung 5-3: Die Objekte To/From File verwenden

In dieser Übung wird beschrieben, wie Testdaten in Dateien geschrieben und daraus gelesen werden. Sie werden in dieser Übung drei allgemeine Testergebnisse speichern und abrufen: einen Testnamen, einen Zeitstempel und einen eindimensionalen Array aus Real-Werten. Der gleiche Prozess gilt für alle VEE-Datentypen.

### Eine Textzeichenfolge an eine Datei senden

1 Wählen Sie **I/O** ⇒ **To** ⇒ **File**. Legen Sie die Einträge wie folgt fest

**Tabelle 20** Eine Textzeichenfolge an eine Datei senden

Operation	Beschreibung
filename	Verwenden Sie die Standarddatei myFile. Sie können die Standarddatei ändern, indem Sie auf das Eingabefeld To File klicken, um eine Liste der Dateien im Home-Verzeichnis anzuzeigen.
Clear File At PreRun & Open	Markieren Sie dieses Feld. Standardmäßig hängt VEE neue Daten an das Ende einer bestehenden Datei an. Wenn dieses Feld markiert ist, wird der Inhalt der Datei vor dem Schreiben der neuen Daten gelöscht.

- 2 Doppelklicken Sie auf den Transaktionsbereich, um das Dialogfeld I/O Transaction zu öffnen. (Sehen Sie sich dazu gegebenenfalls Abbildung 128 und Abbildung 129 an.)

WRITE TEXT an EOL is the default transaction. Sie schreibt Daten an Pin **a** mit einer TEXT-Codierung und einer angegebenen EOL-Sequenz. VEE unterscheidet nicht zwischen Groß- und Kleinbuchstaben. Sie können für die Namen der Dateneingangs- und Datenausgangsanschlüsse Klein- oder Großbuchstaben verwenden.

Legen Sie die Einträge wie folgt fest:

**Tabelle 21** Die Standardtransaktion schreiben

Eintrag	Beschreibung
a (Ausdrucksfeld)	Das Ausdrucks-Listefeld wird hervorgehoben; es enthält den Standardwert a. Geben Sie "Test1" ein und klicken Sie auf <b>OK</b> . (Sie müssen die Anführungszeichen verwenden, um den Eintrag als Textzeichenfolge zu kennzeichnen. Wenn Sie Test1 ohne Anführungszeichen eingeben, interpretiert VEE diese Angabe als den Namen eines Anschlusses oder einer globalen Variablen.)
WRITE	Verwenden Sie den Standardwert <b>WRITE</b> .
TEXT	Verwenden Sie den Standardwert <b>TEXT</b> . Die Codierung <b>TEXT</b> sendet die Daten als ASCII-Zeichen.
DEFAULT FORMAT	Verwenden Sie <b>DEFAULT FORMAT</b> . <b>DEFAULT FORMAT</b> verwendet ein entsprechendes VEE-Format wie beispielsweise <b>STRING</b> .
EOL ON	Verwenden Sie den Standardwert. Die Standard-EOL-Sequenz ist das Escape-Zeichen für eine neue Zeile ( <code>\n</code> ).

- 3 Klicken Sie auf **OK**, um zum Objekt To File zurückzukehren. Die Transaktionsleiste sollte jetzt die Anweisung WRITE TEXT "Test1" EOL enthalten. Diese Transaktion sendet die Zeichenfolge Test1 an die angegebene Datei.

## Einen Zeitstempel an eine Datei senden

Die Funktion `now()` in der Kategorie **Device** ⇒ **Function & Object Browser** ⇒ **Time & Date** liefert die aktuelle Uhrzeit als Real64 Scalar. Der Wert der Real-Zahl ist die Anzahl der Sekunden seit 00:00 Uhr am 1. Januar des Jahres 1 n.Chr.

`now()` gibt daher einen Wert von ca. 63G zurück. VEE liefert dieses Format, weil es mathematisch leicht verarbeitet werden kann und nur wenig Speicherplatz erfordert. Wenn Sie den Zeitstempel in einem leichter lesbaren Format speichern wollen, verwenden Sie **TIME STAMP FORMAT** im Objekt **To File**. Führen Sie die folgenden Schritte aus, um einen Zeitstempel an eine Datei zu senden.

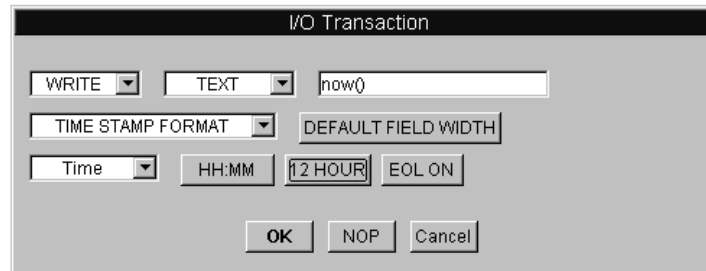
- 1 Doppelklicken Sie im gleichen „To File“-Objekt den Transaktionsbereich, um das Dialogfeld **I/O Transaction** zu öffnen.
- 2 Doppelklicken Sie auf das Eingabefeld „Ausdruckliste“, um das `a` hervorzuheben, und geben Sie `now()` ein. Die Funktion `now()` sendet die aktuelle Uhrzeit aus der Systemuhr im Format **Real**.
- 3 Ändern Sie das Format **Real** in das Format **Time Stamp**. Klicken Sie auf den Pfeil neben **DEFAULT FORMAT**, um die Dropdown-Liste zu öffnen, und wählen Sie **TIME STAMP FORMAT**. Das Dialogfeld **I/O Transaction** zeigt jetzt zusätzliche Einträge an. Legen Sie die Einträge wie folgt fest:

**Tabelle 22** Zeitstempelformate

Zeitstempel	Beschreibung
Date & Time	Wählen Sie in der Dropdown-Liste den Eintrag <b>Time</b> .
HH:MM:SS	Klicken Sie auf <b>HH : MM</b> , um das Zeitformat entsprechend umzuschalten (vom Format Stunde-Minute-Sekunde zum Format Stunde-Minute).
24 HOUR	Klicken Sie auf <b>12 HOUR</b> , um das Zeitformat entsprechend umzuschalten (vom 24-Stunden-Format auf das Format a.m./p.m.).



Das Dialogfeld I/O Transaction sollte jetzt aussehen wie in Abbildung 130.



**Abbildung 130** Das E/A-Transaktionsfeld „TIME STAMP“

- 4 Klicken Sie auf **OK**, um zum Fenster To File zurückzukehren. Die zweite Transaktionsleiste sollte jetzt die Anweisung `WRITE TEXT now() TIME:HM:H12 EOL` enthalten.

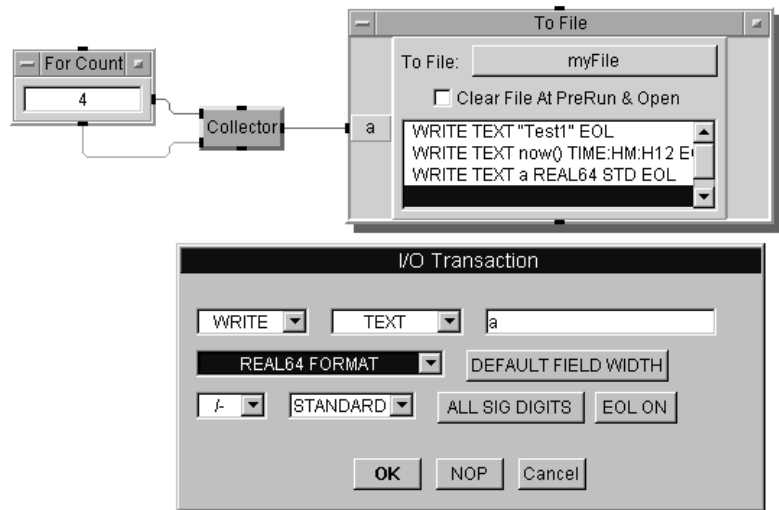
## Ein Real Array an eine Datei senden

Erstellen Sie mit den Objekten For Count und Collector ein ein-dimensionales Array aus vier Elementen und hängen Sie dieses Array an myFile an.

- 1 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Count**. Ändern Sie den Standardwert in For Count in den Wert 4.
- 2 Wählen Sie **Data** ⇒ **Collector**. Doppelklicken Sie auf den Collector, um zur offenen Ansicht zu wechseln. Verbinden Sie den Datenausgang von For Count mit dem Dateneingang des Collector (der obere Eingangs-Pin). Verbinden Sie den Sequenzausgangs-Pin von For Count mit dem Pin XEQ (dem unteren Eingangs-Pin) am Collector. Stellen Sie anschließend den Collector als Symbol dar.
- 3 Der Collector erstellt jetzt das Array `[0, 1, 2, 3]`, das Sie an die Datendatei senden können.
- 4 Verwenden Sie dieses „To File“-Objekt und doppelklicken Sie auf den Transaktionsbereich. Öffnen Sie im Dialogfeld I/O Transaction das Menü **DEFAULT FORMAT** und wählen Sie **REAL64 FORMAT**.

## 5 Testergebnisse speichern und abrufen

- Das Dialogfeld I/O Transaction zeigt weitere Schaltflächen für die Auswahl REAL64 FORMAT an. Sie können alle standardmäßig ausgewählten Optionen beibehalten; sehen Sie sich jedoch die verfügbaren Optionen zur späteren Verwendung an.
- Klicken Sie auf **OK**, um das Dialogfeld I/O Transaction zu schließen. Die Transaktionsleiste im Objekt To File sollte jetzt die Anweisung WRITE TEXT a REAL64 STD EOL enthalten. Beachten Sie, dass VEE außerdem automatisch einen Eingangsanschluss a hinzufügt.
- Verbinden Sie den Ausgang vom Collector mit dem Eingang von a des Objekts To File. Das Programm sollte jetzt aussehen wie in Abbildung 131. (Das konfigurierte „I/O Transaction“-Feld wird ebenfalls angezeigt.)



**Abbildung 131** Daten mit dem Objekt „To File“ speichern

## Daten mit dem Objekt „From File“ abrufen

Zum Abrufen von Daten mit einem Objekt From File müssen Sie wissen, wie die Daten gespeichert wurden.

### HINWEIS

Sie können auch Daten mit To DataSet oder From DataSet speichern und abrufen; eine Kenntnis des Datentyps in der Datei ist in diesem Fall nicht erforderlich. DataSets werden im Abschnitt „DataSets zum Speichern und Abrufen von Records verwenden“ auf Seite 238 beschrieben.

In diesem Beispiel ist der Name eines Tests in einem String-Format gespeichert, gefolgt von einem Zeitstempel im Time-Stamp-Format und einem Array aus Real64-Zahlen. Sie erstellen drei Transaktionen in From File zum Wiedereinlesen der Daten in VEE.

- 1 Wählen Sie **I/O** ⇒ **From** ⇒ **File** und platzieren Sie dieses Objekt unter dem Objekt To File.
- 2 Verbinden Sie den Sequenzausgangs-Pin des Objekts To File mit dem Sequenzeingangs-Pin des Objekts From File.

Über diese Sequenzverbindung wird sichergestellt, dass das Objekt To File das Senden von Daten an myFile abgeschlossen hat, bevor From File mit dem Extrahieren von Daten beginnt.

- 3 Behalten Sie im Objekt From File den Standardwert für die Datendatei myFile bei. Doppelklicken Sie auf den Transaktionsbereich, um das Dialogfeld I/O Transaction zu öffnen. Klicken Sie auf REAL64 FORMAT und ändern Sie die Angabe in STRING FORMAT, wie in Abbildung 132 gezeigt.

## 5 Testergebnisse speichern und abrufen

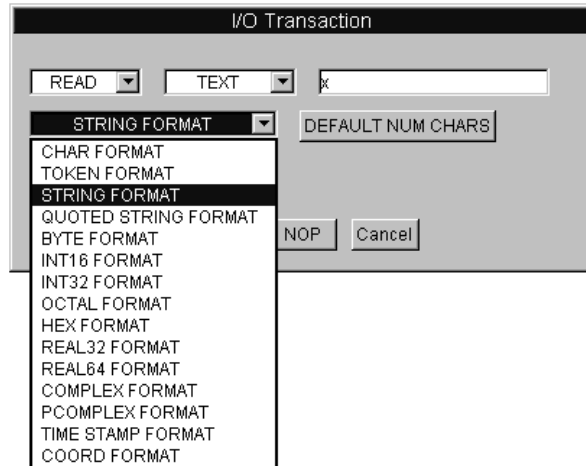


Abbildung 132 Das String-Format wählen

- Alle weiteren Standardwerte sind in korrekt; klicken Sie daher auf **OK**, um das Dialogfeld I/O Transaction zu schließen. Die Transaktionsleiste im Objekt From File sollte jetzt die Anweisung READ TEXT x STR enthalten.

Fügen Sie jetzt zwei weitere Transaktionen hinzu, um den Zeitstempel und den Real-Array wieder einzulesen.

- Doppelklicken Sie im gleichen „From File“-Objekt unterhalb der ersten Transaktionsleiste. Das Dialogfeld I/O Transaction wird geöffnet. Doppelklicken Sie auf das Eingabefeld für die Ausdrucksliste, um **x** hervorzuheben, und geben Sie **y** ein, damit die zweite Transaktion die Daten wieder an Pin **y** einliest. (Wenn dieser Pin als „**x**“ beibehalten wird, überschreibt die zweite Transaktion die Daten, die die erste Transaktion in „**x**“ abgelegt hat, statt sie anzuhängen.) Ändern Sie REAL64 FORMAT in STRING FORMAT und klicken Sie anschließend auf **OK**.

### HINWEIS

Verwenden Sie STRING FORMAT, um den Zeitstempel als Textzeichenfolge wieder einzulesen. TIME STAMP FORMAT wandelt die Zeitstempeldaten in eine Real-Zahl um.

- 6 Doppelklicken Sie im gleichen „From File“-Objekt unterhalb der zweiten Transaktionsleiste, um das Dialogfeld I/O Transaction zu öffnen. Legen Sie die Einträge wie folgt fest:

**Tabelle 23** E/A-Transaktionseinträge

Eintrag	Beschreibung
(Ausdrucksfeld)	Ändern Sie <b>x</b> in <b>z</b> , sodass das Real-Array am Ausgangsanschluss <b>Z</b> wieder eingelesen wird.
SCALAR	Ändern Sie <b>SCALAR</b> in <b>ARRAY 10</b> .
SIZE:	Das Dialogfeld I/O Transaction fügt jetzt eine Schaltfläche <b>SIZE</b> hinzu. In diesem Fall hat das Array vier Elemente. Ersetzen Sie <b>10</b> durch 4 und klicken Sie auf <b>OK</b> .

**HINWEIS**

Wenn Sie die Größe des Array nicht kennen, können Sie **SIZE** auf **TO END** setzen. Dadurch werden die Daten bis zum Ende der Datei gelesen, ohne dass VEE die genaue Dateigröße kennt. Sie können diese Funktion beispielsweise verwenden, um den gesamten Inhalt einer Datei als Zeichenfolge-Array einzulesen und ihren Inhalt zu untersuchen.

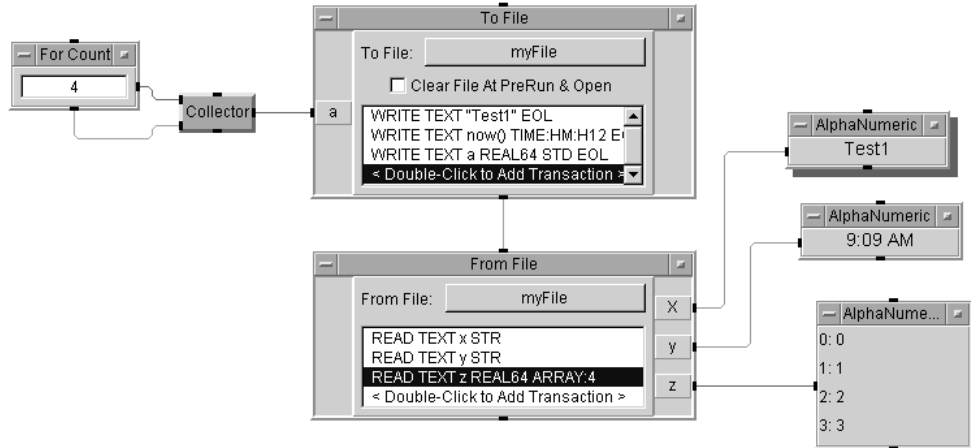
Die Transaktionsleiste im Objekt From File sollte jetzt die Anweisungen READ TEXT y STR und READ TEXT z REAL64 ARRAY:4 enthalten. Beachten Sie, dass VEE automatisch die Datenausgangsanschlüsse für x, y und z hinzufügt. Sie können auch im Objektmenü über ⇒ **Add Terminal, Delete Terminal** oder mit den Tastenkürzeln **Strg+A** und **Strg+D** Ein- und Ausgangsanschlüsse hinzufügen.

- 7 Wählen Sie **Display** ⇒ **AlphaNumeric** und klonen Sie dieses Objekt zwei Mal, um drei Anzeigen zu erhalten. Verbinden Sie die AlphaNumeric-Objekte mit den drei Datenausgangs-Pins an From File. Vergrößern Sie die Array-Anzeige durch Klicken und Ziehen des Objekts an einer beliebigen Ecke.

Sie können die Größe der AlphaNumeric-Anzeigen auch ändern, indem Sie bei der Platzierung der Objekte auf die Objektumrisse klicken und diese ziehen.

## 5 Testergebnisse speichern und abrufen

8 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 133.



**Abbildung 133** Daten mit dem Objekt „From File“ abrufen

Beachten Sie, dass das erste Alphanumeric-Objekt den Titel anzeigt, das zweite den Zeitpunkt des Tests und das dritte die Zahlen in dem Array.

## Records (Datensätze) zum Speichern gemischter Datentypen verwenden

Der Datentyp „Record“ (Datensatz) kann verschiedene Typen von Daten in einem einzigen Datencontainer speichern. Ein Record kann jeden beliebigen VEE-Datentyp enthalten. Die Daten können die Form eines Skalars oder eines Array haben. Sie können einen Testnamen, einen Zeitstempel und ein Real-Array in einer einzigen Datenstruktur speichern.

Die einzelnen Elemente in einem Record sind als Felder gespeichert und werden über eine Punktnotation aufgerufen. Rec.Name greift beispielsweise auf das Feld Name in einem Record mit dem Namen Rec zu. In einem Array von Records kennzeichnet Rec[2].Name das Feld Name im dritten Record des Array. Die Indizierung aller Arrays beginnt bei Null.

Die Strukturierung der Testdaten mit dem Datentyp „Record“ hat verschiedene Vorteile:

- Sie können logische Gruppierungen gemischter Datentypen in einem einzigen Container erstellen; die Entwicklung und Pflege des Programms wird dadurch vereinfacht. Sie können beispielsweise die folgenden Felder für einen Record zum Speichern von Testdaten verwenden: test name, value returned, pass or fail indicator, time stamp, nominal value expected, upper pass limit, lower pass limit sowie eine Beschreibung des Tests.
- Sie brauchen auf diese Weise nur einen einzigen Datencontainer zu bearbeiten statt acht separater Container. Das Programm wird dadurch einfacher und übersichtlicher.
- In VEE können Sie Records in DataSets speichern und daraus abrufen. Ein DataSet ist eine spezielle Datei, die zum Speichern von Records erstellt wurde. Wenn Sie Records aus einem DataSet abrufen, brauchen Sie die Datentypen nicht zu kennen. VEE bietet Objekte zum Abrufen, Sortieren und Suchen der in DataSets gespeicherten Informationen.

### Übung 5-4: Records verwenden

In dieser Übung wird die Verwendung des Datentyps „Record“ (Datensatz) beschrieben. Sie lernen, wie Sie einen Record erstellen, wie Sie ein bestimmtes Feld in diesem Record abrufen, wie Sie einen Wert in ein ausgewähltes Feld schreiben und wie Sie den gesamten Record in einem einzigen Arbeitsschritt auflösen. Diese Übung verwendet auch die Zeitstempelfunktion `now()` in einer anderen Weise.

#### Einen Record erstellen

Erstellen Sie einen Record mit drei Feldern: dem als String gespeicherten Namen eines Tests, einem als Real Scalar gespeicherten Zeitstempel und einem simulierten Testergebnis, das als vier Elemente in einem Array of Reals gespeichert ist. Wenn Sie diese Felder in der nächsten Übung abrufen, werden Sie sehen, dass Sie den Zeitstempel in eine Reihe verschiedener Anzeigeformate umwandeln können.

Erstellen Sie den Testnamen, indem Sie **Data** ⇒ **Constant** ⇒ **Text** wählen und in das Eingabefeld `Test1` eingeben. Benennen Sie das Objekt in `Text Constant` um. Minimieren Sie `Text Constant`.

- 9 Wählen Sie **Device** ⇒ **Function & Object Browser**. Klicken Sie unter **Category** im Abschnitt **Type and Time & Date** auf **Built-in Functions**, wählen Sie unter **Functions** den Eintrag `now` und klicken Sie auf **Create Formula**. Platzieren Sie das Objekt unter `Text Constant`.
- 10 Wählen Sie **Data** ⇒ **Constant** ⇒ **Real64** und platzieren Sie das Objekt unter `now()`.
- 11 Sie können dieses Scalar Real64-Objekt in ein Array 1D umwandeln, indem Sie im Real64-Objektmenü auf **Properties...** klicken und **1D Array** wählen.
- 12 Öffnen Sie das Fenster **Constant Properties**, indem Sie doppelt auf die **Real64**-Titelleiste klicken. Wählen Sie unter **Configuration** den Eintrag **1D Array**, ändern Sie **Size** in **4** und klicken Sie auf **OK**.



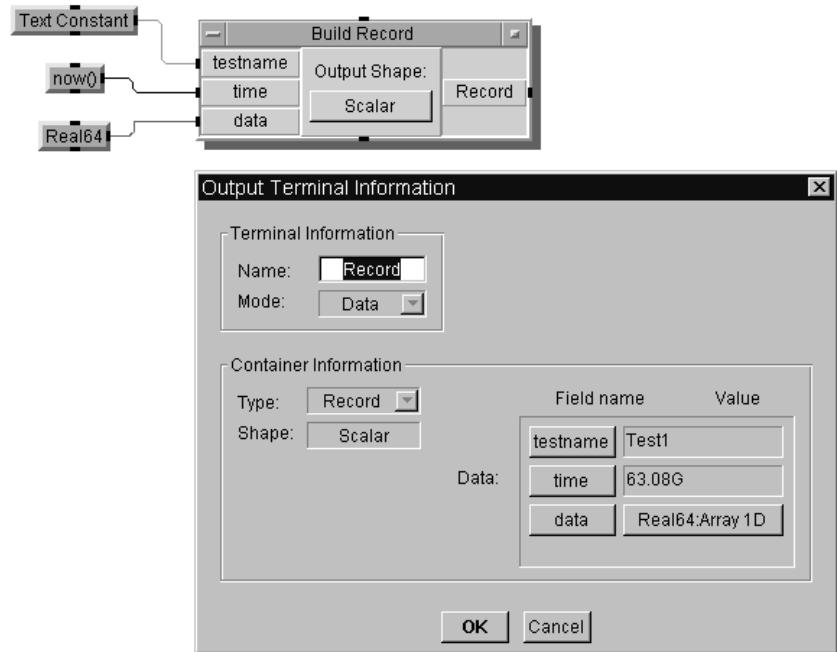
Geben Sie vier Werte in diesen Array ein, indem Sie neben dem Element 0000 doppelklicken, um den ersten Eintrag hervorzuheben und anschließend die Werte 2.2, 3.3, 4.4, 5.5 eingeben. Wechseln Sie mit der Tabulatortaste zu den einzelnen Einträgen. Minimieren Sie Real64.

- 13 Wählen Sie **Data** ⇒ **Build Data** ⇒ **Record** und platzieren Sie das Objekt rechts von den drei anderen Objekten. Fügen Sie einen dritten Dateneingangsanschluss hinzu, sodass Sie Daten über drei Felder eingeben können. Öffnen Sie die einzelnen Anschlüsse, indem Sie auf die Anschlüsse doppelklicken, und benennen Sie die drei Eingangsanschlüsse um in `testname`, `time` und `data`.

Das Feld Output Shape (Ausgabeform) des Objekts Build Record kann umgeschaltet werden zwischen Scalar und Array. Der Standardwert Scalar ist für die meisten Situationen die richtige Wahl. (Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

- 14 Verbinden Sie das Objekt Text Constant mit dem Anschluss `testname`, das Objekt `now()` mit dem Anschluss `time` und das Objekt Real64 mit dem Datenanschluss am Objekt Build Record.
- 15 Führen Sie das Programm aus. Doppelklicken Sie auf den Datenausgangsanschluss von Record, um den Record zu prüfen. Es sollte aussehen wie in Abbildung 134.

## 5 Testergebnisse speichern und abrufen



**Abbildung 134** Informationen zum Ausgangsanschluss an einem Record

Sie sehen die drei Felder und ihre Werte. Wenn Sie auf die Schaltfläche **Real64: Array 1D** klicken, zeigt ein Listenfeld die tatsächlichen Werte an. Beachten Sie, dass der Zeitstempel als Real64 Scalar gespeichert wurde. In der nächsten Übung wandeln Sie den Zeitstempel in eine leichter lesbare Form um. Klicken Sie auf **OK**, um das Dialogfeld Output Terminal Information zu schließen. Speichern Sie das Programm als `records.vee`.

## Ein Feld aus einem Record abrufen

Verwenden Sie das Objekt **Get Field** zum Extrahieren der drei Felder aus dem Record, und zeigen Sie anschließend die Werte für die Felder an.

- 1 Öffnen Sie das Programm **records.vee**.
- 2 Wählen Sie **Data** ⇒ **Access Record** ⇒ **Get Field**. Das Objekt erscheint mit dem Titel **rec.field**.

Der Dateneingang mit der Beschriftung **rec** funktioniert mit jedem Record unabhängig von der Anzahl und dem Typ der Felder. **rec.field** ist der Standardwert im Eingangsfeld; Sie können diese Angabe jedoch ändern, um ein beliebiges Feld abzurufen. **rec** bezieht sich auf den Record am Dateneingangsanschluss dieses Namens. (Denken Sie daran, dass VEE *nicht* zwischen Groß- und Kleinbuchstaben unterscheidet.)

### HINWEIS

Das Objekt **Get Field** ist ein mit Eingängen und einem Ausdruck konfiguriertes Formula-Objekt wie die Formeln im Function & Object Browser.

- 3 Klonen Sie **rec.field** zwei Mal und platzieren Sie die Objekte rechts von **Build Record**.
- 4 Verbinden Sie den Datenausgang von **Build Record** mit allen drei Objekten **rec.field**.
- 5 Da die drei Felder als **testname**, **time** und **data** gespeichert wurden, müssen Sie die Objekte **rec.field** ändern, um das entsprechende Feld zu erhalten.
- 6 Ändern Sie die drei Objektausdrucksfelder **rec.field** in **rec.testname**, **rec.time** und **rec.data**.
- 7 Wählen Sie **Display** ⇒ **AlphaNumeric** aus und klonen Sie dieses Objekt zwei Mal. Verbinden Sie die drei Anzeigen mit den drei Objekten **rec.field**. Ändern Sie die Größe der dritten Anzeige, sodass sie genügend Platz für den Real-Array bietet (ungefähr dreimal so lang wie die anderen Objekte).
- 8 Öffnen Sie das Objektmenü des zweiten **AlphaNumeric**-Objekts und wählen Sie **Properties** und anschließend den

## 5 Testergebnisse speichern und abrufen

Ordner Number. Klicken Sie links von Global Format, um das Häkchen zu entfernen.

Legen Sie das Anzeigeformat fest. Öffnen Sie das Menü Standard im Abschnitt Real. Wählen Sie Time Stamp und klicken Sie auf **OK**.

- 9 Klicken Sie auf **HH:MM:SS**, um auf das Format **HH:MM** umzuschalten. Klicken Sie auf **24 HOUR**, um auf das Format **12 HOUR** umzuschalten. Siehe Abbildung 135.

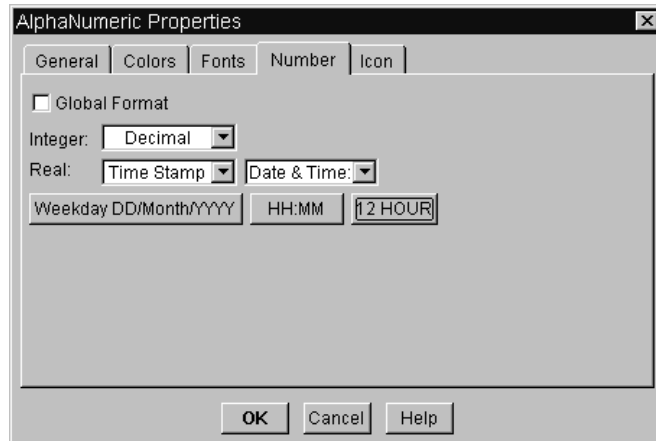
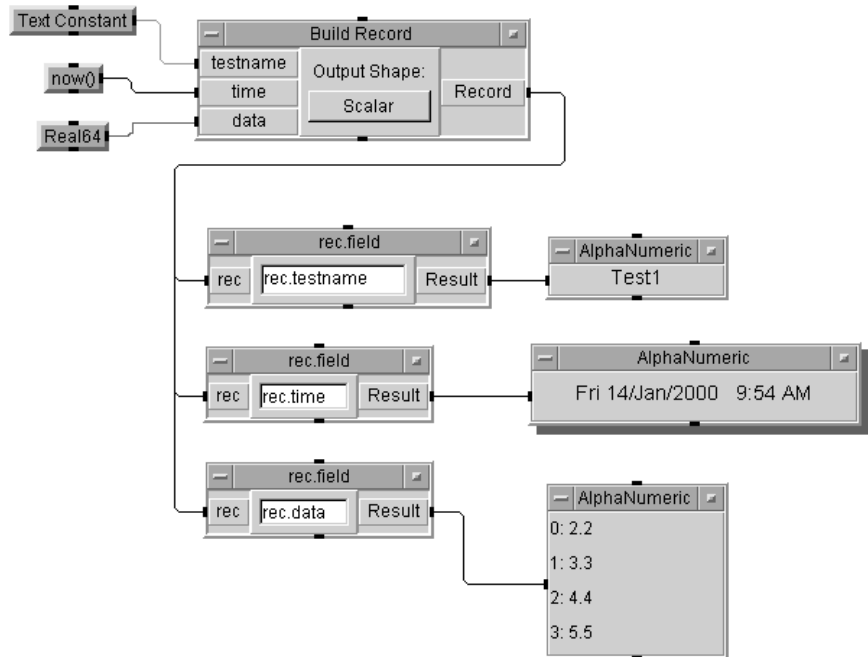


Abbildung 135 Feld „AlphaNumeric Properties“

- 10 Führen Sie das Programm aus und speichern Sie es als **getfield.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 136.



**Abbildung 136** Das Objekt „Get Field“ verwenden

Beachten Sie, dass die zweite Anzeige den Wochentag, das Datum und die Uhrzeit in Stunden, Minuten und der Angabe „a.m./p.m.“ (Vor- oder Nachmittag) auflistet.

### Ein Feld in einem Record setzen

Diese Übung zeigt, wie Daten in spezifischen Feldern eines Records geändert werden.

#### **HINWEIS**

Sie können diesen Record mit verschiedenen Tests wieder verwenden.

- 1 Öffnen Sie das Programm **getfield.vee**.
- 2 Löschen Sie alle Objekte nach Build Record, indem Sie die Objekte markieren und **Strg+X** drücken.
- 3 Wählen Sie **Data** ⇒ **Access Record** ⇒ **Set Field** und platzieren Sie das Objekt rechts von **Build Record**. Verbinden Sie den Ausgang von Build Record mit dem Eingang **rec** des Objekts Set Field. Der Titel lautet **rec.field = b**.

Set Field funktioniert durch Zuordnen des Ausdrucks auf der rechten Seite des Zuordnungssymbols (=) zur linken Seite. Das angegebene Feld von **rec** wird daher so geändert, dass es Werte von der rechten Seite enthält. Der Rest des Records bleibt unverändert. Sie verbinden den ankommenden Record mit **rec** und den ankommenden neuen Wert mit **b**. Der geänderte Record wird an dem Datenausgangsanschluss **rec** angelegt.

### HINWEIS

Das Objekt **Set Field** ist ein mit Eingängen und einem Ausdruck konfiguriertes Formula-Objekt wie die Formeln im Function & Object Browser.

- 4 Ändern Sie den Ausdruck in **rec.data[\*]=b**, um den Wert des Vier-Element-Arrays im Datenfeld zu ändern. (Sie müssen die Array-Notation **[\*]** verwenden, da Sie das gesamte Array im Feld dieses Records ändern.) Die neuen Werte für das Array werden am Eingangsanschluss **b** angelegt.
- 5 Wählen Sie **Data** ⇒ **Constant** ⇒ **Real64** und platzieren Sie dieses Objekt unter dem Objekt Build Record. Öffnen Sie das Objektmenü und wählen Sie Properties. Wählen Sie unter Configuration den Eintrag **1D Array**, ändern Sie **Size** in 4 und klicken Sie auf **OK**.

Wenn die neuen Werte für das Record-Feld in einem Array enthalten sind, muss dieses die gleiche Größe wie das aktuelle Array haben.

Geben Sie die Werte **1, 2, 3, 4** in Real64 ein, indem Sie den ersten Eintrag hervorheben und mit der Tabulatortaste zu den weiteren Einträgen wechseln. (Drücken Sie

nach dem letzten Eintrag nicht die Tabulatortaste.) Verbinden Sie Real64 mit dem Eingang b von Set Field (mit dem Titel rec.field=b).

Verwenden Sie jetzt das Objekt Get Field zum Extrahieren des Felds rec.data aus dem Record und zeigen Sie die Ergebnisse an.

- 6 Wählen Sie **Data** ⇒ **Access Record** ⇒ **Get Field** und platzieren Sie das Objekt unter dem Objekt Set Field (rec.field=b). Ändern Sie den Get Field-Objektausdruck von rec.field in rec.data. Verbinden Sie den Datenausgang von **rec.field = b** mit dem Dateneingang von rec.field.

**HINWEIS**

Sie können auch ein **Formula**-Objekt mit **A.data** im Ausdrucksfeld verwenden.

- 7 Wählen Sie ein AlphaNumeric-Objekt, ändern Sie dessen Größe entsprechend dem Array und verbinden Sie es mit dem Ausgangs-Pin von rec.field.
- 8 Führen Sie das Programm aus und speichern Sie es als **setfield.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 137.

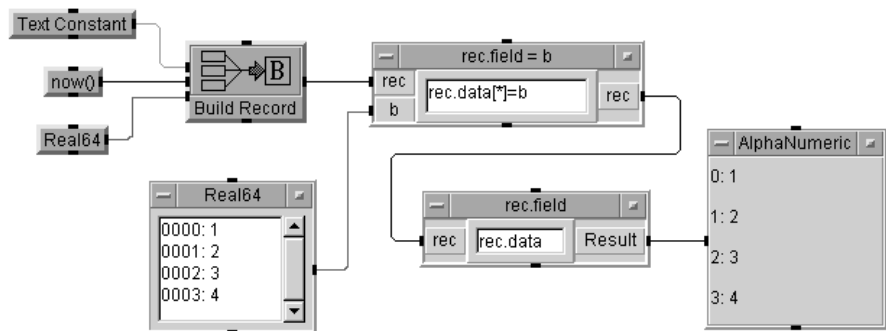


Abbildung 137 Das Objekt „Set Field“ verwenden

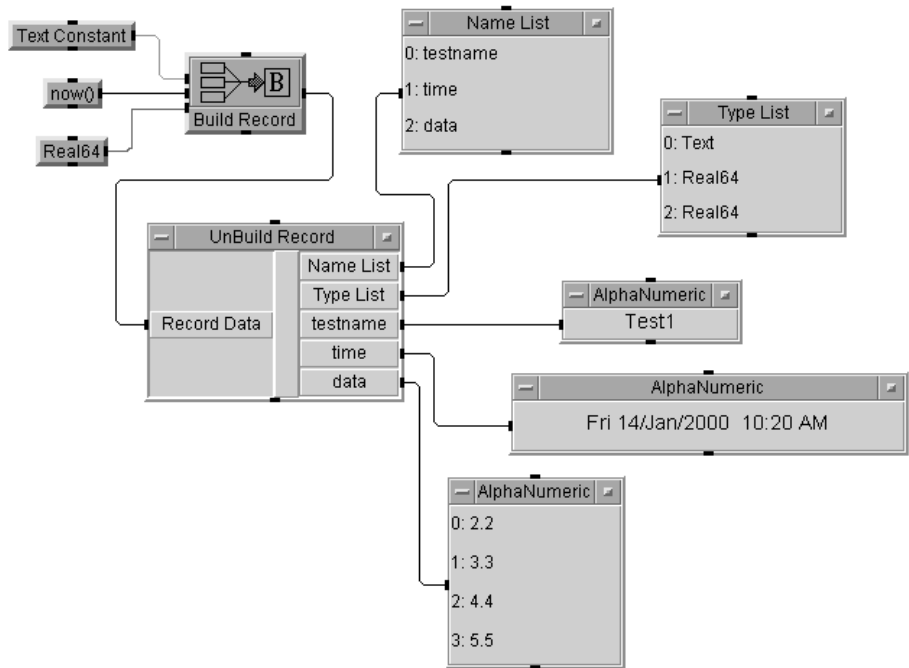
Sie können jedes beliebige Record-Feld wie in diesem Beispiel gezeigt ändern. Außerdem können Sie Felder auch teilweise ändern. Versuchen Sie beispielsweise, den Ausdruck in **rec.field = b** in den Ausdruck `rec.data[1]=20` zu ändern. Löschen Sie anschließend den Eingang **b** von **rec.field = b**. Führen Sie das Programm erneut aus. Das Array sollte wie folgt angezeigt werden: **2.2, 20, 4.4, 5.5**.

### Einen Record in einem einzigen Arbeitsschritt auflösen

Zum Extrahieren aller Felder eines Records und zum Abrufen einer Liste von Feldnamen und ihrer Typen verwenden Sie das Objekt UnBuild Record.

- 1 Öffnen Sie das Programm **setfield.vee**. Löschen Sie alle Objekte nach Build Record.
- 2 Wählen Sie **Data** ⇒ **UnBuild Data** ⇒ **Record**, platzieren Sie dieses Objekt unter Build Record, wechseln Sie zur offenen Ansicht und verbinden Sie den Ausgang von Build Record mit dem Eingang von UnBuild Record. Fügen Sie UnBuild Record einen weiteren Datenausgangs-Pin hinzu und benennen Sie seine Ausgänge **A**, **B**, und **C** in die Feldnamen um: `testname`, `time` und `data`.
- 3 Wählen Sie ein AlphaNumeric-Objekt und klonen Sie es vier mal. Verbinden Sie die fünf Objekte mit den fünf Ausgangsanschlüssen an UnBuild Record. Sie müssen die Objekte für Name List, Type List und Data vergrößern, damit genügend Platz für die Arrays zur Verfügung steht. Konfigurieren Sie außerdem die Anzeige von time auf die aktuelle Zeit im Format Tag/Monat/Jahr und Stunde, Minute im 12-Stunden-Format um.
- 4 Führen Sie das Programm aus und speichern Sie es als **unbuild.vee**. Es sollte aussehen wie in Abbildung 138.





**Abbildung 138** Das Objekt „UnBuild Record“ verwenden

Beachten Sie, dass der Pin **Name List** die Namen der drei Felder **testname**, **time** und **data** des Records liefert; **Type List** kennzeichnet **testname** als Typ **Text** sowie **time** und **data** als Typ **Real64**.

## DataSets zum Speichern und Abrufen von Records verwenden

DataSets können ein oder mehrere Records abrufen. VEE-Objekte entpacken die Records. Durch das Speichern von Records in DataSets statt in Dateien brauchen Sie sich die Datentypen nicht zu merken. Sie können auch Sortier- und Suchoperationen mit den Daten ausführen und auf diese Weise Ihre eigene angepasste Testdatenbank erstellen.

### Übung 5-5: DataSets verwenden

Ein DataSet ist einfach ein Array von Records, die in einer Datei gespeichert wurden. Diese Übung zeigt, wie die Daten in einem DataSet gespeichert und von dort wieder abgerufen werden.

#### Einen Record in einem DataSet speichern und von dort einlesen

In dieser Übung wird ein Array aus zehn Records erstellt; jeder dieser Records enthält drei Felder mit einem Testnamen, einem Real64 Scalar und einem Array von Real-Zahlen. Das Array von Records wird in einem DataSet gespeichert; anschließend werden die Records eingelesen und angezeigt.

- 1 Wählen Sie **Flow**  $\Rightarrow$  **Start**. Wählen Sie **Flow**  $\Rightarrow$  **Repeat**  $\Rightarrow$  **For Count** und platzieren Sie das Objekt unter Start. Wählen Sie **Device**  $\Rightarrow$  **Formula** und platzieren Sie das Objekt rechts von For Count. Verbinden Sie Start mit dem Sequenzeingangs-Pin an For Count; verbinden Sie den Datenausgangs-Pin von For Count mit dem Dateneingangs-Pin von Formula.
- 2 Doppelklicken Sie auf das Ausdrucksfeld Formula, um den Standardausdruck hervorzuheben, und geben Sie anschließend "test" + a ein.

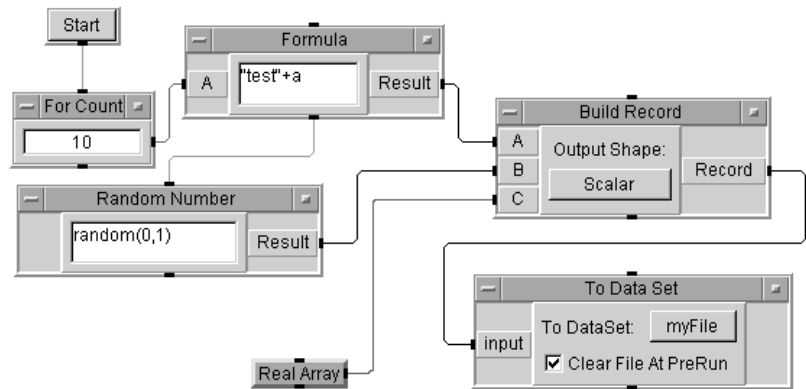
Wenn Sie auf Start klicken, gibt das Objekt For Count sequenziell ganze Zahlen von Null bis Neun an den Pin **A** von Formula aus. Im Objekt Formula werden die ganzen Zahlen dem Wort "test" hinzugefügt und als **Text Scalars ausgegeben: test0, test1, test2, ..., test9**. Diese Werte bilden die ersten Felder in den zehn Records.

- 3 Wählen Sie **Data**  $\Rightarrow$  **Build Data**  $\Rightarrow$  **Record** und platzieren Sie das Objekt rechts von Formula. Fügen Sie einen Daten-

eingangs-Pin hinzu. Verbinden Sie den Datenausgang von Formula mit dem Eingang **A** von **Build Record**.

- 4 Klicken Sie in der Symbolleiste auf die Schaltfläche Function & Object Browser.
  - a Wählen Sie Built-in Functions, Probability & Statistics und random, um das Objekt random (low, high) zu erstellen. Platzieren Sie das Objekt unter dem Objekt Formula.
  - b Löschen Sie die Eingangsanschlüsse und ändern Sie die Eingangsparameter von low in 0 sowie von high in 1.
  - c Benennen Sie das Objekt Random Number um und verbinden Sie seinen Datenausgang mit dem Anschluss **B** von Build Record.
- 5 Verbinden Sie den Sequenzausgangs-Pin des Objekts Formula mit dem Sequenzeingangs-Pin von Random Number. Durch das Verbinden der Sequenz-Pins wird sichergestellt, dass jede Iteration des Programms eine neue Zufallszahl in das Feld **B** des jeweiligen Records stellt.
- 6 Wählen Sie **Data** ⇒ **Constant** ⇒ **Real64**. Platzieren Sie das Objekt Real64 unter dem Objekt Formula .
  - a Öffnen Sie das Objektmenü und wählen Sie **Properties**. Geben Sie Real Array als Titel ein, klicken Sie unter Configuration auf die Option **1D Array** und ändern Sie **Size** in 3. Klicken Sie auf **OK**.
  - b Heben Sie alle Einträge in dem Array hervor, indem Sie doppelt darauf klicken, und geben Sie die Zahlen 1, 2 und 3 ein.
  - c Verbinden Sie den Datenausgang des Real Array mit dem Anschluss C des Objekts Build Record.
- 7 Wählen Sie **I/O** ⇒ **To** ⇒ **DataSet** und platzieren Sie dieses Objekt unter Build Record. Verbinden Sie den Datenausgang von Build Record mit seinem Dateneingang. Behalten Sie den Standarddateinamen **myfile** bei und markieren Sie Clear File At PreRun.
- 8 Führen Sie das Programm aus. Das Programm sollte ein Array aus zehn Records in das DataSet **myFile** schreiben (siehe Abbildung 139).

## 5 Testergebnisse speichern und abrufen



**Abbildung 139** Ein Array von Records in einem DataSet speichern

Lesen Sie jetzt das Array von Records ein und zeigen Sie seinen Inhalt mit den Objekten From Data Set und Record Constant an.

- 9 Wählen Sie **I/O** ⇒ **From** ⇒ **Data Set** und platzieren Sie dieses Objekt unter **For Count**. Behalten Sie den Standarddateinamen **myFile** bei. Klicken Sie auf das Feld **Get Records**, um es von One auf All umzuschalten. Behalten Sie schließlich den Standardwert 1 im Ausdrucksfeld am unteren Rand bei.

Mit diesen Einstellungen durchsucht VEE das DataSet in **myFile** und findet alle Records, die dem Kriterium in dem Ausdrucksfeld entsprechen. Wenn Sie Get Records auf One setzen, gibt VEE zunächst den ersten Record aus, der dem Kriterium in dem Ausdrucksfeld entspricht. Die 1 kennzeichnet eine **TRUE**-Bedingung; dies bedeutet, dass alle Records dem Kriterium entsprechen, sodass das gesamte Array von Records in der Datei an den Ausgangs-Pin Rec angelegt wird. Weitere Anwendungsmöglichkeiten des Ausdrucksfelds werden in anderen Übungen beschrieben. Schlagen Sie weitere Informationen unter Help im Objektmenü nach.

Verbinden Sie den Sequenzausgangs-Pin von For Count mit dem Sequenzeingang am Objekt From Data Set. Auf diese Weise wird sichergestellt, dass der Teil des Programms, der Daten an **myFile** sendet, ausgeführt wird, bevor die Daten aus der Datei gelesen werden. Sie können Show Data Flow einschalten, um die Reihenfolge der Ereignisse anzuzeigen.

- 10** Wählen Sie **Data** ⇒ **Constant** ⇒ **Record** platzieren Sie dieses Objekt unter **To Data Set**. Öffnen Sie das Objektmenü und wählen Sie **Add Terminal** ⇒ **Control Input**. Wählen Sie in dem angezeigten Listenfeld den Eintrag Default Value und klicken Sie dann auf **OK**. Vergrößern Sie das Objekt Record, sodass Sie bei der Ausführung des Programms die Ergebnisse sehen können.

Der empfangene Record wird zum Standardwert. In diesem Fall empfängt Record von dem Objekt From Data Set ein Array von Records und formatiert sich selbst, um dieses Array von Records anzuzeigen.

- 11** Verbinden Sie den Ausgangs-Pin Rec von From Data Set mit dem Pin Default Value an Record. Wenn Sie diesen Anschluss anzeigen wollen, öffnen Sie das Objektmenü und wählen Sie Properties, dann Show Terminals und schließlich **OK**. Es erscheint eine gepunktete Linie zwischen From Data Set und Record.

**HINWEIS**

Eine gepunktete Linie zwischen zwei Objekten kennzeichnet eine Steuerlinie.

---

- 12** Führen Sie das Programm aus und speichern Sie es als **dataset1.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 140.

## 5 Testergebnisse speichern und abrufen

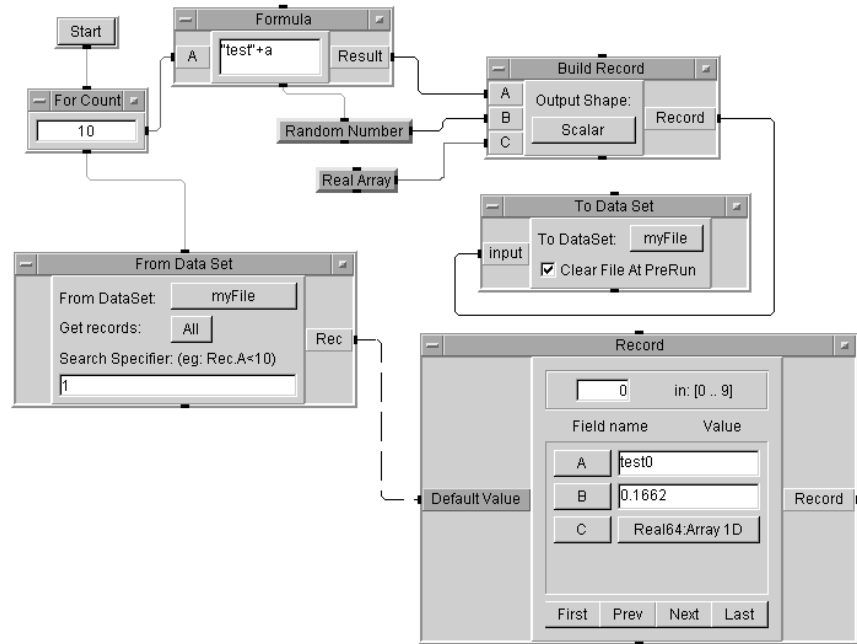


Abbildung 140 Daten mit DataSets speichern und einlesen

### HINWEIS

Ein Objekt From Data Set muss mindestens einen Record enthalten, der dem Kriterium entspricht; ansonsten gibt VEE eine Fehlermeldung aus. Um einen Fehler zu vermeiden, fügen Sie dem Objekt einen Ausgangs-Pin EOF (end-of-file) hinzu; dieser Pin wird aktiviert, wenn kein Record das Kriterium erfüllt. Sie können dann dem Programm weitere Aktionen hinzufügen für den Fall, dass EOF auslöst.

## Eine einfache Testdatenbank anpassen

Sie können in einem DataSet Informationen suchen und sortieren, z. B. Testnamen, Zeitstempel, Testparameter, Testwerte, Kennzeichen für eine erfolgreiche oder fehlgeschlagene Ausführung sowie Testbeschreibungen. DataSet-Records können somit als Testdatenbank verwendet werden. Zum Suchen von Informationen können Sie das Objekt From Data Set wie folgt verwenden:

- Das Ausdrucksfeld im Objekt From Data Set wird für Suchoperationen verwendet.
- Die Funktion `sort()` kann zum Sortieren von Records nach einem angegebenen Feld verwendet werden.

### Übung 5-6: Such- und Sortieroperationen mit DataSets verwenden

In dieser Übung lernen Sie, wie Sie ein DataSet nach Informationen durchsuchen, eine Benutzerschnittstelle für die Suchoperation erstellen und eine Sortieroperation programmieren.

#### Eine Suchoperation mit DataSets durchführen

- 1 Öffnen Sie das Programm **dataset1.vee**.
- 2 Doppelklicken Sie auf das Ausdrucksfeld am unteren Rand des Objekts From Data Set, um den aktuellen Ausdruck **1 hervorzuheben**. Geben Sie **Rec.B>=0.5** ein. Das Objekt gibt jetzt alle Records aus, wobei Feld **B** (die Zufallszahl in unserem Code) größer oder gleich 0.5 ist.
- 3 Fügen Sie einen EOF-Pin hinzu, der auslöst, wenn keine Records dem Kriterium im Ausdrucksfeld entsprechen. Zeigen Sie auf den Datenausgangsbereich des Objekts From Data Set und drücken Sie `Strg+A`. Ein EOF-Ausgang-Pin wird dem Objekt From Data Set hinzugefügt (siehe Abbildung 141).

#### HINWEIS

Zum Hinzufügen eines EOF-Pin könnten Sie auch das Objektmenü öffnen und **Add Terminal** ⇒ **Data Output...** wählen.

## 5 Testergebnisse speichern und abrufen

### 4 Führen Sie das Programm aus und speichern Sie es als **dataset2.vee**.

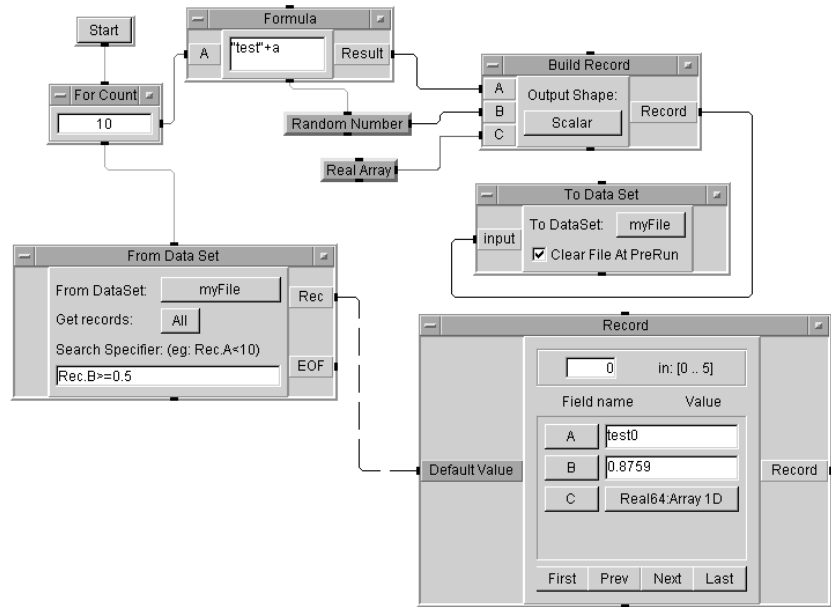


Abbildung 141 Eine Suchoperation mit DataSets

## Eine Benutzerschnittstelle für eine Suchoperation erstellen

In dieser Übung wird ein Menü für einen Benutzer hinzugefügt, um Daten aus der Datenbank der Testergebnisse zu extrahieren. Die Benutzerschnittstelle ist gesichert, um versehentliche Änderungen an dem Programm zu verhindern.

Die Spezifikation des Programms lautet wie folgt:

- Bereitstellen eines Testmenüs, das es dem Benutzer ermöglicht, einen bestimmten Test aus **test0** bis **test9** auszuwählen, aus dem er alle entsprechenden Testdaten abrufen kann.
- Anzeigen der angegebenen Testergebnisse mit den beschrifteten Feldern und Werten. Der Benutzer soll mit der Anzeige interagieren können, um ausführlichere Informationen abzurufen.
- Einbeziehen klarer Bedienungshinweise.



Führen Sie zum Erstellen des Programms die folgenden Schritte durch:

**1** Öffnen Sie das Programm **dataset2.vee**.

Fügen Sie einen Steuereingang hinzu, der die programmgesteuerte Eingabe des Ausdrucks im Objekt From Data Set ermöglicht.

**2** Öffnen Sie das „From Data Set“-Objektmenü und wählen Sie **Add Terminal... ⇒ Control Input...** Wählen Sie in dem angezeigten Menü den Befehl Formula. Ein Formula-Eingangsanschluss wird angezeigt. Klicken Sie auf das Feld Get records, um von All auf One umzuschalten, damit ein Test-Record nach dem anderen aufgerufen wird.

Der Benutzer soll einen bestimmten Testnamen wählen können. Die Testnamen befinden sich im Feld **A** aller Records. Fügen Sie den folgenden Ausdruck hinzu:

**Rec.A==<Testname in Anführungszeichen>**

**Rec.A** gibt den Record aus, in dem Feld **A** dem vom Benutzer ausgewählten Testnamen entspricht. Wenn der Benutzer beispielsweise **test6**, auswählt, muss der Ausdruck lauten: **Rec.A==„test6“**. Das Objekt extrahiert den Test-Record, der anschließend angezeigt werden kann.

Erstellen Sie ein Menü, das es dem Benutzer ermöglicht, auf eine Schaltfläche neben der gewünschten Auswahl zu klicken.

**3** Wählen Sie **Data ⇒ Selection Control ⇒ Radio Buttons** und platzieren Sie das Objekt links von For Count.

**a** Öffnen Sie das Objektmenü und wählen Sie Edit Enum Values.... Heben Sie **0000: Item 1** hervor und geben Sie test0 ein. Drücken Sie die Tabulatortaste Tab, um zu **0001: Item2** zu wechseln, und geben Sie test1 ein. Wenn Sie nach dem dritten Eintrag (test2) die Tabulatortaste drücken, erscheint automatisch ein weiterer Eintrag. Fahren Sie mit der Eingabe von Werten fort, bis Sie bei test9 angekommen sind. Klicken Sie auf **OK**, um alle zehn Einträge von test0 bis test9 anzuzeigen.



und/oder ihre Größe ändern.) Geben Sie im neuen Formula-Objekt den folgenden Ausdruck ein:

```
"Rec.A==" + "\" + A + "\""
```

**Tabelle 24** Eine Formel interpretieren

Element	Beschreibung
„Rec.A==“	"Rec . A==" sendet einen Text-Datentyp an den Ausdruckseingang From Data Formula. (Die Anführungszeichen kennzeichnen eine Textzeichenfolge.)
A	VEE sucht das erste Feld <b>A</b> aller Records in der DataSet-Datei und wählt den ersten Record, der dem ausgewählten Testnamen entspricht.
„\“	Das Escape-Zeichen für Anführungszeichen lautet \". Das Escape-Zeichen wird anschließend in Anführungszeichen gesetzt, um eine Textzeichenfolge zu kennzeichnen.  Der <i>test name</i> wird vom Test Menu als Enum-Datentyp empfangen. Anführungszeichen sind erforderlich, um die richtige Formel im Objekt From DataSet abzulegen.

Wenn beispielsweise **test6** gewählt wird, lautet die endgültige Formel **Rec.A==„test6“**. Das Objekt From Data Set gibt dann den ersten gefundenen Record aus, dessen „**A**“-Feld den Wert „**test6**“ enthält.

- 7 Verbinden Sie den Datenausgangs-Pin Test Menu Enum mit dem Dateneingangs-Pin am Formula-Objekt . Minimieren Sie das Formula-Objekt.
- 8 Verbinden Sie den Datenausgangs-Pin von Formula mit dem Steuereingangs-Pin am Objekt From Data Set mit der Beschriftung Formula.
- 9 Löschen Sie die Sequenzlinie zwischen For Count und From Data Set, um sicherzustellen, dass die alten Daten von Formula nicht erneut verwendet werden. Verbinden Sie den Sequenzausgangs-Pin des Objekts For Count mit dem Sequenzeingangs-Pin von Formula.

## 5 Testergebnisse speichern und abrufen

- 10 Verbinden Sie den Sequenzausgangs-Pin von Formula mit dem Sequenzeingangs-Pin von From Data Set. Auf diese Weise wird sichergestellt, dass die richtigen Daten von Formula verwendet werden.
- 11 Erstellen Sie ein Fenster mit Anleitungen für den Benutzer. Wählen Sie **Display** ⇒ **Note Pad**. Ändern Sie den Titel in Test Results Database Instructions. Entfernen sie jegliche Vorlageninformationen, die möglicherweise vorhanden sind. Klicken Sie auf den Note Pad-Eingabebereich und geben Sie ein: Select the test results you want from the Test Menu.
- 12 Benennen Sie das Objekt Record Constant um in Test Results.
- 13 Das Programm sollte jetzt aussehen wie in Abbildung 143. Führen Sie das Programm einige Male aus, um sicherzustellen, dass es funktioniert. Da für das Objekt Test Menu die Option AutoExecute eingeschaltet wurde, erstellen Sie eine Menüauswahl zur Ausführung des Programms.

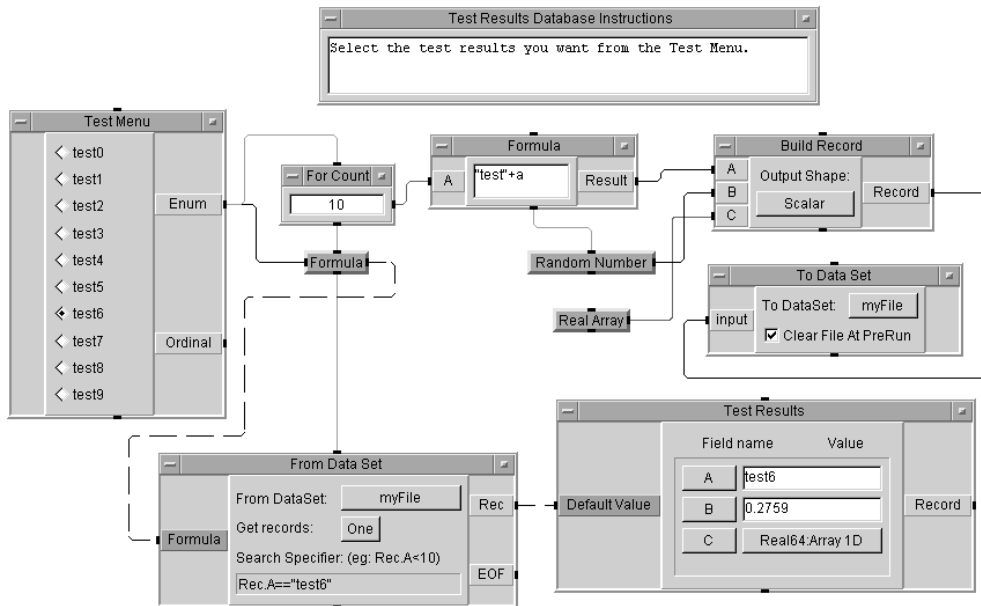


Abbildung 143 Ein Menü zur Suchoperation hinzufügen

Erstellen Sie als nächstes die Benutzerschnittstelle.

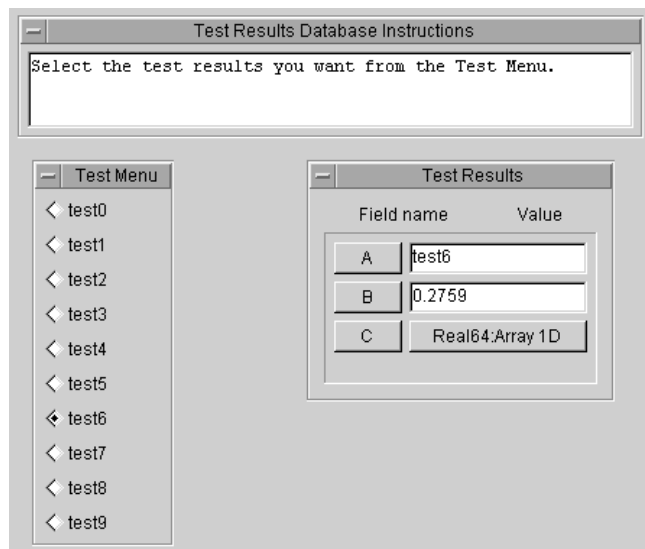
- 14** Drücken Sie `Strg` und klicken Sie auf die folgenden Objekte: **Test Menu**, **Test Results Database Instructions** und **Test Results**.

Alle markierten Objekte zeigen einen Schatten. Vergewissern Sie sich, dass keine weiteren Objekte markiert sind.

Wählen Sie anschließend **Edit** ⇒ **Add to Panel**. Die Benutzerschnittstelle wird als Fensteransicht angezeigt. Sie können jetzt die Objekte verschieben und ihre Größe ändern. Ein Layout ist in Abbildung 144 dargestellt.

**HINWEIS**

Wenn die Auswahl Add to Panel abgeblendet ist, bedeutet, dies, dass im Arbeitsbereich keine Objekte markiert sind.



**Abbildung 144** Die Benutzerschnittstelle für die Datenbank

- 15** Führen Sie das Programm einige Male aus, indem Sie im Test Menu eine Auswahl treffen. Speichern Sie das Programm als **database.vee**.

Beachten Sie, dass Sie ausführlichere Informationen zu jedem bestimmten Record abrufen können, indem Sie auf den entsprechenden Feldnamen oder auf die Werte im Objekt Record Constant (mit den Namen Testergebnisse) klicken.

### HINWEIS

Zum Sichern der Benutzerschnittstelle und des Programms gegen Änderungen wählen Sie **File** ⇒ **Create RunTime Version....** Geben Sie einen Namen für das Programm ein. VEE fügt automatisch die Erweiterung \*.vxe hinzu, um das Programm von ungeschützten Versionen zu unterscheiden.

## Eine Sortieroperation mit einem Record-Feld ausführen

Diese Übung verwendet das Programm **dataset2.vee** aus einer früheren Übung. Das Programm **dataset2.vee** setzt eine Bedingung im Objekt From DataSet wie beispielsweise **Rec.B>=0.5**, und VEE extrahiert alle Records, die diese Voraussetzung erfüllen. Das Array der resultierenden Records wird im Objekt Record Constant angezeigt.

In dieser Übung wird **dataset2.vee** zur Sortierung der resultierenden Records geändert, um zu ermitteln, welche Tests mit den größten Abweichungen fehlgeschlagen sind. Die Tests sind nach dem zweiten Feld in absteigender Reihenfolge sortiert.

- 1 Öffnen Sie das Programm **dataset2.vee**.
- 2 Wählen Sie **Device** ⇒ **Formula** und verbinden Sie den Datenausgangs-Pin Rec von From Data Set mit dem Dateneingangs-Pin des Formula-Objekts. Doppelklicken Sie auf das Ausdrucksfeld Formula, um die Standardformel hervorzuheben, und geben Sie anschließend `sort(a, 1, "B")` ein.

Das Objekt Sort befindet sich in den Funktionen Function & Object Browser, Array Category. Sie können ausführliche Informationen zu seinen Funktionen im Objektmenü unter Help nachschlagen. Die Funktion `sort()` wird vom Formula-Objekt aufgerufen.

Der erste Parameter sortiert die Daten am Pin A des Formula-Objekts in einem Array von Records. Der zweite Parameter kennzeichnet die Richtung der Sortierung: Jede Zahl ungleich Null kennzeichnet eine aufsteigende Sortierfolge, Null kennzeichnet eine absteigende Sortierung. Die Stan-

dardrichtung ist aufsteigend. Der dritte Parameter bei einem Datentyp Record kennzeichnet den Namen des Felds, nach dem sortiert werden soll. In diesem Beispiel wird also eine aufsteigende Sortierung nach dem Feld B in dem Arrays der Records durchgeführt.

- 3 Wählen Sie **Display** ⇒ **AlphaNumeric** und verbinden Sie das Objekt mit dem Datenausgangs-Pin des Formula-Objekts.
- 4 Führen Sie das Programm mehrmals aus. Es sollte jetzt aussehen wie in Abbildung 145. Beachten Sie, dass das Programm alle von der Datei DataSet zurückgegebenen Records in aufsteigender Reihenfolge nach dem Feld B sortiert.

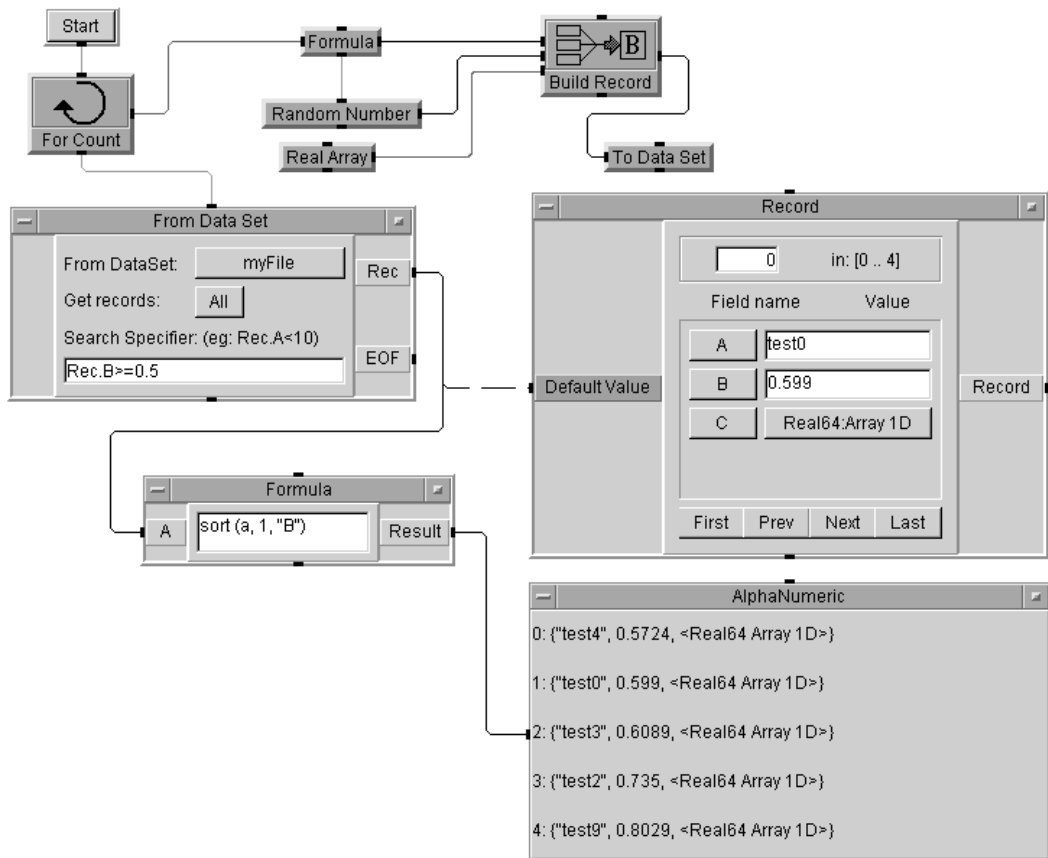


Abbildung 145 Eine Sortieroperation mit einem Record-Feld

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Wenden Sie sich ggf. den Themen dieses Kapitels erneut zu, bevor Sie mit dem nächsten Kapitel fortfahren.

- Basisnotation zur Verwendung von Arrays erläutern
- Ein Array mit dem Objekt Collector erstellen.
- Elemente aus einem Array mit dem Formula-Objekt extrahieren.
- Eine Zeichenfolge, einen Zeitstempel und ein Real-Array in eine Datei schreiben.
- Eine Zeichenfolge, einen Zeitstempel und ein Real-Array aus einer Datei lesen.
- Die Funktion `now()` für einen Zeitstempel verwenden.
- Zeitstempel für die Anzeige auf verschiedene Arten formatieren.
- Einen Record erstellen und auflösen.
- Felder in einem Record lesen und setzen.
- Einen Record in einem DataSet speichern.
- Einen Record aus einem DataSet einlesen.
- Eine Suchoperation mit einem DataSet durchführen.
- Eine Sortieroperation mit einem Record-Feld durchführen.
- VEE-Tools zum Erstellen einer einfachen Datenbank kombinieren.



## 6 Berichte mit ActiveX bequem erstellen

Überblick 255

ActiveX-Automatisierung in Agilent VEE 256

Agilent VEE-Daten an MS Excel senden 261

Erstellen einer Agilent VEE-Vorlage für MS Excel 271

MS Word für Agilent VEE-Berichte verwenden 276

Kapitel-Checkliste 285

## **Berichte mit ActiveX bequem erstellen**

*Dieses Kapitel enthält Informationen zu folgenden Themen:*

- ActiveX-Automatisierung in VEE
- Berichte in MS Excel mit ActiveX erstellen
- Berichte in MS Word mit ActiveX erstellen

*Erforderliche Zeit für dieses Kapitel: 1,5 Stunden*

## Überblick

In diesem Kapitel lernen Sie, in anderen Anwendungen wie beispielsweise MS Excel Berichte zu generieren, indem Sie Daten von einem VEE-Programm an Excel senden. VEE verwendet die ActiveX-Automatisierung zur Steuerung anderer Anwendungen; dies ermöglicht ein rasches Erstellen ausführlicher und effektiver Berichte.

In der ersten Übung wird beschrieben, wie Sie Daten an eine MS Excel-Tabelle senden, die automatisch die ActiveX-Automatisierung verwendet. Die zweite Übung beschreibt eine allgemeine Vorlage zum Generieren von Berichten und das Erweitern der Funktionalität der Basisvorlage. Die letzte Übung verwendet ActiveX in VEE zum Senden eines Screenshot und der Testdaten an ein MS Word-Dokument. (Das selbe Prinzip gilt für andere Tabellenkalkulationen und Textverarbeitungsprogramme mit Unterstützung der ActiveX-Automatisierung.)

### HINWEIS

ActiveX ersetzt die Verwendung von DDE in VEE. DDE wird dennoch in VEE weiterhin unterstützt. Informationen zur Verwendung von DDE in herstellereigenen Anwendungen finden Sie in der zweiten Ausgabe von *Visual Programming with VEE*.

---

## ActiveX-Automatisierung in Agilent VEE

In diesem Kapitel bezeichnet der Begriff ActiveX-Automatisierung die Möglichkeit von VEE, als Automation Controller von Automatisierungs-Server-Anwendungen wie MS Excel, MS Word oder MS Access zu arbeiten. Der Schwerpunkt dieser Übung liegt auf der praktischen Anwendung der Microsoft ActiveX-Technologie zum Generieren von Programmberichten aus dem Bereich Test- und Messprogramme.

### HINWEIS

Dieses Handbuch enthält weitere hierfür relevante Übungen in den Abschnitten „Übung 11-4: Ein ActiveX-Steuerelement verwenden“ auf Seite 453 und „Der aufrufbare VEE ActiveX-Automatisierungs-Server“ auf Seite 494. Ausführliche Informationen zu Konzept und Terminologie der Automatisierung finden Sie im Handbuch *VEE Pro Advanced Techniques*.

---

### Typenbibliotheken zur ActiveX-Automatisierung auflisten

Zum Auflisten der Automatisierungsobjekte auf Ihrem Computer klicken Sie auf **Devices** ⇒ **ActiveX Automation References**.

### HINWEIS

Informationen zu ActiveX Steuer-Referenzen finden Sie in Kapitel, „Benutzerschnittstellen verwenden“. Schlagen Sie außerdem nach in Kapitel, „Überblick“.

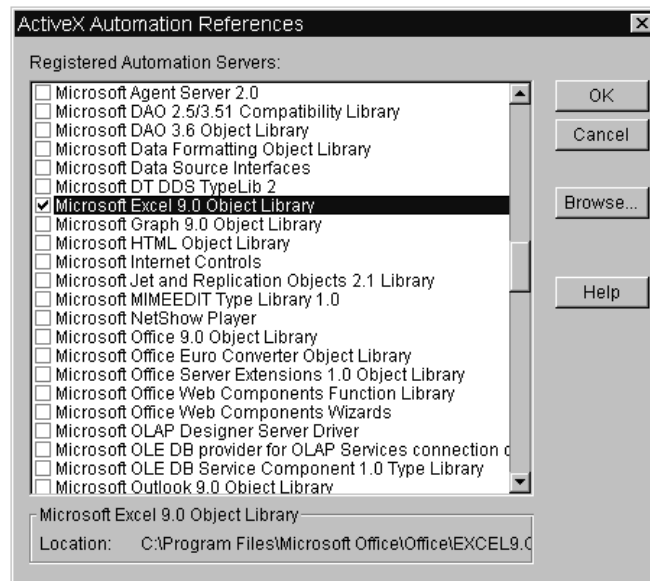
---

**Devices** ⇒ **ActiveX Automation References** listet die auf Ihrem PC installierten Typenbibliotheken auf. Jede Anwendung und ActiveX-Komponente, die als Automatisierungs-Server eingesetzt werden kann, registriert eine Typenbibliothek. VEE zeigt die auf Ihrem PC verfügbaren Bibliotheken an. Sie enthalten Informationen zur Funktionalität der Anwendung oder Komponente, die den ActiveX-Clients bereitgestellt wird.

Typenbibliotheken bestehen normalerweise aus einer Gruppe von Klassen. Manche Klassen können vom Programmierer erstellt werden. Andere Klassen werden immer von der Anwendung oder Komponente erstellt. Klassen umfassen Eigenschaften, Methoden und Ereignisse; nicht alle diese Komponenten

müssen vorhanden sein. Die Typenbibliothek bietet dem Programmierer und der VEE-Umgebung die erforderlichen Informationen zur Nutzung der Anwendung oder Komponente mit ActiveX-Schnittstellen.

Wenn Sie im Dialogfeld ActiveX Automation References eine Typenbibliothek aktivieren, sind die Bibliotheksobjekte zur Verwendung in einem VEE-Programm verfügbar. In Abbildung 146 ist beispielsweise Microsoft Excel 9.0 markiert.



**Abbildung 146** Das Dialogfeld „ActiveX Automation References“

## ActiveX-Programme mit Agilent VEE erstellen und verwenden

VEE enthält für ActiveX-Programme den Datentyp Object. Ein VEE-Objekt mit dem Datentyp Object ist ein Zeiger auf eine Komponente oder ein Datenelement, die bzw. das auf dem Automatisierungs-Server vorhanden ist. Ein Object kann beispielsweise auf ein Arbeitsblatt in MS Excel verweisen oder auf eine Zelle in diesem Arbeitsblatt. (Technisch gesehen ist ein Object ein Zeiger auf eine IDispatch-Schnittstelle, der von MS Excel oder dem Server zurückgegeben wird.)

## 6 Berichte mit ActiveX bequem erstellen

Wenn Sie beispielsweise **Data** ⇒ **Variable** ⇒ **Declare Variable** wählen, unter Name App eingeben und den Datentyp als Object definieren, können Sie über die Variable App auf ein ActiveX-Automatisierungsobjekt wie beispielsweise den Excel Automatisierungs-Server verweisen. Abbildung 147 zeigt ein Beispiel des Datentyps Object.

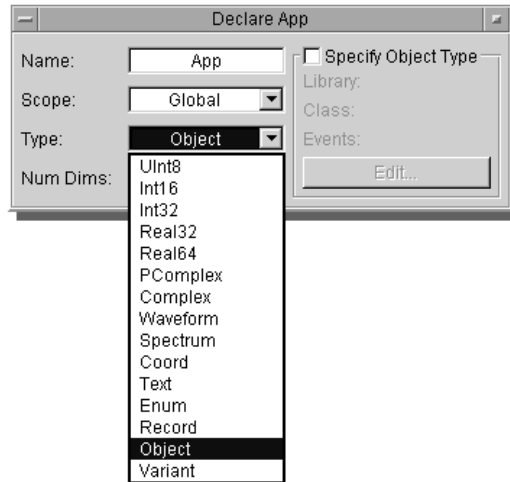
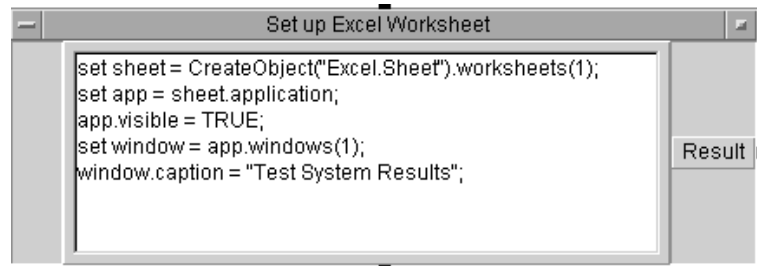


Abbildung 147 Der Datentyp „Object“

### Operationen mit ActiveX-Anweisungen durchführen

Zur Kommunikation mit einem ActiveX Automatisierungs-Server wie beispielsweise dem Excel Automatisierungs-Server geben Sie ActiveX-Befehle in ein VEE-Formel-Objekt ein. Abbildung 148 zeigt ein VEE-Formelobjekt mit dem Namen **Set Up Excel Worksheet**. Es enthält eine Liste von Befehlen, mit denen ein Excel-Arbeitsblatt für die Anzeige von Testergebnissen eingerichtet werden kann.



**Abbildung 148** Einrichten eines Excel-Arbeitsblatts zur Anzeige von Testergebnissen

VEE verwendet die Standardsyntax von Microsoft Visual Basic zum Erstellen von Befehlen oder Anweisungen wie den in Abbildung 148 gezeigten. Die Befehle oder Anwendungen führen drei Arten von Operationen aus: *get properties* (Eigenschaften abrufen), *set properties* (Eigenschaften festlegen) oder *call methods* (Methoden aufrufen).

- *Get property*-Anweisungen beziehen sich normalerweise auf das Abrufen von Daten. Die Syntax lautet *<Objekt>.<Eigenschaft>*. *sheet.application* ruft beispielsweise die Eigenschaft *application* des Objekts *sheet* ab.
- *Set property*-Anweisungen beziehen sich normalerweise auf das Setzen von Daten auf einen bestimmten Wert. Die Syntax lautet *<Objekt>.<Eigenschaft> = <Eigenschaftstyp>*. **object.property = MaxSize** legt beispielsweise eine Eigenschaft fest.
- *Call methods* rufen eine Methode auf. Eine Methode fordert von dem Objekt die Ausführung einer Aktion an. Methoden enthalten Parameter, die das Übergeben und Zurückgeben von Parametern ermöglichen. Die Syntax lautet *<Objekt>.<Methode>(Parameter)*.

## HINWEIS

Die Syntax für den Datentyp *Object* ähnelt der VEE-Syntax zum Abrufen eines Record-Felds *rec.field* und dem Aufruf einer UserFunction, *myLib.func()*. Es ist daher wichtig, dass Sie den Variablen beschreibende Namen zuordnen.

## CreateObject und GetObject verwenden

Beachten Sie, dass eine der Anweisungen in **Set Up Excel Worksheet** in Abbildung 148 den Funktionsaufruf **CreateObject()** enthält. CreateObject() und GetObject() sind Funktionen im VEE-Objekt Function & Object Browser; sie wurden speziell für die Rückgabe eines Zeigers an ein ActiveX-Objekt in VEE konzipiert.

**CreateObject(„Excel.Sheet“)** startet beispielsweise Excel und gibt eine Referenz auf eine darin enthaltene Arbeitsmappe zurück. (Die Microsoft-Anweisung „sheet“ gibt eine Arbeitsmappe zurück.) Verwenden Sie GetObject() zum Abrufen von bereits vorhandenen Daten aus einer aktiven Excel-Tabelle oder zum Laden einer Datei in ein aktives Excel.

Auf CreateObject und GetObject können Sie über **Device** ⇒ **Function & Object Browser**, Type: Built-in Functions, Category: ActiveX Automation zugreifen. Abbildung 149 zeigt je ein Beispiel-Objekt CreateObject und GetObject.

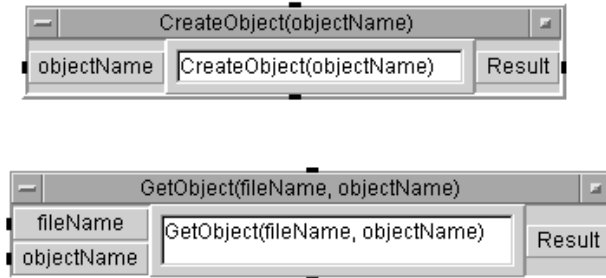


Abbildung 149 CreateObject und GetObject



## Agilent VEE-Daten an MS Excel senden

In diesem Abschnitt werden die VEE-Objekte und MS Excel-Funktionsaufrufe vorgestellt, die zum Generieren von Berichten benötigt werden.

### Übung 6-1: Agilent VEE-Daten an MS Excel senden

In dieser Übung generieren Sie virtuelle Testdaten für MS Excel. (Das Beispiel verwendet MS Office 2000 und die MS Excel 9.0 Objektbibliothek; es sollte jedoch auch mit MS Office 97 und der MS Excel 8.0 Objektbibliothek durchgeführt werden können.) Nach dem Verweisen auf die richtige Automatisierungs-Typenbibliothek deklarieren Sie einige globale Variablen des Typs Object und legen sie in einer UserFunction mit dem Namen globals ab. Die globalen Variablen vereinfachen das Programm und machen es leichter verständlich.

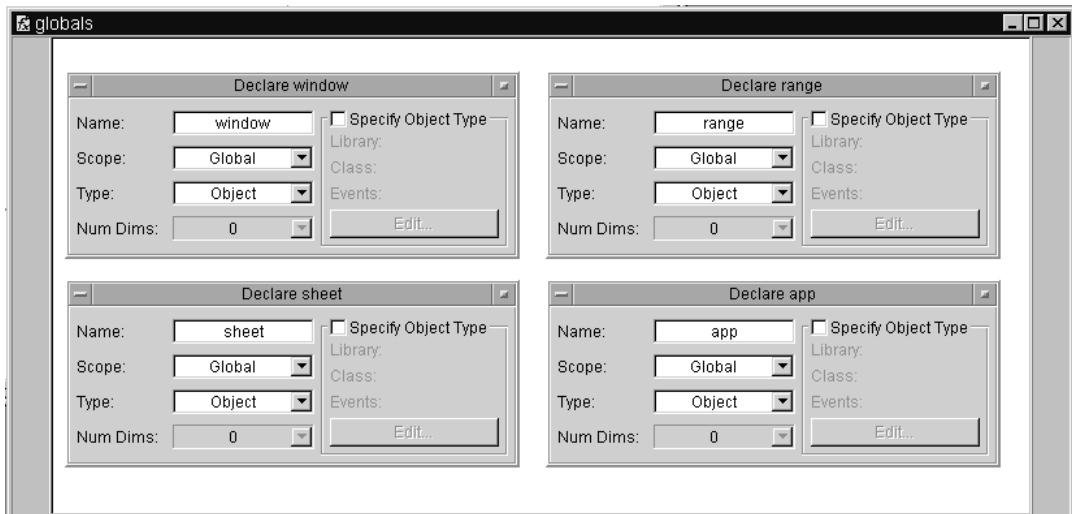
#### HINWEIS

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter **Help** ⇒ **Open Example...** ⇒ **Manual** ⇒ **UsersGuide** enthalten.

- 1 Verweisen Sie auf die Automatisierungs-Bibliothek. Klicken Sie auf **Device** ⇒ **ActiveX Automation References...**, wählen Sie Microsoft Excel 9.0 Object Library und klicken Sie auf **OK**.
- 2 Erstellen Sie eine UserFunction zum Speichern der globalen Variablen. Klicken Sie auf **Device** ⇒ **UserFunction**. Benennen Sie diese UserFunction um in `globals`. (Eine Beschreibung von UserFunctions finden Sie in Kapitel, „Agilent VEE-Funktionen verwenden“ auf Seite 345.)
- 3 Klicken Sie auf **Data** ⇒ **Variable** ⇒ **Declare Variable** und platzieren Sie dieses Objekt auf der linken Seite innerhalb von `globals`. Ändern Sie den Eintrag Name in `sheet` und wählen Sie unter Type den Datentyp Object. Das Dialogfeld wird nun mit weiteren Elemente angezeigt. In dieser Übung müssen Sie Sub Type, Library und Class nicht näher spezifizieren. (Diese werden in einem anderen Beispiel in diesem Kapitel bearbeitet.)

## 6 Berichte mit ActiveX bequem erstellen

- 4 Klonen Sie dieses Objekt drei Mal und geben Sie unter Name die folgenden Bezeichnungen ein: app, range und window. Passen Sie die Größe an und verschieben Sie die UserFunction globals unter Main. Es sollte aussehen wie in Abbildung 150.
- 5 Gleichen Sie die Einträge mit Abbildung 150 ab und minimieren Sie anschließend die vier Objekte.



**Abbildung 150** Die UserFunction „Globals“

Beachten Sie, dass Sie bei der Verwendung des Datentyps Object in der UserFunction globals auch Sub Type, Library, Class und Events näher bestimmen können. Dies bietet die folgenden Vorteile: ausführliche Typenprüfung und das Erfassen von Ereignissen.

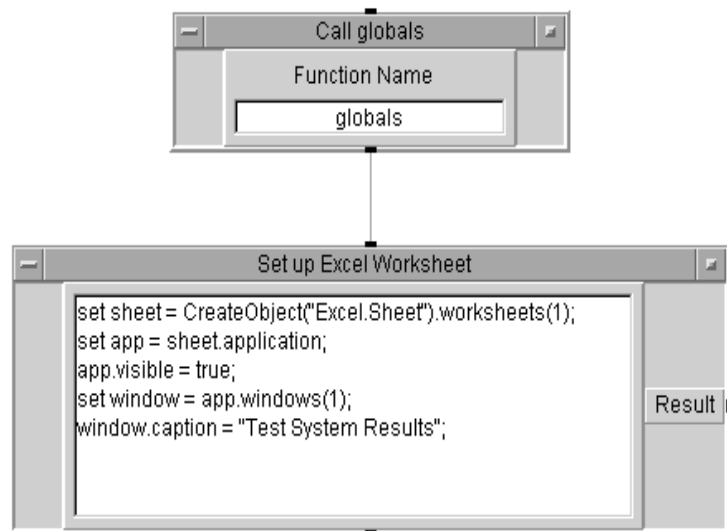
**Ausführliche Typenprüfung:** Wenn Sie beispielsweise die Variable app des Typs Object als Excel.Application festlegen, kann app nur eine Object-Variablen des Typs Excel.Application zugeordnet werden. Eine Zuordnung von Excel.worksheet oder Word.bookmark führt zu einem Fehler.

**Erfassen von Ereignissen:** Sie können eine VEE UserFunction auch verwenden, um verschiedene Ereignisse zu erfassen, die in der Anwendung auftreten können, beispielsweise das Klicken mit der rechten Maustaste in einer Excel-Tabelle. Für alle diese Ereignistypen können Sie eine VEE UserFunction angeben, die das Ereignis verarbeitet und Informationen an Excel zurückgibt.

Ereignisse sind hilfreich für ActiveX-Steuerelemente, bei denen Sie eine Möglichkeit brauchen, die Kommunikation zurück zu VEE zu steuern. (Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

- 6 Öffnen Sie das Objektmenü der UserFunction globals und klicken Sie auf **Generate** ⇒ **Call**. Dadurch wird ein korrekt konfiguriertes Objekt Call globals generiert. Platzieren Sie dieses Objekt links im Hauptfenster; das Objekt globals wird ausgeblendet.
- 7 Klicken Sie auf **Device** ⇒ **Formula** und platzieren Sie dieses Objekt oben in der Mitte des Hauptfensters. Benennen Sie es um in `Set up Excel Worksheet`. Verbinden Sie das Sequenzausgangs-Pin des Objekts Call globals mit dem Sequenzeingangs-Pin des Formelobjekts. Löschen Sie den Eingangsanschluss **A** von **Set Up Excel Worksheet**. (Öffnen Sie hierzu das Objektmenü und wählen Sie **Delete Terminal** ⇒ **Input**.)
- 8 Geben Sie in **Set up Excel Worksheet** den in Abbildung 151 gezeigten Text ein. Beachten Sie, dass für die Abgrenzung der Ausdrücke wie in ANSI C Semikolons verwendet werden.

## 6 Berichte mit ActiveX bequem erstellen



**Abbildung 151** Eine MS Excel-Tabelle einrichten

**Tabelle 25** Eine MS Excel-Tabelle in einem Formelobjekt einrichten

Befehl	Beschreibung
set sheet =	<p>Mit dem Schlüsselwort <b>set</b> wird der Variablen auf der linken Seite des Ausdrucks die Angabe rechts vom Zuordnungsoperator (in diesem Fall das Gleichheitszeichen) zugeordnet. set app definiert beispielsweise sheet.application, die als Excel-Tabelle definiert wurde.</p>
CreateObject („Excel.Sheet“).	<p>Erstellt eine neue Instanz des Automatisierungs-Servers (in diesem Fall MS Excel) und eine neue Arbeitsmappe. Jeder Automatisierungs-Server hat seine eigene Terminologie; die Syntax ist jedoch für alle gleich. Zum Erstellen eines Berichts in MS Word geben Sie beispielsweise <code>CreateObject ("Word.Document ")</code> ein, um Word zu starten und ein leeres Dokument zu erstellen.</p> <p>Bei Verwendung des Schlüsselworts set wird die rechte Seite des Objektzeigers der Variablen auf der linken Seite zugeordnet. Wenn set nicht angegeben ist, wird die Standardeigenschaft (häufig der Name) von der rechten Seite der linken Seite zugeordnet. Weitere Informationen hierzu finden Sie im Handbuch <i>VEE Pro Advanced Techniques</i>.</p>
worksheets(1);	<p>Excel wird jetzt durch <b>CreateObject(„Excel.Sheet“</b> mit einer neuen Arbeitsmappe ausgeführt. Sie möchten nun das erste Tabellenblatt adressieren. Fügen Sie der Anweisung <code>worksheets(1)</code> hinzu, sodass die gesamte Anweisung nun lautet:</p> <p><b>setsheet = CreateObject(„Excel.Sheet“).worksheets(1);</b></p> <p>Dadurch wird sheet für den Bericht auf Tabelle 1 festgelegt. (Sie können hierzu ein Beispiel anzeigen, indem Sie MS Excel öffnen und <b>Datei</b> ⇒ <b>Neu</b> wählen, um eine neue Arbeitsmappe zu erstellen. Sie werden feststellen, dass darin mehrere Blätter mit den Namen Tabelle1, Tabelle2 etc. vorhanden sind. Sie arbeiten auf Tabelle1.)</p>

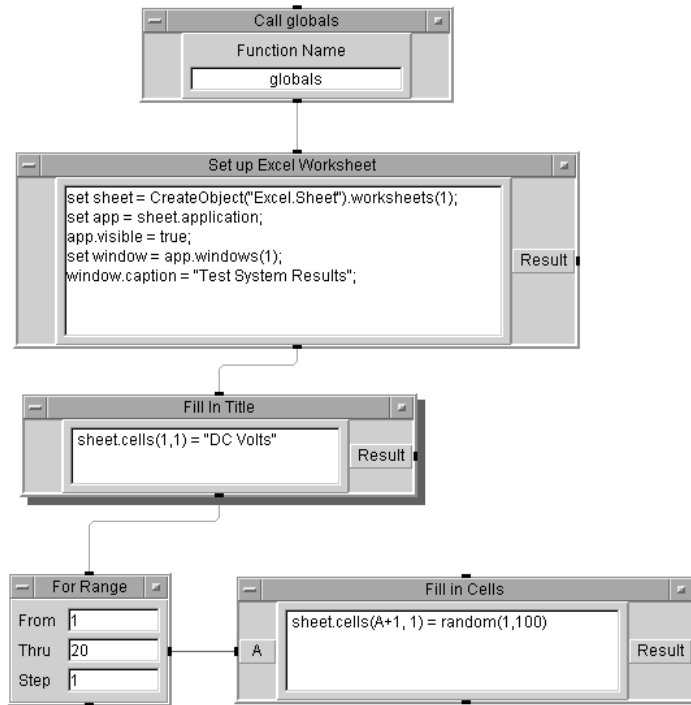
**Tabelle 25** Eine MS Excel-Tabelle in einem Formelobjekt einrichten

Befehl	Beschreibung
set app = sheet.application;	Ruft aus Excel einen Zeiger auf die gesamte Anwendung ab und nicht nur auf die aktuelle Arbeitsmappe. Hierzu wird die Eigenschaft Application der Tabelle abgefragt und auf die Variable app gesetzt.
app.visible = true;	Setzt die Sichtbarkeitseigenschaft app auf den Wert true. Excel wird für Sie sichtbar auf dem Bildschirm angezeigt.
set window = app.windows(1);	Verweist auf das erste Fenster.
window.caption = „Test System Results“;	Setzt die Legende für das erste Fenster auf „Test System Results.“

### HINWEIS

Weitere Informationen zu den Bibliotheken des Anwendungs-Servers finden Sie in einer Vielzahl von Büchern über die ActiveX-Automatisierung und MS Visual Basic. Informationen zur Bestellung von Büchern wie dem Office 2000 oder Office 97 Visual Basic Programmer's Guide finden Sie unter anderem im World Wide Web. Diese Bücher sind auch für VEE hilfreich, da die VEE-Syntax der Syntax von MS Visual Basic stark ähnelt.

- Erstellen Sie ein Formula-Objekt (unter **Device** ⇒ **Formula**). Klonen Sie es, um ein zweites Formula-Objekt zu erstellen. Erstellen Sie das Objekt For Range (unter **Flow** ⇒ **Repeat** ⇒ **For Range**). Benennen Sie die Objekte um, verbinden Sie sie und konfigurieren Sie sie wie in Abbildung 152. (Vergewissern Sie sich, dass Sie den Eingangsanschluss am Formula-Objekt **Fill in Title** gelöscht haben.)



**Abbildung 152** Titel und Daten in die Tabelle einfügen

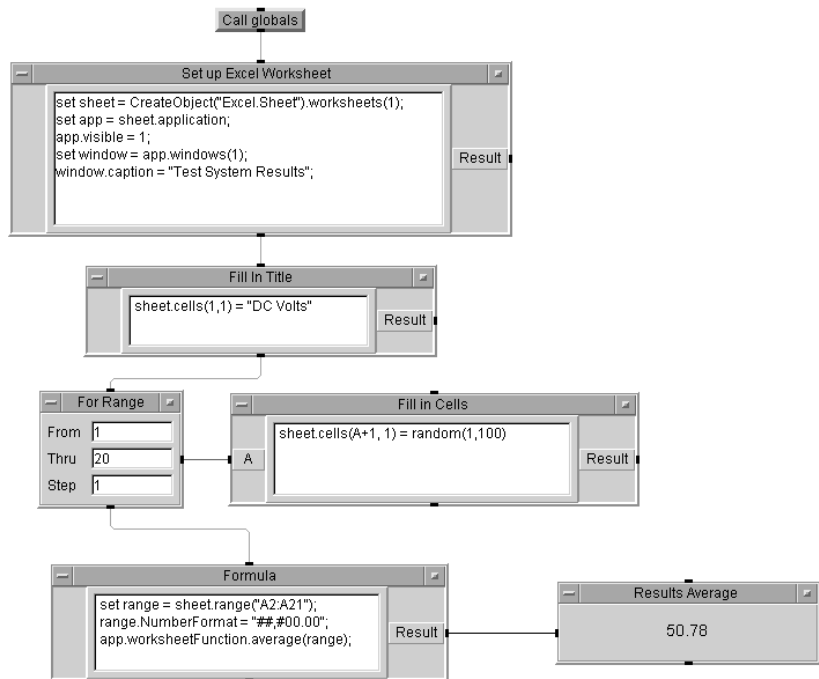
Die Ausdrücke in den Formelobjekten und dem Objekt For Range lauten wie folgt:

**Tabelle 26** Ausdrücke in den Formelobjekten und dem Objekt For Range

Formel	Beschreibung
sheet.cells(1,1) = „DC Volts“	Bezieht sich auf die erste Zeile und Spalte in der Excel-Tabelle. Der Text „DC Volts“ wird hier platziert. Setzt die Standardeigenschaft (den Wert) der Zelle <b>(1,1)</b> auf „ <b>DC Volts</b> “.
sheet.cells(A+1,1) = random(1,100)	Diese Anweisung ist eine Kurzschrift für <b>sheet.cells(A+1,1).value=random(1,100)</b> . Die Tabellenzelle in Zeile <b>A+1</b> , <b>col 1</b> ruft die Zeilennummer ab durch Addieren von <b>1</b> zum Wert des Eingangs-Pin <b>A</b> ; Spalte 1 wird jedoch beibehalten. Der von <b>random</b> zurückgegebene Wert zwischen <b>1</b> und <b>100</b> wird der angegebenen Zelle in der Tabelle zugeordnet.
von 1 bis 20, Schritt 1 (das Objekt „For Range“)	Da das Objekt For Range die Ganzzahlen von <b>1</b> bis <b>20</b> ausgibt, legt <b>Fill in Cells</b> die Zufallszahl in der angegebenen Zelle ab.

**10** Erstellen Sie ein **Formula**-Objekt und ein **AlphaNumeric**-Objekt. Benennen Sie diese Objekte anschließend um und konfigurieren und verbinden Sie sie wie in Abbildung 153.





**Abbildung 153** Das Programm „Results Average“

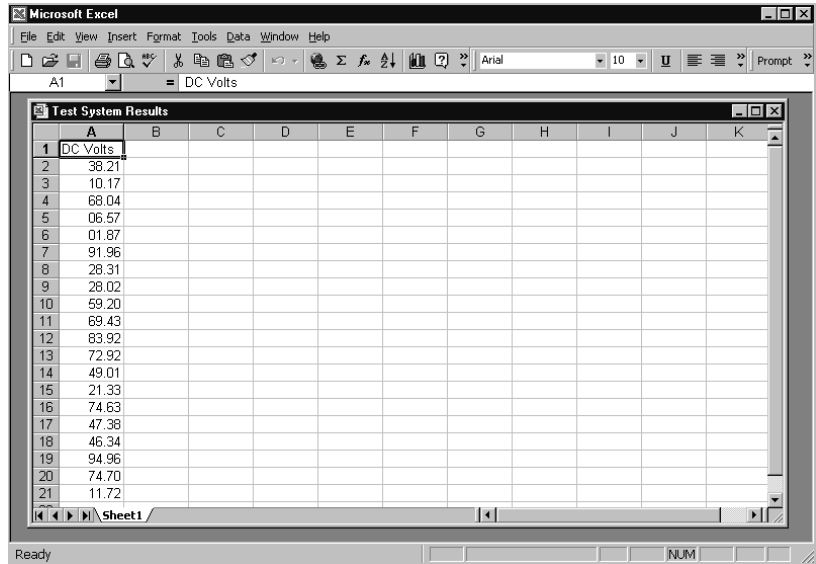
Die Einträge im Formula-Objekt lauten wie folgt:

**Tabelle 27** Formelobjekt-Einträge

Eintrag	Beschreibung
set range = sheet.range(„A2:A21“);	Setzt den VEE-Variablenbereich als Verweis auf den Bereich <b>A2 bis A21</b> der Excel-Tabelle. ( <b>A</b> bezieht sich auf die erste Spalte in einer Tabelle.)
range.NumberFormat = „##,##00.00“;	Ordnet das Format jeder der Zellen mit einem Pfundzeichen ( <b>#</b> ) zu; dies ermöglicht bei Bedarf größere Zahlen.
app.worksheetFunction.average(range);	Ruft die Excel-Methode average() auf, die den Durchschnittswert aus dem angegebenen Bereich von Werten angibt; dieser Durchschnittswert wird in Results Average angezeigt.

## 6 Berichte mit ActiveX bequem erstellen

11 Speichern Sie das Programm als **results\_average.vee**. Führen Sie das Programm aus. MS Excel wird mit einer Tabelle wie der in Abbildung 154 gezeigten aufgerufen.



The screenshot shows a Microsoft Excel window with a spreadsheet titled "Test System Results". The spreadsheet has a single column labeled "DC Volts" in column A, with rows numbered 1 through 21. The data values are as follows:

Row	DC Volts
1	DC Volts
2	38.21
3	10.17
4	68.04
5	06.57
6	01.87
7	91.96
8	28.31
9	28.02
10	59.20
11	69.43
12	83.92
13	72.92
14	49.01
15	21.33
16	74.63
17	47.38
18	46.34
19	94.96
20	74.70
21	11.72

**Abbildung 154** Excel-Tabelle für das Programm „Results Average“

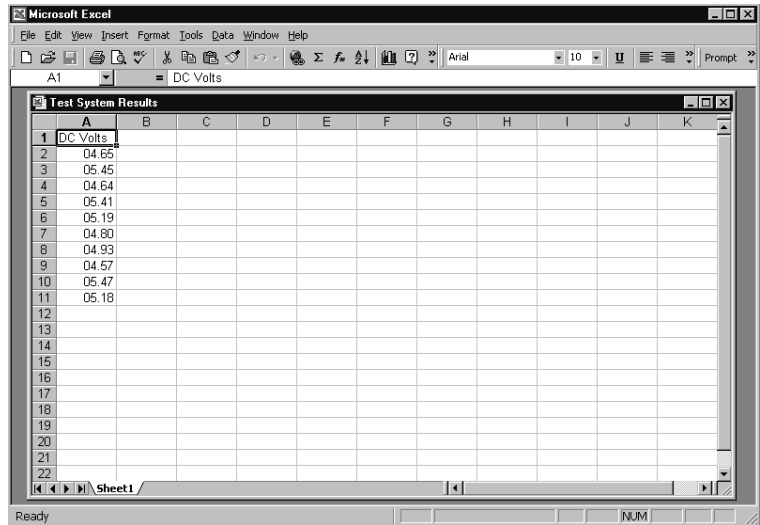
## Erstellen einer Agilent VEE-Vorlage für MS Excel

In dieser Übung erstellen Sie ein Programm zum Anzeigen eines Array von VEE-Testdaten in MS Excel. Sie können dieses Programm als Vorlage zum Anzeigen der Ergebnisse aus anderen Tests in MS Excel-Tabellen verwenden.

### Übung 6-2: Eine Agilent VEE-Vorlage für MS Excel erstellen

- 1 Öffnen Sie **results\_average.vee**.
- 2 Ändern Sie das Objekt For Range, sodass es 10 Mal eine Schleife durchläuft.
- 3 Fügen Sie dem Objekt **Fill in Cells** einen Eingang **B** hinzu und ändern Sie die Anweisung in dem Objekt wie folgt:  
`sheet.cells(A+1,1) = B[A-1]`.
- 4 Klicken Sie auf **Device** ⇒ **Formula**, benennen Sie das Objekt um in `Array of Test Data` und geben Sie die integrierten Funktionen `randomize(ramp(20), 4.5, 5.5)` ein. Sie generieren so ein Zufalls-Array aus 20 Elementen mit Werten von 4,5 bis 5,5. Löschen Sie das Eingangs-Pin und verbinden Sie das Datenausgangs-Pin mit dem Eingang **B** von **Fill in Cells**.
- 5 Ändern Sie den Bereich im Objekt **Formula** am unteren Rand der Anzeige von A21 in A11. Die Anweisung sollte jetzt lauten: **set range = sheet.range(„A2:A11“)**;
- 6 Speichern Sie das Programm als **report\_template.vee** und führen Sie es aus. Vergleichen Sie es mit der Excel-Tabelle in Abbildung 155 und dem fertigen Programm in Abbildung 156.

## 6 Berichte mit ActiveX bequem erstellen



The screenshot shows a Microsoft Excel window with the following data in the 'Test System Results' worksheet:

	A	B	C	D	E	F	G	H	I	J	K
1	DC Volts										
2	04.65										
3	05.45										
4	04.64										
5	05.41										
6	05.19										
7	04.80										
8	04.93										
9	04.57										
10	05.47										
11	05.18										
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											

Abbildung 155 Excel-Tabelle aus dem Array von Testdaten

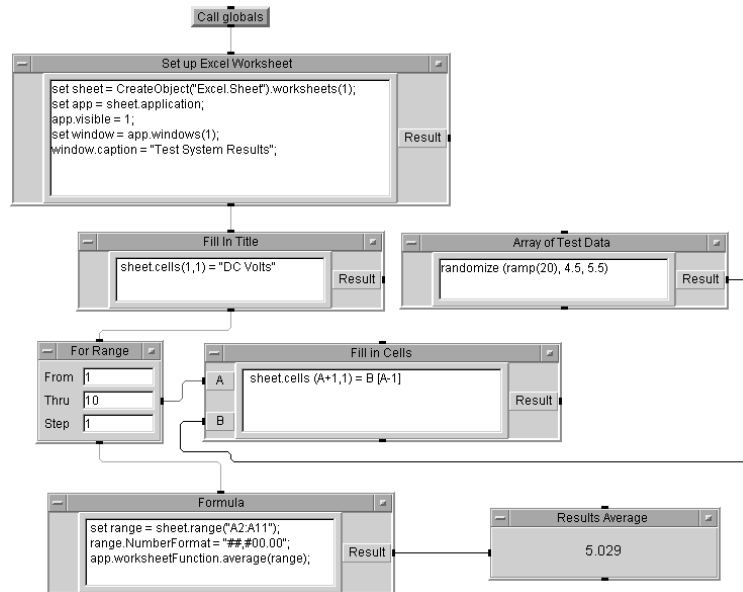


Abbildung 156 Programm für ein Array von Testdaten

Sie können dieses Programm als Vorlage zum Anzeigen von Testergebnissen in MS Excel verwenden. Legen Sie einfach die Testdaten in Arrays ab und ändern Sie den Rest der Vorlage so, dass die entsprechenden Zellen im richtigen Format ausgefüllt werden.

Zusätzliche Methoden und Eigenschaften in der MS Excel-Bibliothek können Sie über den Function & Object Browser anzeigen; wählen Sie Type: Objects, Sub Type: COM und Library: Excel. Sie können eine Class oder ein Member wählen und auf Help klicken, um die Hilfe des Automatisierungsservers aufzurufen, die durch den Hersteller (in diesem Fall Microsoft) zur Verfügung gestellt wird. Weitere ausführliche Informationen über diese Bibliotheken finden Sie in der Microsoft Dokumentation.

## Vertiefung

Generieren Sie eine Wellenform und entfernen Sie die Angabe Time Span, um ein Array zu erhalten. Erstellen Sie ein VEE-Objekt für MS Excel mit einer Tabelle und stellen Sie es auf eine Object-Variable ein. Setzt die Sichtbarkeitseigenschaft der Anwendung auf den Wert true. Legen Sie anschließend das 256-Punkt-Array im Tabellenbereich "A1:A256" in einem Arbeitsschritt ab statt Zelle für Zelle.

Verwenden Sie ein Objekt Unbuild Waveform. Verwenden Sie die Build-Array-Syntax [**a**] zum Erstellen eines 2D-Array aus einem 1D-Array. Rufen Sie anschließend die Funktion transpose() auf, um ein 256 x 1-Array zu generieren statt eines 1 x 256-Array. Somit kann Excel das Array in einem Schritt akzeptieren (siehe Abbildung 157).

## 6 Berichte mit ActiveX bequem erstellen

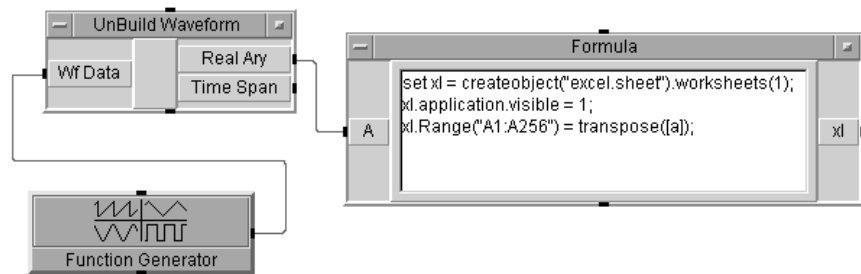


Abbildung 157 Programm zur weiteren Vertiefung

## Die Möglichkeiten mit MS Excel erweitern

Abbildung 158 zeigt ein umfangreicher ausgearbeitetes Programmbeispiel für das Anzeigen von Testergebnissen in MS Excel. Sie sehen, wie die Informationen über einige weitere Aufrufe in der MS Excel-Bibliothek die Vorlage zum Anzeigen der VEE-Daten in MS Excel erweitern.

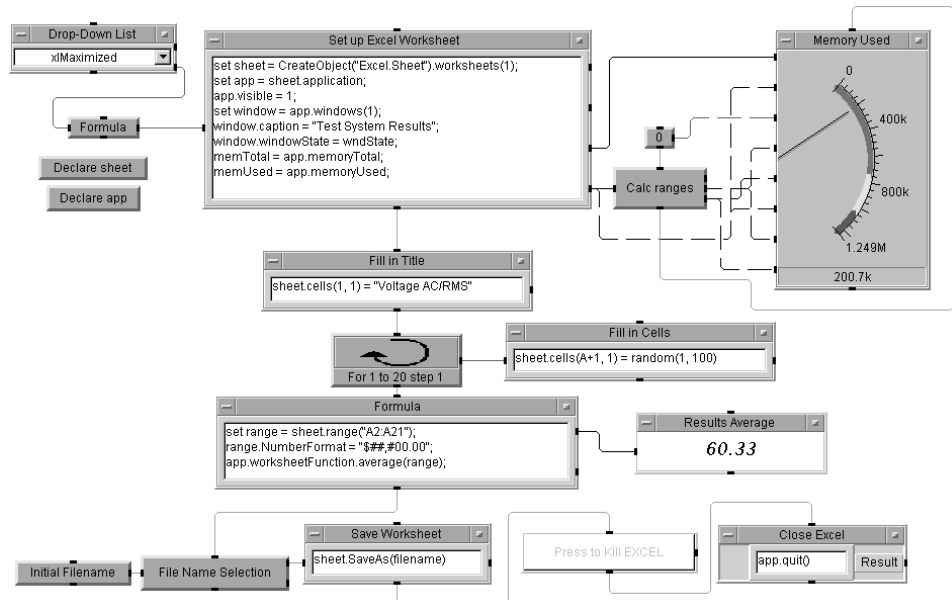


Abbildung 158 Ein Beispielprogramm VEE zu MS Excel

Die Einträge in Abbildung 158 lauten wie folgt:

**Tabelle 28** Beschreibung der Einträge in Abbildung 13

Element	Beschreibung
MS Excel Window Size	Beachten Sie die Dropdown-Liste oben rechts im Arbeitsbereich. Diese Liste ermöglicht die Auswahl einer von drei möglichen Optionen <b>xlMaximized</b> , <b>xlMinimized</b> , <b>xlNormal</b> für die Größe des Tabellenfensters beim Aufruf in Excel. Jeder Fenstergröße ist eine Zahl zugeordnet, die VEE berechnet und in der Variablen <b>wndState</b> ablegt. Dieser Wert wird dann der Eigenschaft <b>windowState</b> in der Excel-Bibliothek zugeordnet.
Memory Tracking	(Klicken Sie in der Ansicht Properties der Objekte Formula und Meter auf <b>Show Terminals</b> .) Beachten Sie die Eigenschaften <b>memoryTotal</b> und <b>memoryUsed</b> in der Excel-Bibliothek, die den VEE-Variablen <b>memTotal</b> und <b>memUsed</b> zugeordnet sind. Diese Werte werden zum Berechnen der Bereiche für die Konfiguration einer VEE-Messanzeige verwendet, bevor diese die von MS Excel verwendete Speichermenge anzeigt.
Number Format	Beachten Sie, wie einfach es ist, dem Zahlenformat ein Dollarzeichen hinzuzufügen.
sheet.SaveAs (Dateiname)	Die Methode SaveAs() wird von der Excel-Bibliothek zum automatischen Speichern der Tabelle aufgerufen. Beachten Sie, dass das Dialogfeld <b>File Name Selection</b> von VEE als Dialogfeld Save As verwendet wird. Sie fügen dieses Objekt über das Menü <b>Data</b> ⇒ <b>Dialog Box</b> ein. Der gewählte Dateiname wird anschließend als Parameter im Aufruf der Methode Excel SaveAs() verwendet.
Press to Kill Excel	Die Bestätigungsschaltfläche (OK) wurde verwendet, um das Schließen von Excel zu bestätigen.
Close Excel	Die Methode quit() wird aufgerufen, um MS Excel zu verlassen.

## MS Word für Agilent VEE-Berichte verwenden

In dieser Übung wird beschrieben, wie VEE-Testinformationen einschließlich Text, Zeitstempel und Screenshots von VEE-Dialogfeldern mit einer XY-Anzeige in einem MS Word-Dokument angezeigt werden können. Die Microsoft Dokumentation enthält ausführliche Beschreibungen, wie MS Word mithilfe der ActiveX-Automatisierung über andere Anwendungen gesteuert werden kann.

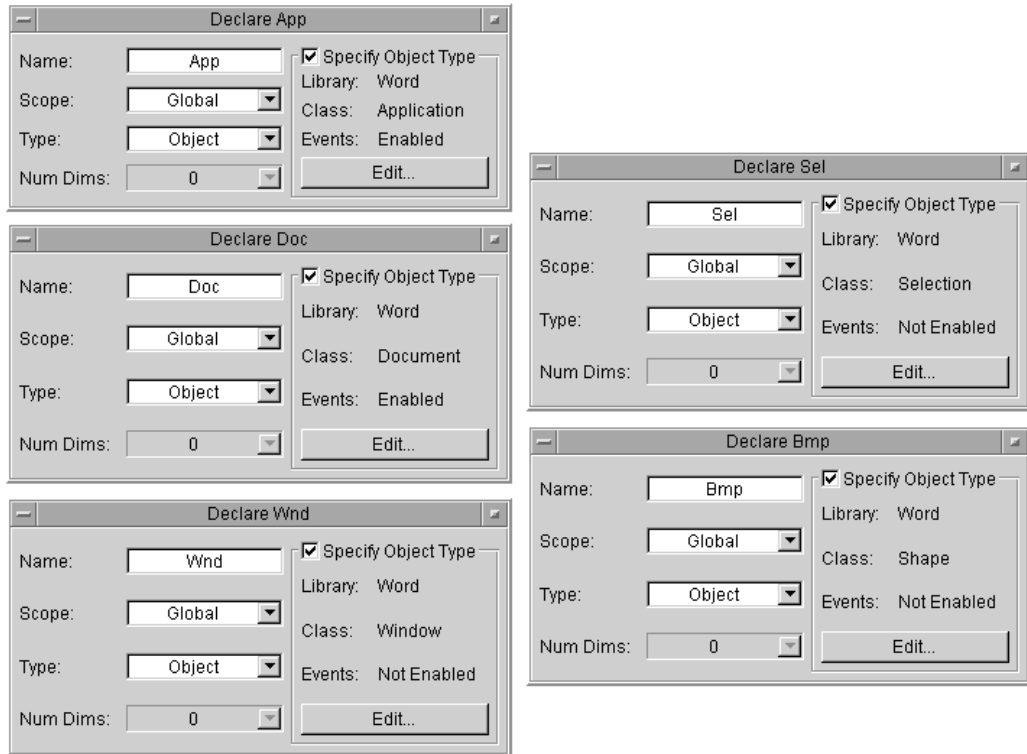
### Übung 6-3: MS Word für Agilent VEE-Berichte verwenden

Führen Sie zunächst die folgenden Schritte aus, um fünf Variablen als Typ Object zu deklarieren.

- 1 Klicken Sie auf **Device** ⇒ **ActiveX Automation References...** und wählen Sie Microsoft Word 9.0 Object Library.
- 2 Klicken Sie auf **Data** ⇒ **Variable** ⇒ **Declare Variable**.
  - a Wählen Sie unter Type den Eintrag Object. Klonen Sie das Objekt vier Mal.
  - b Benennen Sie die fünf Objektvariablen App, Doc, Wnd, Sel und Bmp.
  - c Bearbeiten Sie alle fünf Objekte im Dialogfeld Specify Object Type. Das Festlegen einer bestimmten Klasse innerhalb einer Bibliothek bietet Ihnen mehrere Vorteile: VEE kann eine Typenprüfung auf Programmfehler ausführen, und Sie können Ereignisse vom Automatisierungs-Server erfassen.
  - d Wählen Sie dazu Sub Type: COM und öffnen Sie das Dialogfeld Specify Object Type durch Klicken auf die Schaltfläche Edit. Wählen Sie für jedes der Objekte Library: Word. Wählen Sie die folgenden Klassen:
    - App verwendet Application
    - Sel verwendet Selection
    - Wnd verwendet Window
    - Doc verwendet Document
    - Bmp verwendet Shape



- e Wählen Sie Enable Events, sofern die jeweilige Klasse dies erlaubt. Minimieren Sie die fünf Objekte. Abbildung 159 zeigt die offene Ansicht der Variablen.



**Abbildung 159** Objektvariablen

- Erstellen Sie eine UserFunction mit dem Namen Graph, die eine virtuelle Quelle Function Generator zum Senden einer Sinuskurve an eine Waveform (Time)-Anzeige verwendet. Erstellen Sie eine Fensteransicht der Anzeige. Generieren Sie anschließend ein Objekt Call Graph im Hauptfenster. (Denken Sie daran, dass das UserFunction- Objektmenü eine einfache Möglichkeit zum Generieren eines Aufrufs bietet.)

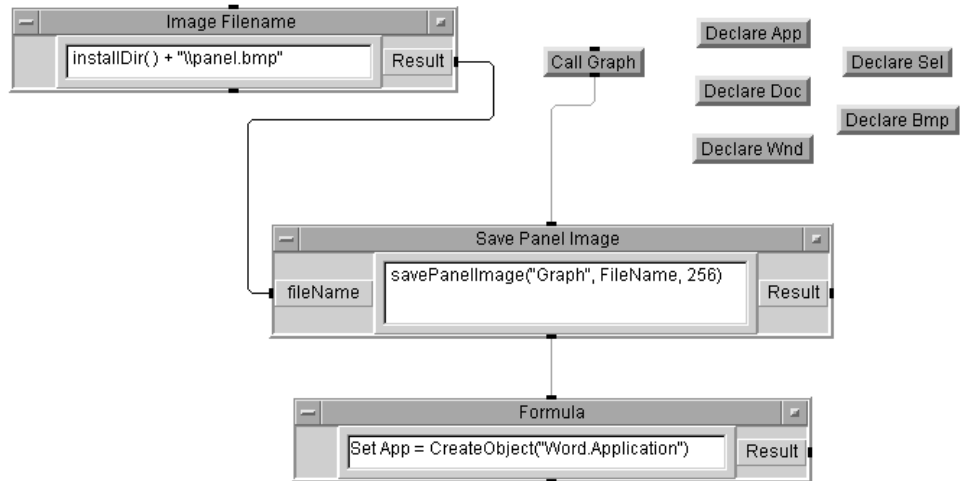
Erstellen Sie jetzt eine Bitmap-Datei des Fensters mit der Waveform-Anzeige zur Verwendung in dem Bericht in MS Word.

- 4 Klicken Sie zum Erstellen eines Dateinamens für die Bitmap-Datei auf **Device** ⇒ **Formula**. Benennen Sie das Objekt um `panel.bmp` im Eingabefeld ein. (Verwenden Sie die Escape-Sequenz `\\` zum Angeben des ASCII-Zeichens `\`.) Löschen Sie den Eingangsanschluss **A**.

Wenn Sie beispielsweise eine Installation in `c:\Program Files\Agilent\` durchgeführt haben, generieren Sie die folgende Textzeichenfolge am Ausgangs-Pin Result:

**C:\Program Files\Agilent\VEE Pro 6.0\panel.bmp.**

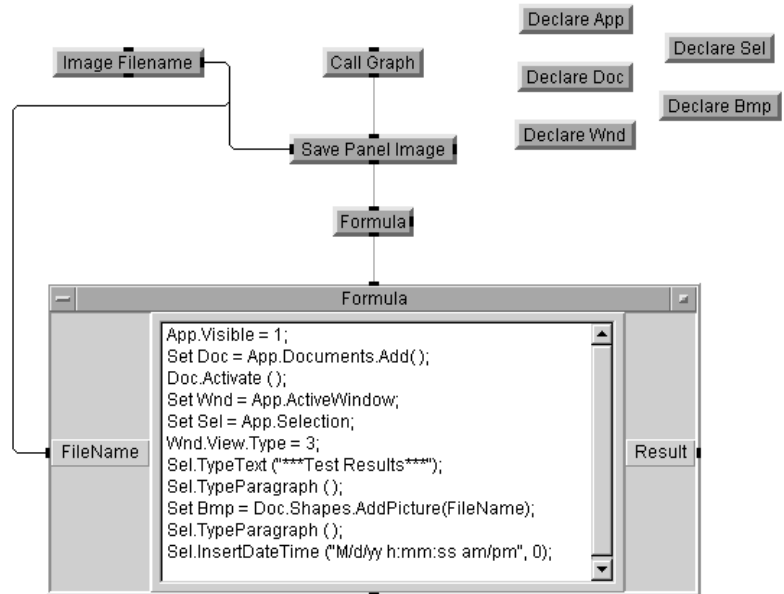
- 5 Erstellen Sie ein weiteres Formelobjekt und geben Sie `savePanelImage("Graph", FileName, 256)` ein. Benennen Sie den Eingangsanschluss um in `FileName`.
- 6 Dadurch wird der Screenshot aus UserFunction Graph in der Datei `panel.bmp` in dem Installationsverzeichnis mit einer Farbtiefe von 256 pro Pixel gespeichert.
- 7 Erstellen Sie ein weiteres Formula-Objekt und geben Sie die folgende Anweisung ein:  
`Set App = CreateObject("Word.Application")`  
Dadurch werden MS Word gestartet und die Objektvariable `app` zugeordnet, sodass sie auf diese Instanz der Anwendung verweist. Löschen Sie den Eingangsanschluss **A**. Verbinden Sie **Call Graph**, **ImageFileName** und **savePanelImage** wie in Abbildung 160 gezeigt.



**Abbildung 160** Anfang des Übungsprogramms 6-3

- 8 Klicken Sie auf **Device** ⇒ **Formula** und geben Sie die in Abbildung 161 gezeigten Anweisungen ein; die Anweisungen werden nachfolgend erläutert. Benennen Sie den Eingangsanschluss **A** um in **FileName**. Verbinden Sie das Dateneingangs- und das Sequenzeingangs-Pin wie in Abbildung 161.

## 6 Berichte mit ActiveX bequem erstellen



**Abbildung 161** ActiveX-Anweisungen hinzufügen

Beachten Sie in Abbildung 161, dass Sie die Eigenschafts- und Methodenaufrufe mit der Punktnotation des Objekts verschachteln können. Schlagen Sie in der ActiveX-Dokumentation die richtigen Eigenschaften für die Zielanwendungen nach. Sie können die in diesem Kapitel beschriebenen Eigenschaften und Methoden verwenden, um mit dem Generieren von Test- und Messungsberichten zu beginnen. Die Einträge im Formelobjekt haben die folgende Bedeutung:

**Tabelle 29** Einträge im Formula-Objekt

<b>Eintrag</b>	<b>Beschreibung</b>
App.Visible = 1;	Zeigt MS Word auf dem Bildschirm an.
Set Doc = App.Documents. Add();	Fügt in MS Word ein Dokument hinzu und ordnet es der Objektvariablen <b>Doc</b> zu. Hinweis: Im Excel-Beispiel wurde Excel über <b>CreateObject(Excel.Sheet)</b> mit einer leeren Tabelle gestartet. In diesem Beispiel wird Word gestartet, und die Methode Add() fügt ein leeres Dokument hinzu. Beide Anwendungen können auf jede dieser beiden Arten gestartet werden.
Doc.Activate();	Aktiviert das Dokument.
Set Wnd = App.Active Window;	Verwendet das Dokument in dem aktiven Fenster und ordnet es der Objektvariablen <b>Wnd</b> zu.
Set Sel = App.Selection;	Setzt den Fokus (die Auswahl) auf das Dokument und ordnet es der Objektvariablen <b>Sel</b> zu. Dies ermöglicht die Eingabe von Text.
Wnd.View.Type = 3;	Gibt den Typ des Fensters zum Anzeigen des Dokuments an. Die 3 kennzeichnet eine normale Fenstergröße. Mit A 1 würde das Fenster als Symbol dargestellt. Hinweis: Hier wird „3“ verwendet statt der Konstanten wdPageView, da die Konstante in der Typenbibliothek von Office 2000 nicht vorhanden ist.
Sel.TypeText (*** Test Results ***), Sel.TypeParagraph();	Fügt den Titel <b>*** Test Results ***</b> in das Dokument ein und gibt ein Zeilenvorschubzeichen aus.
Set Bmp = Doc.Shapes. AddPicture (FileName);	Legt das Bitmap panel.bmp in dem Dokument ab und ordnet diesen Aufruf in der Shapes Class der Objektvariablen Bmp zu.
Sel.TypeParagraph(); Sel.InsertDateTime (M/d/yy h:mm:ss am/pm, 0);	Versieht das Dokument mit einem Zeitstempel.

9 Fügen Sie drei weitere Formula-Objekte und ein Objekt If/Then/Else hinzu. Konfigurieren und verbinden Sie diese Objekte wie in Abbildung 162.

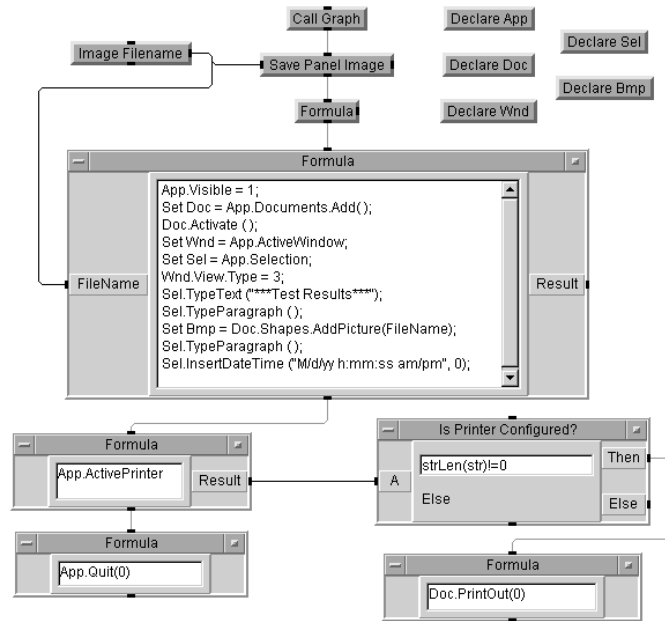


Abbildung 162 Das fertige Programm für den Bericht in MS Word

Die Einträge in den zusätzlichen Objekten haben die folgende Bedeutung:

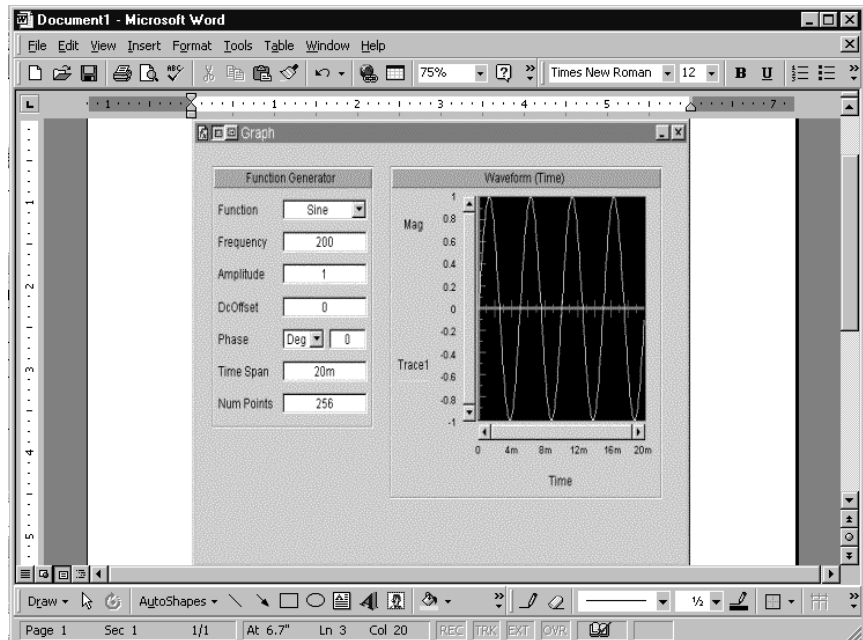
Tabelle 30 Zusätzliche Objekte in Abbildung 17

Object	Beschreibung
App.ActivePrinter	Fordert den Standarddrucker an über eine Zeichenfolge, in der der entsprechende Port angegeben ist.
strLen(str) != 0	Stellt sicher, dass ActivePrinter einen konfigurierten Drucker gefunden hat („if“ – wenn die Zeichenfolge am Eingang nicht Null ist, „then“ ...) und gibt eine 1 (=TRUE) am Pin Then aus. Dadurch erfolgt ein Ping auf das Formula-Objekt mit dem PrintOut-Aufruf.

**Tabelle 30** Zusätzliche Objekte in Abbildung 17

Object	Beschreibung
DocPrintOut(0)	Druckt das Dokument.
App.Quit(0)	Schließt die Anwendung MS Word.

**10** Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 163. (Wenn die Farben des Screenshot optimiert werden müssen, verkleinern Sie alle geöffneten Anwendungen auf Symbolgröße, damit dem PC die gesamte Farbpalette zur Verfügung steht.)



**Abbildung 163** Das mit Übung 6-3 erstellte MS Word-Dokument

Weitere Informationen zum Steuern von MS Excel und MS Word über die ActiveX-Automatisierung finden Sie in der Microsoft Dokumentation. Genauso können Sie aber auch andere Server-Anwendungen, die die ActiveX-Automatisierung unterstützen,

## **6 Berichte mit ActiveX bequem erstellen**

auf diese Weise steuern. Die ActiveX-Automatisierung wird gelegentlich auch als Automation oder OLE-Automation bezeichnet.

Weitere Informationen zu ActiveX-Steuerelementen finden Sie in Kapitel „Benutzerschnittstellen verwenden“. Weitere Informationen zur Verwendung von ActiveX über ein MS Visual Basic-Programm zum Steuern von VEE finden Sie in Kapitel „Überblick“.



## Kapitel-Checkliste

- Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.
- Das Grundkonzept hinter der ActiveX-Automatisierung in VEE beschreiben.
- Daten von VEE an MS Excel senden.
- Eine allgemeine Vorlage zum Senden von Arrays aus Testdaten an eine MS Excel-Tabelle verwenden. (Sie sollten wissen, wie Sie ein Array in einem einzigen Arbeitsschritt an eine Tabelle senden können.)
- Die erweiterten Funktionen der MS Excel-Bibliothek nutzen, z. B. zum Ermitteln des von einem Programm belegten Speichers.
- Text, Zeitstempel und Anzeige-Bitmap von VEE aus an MS Word senden.

## **6    Berichte mit ActiveX bequem erstellen**

# 7

## **.NET mit VEE verwenden**

Was ist .NET?	289
.NET-Terminologie	329
VEE und .NET Framework	290
Einen Namespace in VEE importieren	296
VEE und primäre Interop-Assemblies	300
Programmierverfahren	301
.NET- und IVI-Treiber	324
Die VEE-Laufzeitversion weitergeben	327
VEE- und .NET-Sicherheit	328
.NET-Terminologie	329

## **.NET mit VEE verwenden**

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Grundlegende .NET-Begriffe
- Einen Namespace in VEE importieren
- Beschreibung und Bedeutung einer PIA
- Spezielle .NET-Überlegungen zur Verteilung der VEE-Laufzeitversion
- VEE- und .NET-Sicherheit

*Erforderliche Zeit für dieses Kapitel: 1,5 Stunden*

## Was ist .NET?

Visual Studio .NET ist die neueste Entwicklungsplattform von Microsoft. Im Gegensatz zu früheren Produkten wie Visual Studio 6 wird .NET den Anforderungen sowohl von Desktop-Programmierern als auch von Web-Entwicklern gerecht. Die Stärke von .NET aus der VEE-Perspektive ist das kostenlose .NET Framework, das mit VEE 7.0 verteilt wird. Diese kostenfreie Laufzeitversion ist ein wesentlicher Bestandteil mehrerer neuer VEE-Funktionen.

Die wichtigsten Funktionen und Vorteile des .NET Framework für VEE sind die Framework Class Library (FCL) und die COM-Interop-Technologie. Die Framework Class Library ist insbesondere für VEE-Programmierer hilfreich. Es gibt Hunderte neuer Eigenschaften und Methoden, die jetzt über den Function & Object Browser zur Verfügung stehen. Durch dieses neue Feature kann Ihr Code Funktionalität wie Datei- und Verzeichnisverwaltung, einfachere String-Verwaltung, vereinfachte Ansichten der Betriebssystemumgebung, Manipulation von Webseiten, Zugriff auf Betriebssystemabläufe, Lesen der Registrierungsdateien u.v.m. ableiten.

.NET bietet auch eine native COM-Interop-Technologie. Das bedeutet, dass auf eine COM-Komponente wie auf ein .NET-Objekt zugegriffen werden kann und umgekehrt.

Am Ende dieses Kapitels finden Sie eine Reihe von Definitionen der Microsoft-Terminologie, die hier verwendet wird. Bitte schlagen Sie dort bei Bedarf nach.

### VEE und .NET Framework

Bevor Sie sich der Interaktion des .NET Framework mit VEE zuwenden, informieren Sie sich über das .NET Framework. Schlagen Sie dazu im MSDN-Index unter „.NET Framework – getting started“ unter der folgenden (recht langen) Adresse nach: (<http://msdn.microsoft.com/library/en-us/cpguide/html/cpovrintroductiontonetframeworksdk.asp>).

Im Menü Device ist ein neuer Befehl namens **.NET Assembly References** verfügbar, der dem Befehl **References for ActiveX** sehr ähnelt. Wie bei den Definitionen ausgeführt, wird eine Assembly durch einen Verweis verfügbar gemacht. Klicken Sie auf diesen Menübefehl. Es wird ein Fenster ähnlich wie in Abbildung 164 geöffnet.

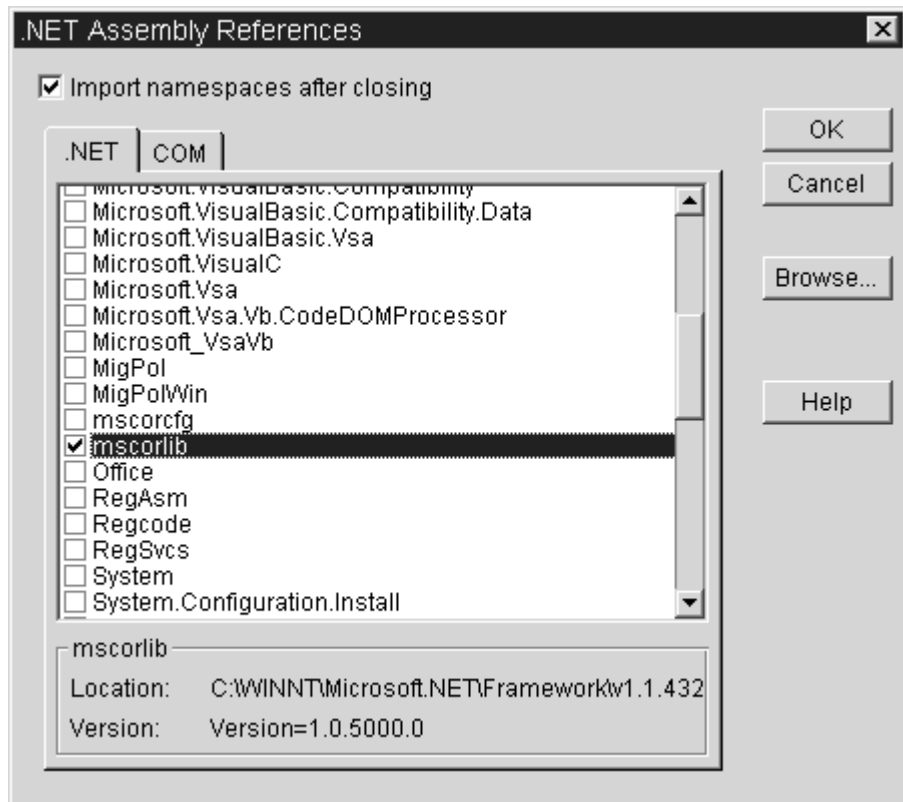


Abbildung 164 Das Dialogfeld Import Namespaces

## .NET-Assembly-Verweise

Es gibt viele Assemblies, unter denen Sie wählen können. Diese befinden sich in dem Verzeichnis, in dem Ihr VEE-Programm gespeichert ist, oder in dem Verzeichnis, in dem .NET Framework installiert ist. Wenn Sie eine Assembly verwenden, die sich in keinem dieser beiden Verzeichnisse befindet, wählen Sie sie mit Hilfe der Schaltfläche **Browse**. Durch Aktivieren des Kontrollkästchens **Import namespaces after closing** werden die gewählten Namespaces beim Schließen des Dialogfeldes importiert. Das Importieren eines Namespace entspricht der Anweisung **using** in C# oder der Anweisung **Imports** in Visual Basic.

In der obigen Abbildung ist die am häufigsten verwendete Assembly, **mscorlib**, markiert. Wählen Sie diese Assembly. **mscorlib** enthält viele Funktionen, die für VEE-Benutzer hilfreich sind, z. B. Dateisystemmanipulation, Auflistungsverwaltung und Dateitypkonvertierung. Eine weitere sehr häufig genutzte Assembly ist **System**, die Funktionen wie Prozessverwaltung, Webanforderung und -antwort und reguläre Ausdrücke enthält.

Auf der Registerkarte COM finden Sie eine Liste aller COM-Typbibliotheken, die aktuell auf Ihrem Computer registriert sind. Hierbei handelt es sich um nahezu dieselbe Liste von Typbibliotheken, die im Dialogfeld ActiveX automation references angezeigt wird.

Wann greifen Sie auf welche der beiden Wahlmöglichkeiten zurück? Sie sollten das Dialogfeld ActiveX automation references verwenden, wenn alle folgenden Bedingungen zutreffen:

- Die Bibliothek wird dort aufgeführt.
- Die Bibliothek ist in den früheren Versionen von VEE voll funktionsfähig.
- Die Bibliothek hat keine primäre Interop-Assembly (PIA).

Durch den direkten Zugriff auf das Dialogfeld wird das Hinzufügen einer weiteren Ebene, der .NET- und COM-Interop-Ebene, vermieden.

In den folgenden Fällen sollten Sie die hier erläuterte .NET/COM-Interop-Funktionalität nutzen:

- Die Bibliothek ist nicht im Dialogfeld ActiveX automation references aufgelistet.
- Nicht die gesamte Funktionalität der Bibliothek wird importiert.
- Sie wissen, dass die COM-Bibliothek eine PIA hat.

Beispiel: Nicht alle IVI-COM-Treiberschnittstellen sind in der Liste ActiveX Objects des Function & Object Browser enthalten. Dank der .NET-Interop-Funktionalität können Sie jedoch alle IVI-Treiberschnittstellen sehen.



Wenn Sie den Menübefehl `.NET Assembly References` verwenden und eine COM-Typbibliothek aus der Liste wählen, wird versucht, die COM-Typbibliothek als `.NET-Assembly` zu importieren. Wenn eine PIA vorhanden ist, nutzt VEE diese. Andernfalls erzeugt VEE eine Interop-Assembly (oder Assemblies, wenn die COM-Typbibliothek auf andere Typbibliotheken verweist) in dem Verzeichnis, in dem das Programm gespeichert ist. Nach Empfang der Interop-Assembly werden alle Typinformationen verfügbar.

### HINWEIS

Um den Systemaufwand zu vermeiden, der entsteht, wenn VEE die generierten Interop-Assemblies aus einem temporären Verzeichnis in Ihr VEE-Programmverzeichnis kopieren muss, speichern Sie zuerst das VEE-Programm. Auf diese Weise werden die generierten Interop-Assemblies direkt in Ihr VEE-Programmverzeichnis gespeichert.

---

Wählen Sie die Registerkarte `COM`. Der Aufruf kann einige Zeit in Anspruch nehmen, da Ihre Registrierungsdatenbank nach `COM-Typbibliotheken` durchsucht wird. Wählen Sie das `Microsoft-ActiveX-Plugin`.

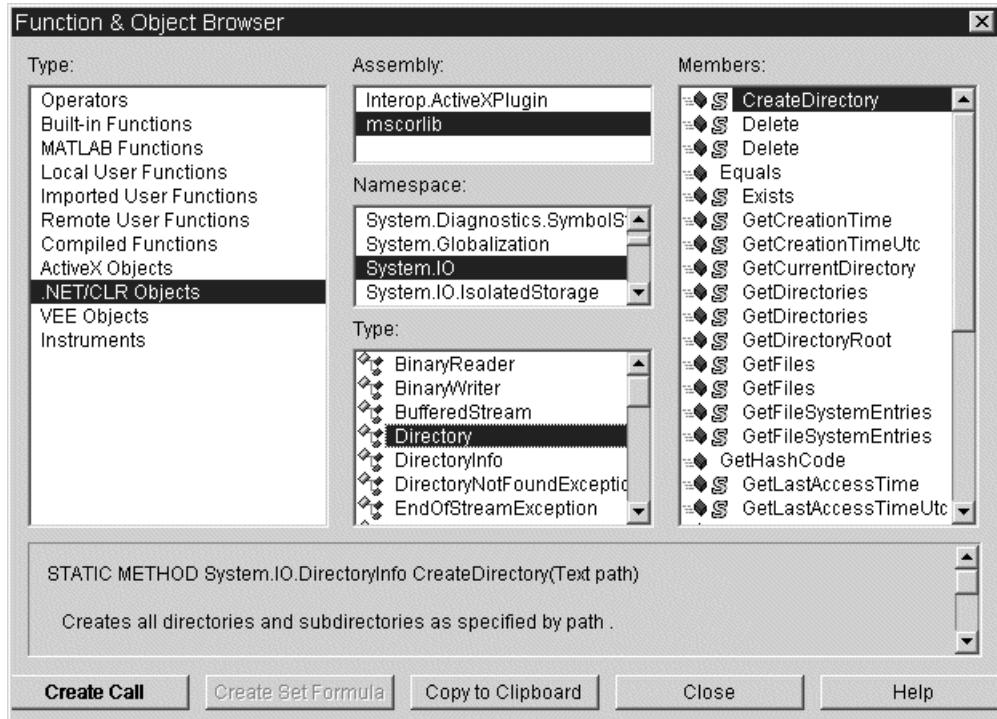
Es wird eine Abfrage der zu importierenden Namespaces eingeblendet. Dies ist ein optionaler Schritt. Wählen Sie nun `Interop.ActiveXPlugin`, `System` und `System.IO`.

### HINWEIS

Das Dialogfeld `Import .NET Namespaces` wird jedes Mal aufgerufen, sofern Sie nicht das entsprechende Kontrollkästchen im Dialogfeld `.NET Assembly References` deaktivieren.

---


Wählen Sie im Menü `Device` den Befehl **Function & Object Browser**. Einer der Menübefehle ist **.NET/CLR Objects**. Wählen Sie diesen Befehl. Beachten Sie, wie das Fenster **Assembly** in Abbildung 165 die beiden früher gewählten Optionen widerspiegelt.




**Abbildung 165** Function & Object Browser – .NET-Objekte

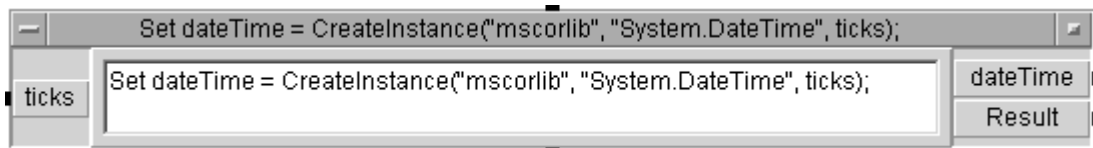
Alle Namespaces der gewählten Assembly werden im Listenfeld **Namespace** aufgeführt. Für jeden Namespace werden alle Typen im Listenfeld **Type** und alle Member jedes Typs im Listenfeld **Members** aufgeführt. Member sind Konstruktoren, Felder, Eigenschaften, Methoden und Enumerationen.

Wenn Sie ActiveX in VEE verwendet haben, werden Sie feststellen, dass die Wahl eines .NET-Klassenmember nahezu derselbe Vorgang ist. Die Unterschiede sind nur gering:


- 1 COM hat keine freigegebenen (statischen) Member. Wenn das .NET-Member ein statisches Member ist, zeigt der Hilfebereich das Schlüsselwort „STATIC“ an, und neben dem Symbol steht ein . Zum Aufrufen eines statischen Member müssen Sie es auf einer Klasse und nicht auf einer Objektinstanz aufrufen. Sie können den Function & Object-Browser

heranziehen, um sich mit der Syntax vertraut zu machen. Ein Beispiel hierfür finden Sie unter „Übung 7-2: .NET zum Ausführen von DateTime-Operationen verwenden“ auf Seite 317.

- 2 Bei Instanzobjektvariablen ist der erste Buchstabe ein Kleinbuchstabe. Dies weist darauf hin, dass Sie eine Methode oder Eigenschaft auf einem Instanzmember eines Objekts aufrufen. Es bedeutet auch, dass Sie ein .NET-Objekt erzeugen müssen.
- 3 Um ein .NET-Objekt zu erstellen, wählen Sie einen Konstruktor (für Konstruktoren und Methoden wird das Symbol  verwendet) in der Memberliste. Wie in der obigen Abbildung ist die Schaltfläche Create Instance verfügbar. Wenn Sie diese Schaltfläche aktivieren, wird eine Formelschablone zum Erstellen eines neuen .NET-Objekts generiert. Beachten Sie das Ausgangs-Pin dateTime in Abbildung 166, das von VEE generiert wird. Sie können dieses Ausgangs-Pin mit dem Eingangs-Pin jedes Formelobjekts verbinden, das dieses .NET-Objekt als Eingang benötigt. Ein Beispiel hierzu finden Sie unter „Übung 7-3: .NET zum Abrufen von Dateinformationen verwenden“ auf Seite 322.



**Abbildung 166** Erstellen eines .NET-Objekts

- 4 Die Enumerationen () in .NET sind stark typisiert, so dass Sie nicht mehr eine Enumeration durch eine ganzzahlige Konstante ersetzen können, wie es mit ActiveX-Objekten möglich war. Sie müssen den vollständigen Enumerationsnamen eingeben (ohne den Namespace, wenn Sie ihn importiert haben) oder auf die Schaltfläche Create Formula zurückgreifen. Enumerationsbeispiele finden Sie im Verzeichnis examples\dotnet.

## Einen Namespace in VEE importieren

Hier noch einmal die Definition für einen Namespace: .NET Framework-Typen verwenden ein Benennungsschema mit Punktmatrix, das eine Hierarchie angibt. Diese Technik gruppiert zugehörige Typen in Namespaces, so dass sie leichter durchsucht werden können und leichter auf sie verwiesen werden kann. Der erste Teil des vollständigen Namens (bis zum Punkt ganz rechts) ist der Namespace-Name. Der letzte Teil des Namens ist der Typname. Ein Beispiel: System.Collections.ArrayList steht für den Typ ArrayList, der zum Namespace System.Collections\* gehört. Die Typen in System.Collections können zur Manipulation von Objektauflistungen verwendet werden.

Das Importieren eines Namespace entspricht der Anweisung **Imports** in Visual Basic oder der Anweisung **using** in C#. In beiden Fällen müssen Sie die Verwendung eines Typs aus diesem Namespace nicht qualifizieren. VEE unterstützt zwei Methoden zum Importieren eines Namespace. Sie können entweder Import .NET Namespaces direkt im Menü Devices wählen oder das entsprechende Kontrollkästchen im Dialogfeld .NET Assembly References aktivieren (siehe Abbildung 168).

Eine vollständige Liste der verfügbaren .NET-Namespaces wird angezeigt. Um einen Namespace zu importieren, aktivieren Sie einfach das zugehörige Kontrollkästchen. Wenn die gewählte *Assembly* keine Namespaces hat, ist die Liste leer.

\*Verschachtelte Klassen bilden eine Ausnahme zu dieser generellen Regel. Weitere Einzelheiten entnehmen Sie bitte der MSDN-Dokumentation.

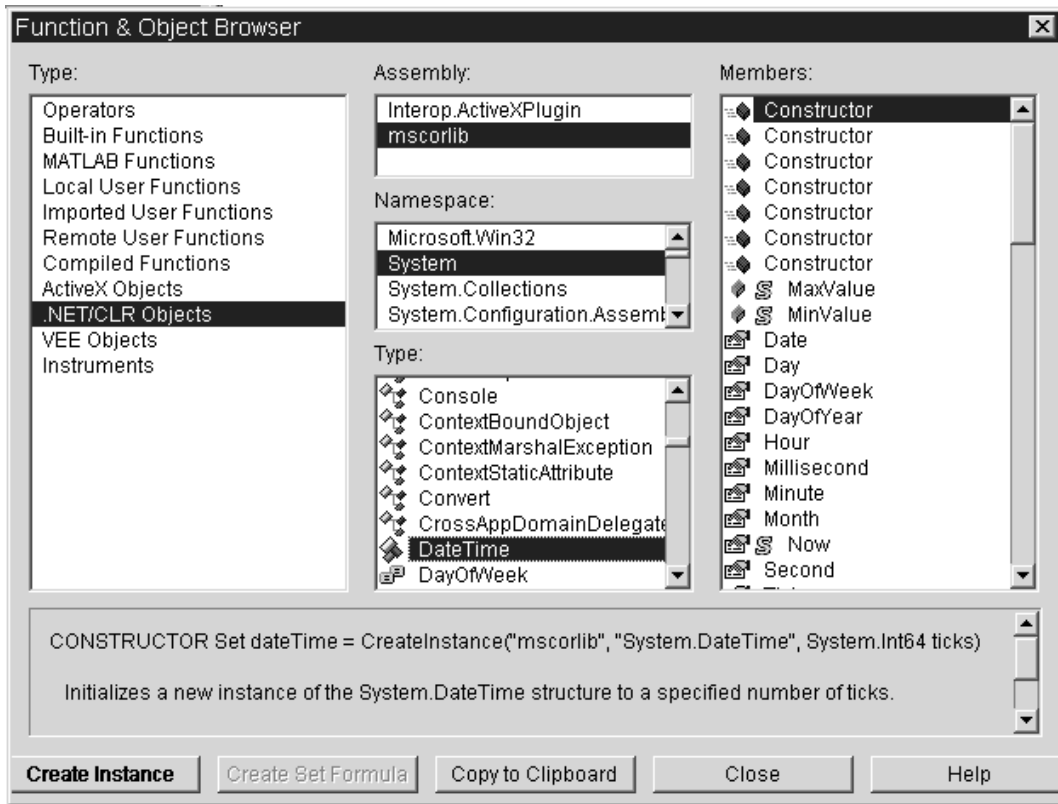


Abbildung 167 Assemblies, Namespaces, Typen und Members

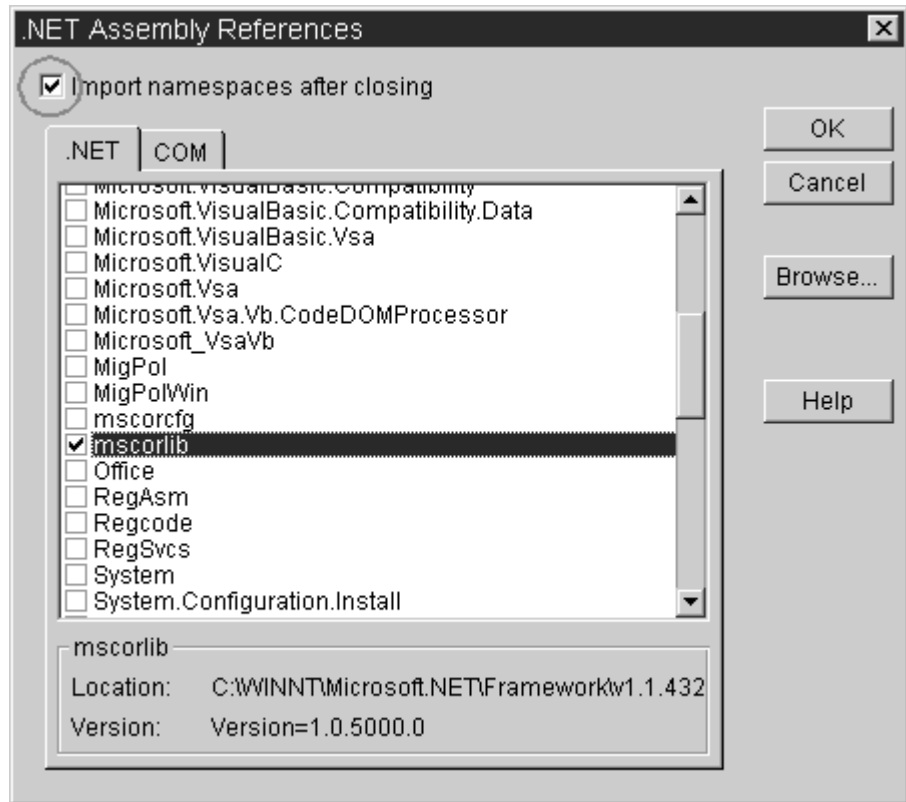
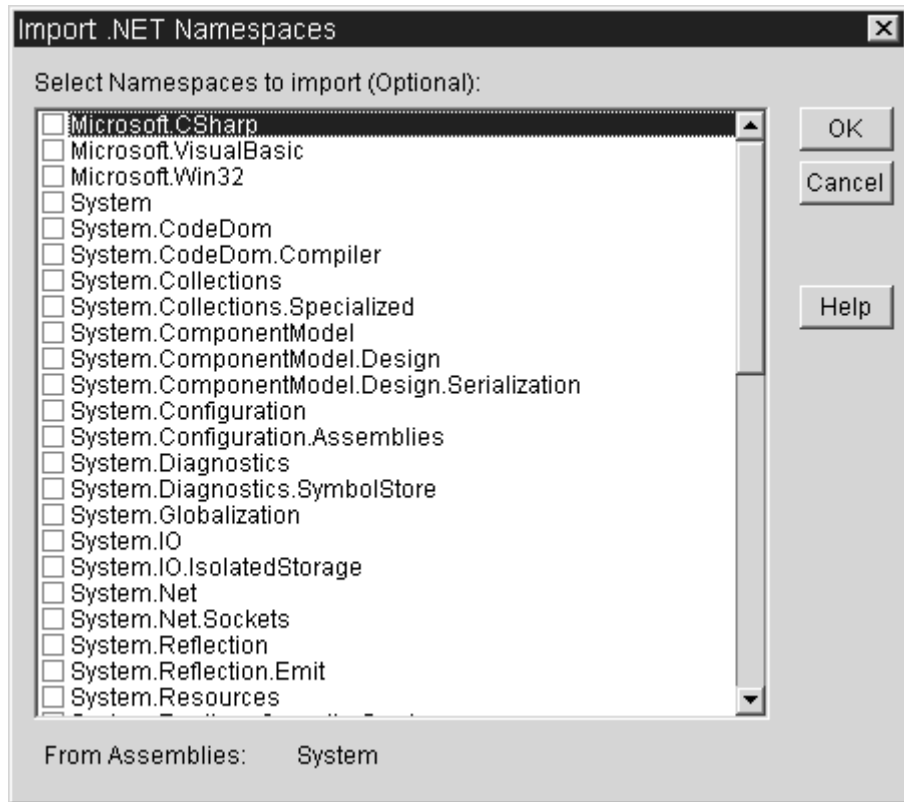


Abbildung 168 .NET Assembly References – Namespaces importieren



**Abbildung 169** Listenfeld zur Wahl von Namespaces

Wenn Sie keinen Namespace für eine Klasse importiert haben, enthält die Formelschablone, die für die statischen Member und Enumerations für diese Klasse generiert wurde, die vollständig qualifizierten Typnamen. Auf diese Weise werden Ihre Formelobjekte viel länger.

## VEE und primäre Interop-Assemblies

Wenn Sie die Registerkarte COM in **Device** ⇒ **.NET Assembly References** wählen, durchsucht VEE die Registrierungsdatenbank und ermittelt alle registrierten COM-Typbibliotheken auf Ihrem System. Wenn bei der Wahl von COM-Typbibliotheken die Bibliothek eine PIA hat und der Schlüssel PrimaryInteropAssemblyCodeBase für die COM-Typbibliothek registriert ist, wird die PIA-Position im Beschreibungsbereich von Function & Object Browser angezeigt. PrimaryInteropAssemblyCodeBase ist nur registriert, wenn die PIA mit der Option /codebase registriert wurde. Nicht alle PIAs werden auf diese Weise registriert. Aktuell sind alle Agilent-IVI-COM-Treiber mit der Option /codebase registriert, viele Assemblies sind dies jedoch nicht. Wenn Sie eine COM-Typbibliothek aktivieren, wird lediglich die Registrierungsdatenbank durchsucht. Eine eventuell vorhandene Interop-Assembly wird nicht geladen. Sobald Sie **OK** wählen, wird automatisch mindestens eine Interop-Assembly generiert, wenn keine PIA vorhanden ist. Die Interop-Assemblies werden auch geladen, nachdem auf die Schaltfläche **OK** geklickt wurde.

Wenn Sie jedoch **Browse** wählen und eine PIA oder eine Interop-Assembly direkt wählen, lädt VEE die Interop-Assembly tatsächlich, um herauszufinden, zu welcher COM-Typbibliothek sie gehört, und ob es sich dabei um eine primäre Interop-Assembly handelt. Dies ist der Hauptunterschied zwischen der Wahl mit Browse und dem Aktivieren in der Liste der Registerkarte COM. Nachdem die Interop-Assembly geladen wurde, ermittelt VEE, zu welcher COM-Typbibliothek sie gehört, und aktiviert dann die entsprechenden COM-Typbibliotheken in der Liste der Registerkarte COM.



## Programmierverfahren

### Datentypen zwischen .NET und VEE konvertieren

VEE konvertiert Datentypen zwischen VEE und .NET automatisch für alle Datentypen, die VEE und .NET nativ unterstützen, z. B. Int16, Int32, Real64 usw. Sie können diese automatische Konvertierung rückgängig machen, um die Daten als echtes .NET-Objekt zu nutzen. In diesen Fällen können Sie die neue Typkonvertierungsfunktion `asClrType()` verwenden, um einen VEE-Datentyp in einen .NET/CLR-Typ umzuwandeln. Ausführliche Informationen hierzu finden Sie in den nachfolgenden Konvertierungstabellen.

.NET-Operationen in VEE akzeptieren normalerweise einen Genauigkeitserweiternden Datentyp (Typ-Erweiterung) als Parameter, nicht jedoch einen Genauigkeitsreduzierenden Datentyp (Typ-Einschränkung). Sie können beispielsweise UInt8 von VEE in eine .NET-Operation übergeben, die einen Parameter vom Typ `System.Int16` benötigt, nicht jedoch umgekehrt. Wenn die .NET-Operation einen Parameter `ByRef` benötigt und Sie sein Ergebnis zurück abrufen möchten, muss auch der Datentyp exakt übereinstimmen. Wenn beispielsweise die .NET-Methode einen Parameter vom Typ `ByRef Int32` benötigt, können Sie nur den VEE-Datentyp `ByRef Int32` übergeben. Falls Ihnen der Rückabruf der Ergebnisse nicht wichtig ist, können Sie das Schlüsselwort `ByRef` überspringen. Dann würden die normalen Parameterregeln gelten.

Die folgenden Tabellen zeigen die Datentyp-Konvertierungsfunktionen zwischen VEE und .NET.

**Tabelle 31** .NET-Skalardatentypen in VEE-Datentypen  
in VEE 7.0 konvertieren

Konvertieren des .NET-Datentyps	Konvertieren in den VEE-Datentyp	Anmerkungen
System.Boolean	Int16	Verwenden Sie <code>isVariantBool()</code> , um zu bestimmen, ob der VEE-Wert Int16 vom Typ System.Boolean war.
System.Byte	UInt8	
System.Char	Object	Sie können eine Methode <code>System.Convert.To*</code> verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. <code>ToInt32()</code> .
System.DateTime	Object	Siehe <code>examples\dotnet\DateTime.vee</code> .
System.Decimal	Object	Sie können eine Methode <code>System.Convert.To*</code> verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. <code>ToInt32()</code> . Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf ( <code>System.OverflowException</code> ) angibt. Die Struktur <code>System.Decimal</code> selbst bietet auch eine Reihe von Konvertierungsmethoden.
System.Double	Real64	
System.Enum	Object	Siehe <code>examples\dotnet\FileInfo.vee</code> .
System.Int16	Int16	
System.Int32	Int32	

System.Int64	Object	Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt.
System.SByte	Object	Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32().
System.Single	Real32	
System.String	Text	VEE konvertiert String automatisch in Text. Wenn Sie die vielen nützlichen Funktionen der Klasse System.String nutzen möchten, können Sie mit asClrType() Text wieder zurück in System.String konvertieren. Beispiel:  Stellen Sie ein: <pre>dotNetString = asClrType(veeText, System.String); ModifiedString = dotnetString.Replace ("","_");</pre> Ein Beispiel finden Sie unter examples\dotnet\stringsplit.vee.
System.UInt16	Object	Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32().

System.UInt32	Object	Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt.
System.UInt64	Object	Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt.
System.Object	Object	

**Tabelle 32** VEE-Datentypen in .NET-Skalardatentypen in VEE 7.0 konvertieren

Konvertieren des VEE-Datentyps	Konvertieren in den .NET-Datentyp	Anmerkungen
Int16	System.Int16	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Int32	System.Int32	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().

Real32	System.Single	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Real64	System.Double	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Text	System.String	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
UInt8	System.Byte	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
<skalar vom Typ*>	System.Boolean	Verwenden Sie asVariantBool() auf jedem VEE-Skalardatentyp, der in einen Datentyp Int16 (UInt8, Int16, Int32, Real32, Real64, Text) umgewandelt werden kann. Sie können auch das generischere asClrType() verwenden, z. B. asClrType(vescalalar, System.Boolean).
Date/Time	System.DateTime	Date/Time von VEE wird als Typ Real64 gespeichert. Verwenden Sie asClrType(veeDateTime, System.DateTime).  Siehe examples\dotnet\DateTime.vee.
Object	System.Object	Wenn VEE einen Zeiger auf ein .NET-Objekt hat.

**Tabelle 33** .NET-Arraydatentypen in VEE-Datentypen in VEE 7.0 konvertieren

Konvertieren des .NET-Datentyps	Konvertieren in den VEE-Datentyp	Anmerkungen
System.Boolean Array	Int16 Array	Verwenden Sie isVariantBool(), um zu bestimmen, ob das Array vom Typ System.Boolean war.
System.Byte Array	UInt8 Array	
System.Char Array	Object	Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.  Sie können die Formel CreateInstance(„mscorlib“,„System.String“,charArray).ToString(); verwenden, um ein Array in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann.
System.Date Time Array	Object	Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.

System.Decimal Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um jedes Array-Member in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt. Die Struktur System.Decimal selbst bietet auch eine Reihe von Konvertierungsmethoden.</p>
Double Array	Real64 Array	
System.Enum Array	Object	Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.
System.Int16 Array	Int16 Array	
System.Int32 Array	Int32 Array	
System.Int64 Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B..ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt.</p>

System.SByte Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. ToInt32().</p>
System.Single Array	Real32 Array	
System.String Array	Text Array	<p>VEE konvertiert String automatisch in Text. Wenn Sie die vielen nützlichen Funktionen der Klasse System.String nutzen möchten, können Sie mit asClrType() Text wieder zurück in System.String konvertieren.</p>
System.UInt16 Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. ToInt32().</p>
System.UInt32 Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. ToInt32(). Wenn der Wert nicht in einen Typ Int32 passt, tritt ein Fehler 751 auf, der eine Ausnahmebedingung aufgrund von Systemüberlauf (System.OverflowException) angibt.</p>



System.UInt64 Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p> <p>Sie können eine Methode System.Convert.To* verwenden, um ihn in einen .NET-Datentyp zu konvertieren, den VEE automatisch konvertieren kann, z. B. ToInt32(). Wenn der Wert nicht in einen Typ Int32 Systemüberlauf (System.OverflowException) angibt.</p>
System.Object Array	Object	<p>Wenn das VEE-Objekt einen Zeiger auf ein .NET-Objekt vom Typ System.Array enthält.</p>

**Tabelle 34** VEE-Arraydatentypen in .NET-Datentypen in VEE 7.0 konvertieren

Konvertieren des VEE-Datentyps	Konvertieren in den .NET-Datentyp*	Anmerkungen
Int16 Array	System.Int16 Array	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Int32 Array	System.Int32 Array	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Real32 Array	System.Single Array	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().
Real64 Array	System.Double Array	Verwenden Sie asClrType(), um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu asClrType().

## 7 .NET mit VEE verwenden

Text Array	System.String Array	Verwenden Sie <code>asClrType()</code> , um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu <code>asClrType()</code> .
UInt8 Array	System.Byte Array	Verwenden Sie <code>asClrType()</code> , um in andere .NET/CLR-Typen zu konvertieren. Siehe Dokumentation zu <code>asClrType()</code> .
< Typ*> Array	System.Boolean Array	Verwenden Sie <code>asVariantBool()</code> auf jedem VEE-Datentyp, der in einen Datentyp <code>Int16</code> ( <code>UInt8</code> , <code>Int16</code> , <code>Int32</code> , <code>Real32</code> , <code>Real64</code> , <code>Text</code> ) umgewandelt werden kann. Sie können auch das generischere <code>asClrType()</code> verwenden, z. B. <code>asClrType veescalar, System.Boolean</code> ).
Date/Time Array	System.DateTime Array	<code>Date/Time</code> von VEE ist eigentlich vom Typ <code>Real64</code> . Verwenden Sie <code>asClrType(veeDateTime, System.DateTime)</code> . Siehe <code>examples\dotnet\DateTime.vee</code> .

**Tabelle 35** .NET-Datentypmodifizierer

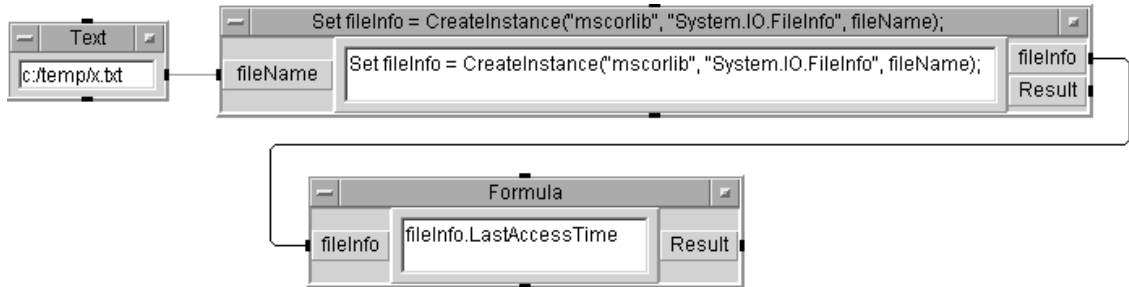
<b>.NET-Datentyp-modifizierer</b>	<b>VEE-Typ</b>	<b>Anmerkungen</b>
ref, out, ByRef	Entweder skalar oder Array des von der Zuordnungstabelle oben angegebenen Typs.	Verwenden Sie das VEE-Schlüsselwort ByRef. Wenn das Schlüsselwort ByRef nicht verwendet wird, tritt eine Ausnahmebedingung bei der .NET-Operation auf. Der übergebene Parameter wird jedoch nicht geändert. Dies ist wahrscheinlich nicht das, was der .NET-Klassendesigner beabsichtigte.

### Instanzmethode aufrufen

Beim Aufrufen einer Instanzmethode müssen Sie zuerst das .NET-Objekt deklarieren und erstellen. Es empfiehlt sich, über Function & Object Browser und die Schaltfläche Create Instance zuerst die Konstruktor-Formelschablone zu erstellen. Abbildung 173 zeigt ein Beispiel.

In Abbildung 170 wird fileInfo als Objekt vom Typ System.IO.FileInfo initialisiert. Sie können das Ausgangs-Pin fileInfo mit dem Eingangs-Pin nachfolgender Formelobjekte verbinden, die diese Objektinstanz benötigen. Beachten Sie auch, dass die Formelobjekte CreateInstance und fileInfo.LastAccessTime automatisch von VEE erzeugt werden. Sie müssen sie lediglich miteinander verbinden und alle eventuell benötigten zusätzlichen Parameter angeben.

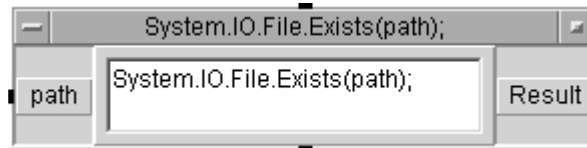
## 7 .NET mit VEE verwenden



**Abbildung 170** Ein .NET-Objekt erstellen und auf sein Instanz-Member zugreifen

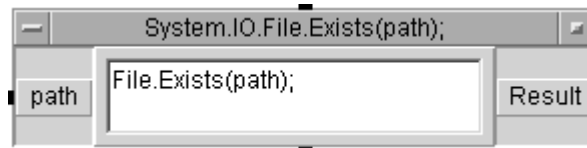
### Eine freigegebene/statische Methode aufrufen

Wenn Sie ein .NET-Member aufrufen und das Member freigegeben/statisch ist, wird das Wort **STATIC** im Beschreibungsfeld in Function & Object Browser angezeigt. Statische Methoden werden direkt auf einer .NET-Klasse aufgerufen, nicht auf einem Objekt. Auf diese Weise müssen Sie nicht zuerst ein .NET-Objekt erstellen. Abbildung 171 zeigt ein Beispiel.



**Abbildung 171** Statische Methode ohne importierten Namespace

Wenn Sie den Namespace System.IO importiert haben, sieht die obige Formelschablone wie in Abbildung 172 aus:



**Abbildung 172** Statische Methode mit importiertem Namespace

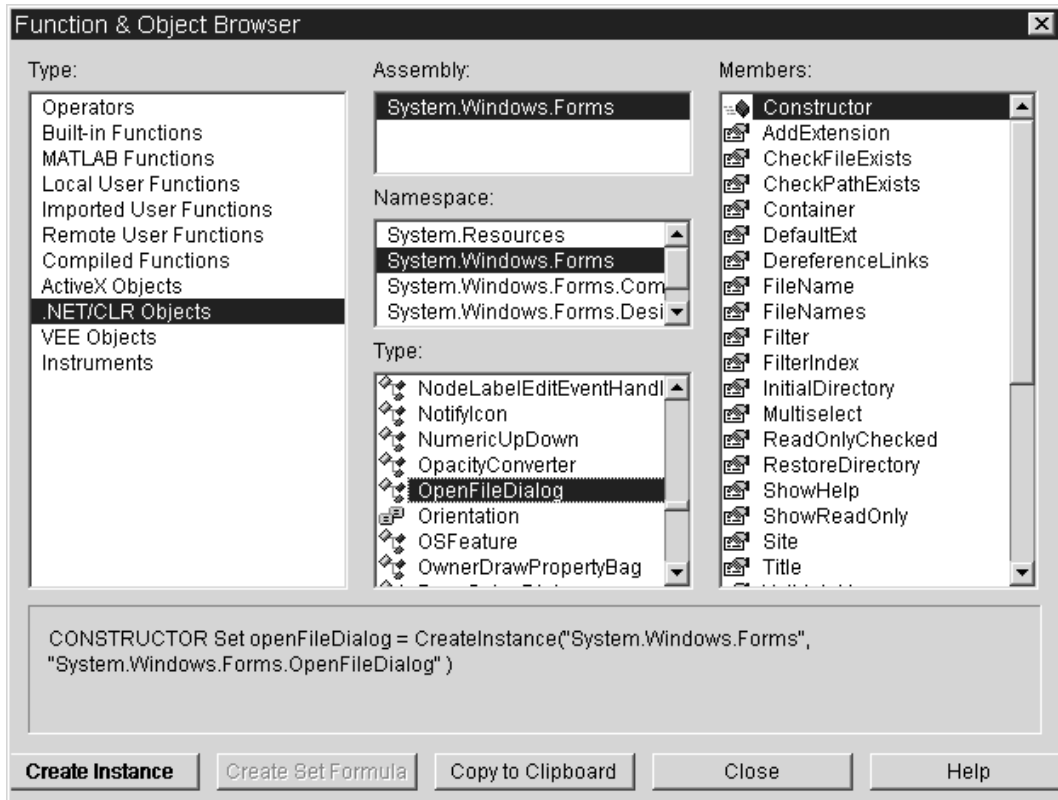
## .NET-Programmiertipps

- Wenn Sie eine .NET-Methode aufrufen, die einen bestimmten numerischen Datentyp benötigt, bei dem der Parameter aus einem Eingangs-Pin stammt (z. B. „A“), können Sie oft die Ausnahmebedingung „Method Not Found“ vermeiden, indem Sie einfach auf das Eingangs-Pin A doppelklicken und den erforderlichen Typ einstellen. Dies ist einfacher als das Aufrufen der integrierten Funktion asClrType oder der .NET-Funktionen System.Convert.To\*.
- Die für Instanz-Member generierte Formelschablone beginnt mit einem Kleinbuchstaben. Statische Member haben einen vollständig qualifizierten Namen (Namespace plus Klassenname), sofern der entsprechende Namespace nicht importiert wurde.
- Wenn Sie ein .NET-Member aufrufen und das Member statisch ist, wird das Wort **STATIC** im Beschreibungsfeld angezeigt. Statische Methoden werden direkt auf einer .NET-Klasse aufgerufen, nicht auf einer Instanz eines Objekts. Daher müssen Sie nicht zuerst ein .NET-Objekt erstellen.
- Enumerationen in .NET unterscheiden sich von Enumerationen in COM. Sie sind stark typisiert. Es gibt zwei Typen von Enumerationen in .NET. Die eine wird als Enumeration von Konstanten behandelt, die andere als Bitfelder. Nur der zweite Enumerationstyp ist für bitweise Operationen gedacht. In VEE wird für beide Typen von Enumerationen der zugrunde liegende Integralwert im Beschreibungsfeld von Function & Object Browser angezeigt. Sie können bitOr, bitAnd usw. am zweiten Enumerationstyp verwenden, wie dies den VEE-Beispielen (examples\dotnet\FileAttributes.vee) zu entnehmen ist.
- Es gibt eine neue Schaltfläche **Copy to Clipboard** in Function & Object Browser. Über diese Schaltfläche können Sie einen Formeltext in die Zwischenablage kopieren und später beliebig in ein Formelobjekt einfügen. Dies ist insbesondere hilfreich, wenn Sie mehrere Formelanweisungen in einem Formelobjekt haben möchten.

## Übung 7-1: .NET zur Wahl von Dateien verwenden

Erstellen Sie ein .NET-Objekt vom Typ `System.Windows.Forms.OpenFileDialog`. Verwenden Sie dieses Objekt, um eine Gruppe von Dateien nach der Dateierweiterung zu filtern und die Dateinamen in einem Dialogfeld anzuzeigen. Das vollständige Beispiel ist in `examples\dotnet\OpenFileDialog.vee` zu finden.

- 1 Wählen Sie im Menü **Device** den Befehl **.NET Assembly References**.
- 2 Aktivieren Sie auf der Registerkarte **.NET** das Kontrollkästchen `System.Windows.Forms`. Wählen Sie **OK**.
- 3 Öffnen Sie **Function & Object Browser** und wählen Sie **.NET/CLR Objects**.
- 4 Wählen Sie den Namespace `System.Windows.Forms`.
- 5 Wählen Sie den Typ `OpenFileDialog` und das Member **Constructor**. Klicken Sie auf **Create Instance**.



**Abbildung 173** Function & Object Browser – Erstellen einer Instanz

- 6 Deklarieren Sie eine globale Variable namens `openFileDialog` und setzen Sie ihren Typ auf `Object`.
- 7 Da Sie `openFileDialog` als globale Variable deklariert haben, löschen Sie das Ausgangs-Pin `openFileDialog` aus dem Formelobjekt **CreateInstance**.

- 8 Ändern Sie den Titel des Formelobjekts **CreateInstance(„System.Windows.Forms“, „System.Windows.Forms.OpenFileDialog“)** in **Open File Dialog**. Führen Sie im Objekt die folgenden Schritte aus:
  - a Geben Sie das Verzeichnis an, in dem das Programm das Dialogfeld öffnet.
  - b Stellen Sie die Eigenschaft **Multiselect** von Open File Dialog auf **True** ein. (Tipp: Sie benötigen **asVariantBool** oder **asClrType**, um **System.Boolean** abzurufen.) Dadurch können Sie mehrere Dateien aus dem Feld **Open File Dialog** wählen.
  - c Stellen Sie den Filter für die Dateierweiterung ein, in diesem Fall **DLL-Dateien**.
  - d Rufen Sie die Methoden **openFileDialog** und **ShowDialog** auf und zeigen Sie das Dialogfeld an.
  - e Rufen Sie die Eigenschaft **FileNames** von **openFileDialog** auf und rufen Sie die Liste der gewählten **DLL-Dateien** ab. Diese Liste wird im Objekt **Logging Alphanumeric** angezeigt.



Das Programm sollte jetzt aussehen wie in Abbildung 174.



Abbildung 174 Das Programm „openFileDialog“

## Übung 7-2: .NET zum Ausführen von DateTime-Operationen verwenden

Verwenden Sie Instanz- und statische Member des .NET-Typs DateTime. Der .NET-Typ DateTime bietet mehr Funktionalität als die DateTime-Funktionen von VEE. In dieser Übung wird der Typ DateTime verwendet, um das aktuelle Datum, die aktuelle Uhrzeit, den aktuellen Wochentag und das aktuelle Jahr

abzurufen und um abzufragen, ob es sich um ein Schaltjahr handelt. Das vollständige Beispiel ist in `examples\dotnet\DateTime.vee` zu finden.

- 1 Wählen Sie **Device** ⇒ **.NET Assembly References**.
- 2 Aktivieren Sie auf der Registerkarte **.NET** das Kontrollkästchen **mscorlib**. Wählen Sie **OK**.
- 3 Aktivieren Sie im Dialogfeld **Import .NET Namespaces** das Kontrollkästchen **System**. Wählen Sie **OK**.
- 4 Öffnen Sie **Function & Object Browser** und wählen Sie **.NET/CLR Objects**.
- 5 Markieren Sie den Namespace **System**.
- 6 Markieren Sie den Typ **DateTime** und die statische Eigenschaft **Now**. Klicken Sie auf die Schaltfläche **Create Get Formula**. Da dies ein statisches Member ist, müssen Sie nicht zuerst eine Instanz des DateTime-Objekts erstellen oder abrufen.
- 7 Wiederholen Sie Schritt 4 und 5. Markieren Sie den Typ **DateTime**. Es gibt mehrere Versionen der ToString-Methoden. Markieren Sie die einfachste Version, die keine Parameter benötigt. Klicken Sie auf die Schaltfläche **Create Call**.
- 8 Verbinden Sie das Ergebnis-Pin des Formelobjekts **DateTime.Now** mit dem Eingangs-Pin **dateTime** des Formelobjekts **dateTime.ToString()**.
- 9 Fügen Sie ein Objekt **Alphanumeric** hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **dateTime.ToString()**.
- 10 Wiederholen Sie Schritt 4 und 5. Markieren Sie den Typ **DateTime** und die Eigenschaft **DayOfWeek**. Klicken Sie auf die Schaltfläche **Create Get Formula**. Die Eigenschaft **DayOfWeek** gibt ein weiteres .NET-Objekt vom Typ **System.DayOfWeek** zurück. Da jedes .NET-Objekt eine ToString()-Methode hat, können Sie es verwenden, um den Wochentag zu formatieren und auszugeben. Bearbeiten Sie die soeben erstellte Formel mit VEE und hängen Sie vor dem Semikolon `.ToString()` an.

- 11 Verbinden Sie das Ergebnis-Pin des Formelobjekts **Date-Time.Now** mit dem Eingangs-Pin `dateTime` des Formelobjekts **dateTime.DayOfWeek**.
- 12 Fügen Sie ein Objekt **Alphanumeric** hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **dateTime.DayOfWeek**.

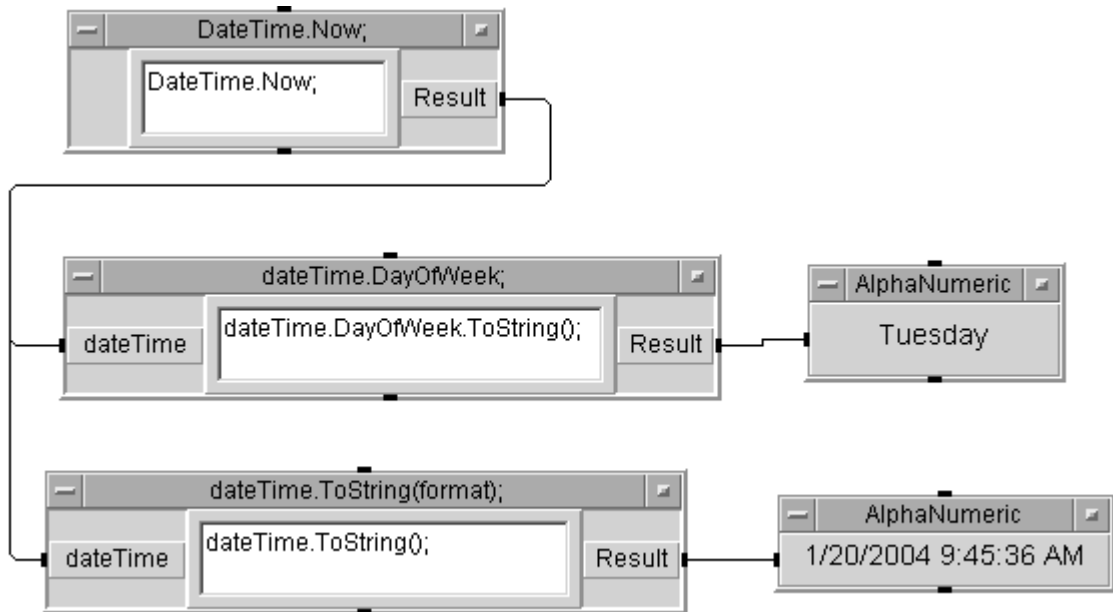


Abbildung 175 Schritt 10 von Übung 7-2

- 13 Wiederholen Sie Schritt 4 und 5. Markieren Sie den Typ **DateTime** und die Eigenschaft **Year**. Klicken Sie auf die Schaltfläche **Create Get Formula**.
- 14 Verbinden Sie das Ergebnis-Pin des Formelobjekts **Date-Time.Now** mit dem Eingangs-Pin `dateTime` des Formelobjekts **dateTime.Year**.
- 15 Fügen Sie ein Objekt **Alphanumeric** hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **dateTime.Year**.

- 16 Wiederholen Sie Schritt 4 und 5. Markieren Sie den Typ **DateTime** und die statische Methode **IsLeapYear**. Klicken Sie auf die Schaltfläche **Create Call**. Da dies ein statisches Member ist, müssen Sie nicht eine Instanz des DateTime-Objekts erstellen oder abrufen.
- 17 Verbinden Sie das Ergebnis-Pin des Formelobjekts **dateTime.Year** mit dem Eingangs-Pin year des Formelobjekts **dateTime.IsLeapYear**.
- 18 Fügen Sie ein Objekt Alphanumeric hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **dateTime.IsLeapYear**.

Die abgeschlossene Übung ist in Abbildung 176 dargestellt.

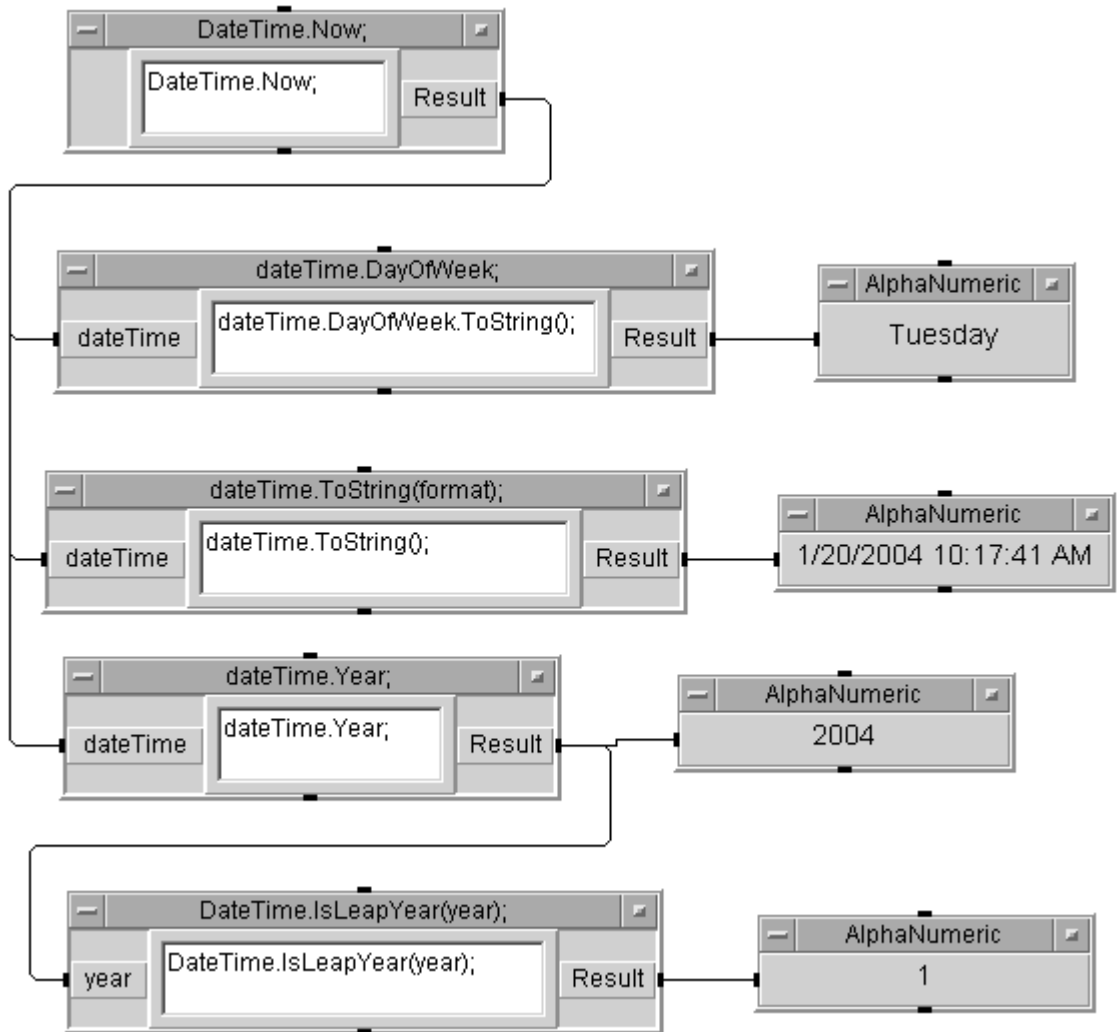


Abbildung 176 Übung 7-2 abgeschlossen

## Übung 7-3: .NET zum Abrufen von Dateiinformationen verwenden

Der .NET-Namespace System.IO bietet eine breite Palette von Funktionen für das Arbeiten mit dem Dateisystem. Diese Funktionalität kann jetzt noch einfacher genutzt werden. In dieser Übung fragen Sie eine Datei nach ihrer Erstellungszeit, dem Zeitpunkt des letzten Zugriffs und der Länge ab. Das vollständige Beispiel ist in `examples\dotnet\FileInfo.vee` zu finden.

- 1 Wählen Sie **Device** ⇒ **.NET Assembly References**.
- 2 Aktivieren Sie auf der Registerkarte **.NET** das Kontrollkästchen **mscorlib**. Wählen Sie **OK**.
- 3 Wenn das Dialogfeld **Import .NET Namespaces** angezeigt wird, können Sie den Namespace **System** aktivieren oder einfach **Cancel** wählen. In dieser Übung verwenden Sie, im Gegensatz zum vollständigen Beispiel, keine statischen oder Enumerationsmember.
- 4 Öffnen Sie **Function & Object Browser** und wählen Sie **.NET/CLR Objects**.
- 5 Markieren Sie den Namespace **System.IO**.
- 6 Markieren Sie den Typ **FileInfo** und das Member **Constructor**. Klicken Sie auf die Schaltfläche **Create Instance**. Sehen Sie sich die generierte Formelschablone an. Ein Ausgangs-Pin namens **fileInfo** wird von VEE erzeugt und auf ein neu erstelltes Objekt **FileInfo** gesetzt.
- 7 Wählen Sie **Data** ⇒ **Dialog Box** ⇒ **File Name Selection**. Verbinden Sie das Ausgangs-Pin **FileName** mit dem Eingangs-Pin **fileName** des Formelobjekts **Create Instance**. Stellen Sie das Objekt **File Name Selection** als Symbol dar.
- 8 Wiederholen Sie Schritt 4 und 5. Markieren Sie den Typ **FileInfo**. Markieren Sie die Eigenschaft **CreationTime**. Klicken Sie auf die Schaltfläche **Create Get Formula**. Da die Eigenschaft **CreationTime** ein .NET-Objekt vom Typ **DateTime** zurückgibt, verwenden Sie die zugehörige Methode **ToString()**, um den Erstellungszeitpunkt zu formatieren und auszugeben. Hängen Sie **.ToString()** an die von VEE generierte Formel an. Sie müssen diese Zeichenfolge vor dem Semikolon anfügen.

- 9 Verbinden Sie das Ausgangs-Pin fileInfo aus dem Formelobjekt **CreateInstance** mit dem Eingangs-Pin fileInfo des Formelobjekts **fileInfo.CreationTime**.
- 10 Fügen Sie ein Objekt Alphanumeric hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **fileinfo.CreationTime**.
- 11 Wiederholen Sie Schritt 8 und 9, um das Formelobjekt für die Eigenschaft **fileInfo.LastAccessTime** zu erstellen und zu verbinden.
- 12 Fügen Sie ein Objekt Alphanumeric hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **fileinfo.LastAccess**.
- 13 Wiederholen Sie Schritt 8 und 9, um das Formelobjekt für die Eigenschaft **fileInfo.Length** zu erstellen und zu verbinden.
- 14 Fügen Sie ein Objekt Alphanumeric hinzu und verbinden Sie es mit dem Ausgangs-Pin des Formelobjekts **fileinfo.Length**.

Die abgeschlossene Übung sollte jetzt aussehen wie in Abbildung 177.

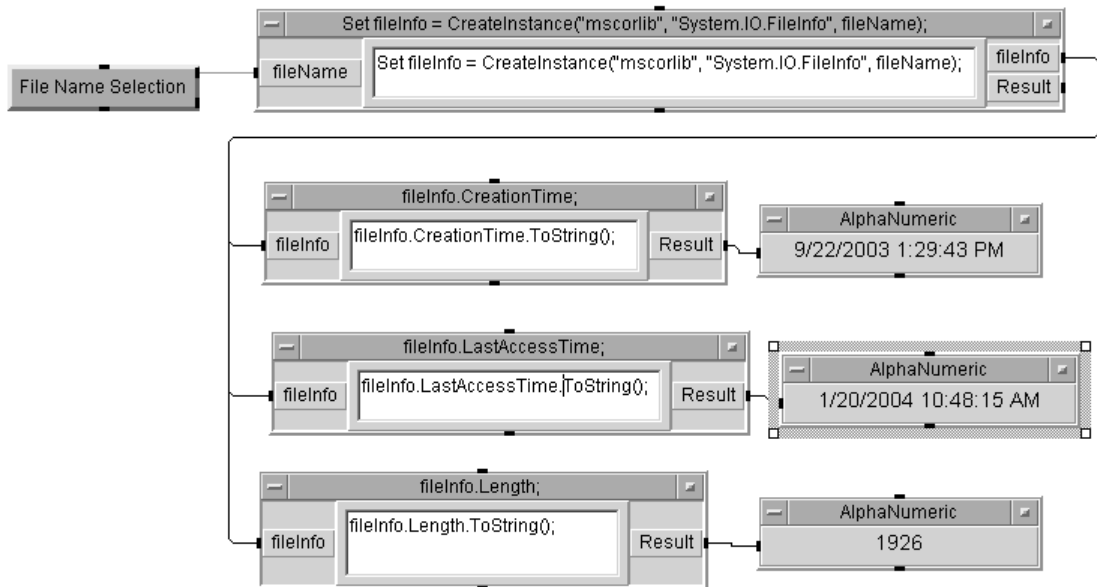


Abbildung 177 Übung 7-3 abgeschlossen

## .NET- und IVI-Treiber

Was ist IVI? IVI ist ein neuer Gerätetreiberstandard, der von der IVI Foundation entwickelt wird. Die IVI Foundation ist ein Konsortium, das zur Entwicklung von Spezifikationen für die Programmierung von Testgeräten gegründet wurde. Diese Spezifikationen bieten eine Reihe von Vorteilen, hauptsächlich jedoch vereinfachen sie die Austauschbarkeit von Geräten. Weitere Informationen zur IVI Foundation finden Sie unter <http://www.ivifoundation.org/>.

Was sind die Vorteile eines IVI-COM-Treibers? Angenommen, zwei Hardwarehersteller bieten einen IVI-COM-Treiber für jeden ihrer DMMs. Um den IVI-Standard zu erfüllen, müssen die Basistreiber austauschbar sein. Wenn Sie also mit dem DMM des einen Herstellers und dem zugehörigen Treiber beginnen, können Sie zum DMM und Treiber des anderen Herstellers wechseln, ohne Änderungen an Ihrem Programm vornehmen zu müssen. Ihr Code ist ohne jegliche Eingriffe wiederverwendbar.

Installieren Sie die IVI-COM-Treiber nach Bedarf. Treiber für Ihre Geräte finden Sie unter <http://www.agilent.com/find/adn>. Wenn Sie ein ADN-Mitglied sind, wählen Sie im Menü ⇒ **Drivers by Driver Type** ⇒ **IVI-COM Drivers and Components** die Option Downloads. Andernfalls ist diese Site eine Quelle für Agilent-Treiber, Evaluierungssoftware, Dokumentation und Faktenpapiere. Um sich für diesen kostenlosen Service anzumelden, füllen Sie das Online-Registrierungsformular für neue Benutzer aus.

Sobald die IVI-COM-Treiber auf Ihrem System registriert sind, stehen sie (wie jede andere COM-Komponente) zur Verfügung und werden (wie die meisten COM-Komponenten) auf der Registerkarte COM in **Device** ⇒ **.NET Assembly References** in VEE aufgelistet.

Wie stellen VEE und .NET IVI-COM-Treiber zur Verfügung? Über die COM-Interop-Funktion von .NET werden die IVI-COM-Treiber in VEE verfügbar. Wenn keine primäre Interop-Assembly für eine gewählte COM-DLL verfügbar ist, versucht VEE, eine Interop-Assembly zu generieren (oder mehrere, wenn die COM-DLL



auf andere Typenbibliotheken verweist). Nachdem auf die Interop-Assembly verwiesen wurde, kann der IVI-COM-Treiber wie jedes andere .NET-Objekt aufgerufen werden.

Weitere Informationen zu IVI-COM finden Sie unter [www.agilent.com/find/adn](http://www.agilent.com/find/adn). Melden Sie sich an, rufen Sie die Knowledge Library auf, und wählen Sie IVI-COM Information. Dort finden Sie viele Faktenpapiere.

## Assemblies

### Installieren einer neuen Assembly

Assemblies sind die Grundbausteine des .NET Framework. Sie haben die Form einer ausführbaren Datei (.exe) oder einer Dynamic Link Library (.dll).

Assemblies können im globalen Assembly-Cache (GAC) installiert werden. Sie benötigen dazu die entsprechenden Berechtigungen und müssen sehr sorgfältig vorgehen, da der GAC eine systemweite Ressource ist. Die im GAC installierten Assemblies werden von allen Anwendungen auf Ihrem Computer gesehen. (Ausführliche Informationen hierzu finden Sie in MSDN.) Sobald die freigegebene Assembly aktualisiert ist, können Sie darauf verweisen, indem Sie auf die Schaltfläche **Browse** klicken und den Speicherort der Assembly suchen, bevor sie in den GAC installiert wurde.

Die meisten Assemblies sind nicht im GAC installiert. Diese werden als private Assemblies bezeichnet. Eine private Assembly sollte sich im VEE-Programm befinden, das darauf verweist. Bei der ersten Verwendung des Dialogfeldes .NET Assembly zum Suchen und Auswählen einer privaten Assembly kopiert VEE diese in dasselbe Verzeichnis wie Ihr aktuelles VEE-Programm, sofern sie sich dort noch nicht befindet. Dieser Kopiervorgang macht auch das Verteilen Ihres VEE-Programms einfacher und wird als xcopy-Einsatz bezeichnet.

### Aktualisieren einer Assembly

Wenn Sie in Ihrem VEE-Programm bereits auf die freigegebene Assembly verwiesen haben, müssen Sie File/New aufrufen oder VEE erneut starten, bevor die neue Version der freigegebenen Assembly geladen wird. Handelt es sich nicht um eine freigegebene, sondern um eine private Assembly, müssen Sie File/New aufrufen oder VEE erneut starten, nachdem Sie die neue Version der privaten Assembly manuell in Ihr aktuelles VEE-Programmverzeichnis kopiert haben. Weitere Informationen hierzu finden Sie im Abschnitt „Die VEE-Laufzeitversion weitergeben“ auf Seite 327.

## Die VEE-Laufzeitversion weitergeben

Wenn Ihr VEE-Programm auf freigegebene Assemblies (die im GAC installiert sind) verweist, müssen Sie diese Assemblies im GAC des Zielsystems installieren. Ist die freigegebene Assembly eine Microsoft .NET Framework-Assembly, sollte sie bei der Installation des Microsoft .NET Framework-Redist-Pakets installiert worden sein.

### HINWEIS

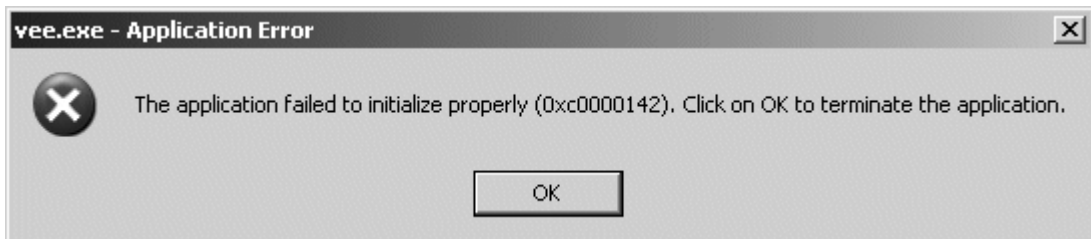
Das .NET Framework-Redist-Paket muss vor der VEE-Laufzeitversion installiert sein.

---

VEE speichert alle privaten Assemblies, auf die verwiesen wird, mit dem VEE-Programm. Wenn Ihr VEE-Programm nur auf private Assemblies verweist, können Sie einfach das gesamte VEE-Programmverzeichnis auf das Zielsystem kopieren. Dies ist viel einfacher als die Weitergabe von VEE-Programmen, die auf COM-Typbibliotheken verweisen, bei der Sie alle COM-Typbibliotheken auf dem Zielsystem registrieren müssen.

## VEE- und .NET-Sicherheit

Durch die Verwendung des .NET Framework in VEE ergibt sich ein Sicherheitsproblem. Wenn das folgende Dialogfeld angezeigt wird, müssen Sie diese Anpassungen an Ihrem .NET Framework-Sicherheitsprofil vornehmen:



- 1 Wechseln Sie zu Ihrem Systemlaufwerk.
- 2 Wechseln Sie in das Verzeichnis `\[winnt oder windows]\Microsoft.NET\Framework\v1.1.4322`.
- 3 Führen Sie an der Eingabeaufforderung den folgenden Befehl aus:

```
Caspol -machine -addgroup „All_Code“ -url File://[VEEInstallationsverzeichnis]/* FullTrust
```

Dabei ist VEEInstallDir Ihr VEE-Installationsverzeichnis auf einem Netzlaufwerk. (Caspol wird mit dem .NET Framework-Redist-Paket ausgeliefert und wird bei der Installation von VEE installiert.) Sie können hierzu auch auf den .NET Framework-Konfigurationsassistenten zurückgreifen. Sie müssen über Administratorberechtigungen verfügen, um diese beiden Tools ausführen zu können.

Wenn Ihr VEE-Programm darüber hinaus auf .NET-Assemblies verweist, muss VEE oft Assemblies generieren und/oder an einem zugreifbaren Speicherort ablegen. Wenn Sie beispielsweise ein VEE-Programm (mit .NET-Verweisen) über E-Mail oder über ein Netzwerk öffnen, ohne es lokal zu speichern, erhalten Sie u.U. eine entsprechende Warnmeldung.

## .NET-Terminologie

Die im Folgenden aufgeführten Begriffe und Bezeichnungen werden im gesamten Kapitel verwendet. Schlagen Sie deshalb diese Microsoft-Definitionen nach, während Sie das Kapitel lesen, oder wenden Sie sich ihnen jetzt zu.

### Assembly

Komponenten sind in Assemblies gepackt. Assemblies sind die wiederverwendbaren, versionierbaren, selbstbeschreibenden Baublöcke von .NET-Anwendungen. Die einfachste Assembly ist ein einzelnes ausführbares Programm, das alle erforderlichen Informationen zum Einsatz und zur Versionierung enthält. Eine Assembly ist der Grundbaustein des .NET Framework. Sie hat die Form einer ausführbaren Datei (.exe) oder einer Dynamic Link Library (.dll).

### Primäre Interop-Assembly (PIA)

Eine primäre Interop-Assembly ist eine eindeutige, vom Hersteller bereitgestellte Assembly, die Typdefinitionen (als Metadatei) von Typen enthält, die mit COM implementiert sind. Es kann nur eine primäre Interop-Assembly geben, die mit einem starken Namen vom Publisher der COM-Typbibliothek unterzeichnet sein muss.

### Namespace

.NET Framework-Typen verwenden ein Benennungsschema mit Punktmatrix, das eine Hierarchie angibt. Diese Technik gruppiert zugehörige Typen in Namespaces, so dass sie leichter durchsucht werden können und leichter auf sie verwiesen werden kann. Der erste Teil des vollständigen Namens (bis zum Punkt ganz rechts) ist der Namespace-Name(\*). Der letzte Teil des Namens ist der Typname. Ein Beispiel: System.Collections.ArrayList steht für den Typ ArrayList, der zum Namespace System.Collections gehört. Die Typen in System.Collections können zur Manipulation von Objektauflistungen verwendet werden.

\*Verschachtelte Klassen bilden eine Ausnahme zu dieser allgemeinen Regel. Weitere Einzelheiten entnehmen Sie bitte der MSDN-Dokumentation.

### Verweis

Zur Verwendung einer Assembly müssen Sie einen Verweis darauf hinzufügen.

### Klasse

Wenn Sie mit objektorientierter Programmierung vertraut sind, wissen Sie, dass eine Klasse die Operationen, die ein Objekt durchführen kann (Methoden, Ereignisse oder Eigenschaften) und einen Wert definiert, der den Status des Objekts (Felder) enthält. Obwohl eine Klasse in der Regel sowohl Definition als auch Implementierung umfasst, kann sie eine oder mehrere Member ohne Implementierung enthalten.

Eine Instanz einer Klasse ist ein Objekt. Sie greifen auf die Funktionalität eines Objekts zu, indem Sie seine Methoden aufrufen und auf seine Eigenschaften, Ereignisse und Felder zugreifen.

### Freigegebene oder statische Member

Freigegebene Member sind Eigenschaften, Prozeduren und Felder, die von allen Instanzen einer Klasse gemeinsam genutzt werden. Einige Programmiersprachen bezeichnen solche Elemente als statische Member.

Freigegebene Felder und Eigenschaften sind nützlich, wenn Sie Informationen haben, die Teil einer Klasse, jedoch nicht für eine bestimmte Instanz einer Klasse spezifisch sind. Normale Felder und Eigenschaften existieren für jede Instanz einer Klasse voneinander unabhängig. Die Änderung des Werts eines Feldes oder einer Eigenschaft, das/die einer Instanz zugeordnet ist, wirkt sich nicht auf die Felder oder Eigenschaften anderer Instanzen der Klasse aus. Wenn Sie dagegen den Wert eines freigegebenen Felds oder einer freigegebenen Eigenschaft für eine Instanz einer Klasse modifizieren, ändern Sie den Wert für alle

Instanzen der Klasse. Auf diese Weise verhalten sich freigegebene Felder und Eigenschaften wie globale Variablen, die nur von Instanzen einer Klasse aufgerufen werden können.

### **Instanzmember**

Ein Instanzmember ist an eine bestimmte Instanz einer Klasse gebunden. Änderungen seiner Werte wirken sich nur auf das zugeordnete Objekt aus.

### Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Wenden Sie sich ggf. den Themen dieses Kapitels erneut zu, bevor Sie mit dem nächsten Kapitel fortfahren.

- Die grundlegende .NET-Terminologie beherrschen.
- Einen Namespace in VEE importieren.
- Eine PIA definieren.
- Die VEE-Laufzeitversion weitergeben
- Die .NET-Sicherheit für VEE anpassen



# 8

## Programme für den PC integrieren

Überblick 335

Die Arbeitsweise des Objekts „Execute Program“ 336

Einen Systembefehl verwenden 339

Checkliste für das Kapitel 343

## **Integrieren von Programmen in anderen Sprachen**

*Dieses Kapitel enthält Informationen zu folgenden Themen:*

- Das Objekt `Execute Program`
- Verwenden der Betriebssystemsbefehle von VEE aus
- Portierbar machen von VEE-Programmen über verschiedene Plattformen hinweg

*Erforderliche Zeit für dieses Kapitel: 30 Minuten*

## Überblick

Dieses Kapitel behandelt die bequemsten Möglichkeiten zur Integration von kompilierten Programmen und Betriebssystembefehlen mit VEE. Einer der entscheidenden Vorteile von VEE ist die Möglichkeit zur Integration mit anderen Anwendungen und Programmen. Darüber hinaus können Sie mit ActiveX auch Komponenten aus anderen Programmen verwenden. (Weitere Informationen hierzu finden Sie in Kapitel, „Berichte mit ActiveX bequem erstellen“.)

In VEE gibt das Objekt Execute Program Programme und Parameter an und verwendet Betriebssystembefehle. Es gibt ein Objekt Execute Program für den PC. Dieses Kapitel enthält eine Übung mit dem Objekt Execute Program für PC.

### Die Arbeitsweise des Objekts „Execute Program“

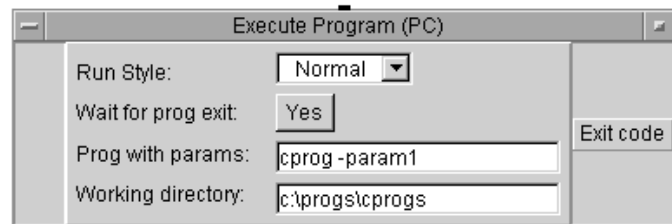
Zusätzlich zur ActiveX-Automatisierung gibt es drei weitere Möglichkeiten, Programme in anderen Sprachen von VEE aus auszuführen:

- 1 Mit dem Objekt `Execute Program` ein anderes Programm, eine andere Anwendung oder einen Betriebssystembefehl außerhalb von VEE ausführen. Diese Methode ist besonders vielseitig und benutzerfreundlich.
- 2 Kompilierte Funktionen in anderen Sprachen mit VEE verbinden („linken“), entweder über gemeinsame Bibliotheken in UNIX-Betriebssystemen oder Dynamic Link Libraries auf dem PC. Diese Methode ist zwar etwas komplizierter, bietet jedoch deutliche Vorteile bei der Geschwindigkeit. Weitere Informationen zu Dynamic Link Libraries finden Sie im Abschnitt „Dynamic Link Libraries verwenden“ auf Seite 475.

Das Objekt `Execute Program` befindet sich im Menü I/O. Es gibt ein Objekt für den PC wie in Abbildung 178 dargestellt. Beachten Sie, dass das Objekt `Execute Program` keine Transaktions-E/A zur Kommunikation mit Programmen verwendet; fügen Sie daher keine Dateneingangs- und -ausgangs-Pins zur Übergabe von Daten an das kompilierte Programm hinzu.

### Mit dem Objekt „Execute Program“ arbeiten

Abbildung 178 zeigt das Objekt `Execute Program` auf dem PC.



**Abbildung 178** Das Objekt „Execute Program“ (PC)

Mit dem Objekt `Execute Program` können Sie folgenden Elemente aus VEE heraus ausführen:

- Kompilierte Programme, die in anderen Sprachen geschrieben wurden
- \*.BAT- oder \*.COM-Dateien
- MS-DOS-Systembefehle, z. B. `dir`
- Beliebige Dokumente oder URLs mit einer bekannten Erweiterung. Die Aktion „open“ wird in den Dateien aufgerufen. Wenn keine Aktion „open“ vorhanden ist, wird mit der Datei die Standardaktion aufgerufen. Ein Beispiel für eine URL ist <http://www.agilent.com/find/vee>.

Das Objekt `Execute Program` enthält die folgenden Elemente:

**Tabelle 36** Felder des Objekts `Execute Program`

Name	Beschreibung
Run Style	Legt die Fenstergröße fest. <code>Normal</code> steht für ein Standardfenster, <code>Minimized</code> für ein Symbol, und <code>Maximized</code> für ein Fenster in Vollbildgröße. <code>Working directory</code> gibt das Arbeitsverzeichnis an, das die Dateien zu dem Programm enthält.
Wait for prog exit	Gibt an, wann das Sequenz-Pin auslösen soll. <ul style="list-style-type: none"> <li>• Wenn diese Angabe auf <code>Yes</code> gesetzt ist, wird das Sequenz-Pin erst ausgelöst, wenn die Ausführung des Programms abgeschlossen ist.</li> <li>• Ist die Angabe auf <code>No</code> gesetzt, löst das Sequenz-Pin aus, bevor die Ausführung des angegebenen Programms abgeschlossen ist. Wenn beim Starten von Dokumenten oder URLs das Dokument oder die Web-Site in eine bereits aktive Anwendung geladen wird, wartet VEE nicht, bis die Anwendung beendet wird.</li> </ul>

**Tabelle 36** Felder des Objekts Execute Program

Name	Beschreibung
Prog with params	<p>(Programm mit Parametern) Dieses Feld enthält die gleichen Befehle, die Sie in auch bei einer DOS-Eingabeaufforderung eingeben. Geben Sie zur Ausführung eines Programms in C beispielsweise den Namen der ausführbaren Datei</p> <ul style="list-style-type: none"><li>– <b>myprog.exe</b> ein.</li></ul> <p>(Sie können die Erweiterung <code>.exe</code> weglassen.)</p> <ul style="list-style-type: none"><li>• Wenn das Programm Parameter hat, werden diese nach dem Namen der ausführbaren Datei und einem Bindestrich angegeben. Beispiel: <b>myprog -param1 -param2.</b></li><li>• Zur Ausführung eines DOS-Systembefehls starten Sie zunächst den DOS-Befehlsinterpreter mit der Option <code>/c</code>. Für Windows 95 und Windows 98 geben Sie beispielsweise den Befehl <code>command.com /c &lt;Systembefehl&gt;</code> ein.</li><li>• Für Windows NT 4.0, Windows 2000 und Windows XP geben Sie den Befehl <code>cmd /c &lt;Systembefehl&gt;</code> ein. Diese Option weist den Befehlsinterpreter an, die Zeichenfolge nach der Angabe <code>/c</code> als Systembefehl einzulesen.</li></ul>

## Einen Systembefehl verwenden

Zum Aufrufen eines kompilierten Programms in einer anderen Sprache können Sie den Namen der ausführbaren Datei und alle erforderlichen Parameter am Objekt `Execute Program` eingeben.

Zur Ausführung eines MS-DOS-Systembefehls müssen Sie allerdings zunächst den DOS-Befehlsinterpreter starten. In dieser Übung starten Sie den DOS-Befehlsinterpreter und führen einen MS-DOS-Systembefehl aus.

### Übung 8-1: Einen Systembefehl verwenden

- 1 Wählen Sie **I/O** ⇒ **Execute Program**. Klicken Sie in das Feld **Prog with params**, um eine Einfügemarke anzuzeigen, und geben Sie den folgenden Befehl ein:

```
command.com /c dir >> c:\bob
```

#### HINWEIS

Ersetzen Sie unter Windows NT 4.0, Windows 2000 oder Windows XP den Befehl `command.com` durch `cmd`. Wenn Sie ein anderes als Laufwerk `c:` verwenden möchten, ersetzen Sie den in diesen Anleitungen angegebenen Laufwerksbuchstaben entsprechend. Unter NT müssen Sie eventuell ein anderes Verzeichnis eingeben, für das Sie die Schreibberechtigung haben.

Eventuell müssen Sie den vollständigen Pfad der ausführbaren Datei **command.com** angeben. Der Befehl startet den DOS-Befehlsinterpreter, der wiederum den Systembefehl zum Anzeigen des aktuellen Verzeichnisses ausführt und die Ausgabe (>) vom Computerbildschirm in die Datei **bob** umleitet.

Behalten Sie für `Wait for prog exit` die Option **Yes** bei. Behalten Sie für `Run Style` den Eintrag **Normal** bei und geben Sie unter `Working Directory` `c:\` für das Arbeitsverzeichnis ein.

- 2 Wählen Sie **I/O** ⇒ **From** ⇒ **File** und platzieren Sie dieses Objekt unter `Execute Program`. Verbinden Sie das Sequen-

## 8 Programme für den PC integrieren

zausgangs-Pin von **Execute Program** mit dem Sequenzeingangs-Pin des Objekts **From File**.

Klicken Sie neben **From File**: auf die Schaltfläche **myFile**, um das Dialogfeld **Read data from what file?** anzuzeigen. Geben Sie **c:\bob** ein und klicken Sie auf **OK**. (Die Datei **bob** wird erstellt.)

- 3 Doppelklicken Sie in die Transaktionszeile, um das Dialogfeld **I/O Transaction** aufzurufen.
  - a Ändern Sie **REAL64 FORMAT** in **STRING FORMAT**.
  - b Ändern Sie **SCALAR** in **ARRAY 1D**.
  - c Klicken Sie auf die Schaltfläche **SIZE: (10)**, um sie auf **TO END** zu setzen (\*), und klicken Sie auf **OK**. Die Transaktionszeile sollte jetzt den folgenden Eintrag enthalten: **READ TEXT x STR ARRAY:\***. Diese Transaktion liest den Inhalt der Datei **bob**.
- 4 Wählen Sie **Display** ⇒ **Logging AlphaNumeric** und verbinden Sie das Dateneingangs-Pin mit dem Datenausgangs-Pin des Objekts **From File**.
- 5 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 179.

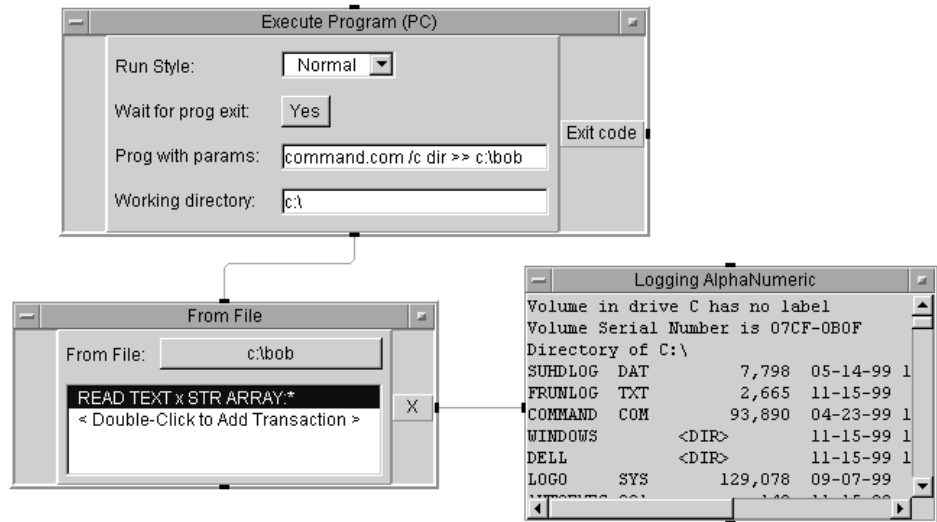
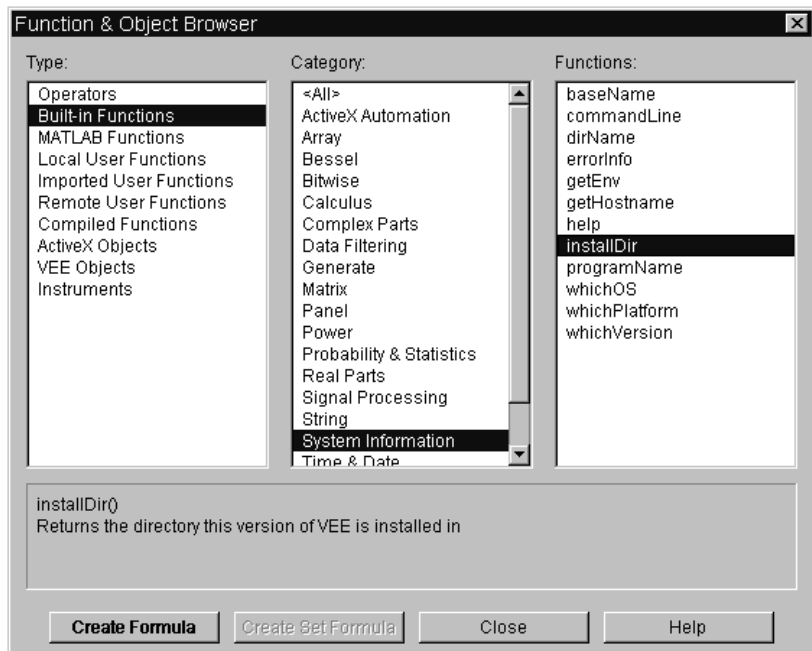


Abbildung 179 Dateiliste in einem Verzeichnis



## Schreiben von einfach portierbaren Programmen

Wenn Sie Programme in anderen Sprachen integrieren möchten, schreiben Sie die VEE-Programme so, dass sie leicht über verschiedene Plattformen hinweg portiert werden können. VEE umfasst Systeminformationsobjekte in **Function & Object Browser** ⇒ **System Information** (siehe Abbildung 180). Diese Objekte können auch als Funktionen verwendet werden.



**Abbildung 180** Systeminformationsfunktionen

Die zur Verbesserung der Portierbarkeit von Programmen verwendeten allgemeinen Systeminformationsfunktionen im Function & Object Browser sind:

**Tabelle 37** Systeminformationsfunktionen

<b>Funktionsname</b>	<b>Beschreibung</b>
installDir	Gibt den Namen des VEE-Installationsverzeichnis an.
whichOS	<p>Gibt das Betriebssystem an und sendet eine der folgenden Zeichenfolgen: <b>Windows_98</b>, <b>Windows_2000</b>, <b>Windows_NT</b>, <b>Windows XP</b>.</p> <p>Das Programm kann beim Integrieren von Programmen aus anderen Sprachen entsprechend diesen Ergebnissen verzweigen. Sehen Sie sich beispielsweise mit <b>manual49.vee</b> im Verzeichnis <b>examples\manual</b> ein Programm an, das <b>whichOS()</b> verwendet, um sicherzustellen, dass es den richtigen Bibliothekstyp importiert. Auf einem PC würde es eine Dynamic Link Library importieren.</p>
whichplatform	Legt das Hardwaresystem fest, auf dem VEE ausgeführt wird, und gibt anschließend eine Zeichenfolge zurück, die diese Plattform kennzeichnet.
whichVersion	Gibt die VEE-Version an, was hilfreich für Wartung und Debugging ist.

## Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Den Zweck des Objekts Execute Program beschreiben.
- Einen Überblick über die Konfigurationseinstellungen am Objekt Execute Program geben.
- Den allgemeinen Prozess beschreiben, mit dem das Objekt Execute Program Daten von einem Programm an eine PC-Plattform sendet und umgekehrt.
- Betriebssystembefehlen aus VEE heraus ausführen.
- Ein Programm erstellen, das die Objekte whichOS(), whichPlatform() oder whichVersion() verwendet, sodass es unter verschiedenen Betriebssystemen verwendet werden kann.

## **8 Programme für den PC integrieren**

# 9

## Agilent VEE-Funktionen verwenden

Überblick 347

Mit Funktionen arbeiten 348

Bibliotheken mit Agilent VEE UserFunctions verwenden 361

Funktionen in großen Programmen finden 373

Agilent VEE-Programme zusammenfügen 375

Kapitel-Checkliste 377

## Agilent VEE-Funktionen verwenden

*Dieses Kapitel enthält Informationen zu folgenden Themen:*

- Eine Benutzerfunktion definieren
- Funktionen erstellen, aufrufen und bearbeiten
- Funktionsbibliotheken erstellen, zusammenführen, importieren und löschen
- Funktionen in großen Programmen finden
- Vorhandene VEE-Programme mit Testprogrammen zusammenführen

*Erforderliche Zeit für dieses Kapitel: 1 Stunde*

## Überblick

Dieses Kapitel beschäftigt sich mit VEE UserFunctions, kompilierte Funktionen und Funktionen auf entfernten Computern. Funktionen sind wiederverwendbare, modulare Codeelemente, mit denen Sie die für die Testentwicklung erforderliche Zeit erheblich verkürzen können. Durch die Wiederverwendung von Funktionen, die bereits in anderen Programmen erstellt wurden, können Sie von bereits erledigten Arbeiten profitieren, die Codegröße Ihrer Programme reduzieren und die Pflege Ihrer Testprogramme erleichtern. Sie können Funktionen auch in Gruppen als Bibliotheken verwenden, die Sie erstellen und in neue Programme einbinden können. Sie können Funktionen programm- und herstellerübergreifend nutzen.

## Mit Funktionen arbeiten

Wie viele andere Programmiersprachen verwendet VEE Funktionen zum Erstellen von Unterprogrammen, die bestimmte Aufgaben ausführen. Die Übungen in diesem Kapitel beschreiben das Erstellen, Aufrufen und Ändern von benutzerdefinierten VEE-Funktionen. Außerdem lernen Sie, wie Sie Funktionsbibliotheken erstellen, die Sie in der Entwicklungsphase oder zur Laufzeit in Programme einbinden können.

### Eine Agilent VEE-Funktion definieren

In VEE gibt es drei Arten von benutzerdefinierten Funktionen. Nachfolgend finden Sie einen Überblick über die verschiedenen Funktionstypen:

#### 1 UserFunctions

- Zum *Erstellen* einer UserFunction wählen Sie **Device** ⇒ **UserFunction**, oder klicken Sie auf **Edit** ⇒ **Create UserFunction**, während verschiedene Objekte gewählt sind.
- Zum *Aufrufen* einer UserFunction von verschiedenen Stellen in einem Programm aus verwenden Sie das Objekt **Call myFunction (Device** ⇒ **Call)** oder einen Ausdruck in einem Objekt (beispielsweise in einem Formelobjekt). Sie können auch mit dem UserFunction-Objektmenü und der Wahl von Befehlen wie **Generate** ⇒ **Call** von der UserFunction aus Aufrufobjekte im Hauptprogramm generieren.
- Zum *Ändern* einer UserFunction klicken Sie auf **Edit** ⇒ **Edit UserFunction...** und wählen die entsprechende UserFunction in dem angezeigten Listenfeld.
- Zum *Übertragen* von UserFunctions aus einem Programm in ein anderes kombinieren Sie die UserFunctions bei der Programmentwicklung oder importieren sie zur Laufzeit (**Device** ⇒ **Import Library**).

#### 2 Kompilierte Funktionen

- Zum *Erstellen* einer kompilierten Funktion arbeiten Sie außerhalb von VEE mit einer kompilierten Sprache. Anschließend legen Sie die Funktionen in einer Bibliothek wie beispielsweise einer DLL ab.



- Zum *Verbinden* („Link“) einer kompilierten Funktion mit einem Programm verwenden Sie das Objekt Import Library, über das die Bibliothek zur Laufzeit in VEE einbinden wird. (Eine ausführlichere Beschreibung finden Sie in Kapitel 12, „Optimizing Agilent VEE Programs“).
- Zum *Aufrufen* einer kompilierten Funktion verwenden Sie das Objekt Call myFunction oder einen Ausdruck in einem VEE-Objekt.

### 3 Funktionen auf entfernten Computern

- Funktionen auf entfernten Computern ähneln UserFunctions, werden jedoch auf einem entfernten, an Ihr Netzwerk angeschlossenen Host-Computer ausgeführt.

## Die Unterschiede zwischen UserObjects und UserFunctions

In den vorigen Kapiteln haben Sie bereits UserObjects erstellt und verwendet. Der Grund für die Existenz von sowohl UserObjects als auch UserFunctions in VEE liegt darin, dass die beiden Komponenten unterschiedliche Merkmale aufweisen und daher für verschiedene Zwecke eingesetzt werden können. Die Unterschiede zwischen einem UserObject und einer UserFunction sind:

Ein UserObject (in **DeviceDevice**  $\Rightarrow$  **UserObject**) ist ein von Ihnen definiertes Objekt, das genau wie jedes andere Objekt in VEE verwendet werden kann. Sie programmieren ein UserObject wie ein Unterprogramm; es bleibt jedoch grafisch auf der Anzeige erhalten. Wenn Sie es an einer anderen Stelle im Programm verwenden möchten, müssen Sie es klonen und alle Kopien verwalten. Wenn Sie ein UserObject oft klonen, wird das Programm dadurch größer und erfordert eine längere Ladezeit. Wenn Sie einem UserObject eine Funktion hinzufügen, müssen Sie diese Funktion auch allen geklonten Kopien hinzufügen, wenn diese Objekte weiterhin identisch sein sollen.

Von einer UserFunction (in **Device**  $\Rightarrow$  **UserFunction**) ist dagegen nur eine einzige Instanz im Speicher vorhanden. Eine UserFunction wird nur dann im Arbeitsbereich in einem eigenen Fenster grafisch dargestellt, wenn Sie dies ausdrücklich angeben. Andernfalls wird sie gespeichert und über das Objekt Call oder ein anderes Objekt, das Ausdrücke akzeptiert, aufgerufen.

Änderungen an einer UserFunction werden von allen Instanzen in dem Programm, die diese UserFunction aufrufen, übernommen. Sie können auch Bibliotheken von UserFunctions erstellen und so den Code zusätzlich wiederverwenden.

### Übung 9-1: Operationen mit UserFunctions

In dieser Übung wird beschrieben, wie Sie eine UserFunction mit dem Namen ArrayStats erstellen, die ein Array akzeptiert, zu diesem Array den Maximalwert, den Minimalwert, den Durchschnittswert und die Standardabweichung berechnet und die Ergebnisse an eigenen Ausgangs-Pins anlegt.

#### Erstellen einer UserFunction

- 1 Wählen Sie **Device**  $\Rightarrow$  **Formula**, löschen Sie das Standard-Eingangs-Pin und ändern Sie den Ausdruck in `ramp(1024,1,1024)`.  
  
Dadurch wird ein Array aus 1024 Elementen mit Werten von 1 bis 1024 erstellt.
- 2 Wählen Sie **Device**  $\Rightarrow$  **UserFunction**. Benennen Sie das Objekt um in ArrayStats.
  - a Fügen Sie einen Dateneingangsanschluss für das Array hinzu.
  - b Fügen Sie vier Datenausgangsanschlüsse für die Ergebnisse hinzu.
  - c Benennen Sie die Ausgangsanschlüsse um: Max, Min, Mean und Sdev. Wählen Sie im Function & Object Browser in der Kategorie Probability & Statistics die Einträge max, min, mean und sdev.
  - d Platzieren Sie sie in ArrayStats und verbinden Sie ihre Dateneingänge mit **A** und ihre Datenausgänge mit den entsprechenden Ausgangsanschlüssen. Verkleinern Sie das Fenster ArrayStats, um sowohl das Hauptfenster als auch das Fenster ArrayStats anzuzeigen. Siehe Abbildung 181.

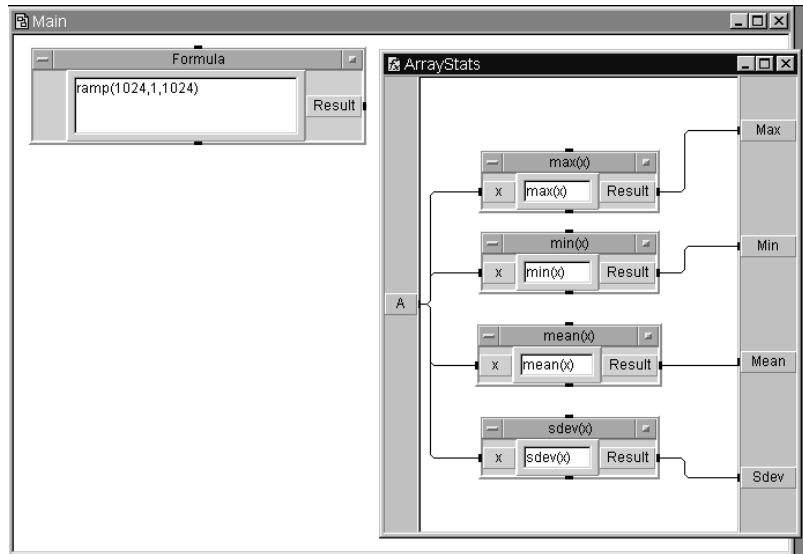


Abbildung 181 Hauptfenster und Fenster „ArrayStats“

- 3 Minimieren Sie ArrayStats. Das Symbol wird am unteren Rand des Arbeitsbereichs angezeigt.
- 4 Klicken Sie auf **Device** ⇒ **Call**, öffnen Sie das Objektmenü und klicken Sie auf Select Function, wie in Abbildung 182 gezeigt. Klicken Sie anschließend auf **OK**. Beachten Sie, dass VEE das Objekt automatisch umbenennt und die richtigen Pins hinzufügt.

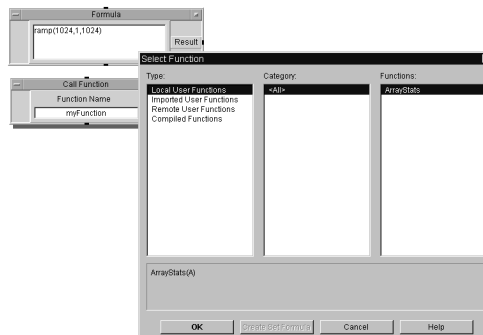
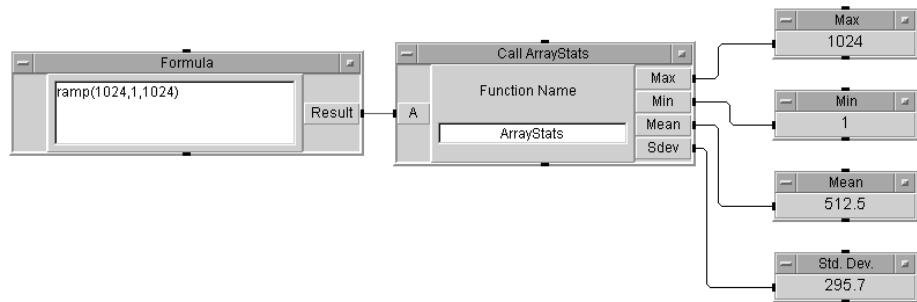


Abbildung 182 Die Pins für Call myFunction konfigurieren

- 5 Verbinden Sie den Ausgang von Formula mit dem Eingang von Call ArrayStats. Wählen Sie **Display** ⇒ **AlphaNumeric**, klonen Sie dieses Objekt drei Mal und verbinden Sie die Anzeigen mit den Ausgangs-Pins von Call ArrayStats. Benennen Sie die Anzeigen um.
- 6 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 183. Speichern Sie das Programm als **array\_stats.vee**.



**Abbildung 183** Aufruf der User Function ArrayStats

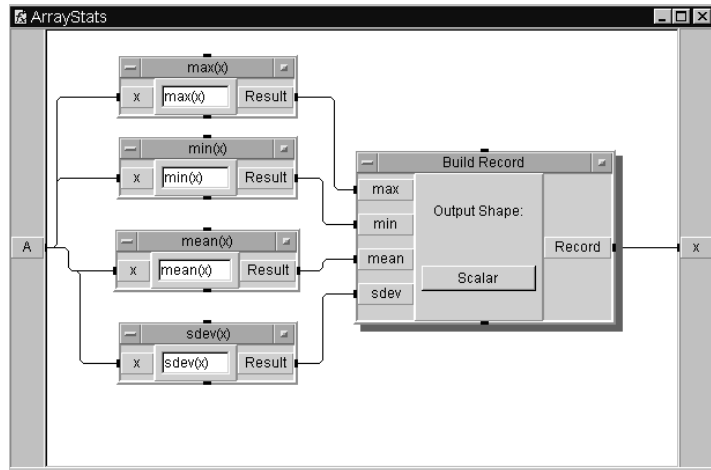
Zur Verwendung von ArrayStats an einer anderen Stelle im Programm klicken Sie auf **Device** ⇒ **Call**, öffnen das Fenster Select Function über das Objektmenü und wählen ArrayStats. VEE benennt in diesem Fall das Objekt Call ArrayStats automatisch um und fügt die erforderlichen Ein- und Ausgangsanschlüsse hinzu.

Wählen Sie im UserFunction-Objektmenü **Generate** ⇒ **Call**, um das Objekt Call ArrayStats zu öffnen. (Vergewissern Sie sich, dass bei dieser Aktion die UserFunction nicht auf den gesamten Arbeitsbereich erweitert wurde.)

## Ändern einer UserFunction

In dieser Übung ändern Sie ArrayStats so, dass es einen Datensatz mit vier Feldern liefert, die die Array-Statistik angeben.

- 1 Löschen Sie die vier AlphaNumeric-Anzeigen.
- 2 Wählen Sie **Edit** ⇒ **Edit UserFunction...** und anschließend ArrayStats im Listenfeld Edit UserFunction. Alle UserFunctions in dem Programm werden angezeigt.
- 3 Öffnen Sie das ArrayStats-Objektmenü, klicken Sie **size** an und vergrößern Sie das Bearbeitungsfenster. Wenn Sie die Größe eines Objekts ändern müssen, klicken und ziehen Sie eine beliebige Ecke des Objekts.
- 4 Löschen Sie die vier Linien zu den Ausgangsanschlüssen. (Drücken Sie die Tastenkombination **Strg+Umschalttaste** und klicken Sie auf die Linie, die Sie löschen möchten.)
- 5 Wählen Sie **Data** ⇒ **Build Data** ⇒ **Record** und platzieren Sie das Objekt rechts im Fenster ArrayStats.
  - a Fügen Sie zwei Dateneingangsanschlüsse hinzu.
  - b Beschriften Sie die vier Anschlüsse nach den statistischen Funktionen: Max, Min, Mean und Sdev.
  - c Verbinden Sie die vier Ausgänge der Formula-Objekte mit den Eingängen von Build Record.
  - d Benennen Sie den Ausgangsanschluss Max in X um, indem Sie auf Max doppelklicken, den neuen Namen eingeben und auf **OK** klicken.
  - e Löschen Sie die weiteren Datenausgangsanschlüsse von ArrayStats.
  - f Verbinden Sie den Ausgang von Build Record mit dem Ausgangsanschluss **X** des User Function-Bearbeitungsfensters. Das Programm sollte jetzt aussehen wie in Abbildung 184. Klicken Sie anschließend auf die Schaltfläche „Minimieren“.



**Abbildung 184** Die UserFunction ArrayStats ändern

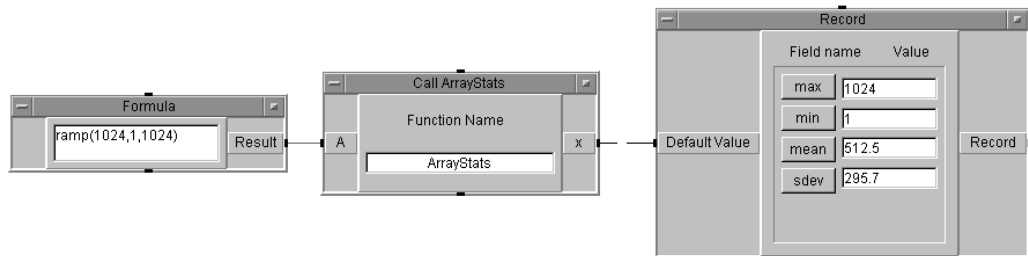
- Öffnen Sie das Call ArrayStats-Objektmenü und klicken Sie auf **Configure Pinout**. Dadurch wird die Anzahl der Pins entsprechend den letzten Änderungen angepasst.

### HINWEIS

Zum Aktualisieren der Anzahl von Pins müssen Sie das Objekt öffnen und auf **Configure Pinout** klicken, wann immer Sie die Anzahl von Eingängen und/oder Ausgängen in einer UserFunction ändern. Sie können die Ein- und Ausgangs-Pins des Call-Objekts auch manuell aktualisieren; die Verwendung von **Configure Pinout** ist jedoch erheblich einfacher. Sie können mit Find alle Call-Objekte und Ausdrücke finden, die eine UserFunction aufrufen. Weitere Informationen hierzu finden Sie unter „Funktionen in großen Programmen finden“ auf Seite 373.

Zeigen Sie jetzt einen Datensatz mit dem Objekt Record Constant an. Verwenden Sie den Steuereingang Default Value zum Akzeptieren eines Datensatzes von ArrayStats. VEE konfiguriert das Objekt Record Constant automatisch für das Speichern des ankommenden Datensatzes.

- 7 Wählen Sie **Data** ⇒ **Constant** ⇒ **Record** und platzieren Sie dieses Objekt rechts von dem Objekt Call Function.
  - a Öffnen Sie das Reord-Objektmenü und klicken Sie auf **Add Terminal** ⇒ **Control Input...**. Wählen Sie im angezeigten Listenfeld das Element **Value**. Sie können das Menü **Properties** öffnen, um die Anschlüsse anzuzeigen (Show Terminals).
  - b Verbinden Sie jetzt den Datenausgang von Call Function mit dem Steuereingangs-Pin am Record-Objekt. Steuerlinien werden durch gestrichelte Linien gekennzeichnet, um sie von Datenlinien zu unterscheiden.
- 8 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 185.



**Abbildung 185** Ausgabe an einen Datensatz nach dem Ändern von ArrayStats

## Eine UserFunction über einen Ausdruck aufrufen

In dieser Übung lernen Sie, wie Sie ArrayStats von einem Ausdruck im Formula-Objekt aus aufrufen.

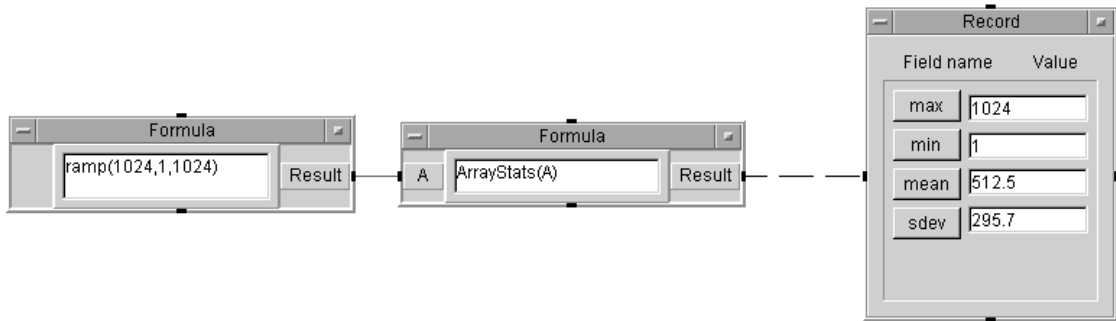
- 1 Wählen Sie **Device** ⇒ **Formula** und ersetzen Sie die Standardformel durch **ArrayStats(A)**. Klicken Sie im Call ArrayStats-Objektmenü auf **Replace**.

Die Statusleiste am unteren Rand der VEE-Anzeige fordert Sie auf, das Ersatzobjekt zu wählen. Klicken Sie auf das Formula-Objekt, das die ArrayStats-Funktion aufruft. VEE ersetzt das Call ArrayStats-Objekt automatisch durch das neue Formula-Objekt und behält die Datenlinien bei.

## 9 Agilent VEE-Funktionen verwenden

Das Formula-Objekt arbeitet mit der Eingabe am Anschluss **A** und sendet sie an die UserFunction ArrayStats. ArrayStats liefert den Datensatz der Statistik an seinen Anschluss **X**. Der erste Ausgangswert von der **UserFunction (X)** wird an das Formula-Objekt zurückgegeben und an seinen Result-Ausgang geliefert.

- 2 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 186.



**Abbildung 186** Die User Function ArrayStats aufrufen

Beachten Sie, dass die Funktionalität von ArrayStats im Objekt Formula exakt dieselbe ist wie zuvor im Objekt Call ArrayStats. Dieses Beispiel verwendet ein Formula-Objekt; Sie können jedoch ArrayStats von *jedem beliebigen* Eingabefeld aus aufrufen, das Ausdrücke akzeptiert, z. B. im Objekt To File.

### HINWEIS

Wenn Sie eine UserFunction von einem Ausdruck aus aufrufen, liefert die UserFunction nur einen einzigen Ausgang (das oberste Datenausgangs-Pin). Wenn Sie alle Ausgänge benötigen oder sie nicht in einem Record abgelegt werden können, verwenden Sie das Objekt Call Function.



**HINWEIS**

Wenn Sie eine UserFunction von einem Ausdruck aus aufrufen, werden Eingangsanschlüsse als Funktionsparameter verwendet, um die Funktion weiterzugeben. Auch wenn keine Daten an die Funktion übergeben werden, müssen Sie leere Klammern nach dem Funktionsnamen eingeben. Andernfalls geht VEE davon aus, dass Sie sich auf eine globale Variable oder einen Eingangsanschluss beziehen. Wenn die UserFunction mit dem Namen MyFunction beispielsweise keine Eingangsparameter hat, müssen Sie MyFunction() in einem Ausdruck angeben. Das Call-Objekt erfordert hingegen keine Klammern, da VEE weiß, dass Sie sich auf eine Funktion beziehen.

## Den Aufruf einer UserFunction generieren

Wenn Sie ein Call-Objekt im Hauptprogramm von einer UserFunction aus generieren und platzieren möchten, können Sie im UserFunction-Objektmenü den Befehl Generate verwenden. Das Untermenü Generate enthält die meisten gängigen Objekte, die eine UserFunction aufrufen. Wenn Sie ein aufrufendes Objekt wählen, kann es mit der richtigen Konfiguration mit Namen und Pins in dem aufrufenden Fenster platziert werden.

In dieser Übung lernen Sie, wie Sie das Objekt ArrayStats im Hauptprogramm von der UserFunction ArrayStats aus generieren.

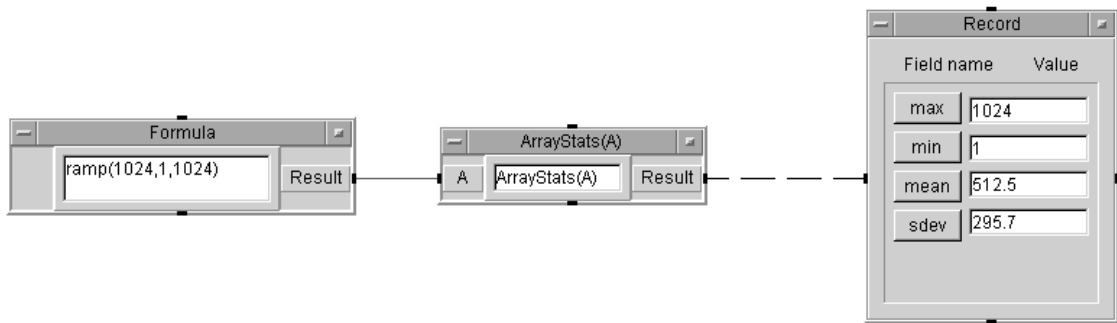
- 1 Doppelklicken Sie in dem in Abbildung 186 gezeigten Beispiel auf das Formelobjekt ArrayStats, um es zu löschen. (Sie können auch im Objektmenü den Befehl „Delete“ wählen.)
- 2 Wählen Sie in der UserFunction ArrayStats das Objektmenü und anschließend **Generate** ⇒ **Formula Call**. Abbildung 187 zeigt das Untermenü Generate in einem UserFunction-Objektmenü.



**Abbildung 187** Das Menü „Generate“ in einer UserFunction

- 3 Platzieren Sie das Objekt im Hauptprogramm. Beachten Sie, dass VEE das neue Objekt automatisch **ArrayStats(A)** benennt und den Ausdruck **ArrayStats(A)** zum Aufruf der UserFunction **ArrayStats** einbindet.
- 4 Verbinden Sie den Ausgang des Formula-Objekts mit **ArrayStats(A)** und den Ausgang von **ArrayStats(A)** mit Record.
- 5 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 188.

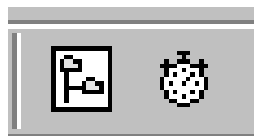
Öffnen Sie ein UserFunction-Objektmenü und wählen Sie das Menü Generate, um die anderen Objekte anzuzeigen, die in einem Programm zum Aufrufen einer UserFunction platziert werden können. Hierzu gehören die Objekte Call, Formula Call (in diesem Beispiel verwendet), If/Then/Else Call, ShowPanel und HidePanel.



**Abbildung 188** Den Aufruf des Objekts ArrayStats(A) von einer UserFunction aus generieren

### UserFunctions und der Program Explorer

UserFunctions und UserObjects gestalten VEE-Programme modularer und machen sie leichter verständlich. Der Program Explorer ist ein hilfreiches Tool zum Navigieren durch komplexe Programme. Die Hierarchie des Programms in Abbildung 189 wird beispielsweise im Program Explorer angezeigt. Klicken Sie zum Anzeigen des Program Explorer auf **View** ⇒ **Program Explorer**, oder klicken Sie in der Symbolleiste auf die Schaltfläche Program Explorer.

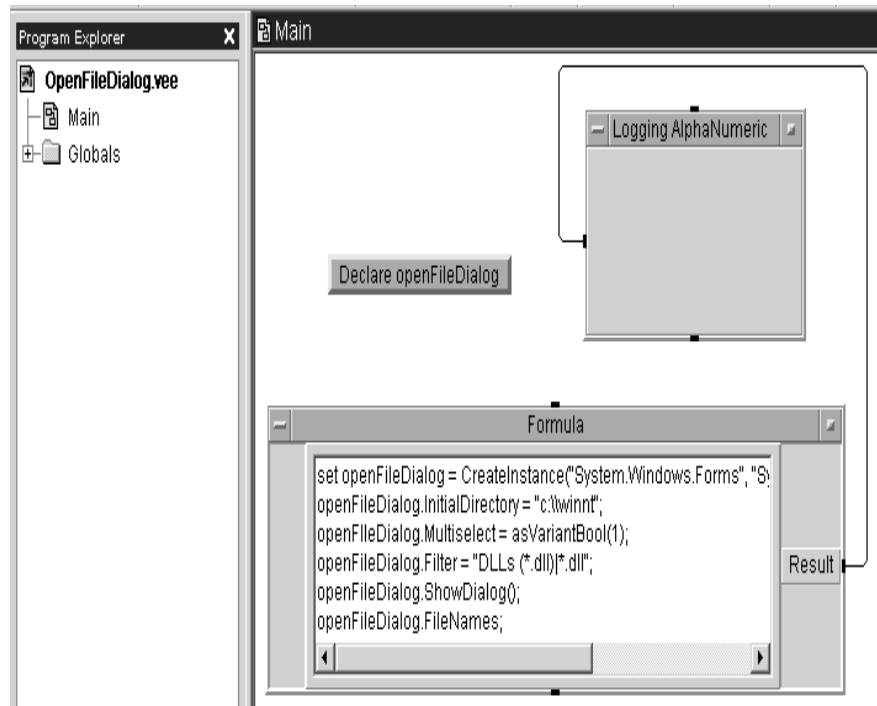


Program Explorer

**Abbildung 189** Die Schaltfläche „Program Explorer“ in der Symbolleiste

## 9 Agilent VEE-Funktionen verwenden

Abbildung 190 zeigt den verwendeten Program Explorer.



**Abbildung 190** Den Program Explorer mit UserFunctions verwenden

## Bibliotheken mit Agilent VEE UserFunctions verwenden

Sie können von vorhandenen VEE-Testprogrammen profitieren, indem Sie UserFunctions wiederverwenden. Wenn Sie ein Programm speichern, werden die UserFunctions automatisch ebenfalls gespeichert. Eine UserFunction kann ein VEE-Programm oder eine Bibliothek aus logisch zusammengehörigen UserFunctions enthalten.

Es gibt zwei Möglichkeiten, vorhandene UserFunctions in ein neues Programm zu bringen:

Legen Sie eine Kopie der Original-UserFunctions mit dem Befehl **File** ⇒ **Merge Library...** im aktuellen Programm ab (wobei Sie die separate Kopie jeder UserFunction beibehalten). Diese zusammengeführten UserFunctions können geändert werden; wenden Sie den Befehl **File** ⇒ **Merge Library...** also an, wenn Sie vorhaben, die UserFunctions zu ändern.

- ODER -

Rufen Sie die Original-UserFunctions mit dem Objekt **Device** ⇒ **Import Library** auf; dieses Objekt greift auf die Originalfunktionen in einer anderen Datei zu, ohne eine Kopie davon zu erstellen. Diese UserFunctions werden zur Laufzeit importiert. Dadurch wird die Ladezeit besser verteilt sowie Platten- und Speicherplatz gespart. Importierte UserFunctions können angezeigt werden (etwa zur Fehlerbehebung), aber nicht geändert. Sie können jedoch die entsprechenden Originaldateien ändern. Importierte UserFunctions können auch programmgesteuert über das Objekt **Device** ⇒ **Delete Library** gelöscht werden.

Verwenden Sie daher *merge* UserFunctions, wenn Sie eine neue Kopie der Funktion zum Ändern eines eigenständigen Programms benötigen, oder *import* UserFunctions, wenn Sie eine einzelne Quelle für die Funktion verwenden oder Platz sparen möchten.

## Übung 9-2: Eine Bibliothek von UserFunctions erstellen und zusammenführen

In dieser Übung erstellen Sie ein Berichtsgeneratorprogramm, das eine VEE-Bibliothek von UserFunctions enthält. Anschließend erstellen Sie ein neues Programm, das die Bibliothek der UserFunctions zusammenführt.

### Erstellen einer Bibliothek von UserFunctions

- 1 Erstellen Sie das Programm der obersten Ebene (ohne die Details der UserFunctions zu programmieren).
  - a Erstellen Sie vier UserFunctions: BuildRecAry mit einem Ausgangs-Pin, ReportHeader, ReportBody mit einem Eingangs-Pin und ReportDisplay. Stellen Sie alle UserFunctions als Symbol dar.
  - b Erstellen Sie im Hauptprogramm vier Objekte **Device Call**, konfiguriert und verbunden wie in Abbildung 191 gezeigt. Speichern Sie das Programm als **Report.vee**.

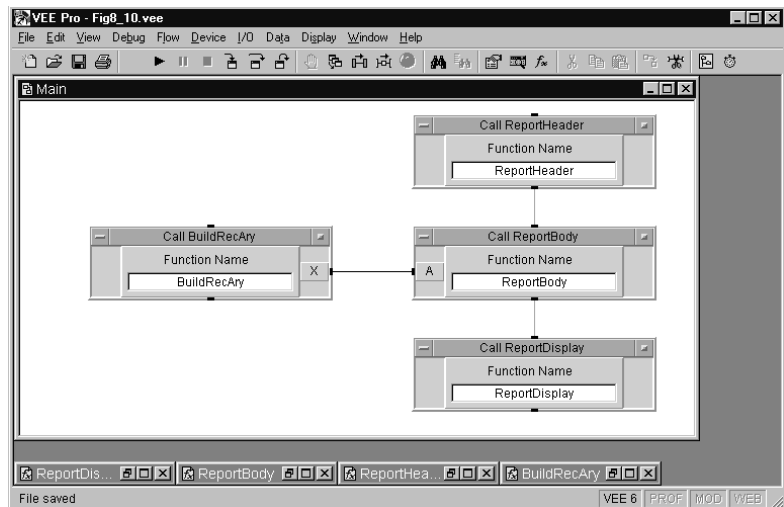


Abbildung 191 Report.vee von der obersten Ebene

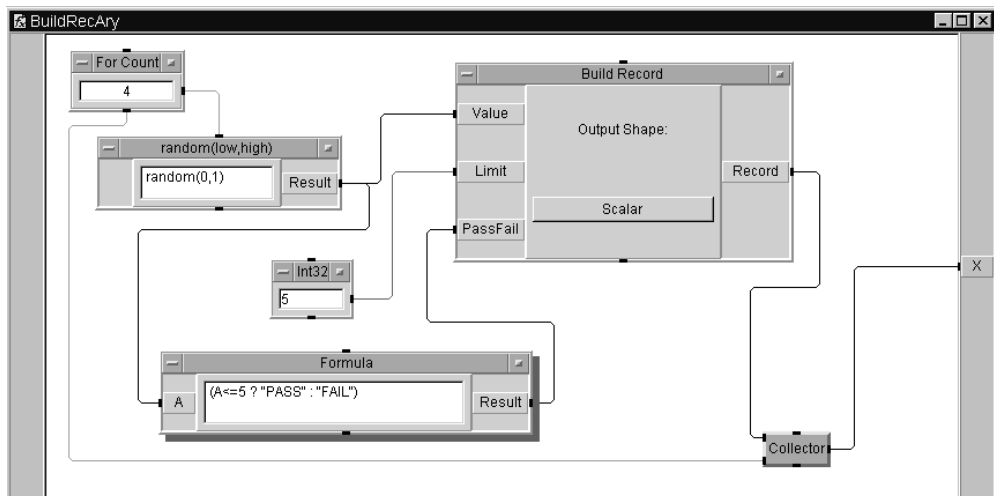
**HINWEIS**

Das Call-Objekt erfordert keine Klammern beim Verweis auf eine User-Function. Wenn Sie die Funktion von einem Formula-Objekt aus aufrufen, müssen Sie die Klammern angeben, ob die Funktion Parameter verwendet oder nicht.

Die vier UserFunctions bilden eine Bibliothek. Sie können sie auflisten, indem Sie auf **Edit** ⇒ **Edit UserFunction...** klicken. Klicken Sie auf **Cancel**, um das Listenfeld zu schließen.

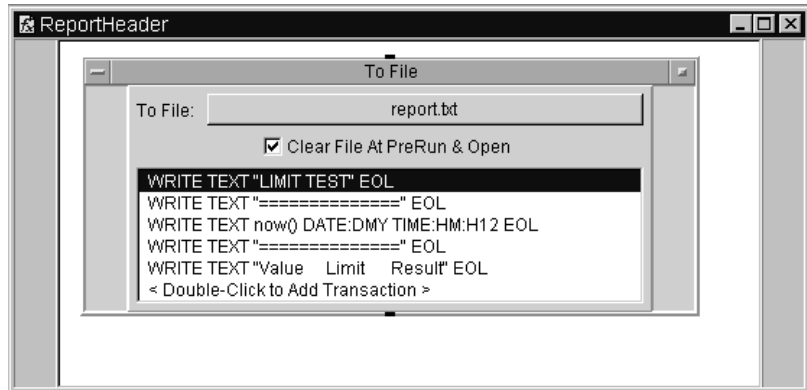
2 Programmieren Sie die vier UserFunctions wie in den folgenden Abbildungen gezeigt.

Abbildung 192 zeigt die UserFunction **BuildRecAry**. Sie enthält ein **Formula**-Objekt mit dem triadischen Ausdruckstest **if A<=5**. Wenn der Ausdruck **TRUE** ergibt, lautet die Ausgabe des Objekts „**PASS**“. Andernfalls gibt das Objekt „**FAIL**“ aus. (Beachten Sie, dass die Klammern erforderlich sind.)



**Abbildung 192** Die UserFunction „BuildRecAry“

Abbildung 193 zeigt die UserFunction **ReportHeader**.

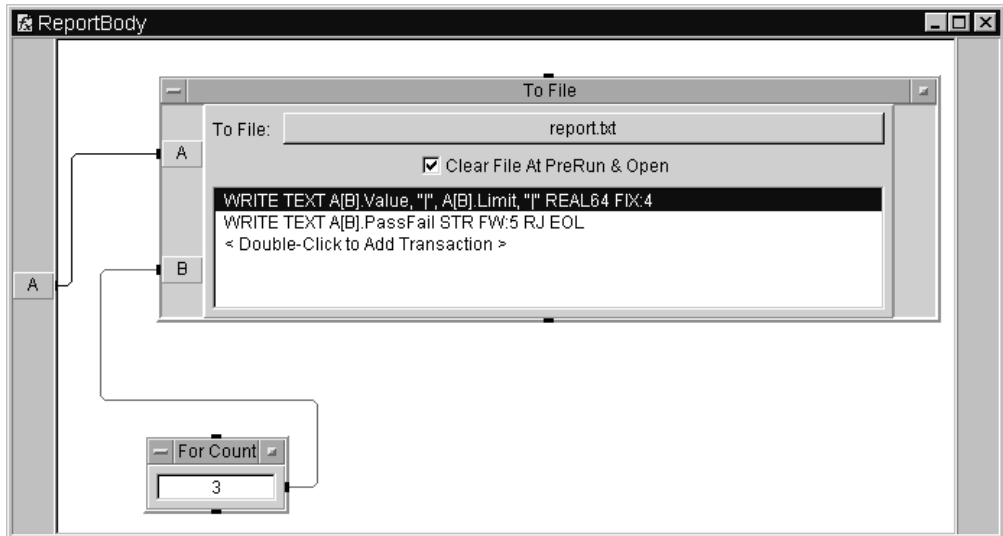


**Abbildung 193** Die UserFunction „ReportHeader“

Abbildung 194 zeigt die UserFunction **ReportBody**. Beachten Sie das Array aus Datensätzen **A[B]**. Wenn sich der Wert von **B** von 0 nach 1 nach 2 ändert, können Sie das jeweilige Feld in diesem Record einschließlich **Value**, **Limit** und **PassFail** über die Notation *<Datensatz>.<Feld>* aufrufen. Beachten Sie, dass die Ausgabe von For Count mit Null beginnt. Beachten Sie außerdem, dass für die erste Transaktion EOL off gilt.

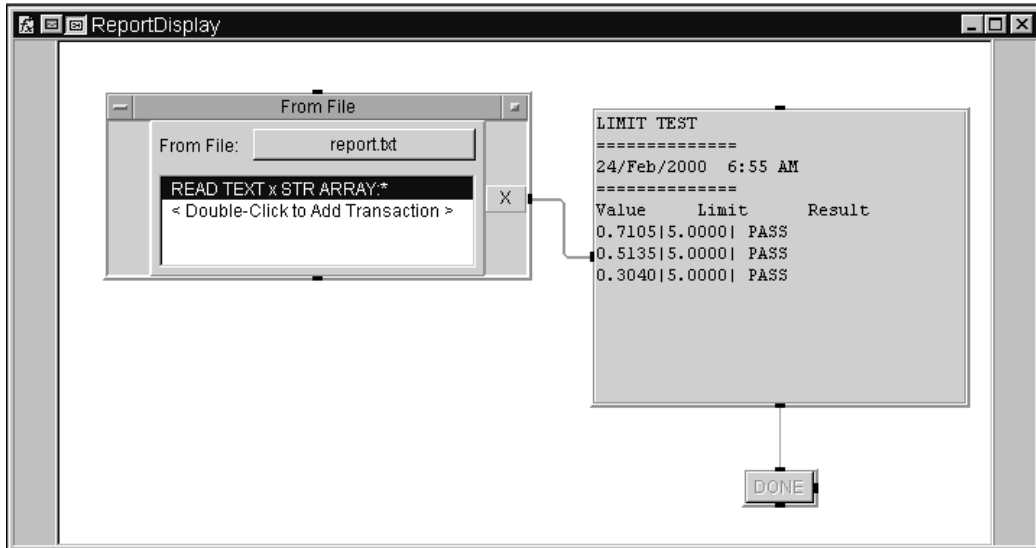
Der vertikale Balken in Hochkommas, "|", steht für eine konstante Zeichenfolge für einen vertikalen Balken. **FW:5** steht für die Breite 5 eines Zeichenfolgefilds, **RJ** steht für rechtsbündig.





**Abbildung 194** Die UserFunction „ReportBody“

Abbildung 195 zeigt die UserFunction ReportDisplay in der Detailansicht. Beachten Sie, dass sie ein Zeichenfolgen-Array bis zum Ende der Datei einliest; dies ist durch das Sternchen (\*) nach der Formatangabe **STR ARRAY** gekennzeichnet.



**Abbildung 195** Die ReportDisplay-Detailansicht

Abbildung 196 zeigt die Fensteransicht der UserFunction ReportDisplay, wobei im Fenster Properties Show Panel on Execute gewählt ist. Im Feld Properties wurde das Element Pop-up Panel Title ebenfalls in ReportDisplay geändert. Wählen Sie zum Erstellen des Fensters die Objekte Confirm (OK) und Logging AlphaNumeric und klicken Sie auf **Edit** ⇒ **Add to Panel**. Beachten Sie, dass in der Anzeige Logging AlphaNumeric die Wahl der Option Show Title Bar aufgehoben wurde. Beachten Sie außerdem, dass die Schaltfläche Confirm (OK) umbenannt wurde in DONE. Die Schaltfläche Confirm (OK) ist enthalten, um die Anzeige auf dem Bildschirm zu halten, bis der Benutzer sie vollständig gelesen hat.

- 3** Führen Sie das Programm aus. Das Fenster ReportDisplay sollte eingeblendet werden und Werte wie in Abbildung 196 anzeigen. Klicken Sie auf **DONE**, um die Ausführung abzuschließen. Speichern Sie das Programm anschließend als **Report.vee**.

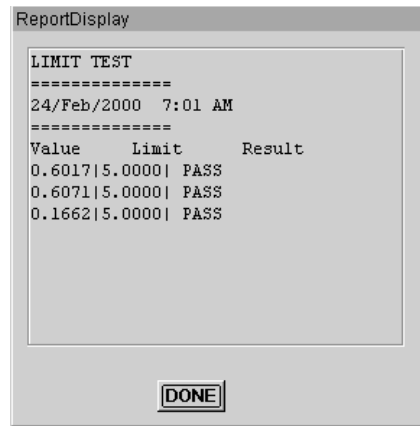


Abbildung 196 Die Fensteransicht ReportDisplay

## Ein weiteres Programm erstellen und in die Bibliothek einfügen

In dieser Übung erstellen Sie ein neues Programm und fügen die Bibliothek in dieses Programm ein. Es wird eine Bibliothek von Funktionen zum Generieren von Berichten erstellt. Das neue Programm enthält ein Note Pad-Objekt, das die einzelnen Funktionen in der Bibliothek erläutert. Der Name des Programms lautet RepGen.

Sie können RepGen wiederverwenden, indem Sie neue User-Functions zur Berichtsgenerierung erstellen, sie in das Programm einfügen und das Objekt Note Pad aktualisieren, um die Änderungen zu protokollieren. Anschließend können Sie den Befehl Merge Library... verwenden, um alle Funktionen von RepGen zu nutzen.

- 1 Wählen Sie **File** ⇒ **New**.
- 2 Klicken Sie auf **File** ⇒ **Merge Library...** Wählen Sie im Listenfeld Merge Library die Datei **Report.vee**. (Wenn Sie in einem anderen Verzeichnis arbeiten, geben Sie den vollständigen Pfadnamen an.)

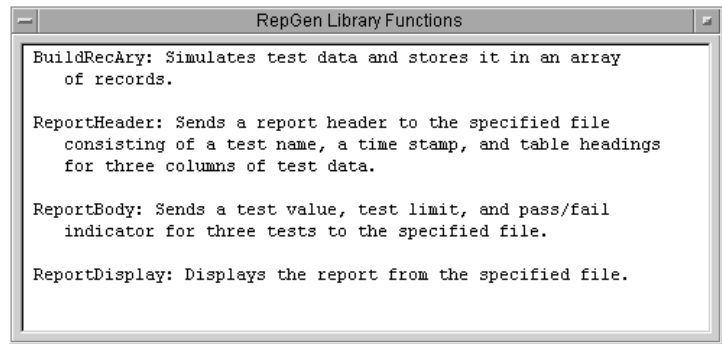
Wählen Sie **Edit** ⇒ **Edit UserFunction** (oder sehen Sie im Program Explorer nach), um sicherzustellen, dass die Bibliothek von **Report.vee** an das neue Programm übertragen wurde.

Wenn Sie den Befehl Merge Library... verwenden, können Sie zusammengefügte Funktionen genau wie lokale Funktionen verwenden.

- 3 Wählen Sie **Display** ⇒ **Note Pad** und geben Sie wie in Abbildung 197 Beschreibungen für die UserFunction ein. Speichern Sie das Programm anschließend als **RepGen.vee**.

### HINWEIS

Sie können ein „Programm“ aus UserFunctions zum Erstellen einer Bibliothek speichern, auch wenn kein tatsächliches VEE-Programm vorliegt, das die Funktionen aufruft.



**Abbildung 197** Die UserFunctions-Bibliothek RepGen.vee

### Übung 9-3: Bibliotheken importieren und löschen

Wenn Sie eine Bibliothek von UserFunctions erstellt haben, möchten Sie sie wahrscheinlich nicht in jedes Programm einbinden. Sie können daher die Bibliothek zur Laufzeit einbinden, einige der Funktionen verwenden und anschließend die Bibliothek löschen, um Speicherplatz zu sparen. Die Objekte Import Library und Delete Library wurden für diese Situation konzipiert.

In dieser Übung importieren Sie Funktionen aus dem Programm RepGen. Anschließend rufen Sie die Funktion Build-RecAry auf, um einige Testdaten zu simulieren und anzuzeigen, und löschen die Bibliothek wieder, um Speicherplatz und Auslagerungsspeicher (Swap Space) freizugeben.

- 1 Wählen Sie **File** ⇒ **New**.
- 2 Wählen Sie **Device** ⇒ **Import Library** und platzieren Sie dieses Objekt im Hauptprogramm. Geben Sie im Objekt Import Library die folgenden Werte ein:

**Tabelle 38** Die Felder des Dialogfelds Import Library

Feld	Beschreibung
Library Type (Typ der Bibliothek)	In der Dropdown-Liste im Feld Library Type können Sie zwischen einer UserFunction, einer kompilierten Funktion (Compiled Function) oder einer Funktion auf einem entfernten Computer (Remote Function) wählen. In diesem Fall möchten Sie mit einer UserFunction-Bibliothek arbeiten; behalten Sie daher den Standardwert bei.
Library Name (Name der Bibliothek)	Der Library Name zeigt <b>myLib</b> als Standardwert an. Dieser Name dient dem VEE-Programm zur Unterscheidung der verschiedenen importierten Bibliotheken. Das Objekt Delete Library verwendet diesen Namen zur Identifikation der zu löschenden Bibliothek. Sie können den Standardnamen beibehalten.
File name (Dateiname)	Über die Schaltfläche zu File Name wird standardmäßig ein Dialogfeld für das Verzeichnis der Benutzerprogramme angezeigt. Geben Sie die Datei an, die die Bibliothek der Funktionen enthält.  Klicken Sie dazu auf den Standardnamen <b>myFile</b> , um das Listenfeld Import library from what file? aufzurufen. Wählen Sie <b>RepGen.vee</b> (aus „Übung 9-2: Eine Bibliothek von UserFunctions erstellen und zusammenführen“ auf Seite 362). Diese Datei befindet sich in dem Verzeichnis, das Sie bei der Installation für Programme angegeben haben.

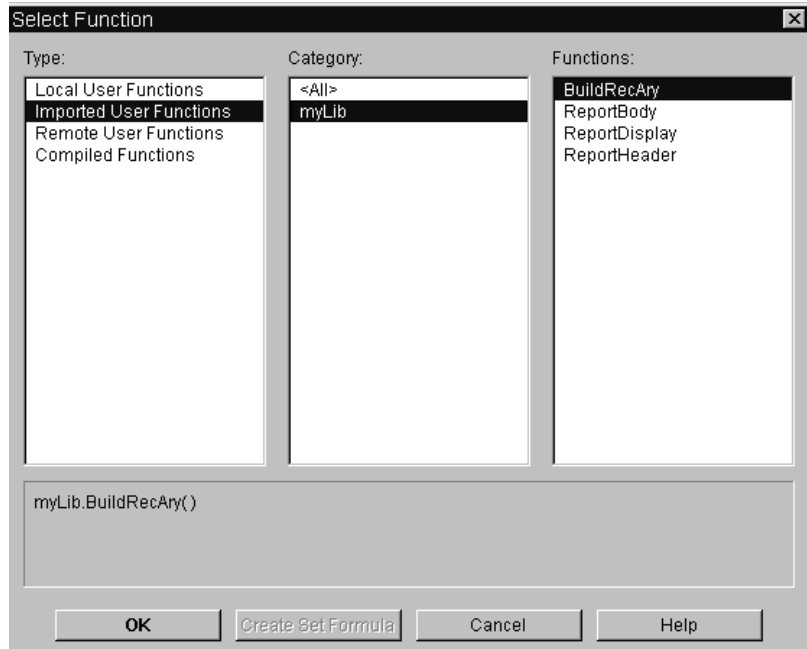
- 3 Öffnen Sie das Objektmenü und wählen Sie anschließend Load Lib, wenn Sie die Bibliothek sofort und nicht erst zur Laufzeit importieren möchten. Dieser Befehl ist der Entwicklungsphase sehr nützlich. (Im nächsten Schritt werden Sie feststellen, dass bei der Wahl von Select Function im Objekt Call die Funktionen mit der Bibliotheks-Identifikation zuerst gekennzeichnet sind, z. B. myLib.BuildRecAry.)

Verwenden Sie zum Anzeigen der Bibliothek der importierten Funktionen den Program Explorer.

### HINWEIS

Da Sie die Bibliothek importiert haben, können Sie die UserFunction nur anzeigen und Unterbrechungspunkte dafür festlegen. Sie können die User Function nicht ändern. Verwenden Sie den Befehl Merge Library..., um einem bearbeitbaren Programm eine UserFunction hinzuzufügen.

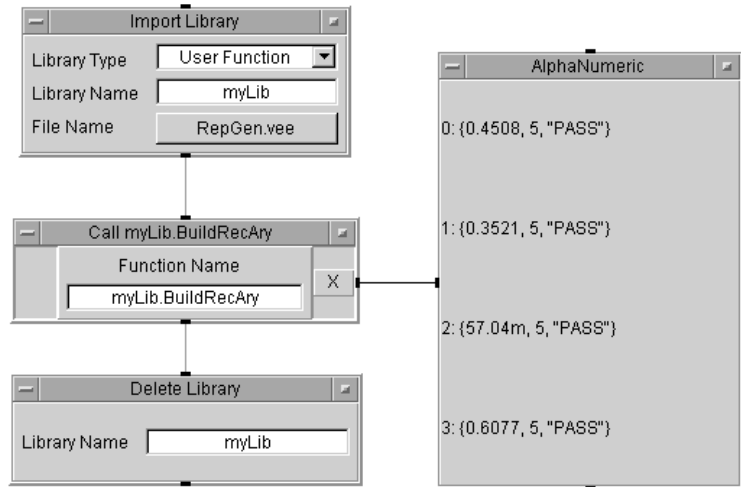
- 
- 4 Wählen Sie **Device** ⇒ **Call** und platzieren Sie dieses Objekt unter dem Objekt Import Library. Verbinden Sie das Sequenzausgangs-Pin von Import Library mit dem Sequenzeingangs-Pin am Objekt Call.
  - 5 Öffnen Sie das Call Function-Objektmenü und klicken Sie auf Select Function, um eine Liste der mit dem Befehl Load Lib importierten Funktionen anzuzeigen. Wählen Sie **myLib.BuildRecAry** (siehe Abbildung 198).



**Abbildung 198** Eine Funktion aus einer importierten Bibliothek wählen

VEE fügt die Funktion automatisch im Feld Function Name ein und fügt den erforderlichen Ausgangsanschluss hinzu. Sie könnten zu diesem Zweck auch `myLib.BuildRecAry` im Feld Function Name eingeben. Verwenden Sie `Select Function`, wenn Sie die Namen der Funktionen in der Bibliothek auflisten möchten.

- 6 Wählen Sie eine AlphaNumeric-Anzeige, vergrößern Sie sie und verbinden Sie sie mit dem Datenausgang von Call.
- 7 Wählen Sie **Device** ⇒ **Delete Library** und platzieren Sie dieses Objekt unter dem Objekt **Call**. Verbinden Sie die Sequenz-Pins, sodass die Bibliothek nach dem Aufruf der Funktion `BuildRecAry` gelöscht wird. Sie können den Standardwert für Library Name beibehalten, da dies der gleiche Name ist, den Sie mit dem Objekt `Import Library` verwendet hatten.
- 8 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 199. Speichern Sie das Programm als **libraries.vee**.



**Abbildung 199** Aufrufen einer Funktion aus einer Bibliothek

Beachten Sie folgende Punkte bei den Namen zusammengefügtter oder importierter Funktionen:

- Wenn eine zusammengefügte Funktion den gleichen Namen hat wie eine lokale Funktion, zeigt VEE einen Fehler an.
- Eine importierte Funktion darf zwar den gleichen Namen haben wie eine lokale Funktion; wenn jedoch nur der Funktionsname verwendet wird, wird die lokale Funktion aufgerufen. Sie können die importierte Funktion mit der Syntax `MyLib_func()` explizit aufrufen, wie im Objekt Call in Abbildung 199.
- Wenn zwei importierte Bibliotheken identische Funktionsnamen enthalten, ist das Ergebnis nicht bestimmt. Beachten Sie, dass das Objekt Call den Namen der Bibliothek `myLib.BuildRecAry` verwendet, somit ist eine Verwechslung ausgeschlossen. Auch wenn es eine andere lokale oder importierte Funktion mit diesem Namen gibt, ist `BuildRecAry` durch die Angabe von Namen und Position eindeutig.



## Funktionen in großen Programmen finden

VEE bietet im Menü eine Funktion Find Edit zum Suchen von Objekten und Text in großen Programmen. Öffnen Sie beispielsweise das Programm **Solitaire.vee** im Verzeichnis **Examples\Games**. Wechseln Sie zur Detailansicht und klicken Sie auf **Edit** ⇒ **Find...**, um das in Abbildung 200 dargestellte Dialogfeld anzuzeigen.

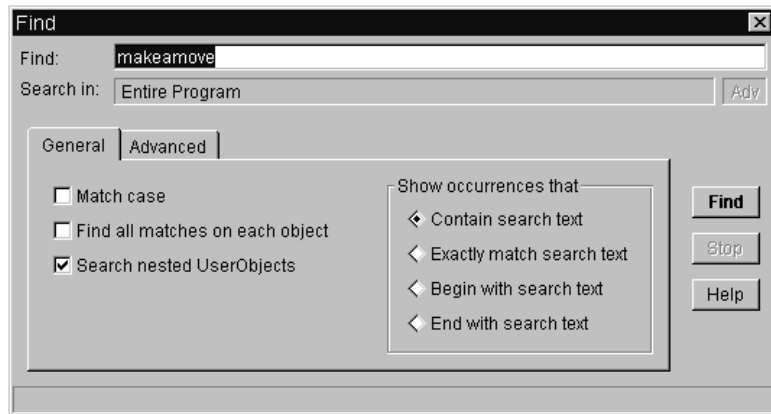


Abbildung 200 Das Dialogfeld „Find“

Geben Sie **makeamove** (eine **UserFunction** im Programm **Solitaire.vee**) wie in der Abbildung gezeigt ein und klicken Sie auf **Find**. VEE findet automatisch die UserFunction mit dem Namen **makeamove** und zeigt den Teil des Programms an, von dem aus sie aufgerufen wurde (siehe Abbildung 201).

Object Location	Object Title	Object Type	Text Found
CheckMoves	Call makeAMove	Call	Call makeAMove
makeAMove	makeAMove	UserFunction	makeAMove
makeAMovePanel	makeAMovePanel	UserFunction	makeAMovePanel
makeAMovePanel	Call makeAMove	Call	Call makeAMove

Abbildung 201 Das Dialogfeld „Find Results“

## 9 Agilent VEE-Funktionen verwenden

Sie können mit Find ein beliebiges Objekt oder einen Text wie beispielsweise Variablen, Titel, Einstellungen von Objekten etc. finden. Doppelklicken Sie auf eine beliebige Linie im Fenster Find Results, um ein Objekt zu suchen.

### HINWEIS

Sie können **Find** auch verwenden, indem Sie den Mauszeiger auf ein Objekt im Program Explorer positionieren und mit der rechten Maustaste klicken. Dadurch wird der Umfang der Suche auf die bestimmte User-Function bzw. das UserObject eingegrenzt.

---

## Agilent VEE-Programme zusammenfügen

Die einfachste Möglichkeit, vorhandene Programme weiterzuverwenden, ist das Zusammenfügen eines älteren Programms mit dem aktuellen Test. Sie können Programme weiterverwenden, indem Sie sie zusammenfügen und entsprechend Ihren Anforderungen anpassen.

Der Befehl **File** ⇒ **Merge...** fügt den Inhalt eines Programms oder einer Gruppe gespeicherter Objekte in den Arbeitsbereich ein, wobei der vorhandene Inhalt des Arbeitsbereichs erhalten bleibt. Standardmäßig zeigt **File** ⇒ **Merge...** ein Verzeichnis von Programmen an, die zusammen mit VEE ausgeliefert werden. Hierzu gehören allgemeine Programme wie beispielsweise die Anzeige eines Balkendiagramms und ein Tastenblock für die Benutzereingabe (z. B. für Kennungen).

### Übung 9-4: Ein Programm zur Anzeige von Balkendiagrammen zusammenfügen

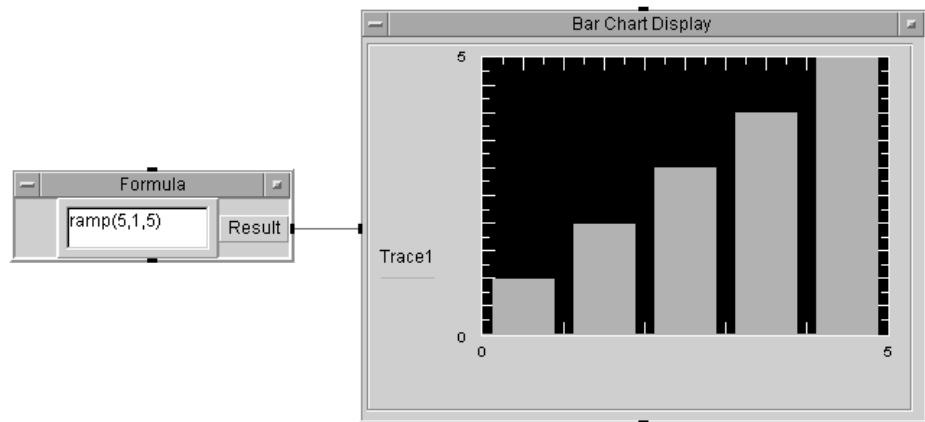
In dieser Übung fügen Sie ein bestehendes Programm mit einem neuen Programm zusammen. Das Beispiel verwendet ein Programm aus dem Verzeichnis VEE Lib; Sie können jedoch jedes beliebige Programm verwenden. Sie erstellen mit der Funktion `ramp()` ein Array aus fünf Werten von 1 bis 5. Anstatt das Array mit einer der internen Anzeigen von VEE darzustellen, fügen Sie das Programm `BarChart` in das aktuell erstellte Programm ein.

- 1 Wählen Sie im Menü `Device` den Befehl `Formula` und platzieren Sie dieses Objekt im linken Arbeitsbereich.
- 2 Löschen Sie den Dateneingangsanschluss.
- 3 Ändern Sie die Standardformel in `ramp(5,1,5)`.

Der erste Parameter ist die gewünschte Anzahl der Elemente im `ramp`-Array. Der zweite Parameter gibt die erste Zahl an, der dritte Parameter die letzte. Da die Formel jetzt den `ramp`-Aufruf enthält, können Sie weitere Informationen zu dieser Funktion anzeigen, indem Sie im `Formula`-Objektmenü den

Befehl Help wählen. (Oder rufen Sie **Help** ⇒ **Contents & Index** auf und verwenden Sie die Suchfunktion auf den Registerkarten Index oder Suchen.)

- 4 Klicken Sie auf **File** ⇒ **Merge...**, um das Dialogfeld Merge File aufzurufen. Wählen Sie **BarChart.vee** und platzieren Sie das Objekt rechts von dem Objekt Formula. Verbinden Sie die beiden Objekte.
- 5 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 202.



**Abbildung 202** Ein Balkendiagramm-Programm zusammenfügen

Beachten Sie, dass die Anzeige des Balkendiagramms ein ein-dimensionales Array erfordert und die Werte als vertikale Balken darstellt. Sie verwendet die erforderliche Anzahl von Balken zum Anzeigen der Werte in dem Array. Öffnen Sie die Detailansicht der Anzeige, um zu sehen, wie das Programm erstellt wird. Sie können Beispiele in der Bibliothek anzeigen, um weitere Anregungen zu Programmen zu erhalten.

### HINWEIS

Der Befehl **File** ⇒ **Merge** wird zum Zusammenfügen von UserObjects und Objekten verwendet. Der Befehl **File** ⇒ **Merge Library** wird zum Zusammenfügen von UserFunctions verwendet.

## Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Eine UserFunction definieren und mit einer kompilierten Funktion und einer Funktion auf einem entfernten Computer vergleichen.
- Eine UserFunction erstellen, aufrufen und bearbeiten.
- UserFunction-Bibliotheken erstellen, zusammenfügen, importieren und löschen.
- Die Find-Funktion in einem der Spielprogramme verwenden.
- Ein vollständiges VEE-Programm mit dem aktuellen Programm zusammenfügen.

## **9 Agilent VEE-Funktionen verwenden**

# 10

## Tests sequenzieren

Überblick	381
Das Sequencer-Objekt verwenden	383
Eine Reihenfolge für die Ausführung eines Tests erstellen	384
Daten im Sequencer weitergeben	397
Daten aus dem Sequencer analysieren	409
Protokollierte Daten speichern und abrufen	413
Kapitel-Checkliste	416

# Tests sequenzieren

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Das Objekt Sequencer
- Konfigurieren eines Tests für den Sequencer
- Erstellen einer Reihenfolge für die Testausführung entsprechend den Ergebnissen zur Ausführungszeit
- Aufrufen der vom Sequencer protokollierten Daten
- Möglichkeiten zum Übergeben von Daten an oder von Sequencer-Tests
- Ausführen einer Analyse mit protokollierten Daten vom Sequencer
- Speichern von Sequencer-Testdaten

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*



## Überblick

In diesem Kapitel lernen Sie die Grundfunktionen zur Verwendung des Objekts Sequencer kennen. Das Sequencer-Objekt kann eine Reihe von Sequenz-Transaktionen ausführen, die wiederum eine UserFunction, eine kompilierte Funktion oder eine Fernfunktion ausführen können. Normalerweise wird der Sequencer zur Ausführung einer Reihe von Tests verwendet.

Einige der Vorteile des Sequencer sind:

- Einfache Entwicklung eines Testplans
- Breites Spektrum von Verzweigungsfunktionen zwischen Tests
- Wichtige Komponente zum Aufbau einer angepassten Testausführung
- Möglichkeit zum Aufruf von Tests in VEE und anderen Sprachen
- Automatische Protokollierung der Testergebnisse

### HINWEIS

Der Sequencer ist eine der leistungsstärksten Funktionen von VEE. Weitere Informationen zum Sequencer finden Sie in der Online-Hilfe.

---

Die erste Übung zeigt, wie Sie einen Test für das Sequencer-Objekt konfigurieren, einen Test im Ausführungsfluss hinzufügen, einfügen oder löschen und auf die Testdaten zugreifen können, die vom Sequencer protokolliert wurden. Die Übung simuliert die Testergebnisse mit der Funktion `random()`.

In der zweiten Übung lernen Sie, wie Sie Daten strukturieren, die mit globalen Variablen an Tests übergeben wurden, wie Sie UserFunctions über den Sequencer aufrufen und wie Sie Sequencer-Dateien in Dateien protokollieren. Als letztes analysieren Sie Teile der Daten.

## 10 Tests sequenzieren

### HINWEIS

Informationen zur Verwendung eines Statusfensters zur Aktualisierung einer Testsequenz finden Sie im Abschnitt „Übung 11-5: Ein Statusanzeige erstellen“ auf Seite 455.

---

### HINWEIS

Zusätzlich zu den Übungen in diesem Kapitel können Sie weitere praktische Erfahrungen mit dem Sequencer sammeln mit der Übung in „Tests sequenzieren“ auf Seite 561 in Anhang „Tests sequenzieren“.

---

## Das Sequencer-Objekt verwenden

Das Sequencer-Objekt führt Tests in einer angegebenen Reihenfolge entsprechend den Ergebnissen zur Ausführungszeit aus. Die einzelnen Tests können VEE UserFunctions, kompilierte Funktionen, Fernfunktion oder andere Ausdrücke sein, die einzelne Ergebnisse zurückgeben. Dieses Ergebnis wird mit der Testspezifikation verglichen, um festzustellen, ob der Test erfolgreich war oder nicht. Der Sequencer verwendet dann ein Kennzeichen „erfolgreich“ oder „fehlgeschlagen“, um zu ermitteln, welcher Test als nächstes ausgeführt werden soll.

Es gibt sechs verschiedene Optionen zum Verzweigen zum nächsten Test. Diese Optionen umfassen die Ausführung des nächsten Tests, die Wiederholung des letzten Tests oder den Rücksprung zu einem früheren Test. Übung 10-1 erläutert diese Verzweigungsoptionen ausführlich. Der Sequencer kann den Benutzer auch zu einer Eingabe auffordern, um zu entscheiden, welche Aktion ausgeführt werden soll.

Nach der Ausführung der angegebenen Tests protokolliert der Sequencer die Testdaten automatisch an einen Ausgangsanschluss. Von dieser Stelle aus können Daten analysiert, angezeigt oder zur späteren Überprüfung in einer Datei gespeichert werden.

### Eine Reihenfolge für die Ausführung eines Tests erstellen

In dieser Übung simulieren Sie Testergebnisse mit der Funktion `random()`, erstellen eine Reihenfolge für die Ausführung der Tests, erfahren mehr über die Änderung dieser Reihenfolge und rufen spezifische Daten aus den protokollierten Ergebnissen ab.

#### Übung 10-1: Einen Test konfigurieren

##### HINWEIS

Das Beispiel erläutert, wie Sie die Funktion `random()` mit einem bestimmten Bereich erwarteter Testergebnisse implementieren; Sie können dasselbe Prinzip jedoch auch beim Konfigurieren anderer Tests anwenden.

---

- 1 Wählen Sie **Device** ⇒ **Sequencer** und platzieren Sie dieses Objekt oben links im Arbeitsbereich.
- 2 Wählen Sie **Display** ⇒ **AlphaNumeric**, platzieren Sie diese Anzeige unter dem Sequencer, verbreitern Sie sie und verbinden Sie den Ausgangsanschluss Log des Sequencer mit dem Dateneingang von AlphaNumeric.
- 3 Doppelklicken Sie auf die Transaktionsleiste Sequencer, um das Dialogfeld Sequence Transaction zu öffnen. Stellen Sie die Felder wie folgt ein:

##### HINWEIS

Denken Sie beim Bearbeiten der Felder daran, auf neue Felder zu klicken, um sie zu ändern, oder, dass Sie mit der Tabulatortaste von Feld zu Feld wechseln können. Verwenden Sie die Kombination Umschalt-Tabulatortaste, um den Eingabefokus in umgekehrter Reihenfolge zu setzen. Drücken Sie die Eingabetaste nur, wenn Sie mit dem Bearbeiten des Dialogfelds fertig sind.

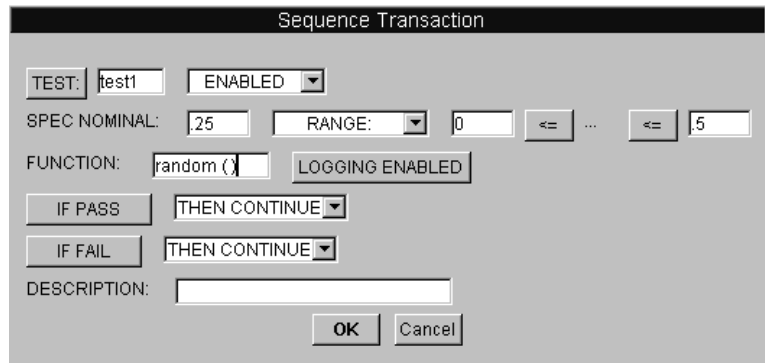
---

**Tabelle 39** Sequenz-Transaktionsfelder

Feldbezeichnung	Beschreibung
TEST:	Der Standardname lautet test1; Sie können diesen Namen verwenden. Dies ist nur die Beschriftung für den Test im Sequencer, nicht der Name der Funktion selbst.
SPEC NOMINAL:	Gibt den erwarteten Testwert an. Der Standardwert ist <b>.5</b> . Ändern Sie diesen Wert in <b>.25</b> , und ändern Sie die Obergrenze im Feld <b>RANGE</b> (ganz rechts) von <b>1</b> in <b>.5</b> .
FUNCTION:	<p>Der Standardeintrag <b>testFunc(a)</b> enthält eine Funktion zur Ausführung des Testes. Ersetzen Sie in diesem Fall das Standardfeld durch die Funktion random(). Random() gibt einen Real64-Wert zwischen <b>0</b> und <b>1</b> zurück, der das Testergebnis simuliert. Dieses Ergebnis wird mit der Spezifikation des Tests verglichen.</p> <p>Das Objekt random(low,high) befindet sich in der Kategorie Probability &amp; Statistics im Fenster Function &amp; Object Browser. Denken Sie daran, dass Sie diese mathematische Funktion von einem beliebigen Ausdrucksfeld aus mit einem Formula-Objekt aufrufen können. Wenn Sie die Parameter low und high nicht wie in diesem Beispiel gezeigt angeben, verwendet die Funktion die Standardparameter <b>0</b> und <b>1</b>.</p>

Sie können die weiteren Standardwerte beibehalten. Diese Konfiguration liefert in ca. der Hälfte aller Fälle das Ergebnis **PASS** (erfolgreich). Das Dialogfeld sollte ungefähr aussehen wie in Abbildung 203.

## 10 Tests sequenzieren



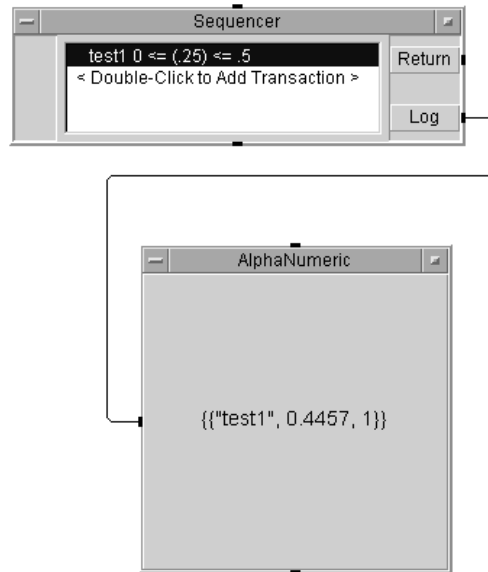
**Abbildung 203** Das Dialogfeld **Sequence Transaction**

Klicken Sie auf **OK**, um das Dialogfeld zu schließen. Sie sehen die Transaktion

**test1 0 <= (.25) <= .5**

in der ersten Transaktionsleiste. Dies bedeutet, dass test1 erfolgreich ist, wenn der zurückgegebene Wert im Bereich zwischen **0** und **.5** (je einschließlich) liegt. Das erwartete Ergebnis liegt bei ca. **.25**.

- 4 Führen Sie das Programm aus. Es sollten den Namen des Tests anzeigen, das Testergebnis und das Kennzeichen Pass/Fail (erfolgreich/fehlgeschlagen) (**1** für **PASS**, **0** für **FAIL**); siehe Abbildung 204.



**Abbildung 204** Einen Test konfigurieren

Vor der Fortsetzung sehen Sie sich Tabelle 203 an, um sich einen Überblick über die verschiedenen Auswahlmöglichkeiten im Dialogfeld Sequence Transaction zu verschaffen. Öffnen Sie das Dialogfeld erneut, indem Sie auf die Transaktionsleiste doppelklicken. Öffnen Sie die verschiedenen Menüs und lesen Sie über die verschiedenen Möglichkeiten nach.

**Tabelle 40** Das Dialogfeld Sequence Transaction

Sequenz- Transaktionsfeld	Beschreibung
TEST:	<p>Eindeutiger Name für den Verweis auf den Test im Sequencer. Die Standardnamen beginnen mit test1 und werden mit jedem Test hochgezählt. Die Auswahl von <b>TEST</b> : bedeutet, dass ein Testergebnis mit den Spezifikationen verglichen wird, und es erfolgt entsprechend der Konfiguration eine Verzweigung auf den nächsten Test.</p> <p>Die Schaltfläche <b>TEST</b> : schaltet um auf <b>EXEC</b>:. Wenn Sie das Feld <b>TEST</b> : auf <b>EXEC</b>: umschalten, wird der Test ohne Vergleich zwischen einem Testergebnis und den Spezifikationen ausgeführt. Sie könnten beispielsweise <b>EXEC</b> : wählen, wenn die UserFunction globale Variablen einrichtet. Mi der Auswahl <b>EXEC</b> : wird außerdem die Protokollierung für den Test ausgeschaltet.</p>
ENABLED	<p>Legt fest, wann ein Test ausgeführt werden soll. Dieses Menü enthält vier Auswahlmöglichkeiten:</p> <ul style="list-style-type: none"> <li>• <b>ENABLED</b> führt den Test unter allen Umständen aus.</li> <li>• <b>ENABLED IF</b> : führt den Test aus, wenn der angegebene Ausdruck <b>TRUE</b> ist. Der Test kann beispielsweise aktiviert sein, wenn der Eingangs-Pin <b>A</b> den Wert <b>1 (A == 1)</b> enthält. Sie können <b>ENABLED IF</b>: zur Testüberwachung verwenden. Auf diese Weise kann beispielsweise ein bestimmter Test immer nach zehn Durchläufen gestartet werden.</li> <li>• <b>DISABLED</b> ist das Gegenteil von <b>ENABLED</b> .</li> <li>• <b>DISABLED IF</b> : ist das Gegenteil von <b>ENABLED IF</b> : .</li> </ul>
SPEC NOMINAL:	Der von dem Test erwartete Wert.



**Tabelle 40** Das Dialogfeld Sequence Transaction

Sequenz- Transaktionsfeld	Beschreibung
RANGE:	<p>Gibt den Bereich der Testwerte an. Dieses Menü enthält vier Auswahlmöglichkeiten:</p> <ul style="list-style-type: none"> <li>• <b>RANGE</b> gibt den Bereich der Testwerte an, die eine Bedingung <b>PASS</b> kennzeichnen. Sie können außerdem unter den üblichen Vergleichen wählen:  <math>&gt;</math>, <math>&gt;=</math>, <math>&lt;</math>, <math>&lt;=</math>, <math>=</math>, <math>!=</math>.</li> <li>• <b>LIMIT</b> verwendet nur einen einzigen Wert für den Vergleich von Testdaten.</li> <li>• <b>TOLERANCE</b> gibt den Bereich der „erfolgreichen“ Werte durch Addieren oder Subtrahieren der angegebenen Toleranz zu bzw. von dem Wert <b>SPEC NOMINAL</b> an.</li> <li>• <b>%TOLERANCE</b> gibt den Bereich der „erfolgreichen“ Werte durch Addieren oder Subtrahieren einer prozentualen Toleranz des Werts <b>SPEC NOMINAL</b> zu bzw. von der normalen Spezifikation an.</li> </ul>
FUNCTION:	<p>Gibt den auszuführenden Test an. Sie können UserFunctions, kompilierte Funktionen oder Fernfunktionen aufrufen, oder Sie können einen Ausdruck eintragen, der ausgewertet werden soll. Das Ergebnis der aufgerufenen Funktion (bzw. des ausgewerteten Ausdrucks) wird mit den Spezifikationen verglichen.</p> <p>Wenn eine UserFunction mehr als einen Wert zurückgibt, geht VEE davon aus, dass der oberste Ausgabe-Pin das zu testende Ergebnis enthält.</p> <p>Funktionen können auch kombiniert und verschachtelt werden: Beispiel: <math>(\text{random}(0, \text{myfunc}()) + 3, 100) * 2</math>.</p>

**Tabelle 40** Das Dialogfeld Sequence Transaction

Sequenz- Transaktionsfeld	Beschreibung
LOGGING ENABLED	<p>Protokolliert Testdaten. Zum Angeben von Protokollierungsoptionen öffnen Sie das Sequencer-Objektmenü, wählen Properties, klicken auf den Ordner Logging und treffen Ihre Auswahl in der Liste. Standardmäßig sind Name, Result und Pass gewählt. Außerdem steht ein Feld zur Verfügung, in dem Sie zwischen Log to Output Pin Only (Nur an Ausgangs-Pin protokollieren) und Log Each Transaction To: (Jede Transaktion protokollieren an:) wählen können. Wenn die Protokollierung aktiviert ist, protokolliert jeder Test einen Record.</p> <p>Schaltet um auf <b>LOGGING DISABLED</b>.</p>
IF PASS	<p>Legt die Verzweigungsanleitungen fest. Wenn der Test erfolgreich war, springt VEE in diese Zeile, um die Verzweigungsanweisungen auszuführen. <b>IF PASS</b> weist VEE an, entsprechend der Auswahl im Dropdown-Menü zu verzweigen.</p> <p>Diese Schaltfläche schaltet außerdem um zu <b>IF PASS CALL</b> : .</p> <p><b>IF PASS CALL</b> : weist VEE an, die angegebene Funktion aufzurufen und anschließend zur Verzweigungs-Menüauswahl zu springen.</p> <p>(Schlagen Sie auch unter <b>THEN CONTINUE</b>, dem nächsten Element in dieser Tabelle, nach.)</p>

Tabelle 40 Das Dialogfeld Sequence Transaction

Sequenz- Transaktionsfeld	Beschreibung
THEN CONTINUE	<p>Legt die Testverzweigung fest. Das Dropdown-Menü <b>THEN CONTINUE</b> (für <b>IF PASS</b> und <b>IF FAIL</b>) enthält sechs Verzweigungsoptionen:</p> <ul style="list-style-type: none"> <li>• <b>THEN CONTINUE</b> führt den nächsten im <b>Sequencer</b> konfigurierten Test aus.</li> <li>• <b>THEN RETURN</b> : weist VEE an, die Ausführung von Tests zu stoppen und den angegebenen Ausdruck am Return-Ausgangs-Pin des Sequencer anzulegen.</li> <li>• <b>THEN GOTO</b> : springt zu dem in diesem Feld angegebenen Test.</li> <li>• <b>THEN REPEAT</b> wiederholt den aktuellen Test bis zu der im <b>MAX TIMES</b> : Feld angegebenen Anzahl. Wenn die Bedingung <b>PASS/FAIL</b> auch nach der maximalen Anzahl von Wiederholungen weiterhin erfüllt ist, fährt VEE mit dem nächsten Test fort.</li> <li>• <b>THEN ERROR</b>: stoppt die Ausführung durch Generieren einer Fehlerbedingung mit der angegebenen Fehlernummer. Ein Fehler kann mit dem Ausgangs-Pin Error des Sequencer erfasst werden. Keiner der anderen Ausgangs-Pins sendet Daten.</li> <li>• <b>THEN EVALUATE</b> : ruft die angegebene User Function auf, die eine Zeichenfolge zurückgeben muss, die wiederum eine Verzweigungs-Menüoption angeben muss. Gültige Zeichenfolgeergebnisse von der UserFunction sind: „Continue“, „Return &lt;expr&gt;“, „Goto &lt;name&gt;“, „Repeat &lt;expr&gt;“, „Error &lt;expr&gt;“, wobei &lt;expr&gt; ein beliebiger gültiger VEE-Ausdruck und &lt;name&gt; der Name eines Tests in einer Sequenz ist. Diese Option gibt Ihnen die Möglichkeit, den Benutzer zu fragen, welche Aktion als nächste ausgeführt werden soll.</li> </ul>

**Tabelle 40** Das Dialogfeld Sequence Transaction

Sequenz- Transaktionsfeld	Beschreibung
IF FAIL	Verzweigungsanleitungen. Wenn der Test fehlgeschlagen ist, springt VEE in diese Zeile, um die Verzweigungsanweisungen auszuführen. <b>IF FAIL</b> schaltet um auf <b>IF FAIL CALL</b> : . Die Optionen sind die gleichen wie für <b>IF PASS</b> .
DESCRIPTION:	Textkommentare zu dem Test. Diese Kommentare werden in der Sequencer-Transaktionsleiste angezeigt. Sie können zusammen mit dem Test-Record gespeichert werden; verwenden Sie hierzu den Ordner Logging im Fenster Properties.

### Einen Test hinzufügen, einfügen oder löschen

In diesem Abschnitt fügen Sie dem Sequencer-Objekt eine weitere Test- Transaktion hinzu. Sie können dieselbe random()-Funktion zum Simulieren eines Testergebnisses verwenden; dieses Mal vergleichen Sie jedoch das Testergebnis mit einem Grenzwert statt mit einem Wertebereich.

- 1 Doppelklicken Sie unter der ersten Transaktionsleiste Sequencer, um das Dialogfeld Sequence Transaction zu öffnen. Füllen Sie die Felder wie folgt aus:

**Tabelle 41** Das Dialogfeld Sequencer Transaction

Element	Beschreibung
test2	Verwenden Sie den Standardwert.
SPEC NOMINAL:	Ändern Sie die Einstellung von <b>.5</b> zu <b>. 25</b> .
RANGE:	Wählen Sie im Dropdown-Menü <b>LIMIT</b> :. Wählen Sie < für den Operator. Ändern Sie <b>1</b> zu <b>. 5</b> für den Grenzwert.
FUNCTION	Ändern Sie das Feld von testFunc(a) in random().

Behalten Sie die weiteren Standardwerte bei und klicken Sie auf **OK**, um zum Sequencer zurückzukehren.

**HINWEIS**

Sie können auch eine Transaktion nach der hervorgehobenen hinzufügen, indem Sie im Objektmenü **Add Trans...** wählen.

Der Sequencer-Testplan sollte jetzt die folgende zweite Transaktion enthalten: **test2 (.25) < .5**. Fügen Sie jetzt eine Transaktion zwischen diesen beiden Tests ein.

- 2 Vergewissern Sie sich, dass die zweite Transaktionsleiste hervorgehoben ist. Öffnen Sie anschließend das Objektmenü und wählen Sie **Insert Trans**. Füllen Sie die Felder wie folgt aus:

**Tabelle 42** Eine Transaktion einfügen

Feld	Beschreibung
TEST:	Ändern Sie das Namensfeld in <b>Insert</b> .
FUNCTION	Ändern Sie die Angabe in <b>random()</b> .

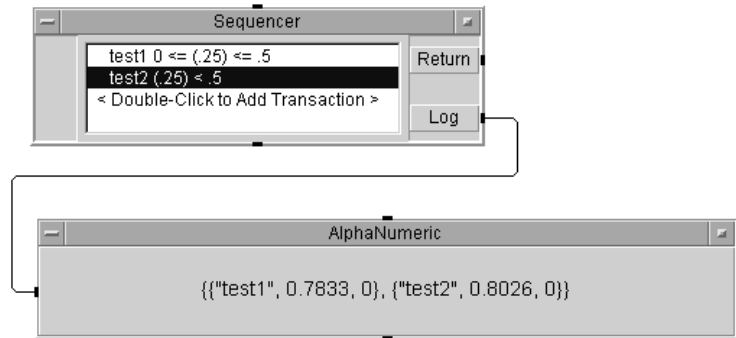
Klicken Sie auf **OK**. In der zweiten Transaktionsleiste wird jetzt **Insert 0 <= (.5) <= 1** angezeigt. Führen Sie das Programm aus, um die drei Records aus den drei Tests anzuzeigen. (Eventuell müssen Sie die Anzeige vergrößern, um alle drei Einträge zu sehen.)

- 3 Löschen Sie jetzt **Insert**, indem Sie auf die Transaktionsleiste **Insert** klicken, mit dem Mauszeiger auf die Transaktionsleiste **Insert** zeigen und **Strg+K** drücken.

**HINWEIS**

Sie können auch auf die Ziel-Transaktionsleiste klicken und im Objektmenü **Cut Trans** wählen. Außerdem können Sie eine zuvor ausgeschnittene Transaktion einfügen, indem Sie im Objektmenü **Paste Trans** wählen (oder die Tastenkombination **Strg+Y** drücken). Auf ähnliche Weise können Sie eine Transaktion mit der Auswahl **Copy Trans** kopieren.

- 4 Führen Sie das Programm aus und notieren Sie die beiden Records aus den beiden Tests. Speichern Sie das Programm als **seq1.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 205.

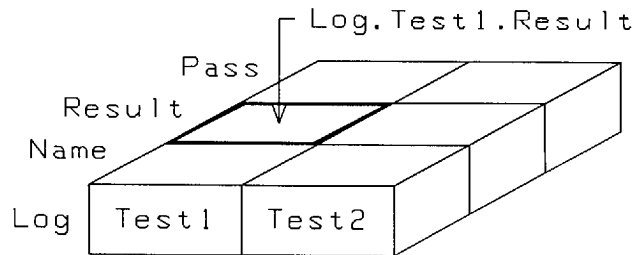


**Abbildung 205** Ein einfaches **Sequencer**-Beispiel

Die Klammern kennzeichnen einen Datentyp Record. Der Sequencer gibt einen Record of Records aus, wie in der Anzeige AlphaNumeric dargestellt. Dies bedeutet, dass Sie den Sequencer in einer Schleife ausführen und die gleiche Testsequenz mehrmals ausführen können, wobei Sie mehrmals ein Array aus Records of Records erhalten.

### Auf die protokollierten Testdaten zugreifen

Der Sequencer gibt einen Record of Records (Datensatz aus Datensätzen). Jeder Test verwendet den Testnamen als seinen Feldnamen im Sequencer-Record. Die Felder in den einzelnen Tests werden entsprechend der Konfiguration für die Protokollierung benannt. Über die Standardkonfiguration mit den Feldern Name, Result und Pass können Sie auf das Ergebnis in test1 mit der Notation `Log.Test1.Result` zugreifen (siehe Abbildung 206).



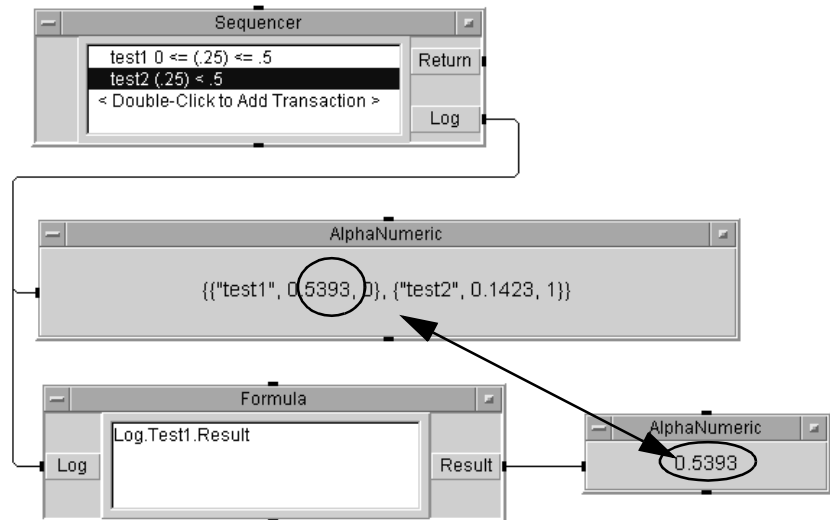
**Abbildung 206** Ein oder mehrere protokollierte Records

Der Zugriff auf die Testergebnisse erfolgt auf die folgende Weise.

- 1 Öffnen Sie **seq1.vee**.
- 2 Wählen Sie **Device** ⇒ **Formula** und platzieren Sie das Objekt unter der Anzeige. Ändern Sie den Ausdruck in **Log.Test1.Result**. (Denken Sie daran, dass in VEE nicht zwischen Groß- und Kleinbuchstaben unterschieden wird.

Ändern Sie den Namen des Eingangsanschlusses von **A** zu **Log**. (Sie können auch den Standardnamen **A** beibehalten; die Formel lautet dann **A.Test1.Result**.) Verbinden Sie den Ausgangsanschluss **Log** des Sequencer mit dem Eingangsanschluss **Log** von Formula.

- 3 Wählen Sie **Display** ⇒ **AlphaNumeric** und verbinden Sie das Objekt mit dem Formula-Ausgang.
- 4 Führen Sie das Programm aus. Es sollte auf das Feld **Result** in **Test1** zugreifen. Speichern Sie das Programm als **seq2.vee**. Das Programm sollte jetzt aussehen wie in Abbildung 207.



**Abbildung 207** Auf protokollierte Daten zugreifen

### HINWEIS

Jeder Test erstellt bei der Ausführung im Sequencer einen Record (Datensatz) mit dem Namen des Tests. Dieser Record kann in späteren Tests verwendet werden. Sie können beispielsweise **test2** aktivieren, wenn test1 erfolgreich war (ENABLED IF: test1.pass == 1). Wenn Sie die Testdaten in einem Ausdrucksfeld aufrufen müssen, während der Test noch aktiv ist, werden die Testdaten in dem temporären Record thistest gespeichert.

- 5 Ändern Sie die Formel, sodass sie Log.test1 lautet, und führen Sie das Programm erneut aus. Es sollte den vollständigen Record für test1 abrufen; dies wird durch die Klammern um die drei Werte in der Anzeige gekennzeichnet.
- 6 Durch die Änderung der Formel können Sie auf result, pass, name und andere Felder in den Records test1 und test2 zugreifen. Wählen Sie im Fenster Properties das Register Logging und fügen Sie den protokollierten Records die Felder Nominal und Time Stamp hinzu. Greifen Sie auf diese neuen Felder mit dem Formula-Objekt zu.



## Daten im Sequencer weitergeben

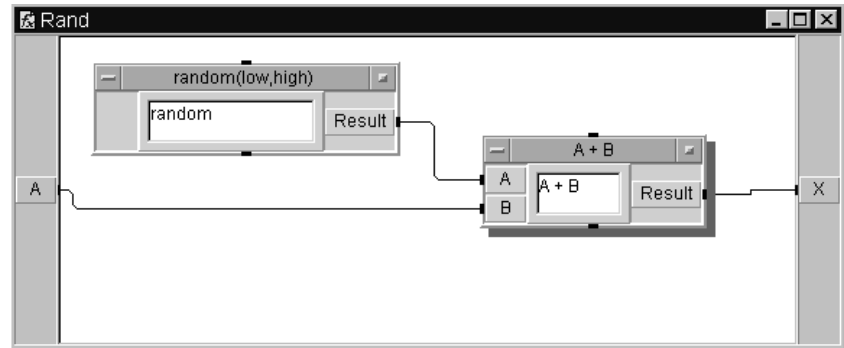
In dieser Übung erstellen Sie eine UserFunction und rufen Sie aus drei verschiedenen Tests auf. Im ersten Teil geben Sie Daten an die UserFunctions über einen Eingangsanschluss am Sequencer weiter. Im zweiten Teil ändern Sie das Programm, sodass es eine globale Variable statt eines Eingangsanschlusses verwendet. Dadurch haben Sie die Möglichkeit, eine Funktion im EXEC-Modus statt im TEST-Modus aufzurufen. Im dritten Teil lernen Sie, wie Sie eine Wellenformausgabe mit einer Maske testen.

### Übung 10-2: Daten mit einem Eingangsanschluss weitergeben

Führen Sie zunächst die Schritte zum Erstellen der UserFunction Rand aus, die einen Messprozess simulieren wird. Rand() fügt dem Ausgang des Objekts random(low,high) einen Eingangsparameter hinzu und legt dieses Ergebnis am Ausgangsanschluss ab. Rand() wird aus drei verschiedenen Tests aufgerufen.

- 1 Wählen Sie **Device**  $\Rightarrow$  UserFunction. Ändern Sie den Namen von UserFunction1 in Rand.
- 2 Rufen Sie die Funktion random(low,high) ab, löschen Sie die Eingangsanschlüsse, löschen Sie die Parameter und platzieren Sie das Objekt in Rand. (Ohne Parameter lauten die Standardwerte 0 und 1.) Platzieren Sie ein Objekt A+B rechts von random(low,high). Verbinden Sie den Ausgang von random(low,high) mit dem linken oberen Eingang des Objekts **A+B**.
- 3 Fügen Sie Rand einen Dateneingangsanschluss hinzu. Verbinden Sie den Eingangsanschluss **A** mit dem linken unteren Eingangsanschluss des Objekts **A+B**.
- 4 Fügen Sie Rand einen Datenausgangsanschluss hinzu. Verbinden Sie den Ausgang des Objekts **A+B** mit dem Ausgangsanschluss von Rand.

Die UserFunction Rand sollte jetzt aussehen wie in Abbildung 208.



**Abbildung 208** Die UserFunction Rand

- Speichern Sie das Programm als **seqdat1.vee**. Schließen Sie das Fenster Rand, indem Sie in der rechten oberen Ecke des Fensters auf die Schaltfläche **x** klicken.

### HINWEIS

Durch das Schließen des Fensters wird die UserFunction nicht entfernt. Sie können dies überprüfen, indem Sie auf **Edit** ⇒ **Edit UserFunction** klicken – Rand wird in einer Liste der zu ändernden UserFunctions aufgeführt. Sie können die Funktion Rand auch minimiert anzeigen; das entsprechende Symbol erscheint am unteren Rand der VEE-Anzeige.

Richten Sie jetzt im Sequencer die drei Tests zum Aufruf von Rand mit einem Sequencer-Eingangs-Pin so ein, dass die Eingangsparameter für Rand geliefert werden.

- Wählen Sie **Device** ⇒ **Sequencer** und platzieren Sie dieses Objekt im Hauptprogramm. Fügen Sie dem Sequencer einen Eingangsanschluss hinzu. Klicken Sie auf die Transaktionsleiste, um das Dialogfeld Sequence Transaction zu öffnen. Ändern Sie das Feld **FUNCTION** von **testFunc(a)** in **rand(a)**. Dadurch wird die UserFunction Rand() aufgerufen und der Wert an den Eingangsanschluss **A** des Sequencer gesendet. Klicken Sie auf **OK**, um zur offenen Ansicht des Sequencer zurückzukehren.

**HINWEIS**

Sie können auch den Namen eines Sequencer-Eingangsanschlusses verwenden, z. B. **A**, um Daten an ein beliebiges Ausdrucksfeld im Fenster Sequence Transaction weiterzugeben. Sie können beispielsweise über **A** Daten an **RANGE:** und **SPEC NOMINAL:** weitergeben.

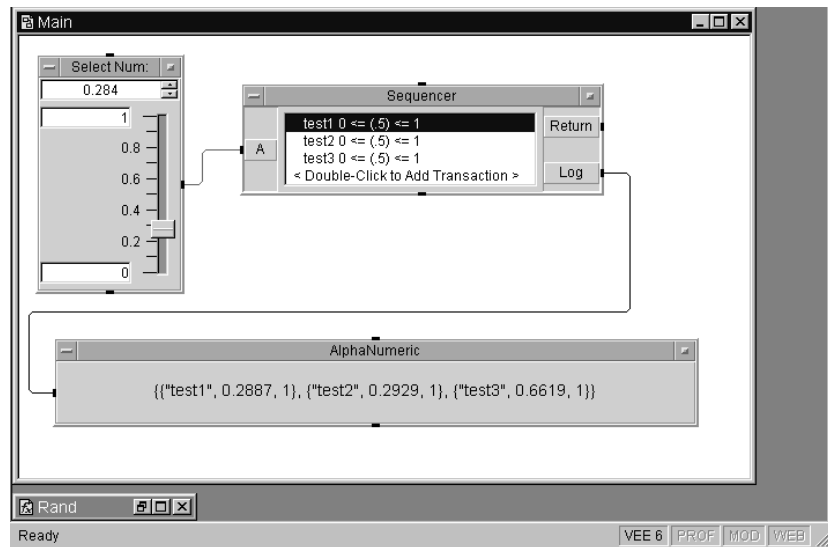
Vergewissern Sie sich, dass die Transaktion hervorgehoben ist, zeigen Sie auf die Transaktionsleiste, drücken Sie `Strg+K`, um den Test auszuschneiden, und drücken Sie dann drei Mal `Strg+Y`, um den Test wieder im Sequencer einzufügen. (Sie können zum Ausschneiden und Einfügen auch die Objektmenüs verwenden.)

Die Standardnamen der Tests lauten `test1x2`, `test1x1` und `test1`. Öffnen Sie die drei Dialogfelder Sequence Transaction, und ändern Sie die drei Namen in `test1`, `test2` und `test3` zur Verdeutlichung.

- 1 Wählen Sie **Data** ⇒ **Continuous** ⇒ **Real64 Slider** und platzieren Sie dieses Objekt links vom Sequencer. Ändern Sie den Namen auf die Eingabeaufforderung `Select Num;`, verkleinern Sie das Objekt und verbinden Sie es mit dem Eingangsanschluss des Sequencer.

Sie können die Größe eines Objekts beim Positionieren ändern, indem Sie auf das Objekt klicken und seine Ecken mit der linken Maustaste ziehen.

- 2 Wählen Sie eine AlphaNumeric-Anzeige, platzieren Sie sie unter dem Sequencer, verbreitern Sie sie und verbinden Sie sie mit dem Ausgangsanschluss `Log` am Sequencer.
- 3 Speichern Sie das Programm als `seqdat1`. Wählen Sie im Real64 Slider-Objekt eine Zahl und führen Sie `seqdat1` aus. Es sollte aussehen wie in Abbildung 209.



**Abbildung 209** Daten mit einem Eingangsanschluss weitergeben

Je größer die Anzahl der Tests wird, desto mehr Eingangs-Pins sind für die Weitergabe von Daten mit einem Eingangsanschluss erforderlich. Sie können die Anzahl der Eingangs-Pins reduzieren, indem Sie Records an Eingangsanschlüsse weitergeben und einzelne Felder in den Records für die verschiedenen Tests verwenden. Sie können auch eine separate UserFunction verwenden, um globale Variablen einzurichten, die dann von anderen UserFunctions oder einem Ausdrucksfeld im Programm aufgerufen werden können. Die nächste Übung verdeutlicht dies.

### Daten mit einer globalen Variablen weitergeben

In der Übung wird das Programm seqdat1 durch Hinzufügen einer globalen Variablen geändert, die den Parameter **a** an die UserFunction Rand weitergibt.

- 1 Löschen Sie das Real64 Slider-Objekt Select Num. Löschen Sie den Eingangsanschluss **A** am Sequencer.
- 2 Heben Sie die test1-Transaktionsleiste hervor, öffnen Sie das Objektmenü und klicken Sie auf **Insert Trans....** Wenn das Feld Sequence Transaction angezeigt wird, klicken Sie auf **TEST**,

um die Auswahl auf **EXEC** umzuschalten, und ändern Sie den Namen in `Setup`.

- 3 Sie verwenden den **EXEC**-Modus, da die User Function nur eine globale Variable einrichtet und kein Ergebnis ausgibt, das mit einer Spezifikation getestet werden muss.
- 4 Ändern Sie das Feld **FUNCTION** in `global()` und klicken Sie auf **OK**, um das Dialogfeld zu schließen.
- 5 Als nächstes erstellen Sie die **UserFunction global()**.
- 6 Wählen Sie **Device** ⇒ **UserFunction**. Ändern Sie den Namen **UserFunction1** in `global`.

Wählen Sie **Data** ⇒ **Continuous** ⇒ **Real64 Slider** und platzieren Sie dieses Objekt in der UserFunction. Ändern Sie den Namen in `Select Num:` und verkleinern Sie das Objekt in der vertikalen Richtung.

Wählen Sie **Data** ⇒ **Variable** ⇒ **Set Variable** und platzieren Sie das Objekt rechts vom Real64 Slider.

Ändern Sie den Namen der globalen Variablen von **globalA** in `a`. Verbinden Sie den Real64 Slider mit dem Objekt Set Variable.

Fügen Sie zum Anzeigen der Funktion auf dem Bildschirm eine Einblend-Fensteransicht hinzu, in der der Benutzer eine Nummer wählen kann. Fügen Sie eine **Confirm**-Schaltfläche (OK) hinzu, sodass das Fenster so lange in der Anzeige erhalten bleibt, bis der Benutzer eine Auswahl getroffen hat. (Sie können diese Aufgaben auch mit einem „Real Input“-Dialogfeld innerhalb der **UserFunction global()** ausführen.)

- 7 Wählen Sie **Flow** ⇒ **Confirm (OK)** und platzieren Sie dieses Objekt über dem Objekt Real64 Slider. Verbinden Sie den Datenausgangs-Pin OK mit dem Sequenzeingangs-Pin von Real64 Slider.

### HINWEIS

Wenn Sie die Schaltfläche OK unter dem Objekt Set Variable platzieren, bewirkt dies einen logischen Fehler. Dies liegt daran, dass VEE den alten Wert am Slider an das Objekt Set Variable sendet und wartet, bis die Schaltfläche OK angeklickt wird. Alle im Einblendfenster neu eingegebenen Werte werden ignoriert. Wenn OK über dem Real64 Slider verbunden wird, wartet VEE mit dem Setzen der globalen Variablen, bis OK gedrückt wurde und verwendet somit den neuen Slider-Wert. Sie können Show Data Flow einschalten, um die Reihenfolge der Ausführung anzuzeigen.

---

- 8 Wählen Sie **Display** ⇒ **Note Pad** und entfernen Sie die Vorlageninformationen. Platzieren Sie das Objekt rechts von der Schaltfläche OK. Geben Sie die folgende Benutzer-Eingabeaufforderung in Note Pad ein:

```
Please select a number for this run of tests
1, 2, and 3.
```

- 9 Wählen Sie das "Note Pad"-Objekt, den Real64 Slider und die Schaltfläche OK, indem Sie `Strg` gedrückt halten und auf diese Objekte klicken. Alle diese Objekte werden jetzt mit einem Schatten versehen, um anzuzeigen, dass sie markiert sind. Klicken Sie auf **Edit** ⇒ **Add To Panel**. (Denken Sie daran, dass das Menü Edit auch über einen Klick mit der rechten Maustaste auf eine leeren Stelle der VEE-Anzeige oder in der Detailansicht eines UserObject bzw. einer UserFunction verfügbar ist.) Verkleinern Sie in der Fensteransicht das Fenster, und positionieren Sie das Note Pad oben, den Real64 Slider in der Mitte und die Schaltfläche OK unten.

### HINWEIS

Wenn Sie die Objekte in der Fensteransicht neu positionieren, hat dies *keine* Auswirkung auf das Layout der Objekte in der Detailansicht.

---

Öffnen Sie das Fenster UserFunction Properties. Setzen Sie im Ordner General unter Pop-up Panel die Eigenschaft ShowPanelonExecute auf True. Abbildung 210 zeigt die UserFunction in der Detailansicht, und Abbildung 211 zeigt die Fensteransicht.

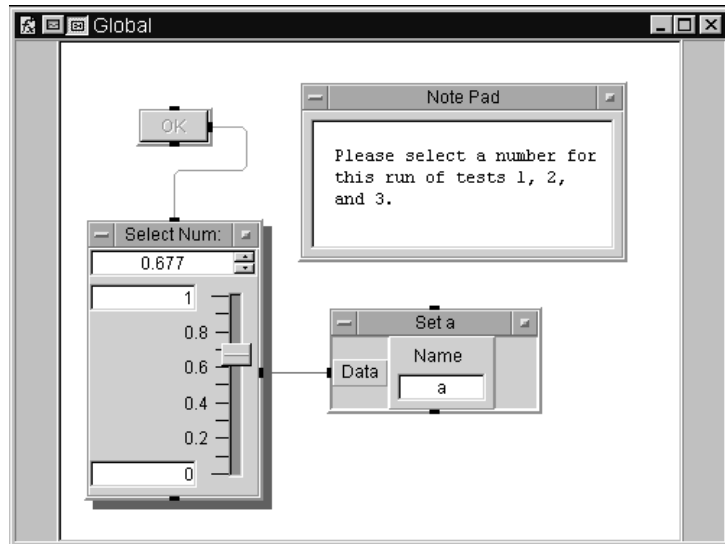


Abbildung 210 Die UserFunction „Global“ (Detailansicht)

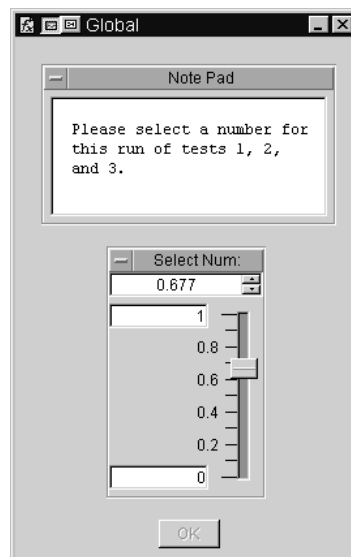
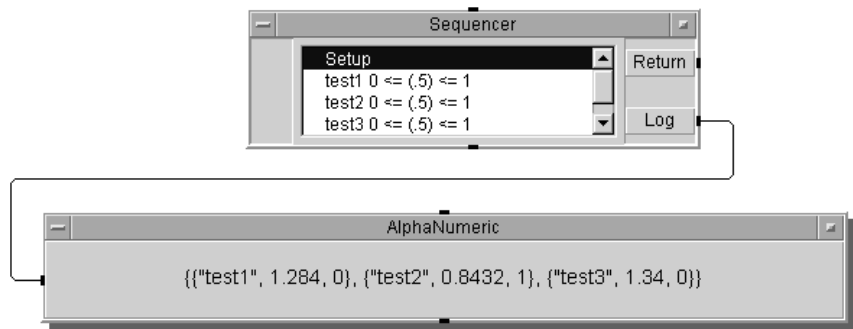


Abbildung 211 Die UserFunction „Global“ (Fensteransicht)

**10** Speichern Sie das Programm als **seqdat2** und führen Sie es aus. Wenn das Einblendmenü angezeigt wird, wählen Sie einen Wert und drücken Sie OK. Es sollte aussehen wie in Abbildung 212.

### HINWEIS

Das Einblendfenster erscheint standardmäßig in der Mitte der Anzeige. Zum Verschieben klicken und ziehen Sie die Titelleiste.



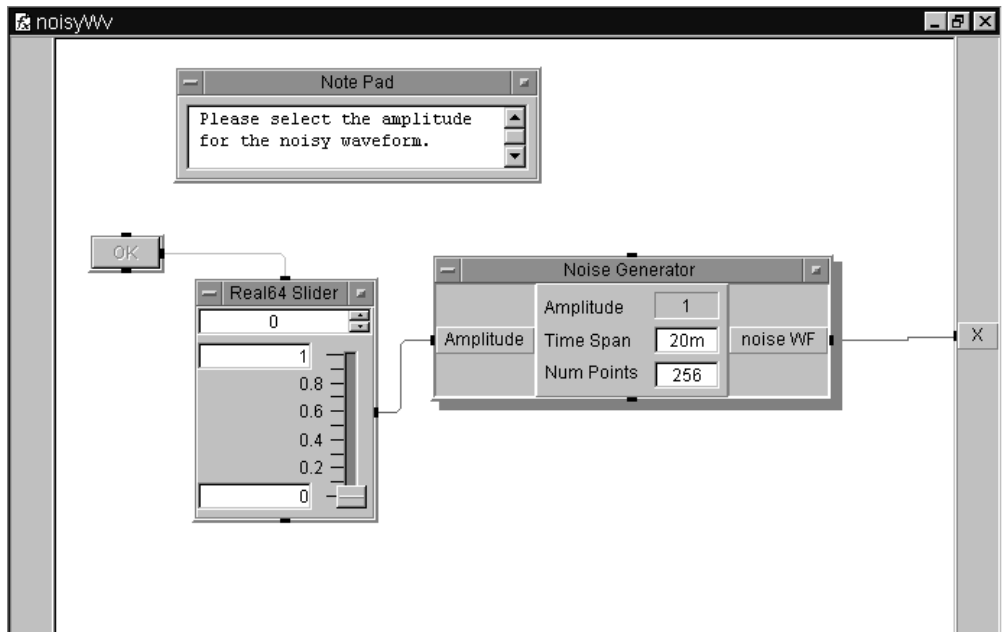
**Abbildung 212** Daten über eine globale Variable weitergeben

### Eine Wellenformausgabe mit einer Maske vergleichen

In dieser Übung erstellen Sie eine UserFunction mit dem Namen `noisyWv` und rufen sie von einer einzelnen Transaktionszeile im Sequencer aus auf. Der Benutzer kann die Amplitude der Welle von 0 bis 1 variieren. Diese Funktion simuliert ein Testergebnis, das eine gestörte Wellenform zurückgibt. Sie verwenden das Objekt `Coord` im Menü **Data** ⇒ **Constant** zum Erstellen einer geraden Linienmaske bei **0.6**, die der Sequencer zum Testen der gestörten Wellenform verwendet.

- 1 Erstellen Sie die UserFunction mit dem Namen `noisyWv` wie in Abbildung 213 in der Detailansicht gezeigt.





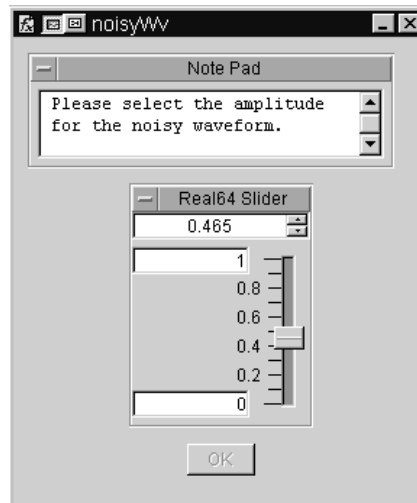
**Abbildung 213** Die UserFunction „noisyWv“ (Detailansicht)

- 2 Drücken Sie `strg` und klicken Sie auf die Schaltfläche OK, den Real64 Slider und Note Pad, um diese Objekte zum Erstellen einer Fensteransicht zu markieren. Wählen Sie **Edit** ⇒ **Add To Panel**.

Wenn die Fensteransicht angezeigt wird, ordnen Sie die Objekte wie gewünscht neu an und passen Sie die Größe des Fensters entsprechend an.

Öffnen Sie das Objektmenü, klicken Sie auf **Properties** und klicken Sie unter Pop-up Panel neben Show Panel on Execute.

Die Fensteransicht sollte aussehen wie in Abbildung 214.



**Abbildung 214** Das UserObject „noisyWv“ (Panel)

- 3 Wählen Sie **Device** ⇒ Sequencer und platzieren Sie dieses Objekt links in der Mitte des Hauptfensters. Fügen Sie einen Dateneingangsanschluss hinzu und nennen Sie ihn `mask`.
- 4 Klicken Sie auf die Transaktionsleiste, um das Dialogfeld Sequence Transaction zu öffnen. Ändern Sie die Felder wie folgt:

**Tabelle 43** Das Dialogfeld Sequence Transaction

Feld	Beschreibung
FUNCTION	Geben Sie <code>noisyWv()</code> ein.
RANGE:	Klicken Sie und wählen Sie im Menü den Eintrag <code>LIMIT</code> . Behalten Sie <code>&lt;=</code> bei, und geben Sie <code>mask</code> als Namen des Anschlusses <code>LIMIT</code> ein. Alle weitere Standardwerte sind in Ordnung; klicken Sie daher auf <b>OK</b> .

**test1** ruft ein Ergebnis von `noisyWv()` ab und vergleicht es mit dem Grenzwert am Eingangsanschluss **mask**. Wenn die gestörte Welle an allen Stellen kleiner oder gleich der Maske ist, gilt der Test als erfolgreich. Andernfalls gilt er als fehlgeschlagen.

- Wählen Sie **Data**  $\Rightarrow$  **Constant**  $\Rightarrow$  **Coord** und platzieren Sie das Objekt über dem Sequencer. Verbinden Sie seinen Ausgang mit dem Eingangsanschluss `mask` des Sequencer.

Öffnen Sie das Coord-Objektmenü, klicken Sie auf **Properties** und stellen Sie die Felder wie folgt ein:

**Tabelle 44** Eigenschaften des Objekts Coord

Eigenschaft	Beschreibung
DataShape	Wählen Sie „1D Array“.
ArraySize	Geben Sie 2 ein (Sie brauchen nur zwei Koordinatenpaare zur Angabe einer geraden Linie.)

- Im Coord- Objekt sehen Sie jetzt zwei Indizes für Koordinatenpaare. Doppelklicken Sie auf den ersten Index, **0000**: um eine Einfügemarke anzuzeigen. Geben Sie die Koordinaten durch Kommata getrennt ein; VEE fügt die Klammern automatisch hinzu. Geben Sie `0, 0.6 Tab 20m, 0.6` ein und klicken Sie anschließend außerhalb des Objekts auf den Arbeitsbereich. Die Einträge lauten wie folgt:

- Die x-Achse (Zeitachse) für den Noise Generator in `noisyWv()` reicht von 0 bis 20 Millisekunden; daher die beiden x- Werte 0 und 20m.
- Die beiden y-Werte sind jeweils **0.6**, da Sie eine Maske mit einer geraden Linie erstellen wollen.

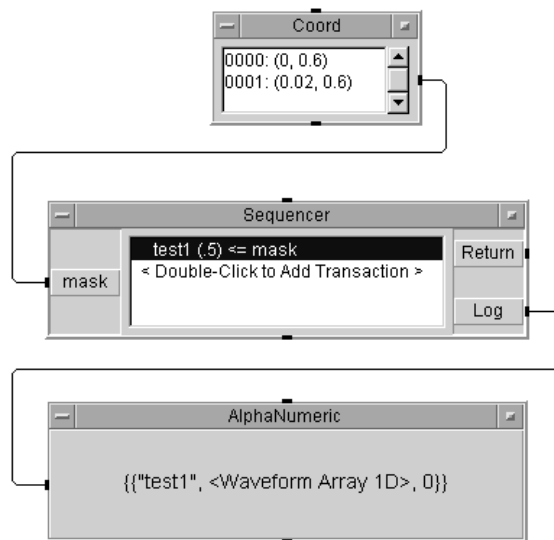
**HINWEIS**

Sie können eine beliebige Masken-Wellenform erstellen, indem Sie die entsprechende Anzahl von Koordinatenpaaren konfigurieren und ausfüllen.

## 10 Tests sequenzieren

Der Vergleichsmechanismus des Sequencer arbeitet genau wie das Objekt Comparator, das den Datentyp Coord zum Testen von Wellenformen akzeptiert. Sie sollten natürlich die beiden Wellenform-Datentypen vergleichen. Drücken Sie Tab, um zwischen den Koordinatenpaaren zu wechseln. Wenn Sie fertig sind, klicken Sie auf den Arbeitsbereich.

- 7 Wählen Sie eine AlphaNumeric-Anzeige, verbreitern Sie sie, und verbinden Sie sie mit dem Ausgang Log des Sequencer.
- 8 Speichern Sie das Programm als **seqdat3** und führen Sie es aus. Es sollte aussehen wie in Abbildung 215.

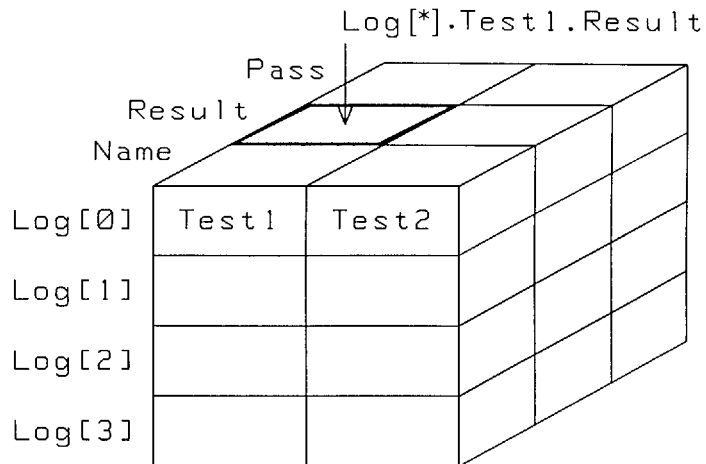


**Abbildung 215** Eine Wellenform mit einer Maske vergleichen

Die Übungen zum Weitergeben von Daten mit dem Sequencer sind damit abgeschlossen. In der nächsten Übung lernen Sie, wie Sie Daten aus verschiedenen Iterationen des Sequencer aufrufen und analysieren.

## Daten aus dem Sequencer analysieren

Wie bereits erwähnt, werden die Sequencer-Daten als Record aus Records ausgegeben. In vielen Fällen jedoch kann der Sequencer eine Testserie mehrmals durchlaufen. Dadurch wird ein Array von Records generiert. Jeder Record stellt einen Durchlauf des Sequencer dar und enthält weitere Records für die einzelnen Tests in einem Durchlauf. Sie können sich dies vorstellen wie einen Quader aus einzelnen kleineren Quadern im Speicher (siehe Abbildung 216).



**Abbildung 216** Ein protokolliertes Array aus Records von Records

Das Array von Records wird als Log (Protokoll) bezeichnet, da dies der Name ist, der dem Ausgangs-Pin des Sequencer zugeordnet wurde. Für den Zugriff auf die Daten eines bestimmten Durchlaufs verwenden Sie die Array-Indizierung mit der Klammernotation.

- **Log[0]** ist der erste Durchlauf am **Sequencer**, **Log[1]** ist der zweite Durchlauf, usw.
- Der Haupt-Record für jeden Durchlauf hat zwei Felder, **Test1** und **Test2**.
- Innerhalb des Records **Test1** gibt es drei Felder: **Name**, **Result** und **Pass**. Dasselbe gilt für den Record **Test2**.

- **Log.Test1.Result** liefert somit ein Array aus vier Werten; jeder dieser Werte steht für einen der vier Durchläufe. **Log[0].Test1.Result** gibt einen Skalarwert aus, **Result** von **Test1** im ersten Durchlauf (**Log[0]**).

Das protokollierte Array aus Records vereinfacht das Analysieren und Untersuchen der Daten. Sie könnten sich beispielsweise dafür interessieren, wie viele Tests in einem bestimmten Durchlauf erfolgreich waren. Oder Sie wollen das Durchschnittsergebnis von Test2 über alle Durchläufe wissen. Oder Sie wollen alle Daten von Test1 im vierten Durchlauf anzeigen. Alle diese Abfragen sind mit dieser Datenstruktur möglich. In der nächsten Übung werden einige Analyseoperationen mit Daten durchgeführt.

### Übung 10-3: Mehrere Durchläufe von Daten aus dem Sequencer analysieren

- 1 Löschen Sie den Inhalt der Anzeige und öffnen Sie das Programm **seqdat1.vee**.

Ändern Sie das Programm **seqdat1.vee**, sodass es den Sequencer drei Mal durchläuft. Führen Sie anschließend die Analyseoperationen mit den Daten durch.

- 2 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Count** und platzieren Sie dieses Objekt über dem Objekt Real64 Slider. Ändern Sie die Anzahl der Iterationen in 3 und verbinden Sie den Datenausgangs-Pin mit dem Sequenzeingangs-Pin des Sequencer.
- 3 Löschen Sie die Datenlinie zwischen dem Sequencer Log-Pin und der Anzeige. Wählen Sie **Data** ⇒ **Collector** und platzieren Sie dieses Objekt rechts vom Sequencer. Verbinden Sie seinen linken oberen Dateneingangs-Pin mit dem Sequencer Log-Pin und seinen Pin XEQ (links unten) mit dem Sequenzausgangs-Pin am Objekt For Count. Verbinden Sie den Datenausgangs-Pin des Collector mit dem Dateneingangs-Pin der AlphaNumeric-Anzeige. Vergrößern Sie die Anzeige vertikal, damit genügend Platz für das Array mit drei Elementen zur Verfügung steht.

Der Sequencer durchläuft jetzt test1 und test2 drei Mal und erfasst die Daten in einem Array mit drei Elementen, von denen jedes einen Record aus Records für jeden Durchlauf enthält. (Sehen Sie sich hierzu die Quaderdarstellung in Abbildung 216 an.)

Führen Sie das Programm an dieser Stelle aus, um die Anzeige der Sequencer-Daten zu sehen.

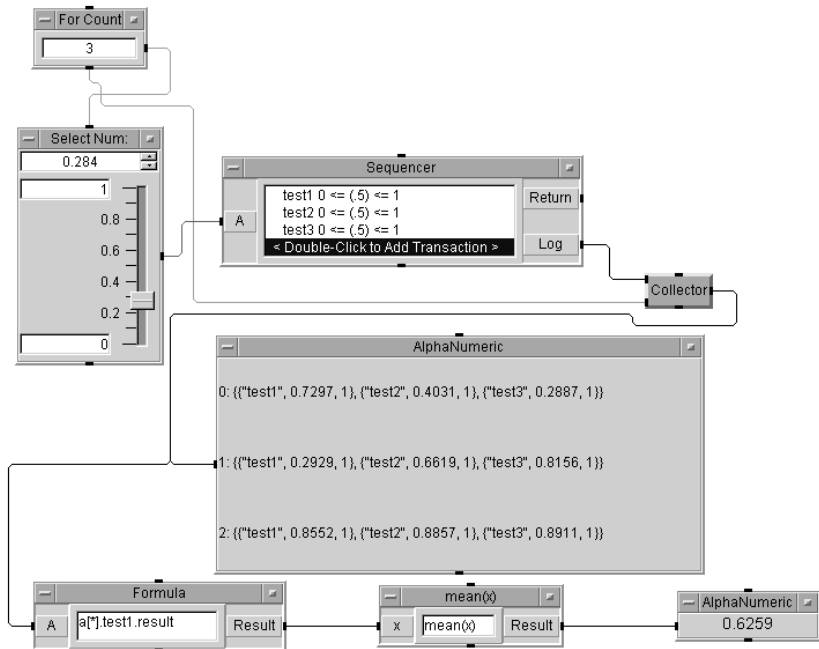
Verwenden Sie jetzt das Formula-Objekt zum Extrahieren eines Teils der zu analysierenden Daten. Diese Übung verwendet die Ergebnisse von test1 für alle drei Durchläufe als Beispiel und findet den Mittelwert aus diesem Array.

- 4 Wählen Sie **Device**  $\Rightarrow$  Formula und platzieren Sie das Objekt unter der Anzeige. Verbinden Sie den Eingangs-Pin von Formula mit dem Ausgang des Collector. Ändern Sie das Eingabefeld Formula wie folgt: `a[*].test1.result`. Verbinden Sie ein Objekt `mean(x)` mit Formula und eine AlphaNumeric-Anzeige mit `mean(x)`.

Das **a** bezieht sich auf das Array am Eingangsanschluss **A**. **Test1.result** ruft das richtige Feld auf. Alle Durchläufe werden in einem Array angezeigt. (**A[0].test1.result** bezieht sich beispielsweise nur auf den ersten Durchlauf.)

- 5 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 217.

## 10 Tests sequenzieren



**Abbildung 217** Mehrerer Durchläufe von **Sequencer**-Daten analysieren

Diese Übung greift nur auf ein einzelnes Array zu; das Prinzip ist jedoch beim Extrahieren weiterer Daten-Arrays aus dem Sequencer-Ausgang das gleiche. Sie können leicht ändern, welche Felder gespeichert werden, indem Sie den Logging folder (Protokoll-Ordner) im Dialogfeld Sequencer Properties öffnen.



## Protokollierte Daten speichern und abrufen

Diese Übung zeigt die Verwendung der Objekte To/From File und To/From DataSet.

### Übung 10-4: Die Objekte „To/From File“ mit protokollierten Daten verwenden

- 1 Öffnen Sie die Datei **seqdat2** und löschen Sie die Datenlinie zu der Anzeige.
- 2 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Count** und platzieren Sie dieses Objekt links vom Sequencer. Ändern Sie die Anzahl in For Count in 3 und verbinden Sie seinen Datenausgangs-Pin mit dem Sequenzeingangs-Pin des Sequencer.
- 3 Vergrößern Sie den Arbeitsbereich vertikal und platzieren Sie die AlphaNumeric-Anzeige im unteren Bereich. Wählen Sie **Data** ⇒ **Collector** und platzieren Sie dieses Objekt im linken Arbeitsbereich. Verbinden Sie den Sequencer Log-Pin mit dem Dateneingangs-Pin des Collector. Verbinden Sie den Sequenzausgangs-Pin des Objekts For Count mit dem Collector XEQ-Pin.

Der Collector erstellt ein Array aus Records von Record aus dem Sequencer. Durch Verwenden der Transaktion WRITE CONTAINER im Objekt To File können Sie sehr leicht einen beliebigen VEE-Daten-Container in eine Datei schreiben.

- 4 Wählen Sie **I/O** ⇒ **To** ⇒ **File** und platzieren Sie dieses Objekt rechts vom Collector. Wählen Sie **I/O** ⇒ **From** ⇒ **File** und platzieren Sie dieses Objekt unter dem Objekt To File. Fügen Sie dem Objekt To File einen Eingangsanschluss hinzu und verbinden Sie den Collector-Ausgang damit. Verbinden Sie den Sequenzausgangs-Pin von To File mit dem Sequenzeingangs-Pin von From File. Verbinden Sie den Datenausgang von From File mit der Anzeige.

Markieren Sie **Clear File At PreRun & Open in To File** und konfigurieren Sie eine Transaktion WRITE CONTAINER a. Konfigurieren Sie eine Transaktion im Objekt From File wie die folgende: READ CONTAINER x.

Sie können die Standard-Datendatei zum Speichern verwenden.

- 5 Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 218.

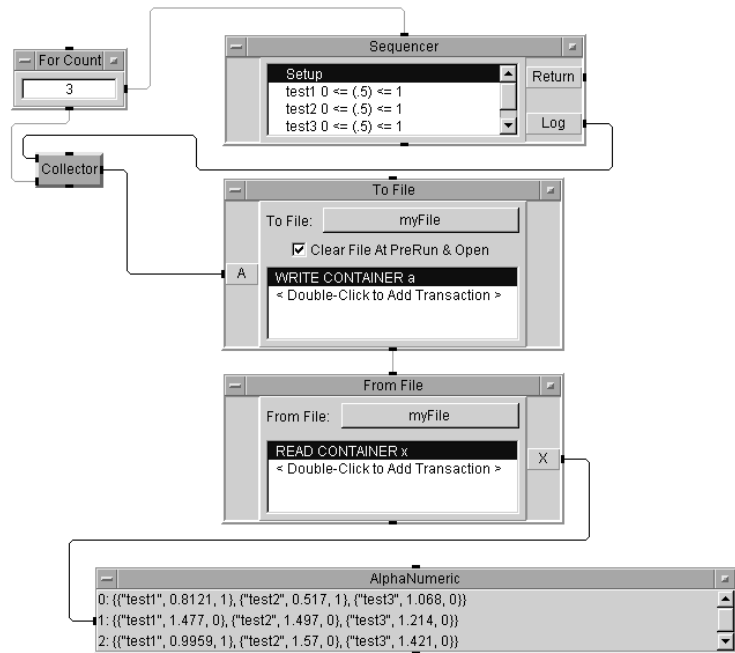


Abbildung 218 Protokolierte Daten mit „To/From File“ speichern

## Die Objekte „To/From DataSet“ mit protokollierten Daten verwenden

Da Sie Testdaten als Records speichern, verwenden Sie eventuell lieber die Objekte To/From DataSet. In diesem Fall brauchen Sie keinen Collector, da Sie jeden Durchlauf des Sequencer an das Ende des DataSet anhängen.

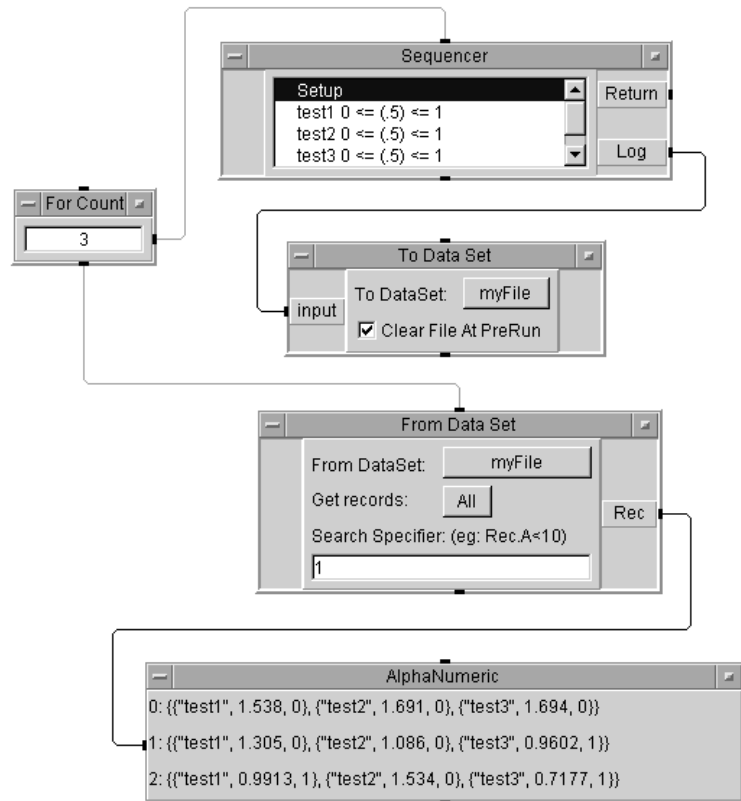
Ändern Sie das letzte Programm, sodass es aussieht wie in Abbildung 219. Die Objekte To/From DataSet befinden sich im Menü I/O. Beachten Sie die Sequenzlinie, die zum From DataSet führt. Die Sequenzlinie wird verwendet, weil vor dem Auslösen

des From DataSet gewartet werden soll, bis alle drei Durchläufe an das DataSet angehängt wurden.

Ein Grund zur Verwendung der Objekte To/From DataSet zum Erfassen von Daten statt der Objekte To/From File liegt darin, dass Sie die Daten mit der Funktion Search Specifier im Objekt From DataSet in nützliche Informationen umwandeln können.

**HINWEIS**

Denken Sie daran, das Feld Get records in From DataSet von **One to All** zu ändern.



**Abbildung 219** Speichern von protokollierten Daten mit „To/From DataSet“

### Kapitel-Checkliste

Ermitteln Sie anhand der folgenden Checkliste, ob Sie einzelne Themen noch einmal ansehen sollten, bevor Sie mit dem nächsten Kapitel fortfahren.

- Das Konzept des Sequencer-Objekts beschreiben.
- Einen Test für den Sequencer konfigurieren.
- Operationen für einen Sequencer-Test hinzufügen, einfügen und löschen.
- Die vom Sequencer protokollierten Daten aufrufen.
- Sequencer-Eingangsanschlüsse zur Datenweitergabe an Tests verwenden.
- Globale Variablen zur Datenweitergabe an Tests verwenden.
- Eine Wellenformausgabe mit einer Maske vergleichen.
- Mehrere Durchläufe von Daten aus dem Sequencer analysieren.
- Daten mit den Objekten To/From File speichern.
- Daten mit den Objekten To/From DataSet speichern.

# 11

## Benutzerschnittstellen verwenden

Überblick 419

Wichtige Überlegungen zum Thema Benutzerschnittstellen 420

Benutzerschnittstellenobjekte verwenden 424

Allgemeine Aufgaben beim Erstellen von Benutzerschnittstellen 439

# Benutzerschnittstellen verwenden

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Benutzerschnittstellen erstellen
- Menüs für einen Benutzer verwenden
- Bitmaps für eine bessere Verständlichkeit importieren
- Testprogramme sichern
- Funktionen von Benutzerschnittstellen
- ActiveX-Steuer-elementen zur Erweiterung des Funktionsumfangs von VEE verwenden

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*

## Überblick

In diesem Kapitel lernen Sie, wie Sie Benutzerschnittstellen erstellen, indem Sie beispielsweise Menüs verwenden, die Programmoberfläche anpassen, Warnsignale hinzufügen und Bitmaps importieren. Dieses Kapitel erweitert die Übungen in den vorangegangenen Kapiteln, in denen Sie Benutzerschnittstellen und Einblendfenster erstellt hatten.

Einige Vorteile bei der Verwendung von Funktionen der VEE-Benutzerschnittstelle sind:

- Optimale Benutzerfreundlichkeit
- Verbesserte Programmleistung
- Sicherheit gegen unberechtigte Änderungen
- Gute Übersichtlichkeit durch visuelle Hilfsmittel

# Wichtige Überlegungen zum Thema Benutzerschnittstellen

Dieser Abschnitt bietet einen Überblick über die Erstellung von Benutzerschnittstellen in VEE.

## Eine Benutzerschnittstelle erstellen

VEE umfasst ein breites Spektrum von Steuerelementen, Dialogfeldern, Anzeiger- und Anzeigeelementen, die Sie bei der Erstellung von Benutzerschnittstellen verwenden können. Steuerelemente sind beispielsweise Schaltflächen, Umschalter, Kontrollkästchen, Dropdown-Listenfelder und Listenfelder. Anzeigerelemente sind beispielsweise „Tankanzeigen“, Thermometer, Füllmarkierungen, VU-Meter und farbige Alarmanzeigen.

Zusätzlich zu den Benutzerschnittstellen-Elementen in VEE können Sie auch Elemente aus anderen Quellen einbinden. Es stehen Tausende von Benutzerschnittstellen-Elementen zur Verfügung, die Sie aus dem World Wide Web laden können. Zudem erhalten Sie über ActiveX-Steuerelemente Zugriff auf weitere Benutzerschnittstellen-Elemente. (Manche dieser Elemente stehen kostenlos zur Verfügung, für andere müssen Sie eine Gebühr entrichten.)

Ob Sie die Objekte für Benutzerschnittstellen aus VEE oder andere Elemente verwenden, der Prozess zum Erstellen einer Benutzerschnittstelle ist immer der gleiche.

Wenn Sie Benutzerschnittstelle für ein VEE-Programm erstellen, dann erstellen Sie eine Fensteransicht des jeweiligen Programms.

- 1 Wählen Sie in der Fensteransicht das bzw. die gewünschten Objekte, indem Sie die Taste `Strg` gedrückt halten und auf die einzelnen Objekt klicken, um sie zu markieren.
- 2 Wählen Sie **Edit** ⇒ **Add To Panel**. Die Anzeige wechselt in die Fensteransicht (normalerweise blau angezeigt); diese Ansicht umfasst die Objekte, die Sie in der Detailansicht markiert haben.



Sie haben jetzt eine Ansicht des VEE-Programms, die Sie so anpassen können, dass Sie nur die Komponenten anzeigt, die der Benutzer sehen muss.

## Wechseln zwischen der Fensteransicht und der Detailansicht

Zum Wechseln zwischen der Fensteransicht und der Detailansicht eines VEE-Programms klicken Sie auf das Fenster oder auf das Detailsymbol in der *Titelleiste* des Fensters, wie in Abbildung 220 gezeigt.

### HINWEIS

Sie müssen eine Fensteransicht des Programms erstellen, damit die Schaltfläche zum Wechseln in die Fensteransicht in einer Fenster-Titelleiste angezeigt wird.

Normalerweise entwickeln Sie das Programm in der Detailansicht und erstellen anschließend eine Fensteransicht für die Benutzerschnittstelle. Die Fensteransicht für die Schaltfläche kann in der Titelleiste eines UserObject-Fensters, eines User-Function-Fenster oder des Hauptfensters angezeigt werden.



Die Schaltfläche  
für die Fensteransicht

Die Schaltfläche  
für die Detailansicht

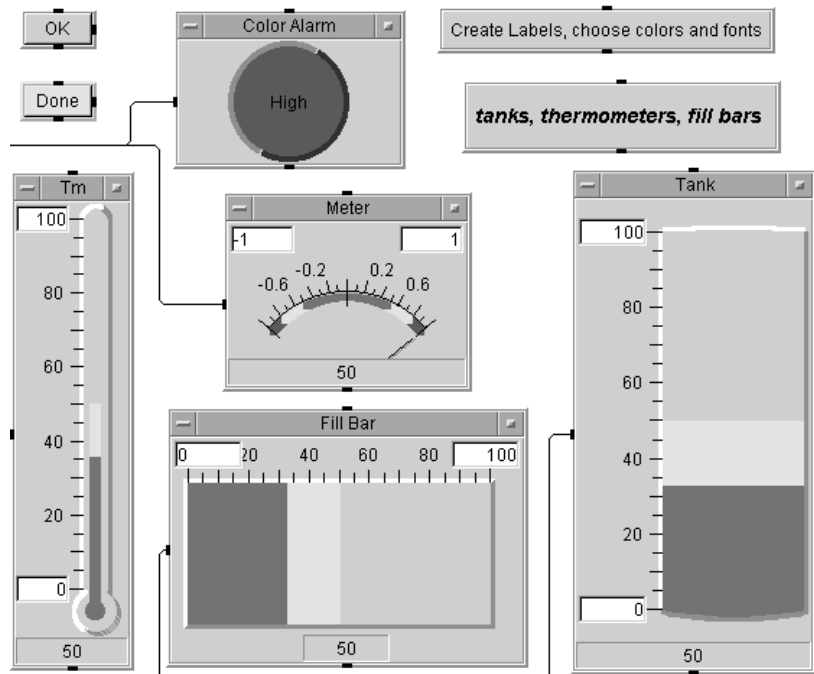
**Abbildung 220** Schaltflächen für die Fenster- und die Detailansicht in der Titelleiste

### Eine Benutzerschnittstelle anpassen

In der Fensteransicht eines VEE-Programms können Sie die Größe von Objekten ändern, Objekte neu anordnen und die Art der Anzeige von Objekten ändern, ohne *dass sich dies auf die Detailansicht dieser Objekte auswirkt*. Sie können beispielsweise die Titelleiste und die Skalen aus der Fensteransicht einer „Waveform (Time)“-Anzeige entfernen, ohne dass sich die Detailansicht dieser „Waveform (Time)“-Anzeige dadurch ändert. Wenn Sie jedoch ein Objekt in der Detailansicht löschen, wird es auch aus der Fensteransicht entfernt.

In der Fensteransicht können Sie verschiedene Farben und Schriftarten wählen, um bestimmte Komponenten von anderen abzuheben. Außerdem können Sie zur Verdeutlichung skalierbare Bitmap-Dateien einfügen. Sie können die Fensteransicht auch für den Benutzer dokumentieren, indem Sie die Titelleisten über die Objekte Note Pad und Label bearbeiten, und Sie können in den Objektmenüs den Befehl Description verwenden.

Abbildung 221 zeigt einige der in VEE verfügbaren Anzeigerelemente.



**Abbildung 221** Eine Auswahl der VEE-Anzeigerelementen

# Benutzerschnittstellenobjekte verwenden

In diesem Abschnitt werden die Benutzerschnittstellenobjekte und die in VEE verfügbaren Optionen vorgestellt. Sie können diesen Abschnitt durchblättern, um sich einen Überblick darüber zu verschaffen, welche Elemente Sie beim Erstellen von Benutzerschnittstellen für Programme verwenden und wie Sie diese Elemente anpassen können. Führen Sie anschließend die Übungen durch, um zu sehen, wie Benutzerschnittstellen für einige typische Aufgaben erstellt werden.

## Farben, Schriftarten und Anzeigerelemente

### Farben und Schriftarten

Sie können die Farben und Schriftarten über geeignete Auswahlen in **File** ⇒ **Default Preferences** oder über das im jeweiligen Objektmenü erreichbare Properties-Fenster konfigurieren. Die Auswahl der Farben und Schriftarten hängt von dem Betriebssystem und den installierten Schriftarten ab.

### Farbige Alarmmeldungen

Wählen Sie farbige Alarmobjekte über **Display** ⇒ **Indicator**. Farbige Alarmobjekte können in der Form von Quadraten oder Kreisen dargestellt werden. Darüber hinaus kann jedes Alarmobjekt für drei verschiedene Alarmstufen farblich gekennzeichnet sein, wobei jeder Stufe zusätzlich eine eigene Textmeldung zugewiesen sein kann. Alarme werden häufig zum Simulieren einer „LED“ verwendet oder, um Benutzer über eine Situation zu informieren, die einen Eingriff erfordert.

### Tankanzeigen, Thermometer, Füllanzeigen, Messelemente

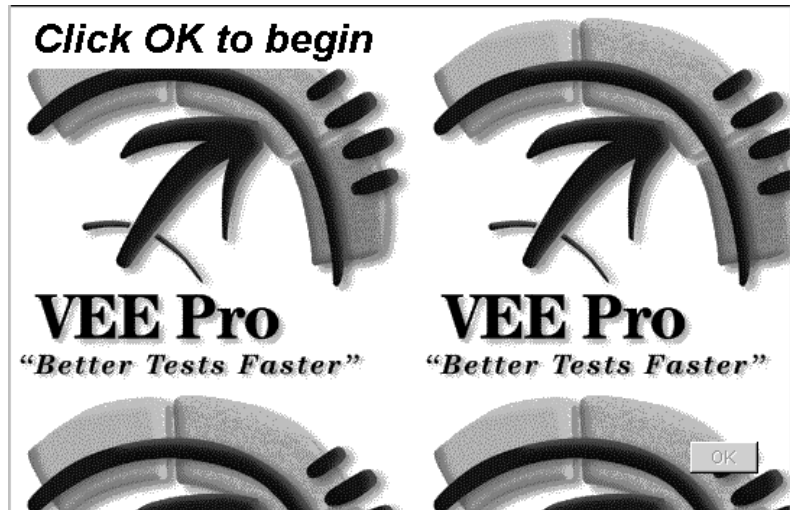
Diese Objekte werden im Untermenü **Display** ⇒ **Indicator** gewählt. Ihre Farben und Beschriftungen können angepasst werden. Diese Anzeigerelemente können auf eine horizontale oder eine vertikale Darstellung eingestellt werden, und sind für die Anzeige dreier Abstufungen ausgelegt, die Sie über den Befehl Properties im jeweiligen Objektmenü konfigurieren können.

## Grafikbilder

Klicken Sie in der Fensteransicht auf eine leere Stelle und setzen Sie im Properties-Fenster die Eigenschaft PanelBackgroundPicture, wenn Sie ein Bitmap in die Fensteransicht importieren möchten. VEE kann Dateien in den Formaten \*.jpeg, \*.png, \*.wmf, \*.GIF und \*.bmp als Hintergrundbild für das Hauptfenster oder die Fenster UserObject und User-Function importieren.

Andere VEE-Objekte werden durch ein Hintergrund-Bitmap nicht verdeckt. (Weitere Informationen hierzu finden Sie im Abschnitt „Übung 11-2: Bitmaps für den Fensterhintergrund importieren“ auf Seite 445.) Bilder können skaliert, nebeneinander angeordnet, zurechtgeschnitten oder zentriert werden.

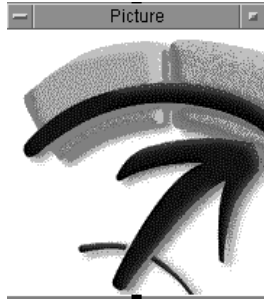
Abbildung 222 zeigt ein Hintergrundbild, bei dem das Optionsfeld Tiled (Nebeneinander anordnen) gewählt ist.



**Abbildung 222** Hintergrundbild, nebeneinander angeordnet

## 11 Benutzerschnittstellen verwenden

Über das Menü **Display** können Sie außerdem das Objekt **Picture** wählen, mit dessen Hilfe Sie eine Bitmap-Datei in einem Programm platzieren können. Abbildung 223 zeigt ein Bild, das mit **Display** ⇒ **Picture** eingebunden und anschließend in VEE zurechtgeschnitten wurde.



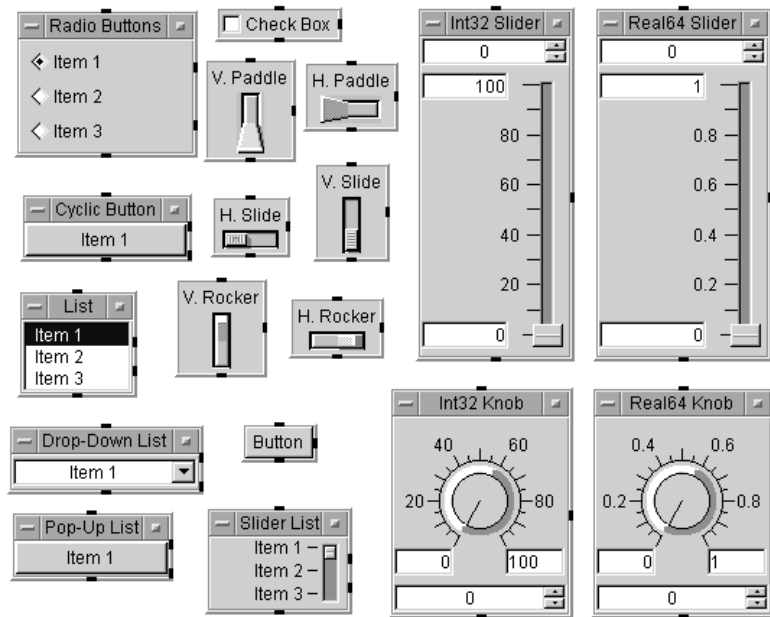
**Abbildung 223** Ein in VEE zurechtgeschnittenes Bild

### HINWEIS

Über die Registerkarte **Properties** ⇒ **Icon** können Sie auch Bitmap-Bilder für beliebige Symbole ändern.

## Ein Steuerelement für die Benutzereingabe anzeigen

Es gibt verschiedene Möglichkeiten, ein Programm so einzurichten, dass es über Benutzereingaben gesteuert werden kann. Sie können die Benutzereingabe über Einblendfenster, beliebige Datenkonstanten, Regler („Slider“) oder Knöpfe („Knobs“) abfragen. Zur Auswahl eines Steuerelements sehen Sie in den Menüs wie **Data** ⇒ **Selection Control**, **Data** ⇒ **Toggle Control** und **Data** ⇒ **Continuous** nach. Abbildung 224 zeigt eine Sammlung der Objekte, die Sie zur übersichtlicheren Darstellung einer Programmoberfläche für den Benutzer anzeigen können.



**Abbildung 224** Steuerelemente aus verschiedenen Data-Unteremenüs

Sie können Aussehen und Verhalten aller in Abbildung 224 dargestellten Objekte verändern. Sehen Sie sich beispielsweise das Properties-Dialogfeld für das Objekt Real64 Knob in Abbildung 225 an. Wählen Sie zum Konfigurieren des Objekts eine Eigenschaft wie beispielsweise ForeColor und treffen Sie dann Ihre Auswahl.

## HINWEIS

Dank ActiveX können Sie auch Steuerelemente und Anzeigen aus anderen Anwendungen verwenden, wie im Beispiel in „Übung 11-4: Ein ActiveX-Steuerelement verwenden“ auf Seite 453 gezeigt.

## 11 Benutzerschnittstellen verwenden

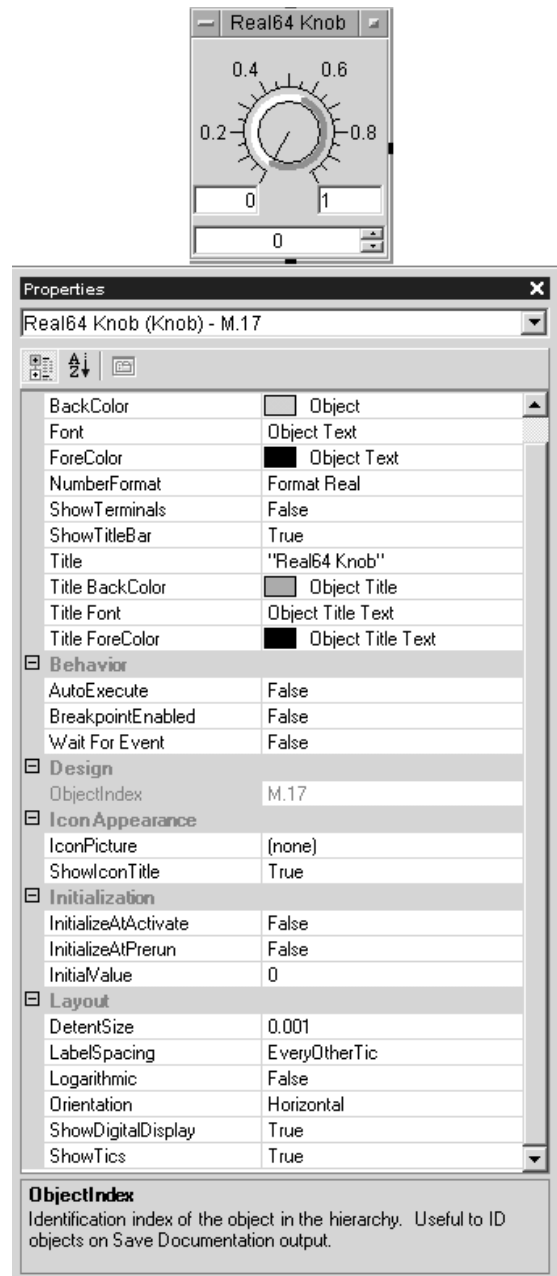


Abbildung 225 Das Dialogfeld Properties



## Eins Dialogfeld für die Benutzereingabe öffnen

VEE stellt vordefinierte Dialogfelder mit automatischer Fehlerprüfung der Eingabe, Eingabeaufforderungen und Fehlermeldungen bereit. Der Zugriff auf dieser Dialogfelder erfolgt über **Data** ⇒ **Dialog Box**.

Ein Programm kann beispielsweise bei seiner Ausführung vom Benutzer die Eingabe einer realen Zahl anfordern. Sie können in das Programm beispielsweise ein „**Real64 Input**“-Objekt einbinden, das bei der Programmausführung automatisch ein „**Real64 Input**“-Dialogfeld für die Benutzereingabe öffnet. Das Dialogfeld **Real64 Input** zeigt außerdem automatisch eine Fehlermeldung an, wenn der Benutzer fehlerhafte Informationen in das Eingabefeld eingibt. Abbildung 226 zeigt das in das Programm einzubindende Objekt und das bei der Ausführung des Programms geöffnete Dialogfeld **Real64 Input**.

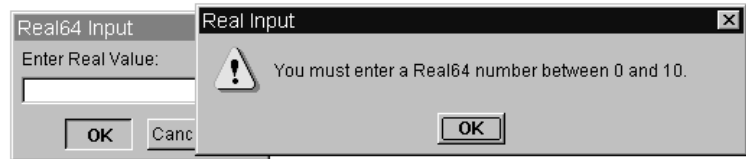
Objekt in das Programm einbinden und entsprechend anschließen

Bei der Programmausführung wird das Dialogfeld für die Benutzereingabe geöffnet

**Abbildung 226** Ein Texteingabefenster

Abbildung 227 zeigt die konfigurierbare Fehlermeldung, die bei der Programmausführung ausgegeben wird, wenn der Benutzer auf **OK** klickt, ohne die richtigen Informationen im Dialogfeld **Real64 Input** eingegeben zu haben.

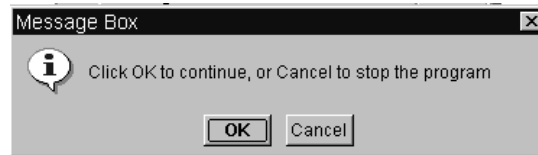
## 11 Benutzerschnittstellen verwenden



**Abbildung 227** Ein Beispiel für die automatische Fehlerprüfung

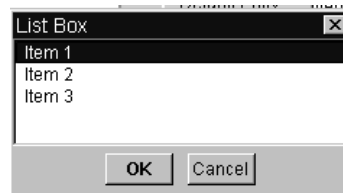
Die Dialogfelder für Int32 und Text, die Sie auch unter **Data** ⇒ **Dialog Box** finden, ähneln dem „Real64 Input“-Dialogfeld. Außerdem enthält das Menü **Data** ⇒ **Dialog Box** Einträge für die Auswahl von Meldungsfenstern, Listefeldern und DialogfelderN zur Dateiauswahl.

Abbildung 228 zeigt ein Dialogfeld, das zur Anzeige einer Meldung geöffnet wird.



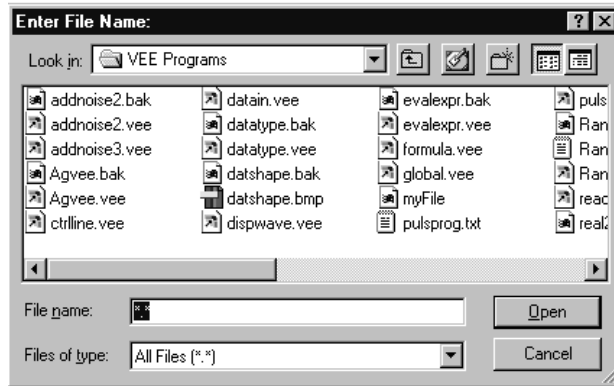
**Abbildung 228** Ein Meldungsdialogfeld

Abbildung 229 zeigt ein Dialogfeld, das geöffnet wird, wenn der Benutzer aus einer Liste wählen soll.



**Abbildung 229** Das Dialogfeld zur Listenauswahl

Abbildung 230 zeigt ein Dialogfeld, das geöffnet wird, wenn der Benutzer einen Dateinamen wählen soll.



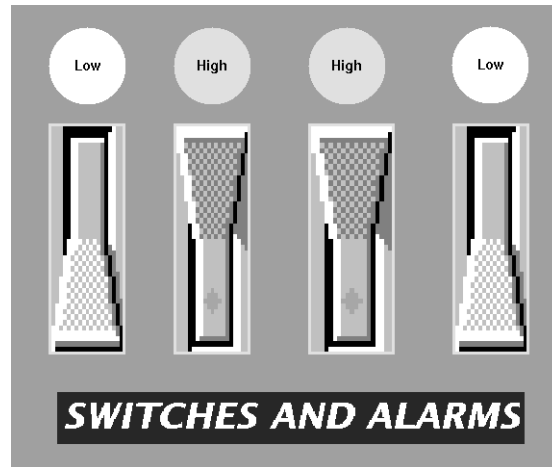
**Abbildung 230** Ein Dialogfeld zur Dateiauswahl

## Ein Schalterelement für den Benutzer anzeigen

VEE hält eingebaute Steuerelemente bereit, die zum Senden der Werte 0 oder 1 verwendet werden können. Zur Verwendung eines Schalterelements legen Sie seinen Anfangsstatus fest und führen ein Unterprogramm aus, wenn das Schalterelement betätigt wird. Sie können auch eigene Bitmaps für Schalterelemente verwenden.

Wenn Sie beispielsweise ein Programm entwickeln, bei dem der Benutzer Schalter oder Alarme einstellen muss, können Sie Schalter-Steuerelemente verwenden. Abbildung 231 zeigt ein Fenster, in dem der Benutzer Schalter einstellen kann.

## 11 Benutzerschnittstellen verwenden



**Abbildung 231** Kombinierte Schalter und Alarmer

### Objekte in der Benutzerschnittstelle ausrichten

Gemäß Voreinstellung wird das Ausrichtungsgitter in der Fensteransicht angezeigt. Sie können das Gitter ausblenden, indem Sie die Eigenschaft `PanelDrawGrid` der Fensteransicht auf `False` setzen. Darüber hinaus gibt es die Funktion „Snap-to-Grid“ (An Gitter einrasten), die das Ausrichten von Objekten erleichtert. Sie können das Gitter einblenden und die Eigenschaft `PanelGridSize` von 10 bis 1 ändern (10 ist der Standardwert), um eine exakte Ausrichtung zu ermöglichen (siehe Abbildung 232). Mit dieser Funktion geben Sie Ihrem Programm einen professionellen Anstrich. Sie finden die Funktion „Snap-to-Grid“ im Ordner `Panel` in der Auswahl `Properties` des Menüs `UserObject` bzw. `UserFunction`. (Denken Sie daran, dass Sie eine Fensteransicht für die Auswahl des Ordners `Panel` erstellt haben müssen, damit das Dialogfeld angezeigt werden kann.)

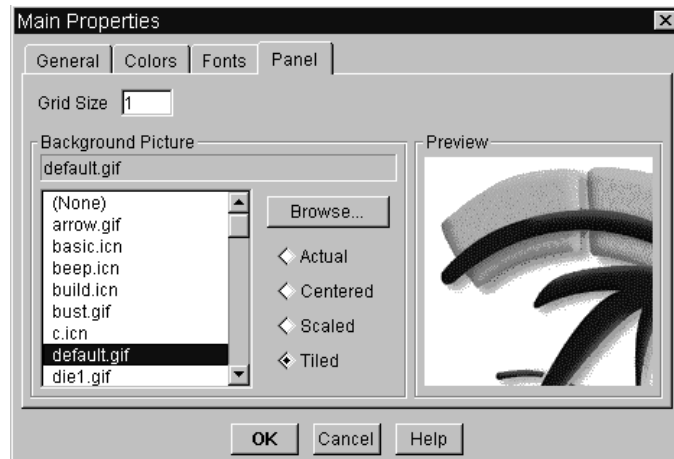


Abbildung 232 Fenstereigenschaften konfigurieren

## Andere Funktionen für die Fenstergestaltung

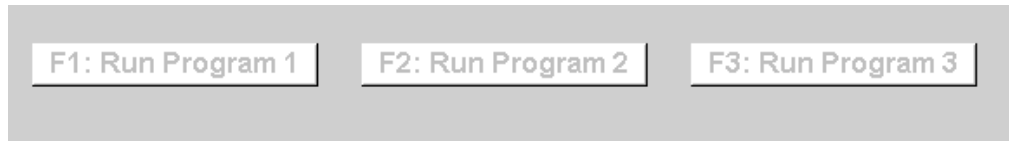
Zusätzliche Funktionen unterstützen Sie dabei, Ihr Fensters professioneller zu gestalten: Tabulatorfolge und Vordergrund/Hintergrund. Jedem Element in Ihrem Fenster ist eine Nummer zugewiesen, die bestimmt, in welcher Reihenfolge die Fensterelemente den Eingabefokus erhalten, wenn Sie die Tabulator-taste drücken. Sie können diese Reihenfolge beliebig ändern, um dem Benutzer die Interaktion mit den Fensterelementen zu erleichtern. Die Funktion Vordergrund/Hintergrund bestimmt, welches Element am unteren Ende des Stapels der Fensterelemente und welches am oberen Ende, also im Vordergrund, platziert ist. Wenn Sie diese Funktion auf Elemente anwenden, die übereinander platziert sind, können Sie bestimmen, welches Element die anderen überlagert bzw. verdeckt.

## Einer Benutzerschnittstelle für die Bedienung mit der Tastatur erstellen

Sie können mit VEE auch Schnittstellen erstellen, die der Benutzer ausschließlich über die Tastatur steuern kann. Diese Schnittstellen erfordern keine Maus.

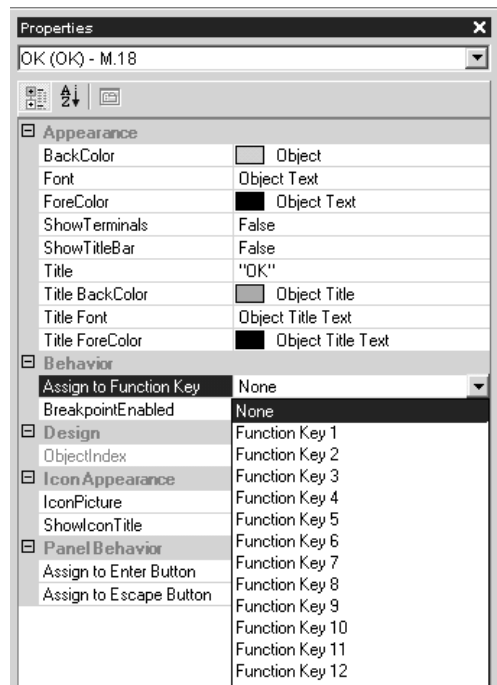
## 11 Benutzerschnittstellen verwenden

Beispielsweise können Sie das Objekt OK als „Softkey“ konfigurieren. Normalerweise konfigurieren Sie dieses Objekt so, dass es mit einer der Funktionstasten verbunden ist. Der Benutzer kann dann zur Steuerung des Programms die entsprechenden Funktionstasten drücken, wie in Abbildung 233 gezeigt.



**Abbildung 233** Ein Softkey zur Ausführung einer UserFunction

Abbildung 234 zeigt, wie ein Objekt OK über das Properties-Fenster so konfiguriert wird, dass es mit einer Funktionstaste, der Eingabetaste oder der Taste Esc verbunden ist.



**Abbildung 234** Das Objekt Confirm (OK) als Softkey konfigurieren

Darüber hinaus kann das Programm in der Fensteransicht über die Tastatur gesteuert werden. VEE hebt automatisch eine Schaltfläche für das Fenster mit einer gepunkteten Linie hervor. Wenn der Benutzer die Eingabetaste drückt, wird diese Schaltfläche „gedrückt“. Ändert der Benutzer einen Texteingabebereich, wird mit der Eingabetaste diese Änderung übernommen; mit der Taste Esc wird die Bearbeitung abgebrochen. Die Tabulatortaste wechselt zwischen den verschiedenen Auswahlmöglichkeiten und hebt das jeweils aktive Objekt hervor. Die Tastenkombination Umschalt+Tabulatortaste wechselt den Eingabefokus in entgegengesetzter Richtung, also rückwärts. Verwenden Sie die folgenden Tastenkombinationen zum Steuern der Programmausführung

**Tabelle 45**

<b>Tastenkombination</b>	<b>Beschreibung</b>
Strg+G	Ausführen oder fortsetzen
Strg+P	Pause
Strg+T	Schritt

## Bildschirmfarben wählen

Verwenden Sie zur Auswahl der Bildschirmfarben das Dialogfeld **File** ⇒ **Default Preferences**. Stellen Sie die VEE-Umgebung wie gewünscht ein und speichern Sie die Änderungen. Abbildung 235 und Abbildung 236 zeigen, wie die Bildschirm-elemente auf die gewünschte Farbe geändert werden.

## 11 Benutzerschnittstellen verwenden

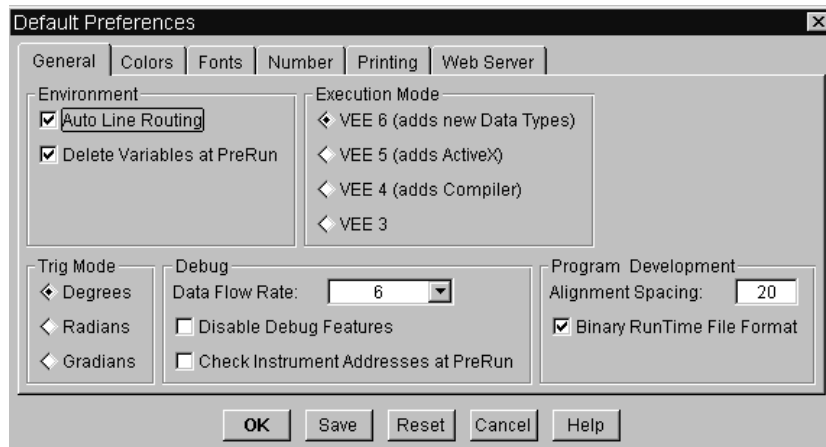


Abbildung 235 Das Dialogfeld Default Preferences

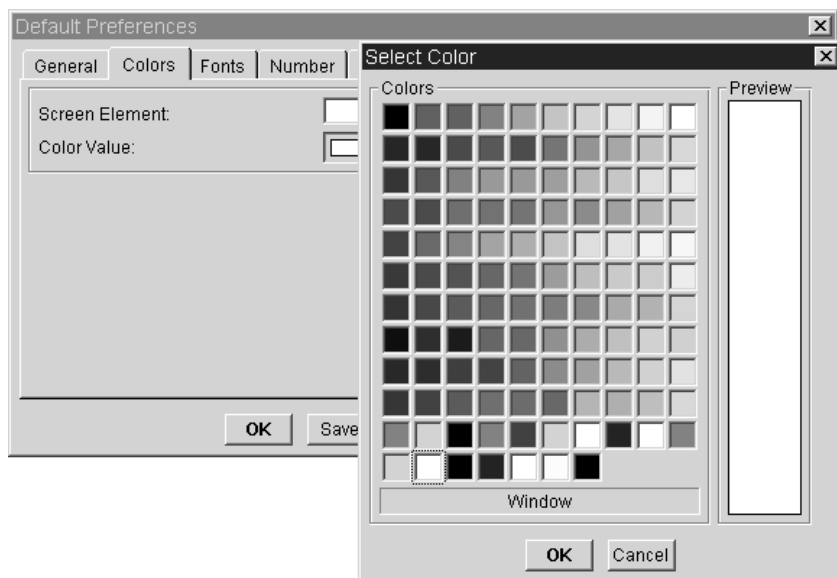


Abbildung 236 Farbauswahl für Bildelemente



## Ein Programm sichern (Laufzeitversion erstellen)

Damit ein Benutzer nicht versehentlich ein Programm ändern kann oder andere Personen sehen können, wie ein Programm arbeitet (durch Anzeige in der Detailansicht), können Sie eine Laufzeitversion eines VEE-Programms erstellen. *Speichern Sie das Originalprogramm und die Laufzeitversion in separaten Dateien.*

### HINWEIS

Wenn Sie eine Laufzeitversion eines VEE-Programms erstellen, kann die Laufzeitversion *nicht* geändert werden. (Sie können die Detailansicht nicht anzeigen.) Erstellen Sie daher Kopien des Originalprogramms, bevor Sie mit diesem Prozess beginnen, und gehen Sie genau anhand der Anweisungen vor.

Führen Sie zum Erstellen einer Laufzeitversion eines VEE-Programms die folgenden Schritte aus:

- 1 Erstellen Sie eine Fensteransicht für die Laufzeitversion des Programms.
- 2 Speichern Sie das Programm, um eine Kopie zu haben, die Sie ändern können.
- 3 Wählen Sie **File** ⇒ **Create RunTime Version...** VEE verwendet automatisch die Erweiterung **\*.vxe**, um die Laufzeitversion als solche zu kennzeichnen.

## Ein Einblendfenster während der Ausführung anzeigen

Sie können bei der Ausführung eines UserObject oder einer UserFunction in einem Programm ein Fenster einblenden. Wählen Sie zum Anzeigen eines Einblendfensters im Properties-Fenster des Objekts die Eigenschaft ShowPanelonExecute. Damit dieses Fenster geöffnet bleibt, bis der Benutzer zur Fortsetzung bereit ist, fügen Sie ein Objekt Confirm (OK) hinzu. Andernfalls wird das Fenster geschlossen, wenn die Ausführung des UserObject bzw. der UserFunction abgeschlossen ist.

Damit ein Einblendfenster auch über mehrere Aufrufe einer UserFunction hinweg angezeigt wird, verwenden Sie die Funktionen ShowPanel() und HidePanel(). Sie können beispielsweise

## 11 Benutzerschnittstellen verwenden

das angezeigte Einblendfenster als Statusanzeige beibehalten, während das Programm ausgeführt wird. Ein Beispiel hierzu finden Sie im nächsten Abschnitt.

### Eine Statusanzeige erstellen

In VEE können Sie Statusanzeigen zur Überwachung der mehrfachen Tests oder Funktionen implementieren. Dies wird mit den Funktionen ShowPanel() und HidePanel() implementiert, wie in Abbildung 237 gezeigt. Weitere Informationen hierzu finden Sie unter „Übung 11-5: Ein Statusanzeige erstellen“ auf Seite 455.

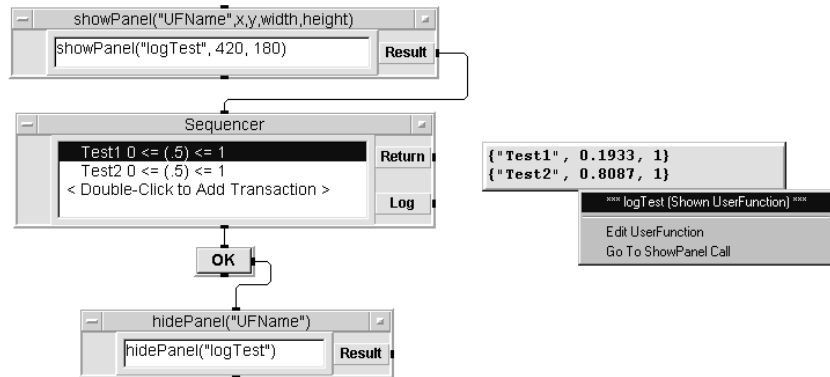


Abbildung 237 Eine Statusanzeige erstellen

## Allgemeine Aufgaben beim Erstellen von Benutzerschnittstellen

In den folgenden Übungen lernen Sie, wie Sie eine Reihe von Funktionen für die Benutzerschnittstelle implementieren. Insbesondere lernen Sie, wie Sie Menüs, Warnungen und Statusanzeigen erstellen und Bitmaps importieren, um Ihren Programmen optisch mehr Eindruck zu verleihen. In allen diesen Übungen haben Sie die Möglichkeit, die Schnittstellen Ihren eigenen Bedürfnissen gemäß anzupassen.

### Übung 11-1: Menüs verwenden

In dieser Übung erstellen Sie eine Benutzerschnittstelle, die ein Menü mit drei Befehlen enthält: **die1**, **die2**, und **die3**. Wenn der Benutzer einen Befehl wählt, wird eine jeweils gleichnamige Funktion aufgerufen, die einen Würfel mit einem, zwei oder drei Punkten anzeigt. Dieses Programm simuliert eine Situation, in der der Benutzer einen Test aus einem Menü wählen muss. Sie lernen außerdem, wie Sie ein Bitmap zum Ändern des Erscheinungsbilds eines Symbols importieren. Das Programm erhält den Namen „Dice“ (Würfel).

Beginnen Sie mit dem Erstellen der drei UserFunctions.

**1** Wählen Sie **Device** ⇒ **UserFunction**.

Sie könnten jedes beliebige Symbol zum Anzeigen des importierten Bitmap verwenden; in diesem Beispiel verwenden wir das Objekt Picture.

**2** Wählen Sie **Display** ⇒ **Picture** und platzieren Sie dieses Objekt in der UserFunction.

**3** Öffnen Sie das Objektmenü **Picture**, klicken Sie auf **Properties** und setzen Sie dann die Eigenschaft ShowTitleBar auf False. Wählen Sie unter Picture **die1.gif**, klicken Sie auf **Scaled** und anschließend auf **OK**.

#### HINWEIS

Klicken Sie zum Öffnen eines Objektmenüs, wenn Show Title Bar deaktiviert ist, mit der rechten Maustaste auf das Objekt.

### HINWEIS

Obwohl VEE standardmäßig das Bitmaps-Unterverzeichnis verwendet, können Sie auch Bitmaps aus anderen Verzeichnissen verwenden.

---

Sie sollten jetzt ein Bild eines Würfels mit einem Punkt auf der Oberseite haben.

- 4 Wählen Sie **Flow** ⇒ **Confirm (OK)** und platzieren Sie das Objekt unter dem Würfel. Verbinden Sie den Sequenzausgangs-Pin von Picture mit dem Sequenzeingangs-Pin von OK.
- 5 Wählen Sie die Objekte Picture und OK (drücken Sie `Strg` und klicken Sie auf die Objekte, um sie zu markieren). Öffnen Sie das Kontextmenü Edit, indem Sie auf den Hintergrund zeigen und die rechte Maustaste drücken. Wählen Sie Add to Panel.
- 6 Ändern Sie die UserFunction Title und Panel Title in die1. Ordnen Sie die Objekte an und ändern Sie ihre Größe nach Bedarf.

### HINWEIS

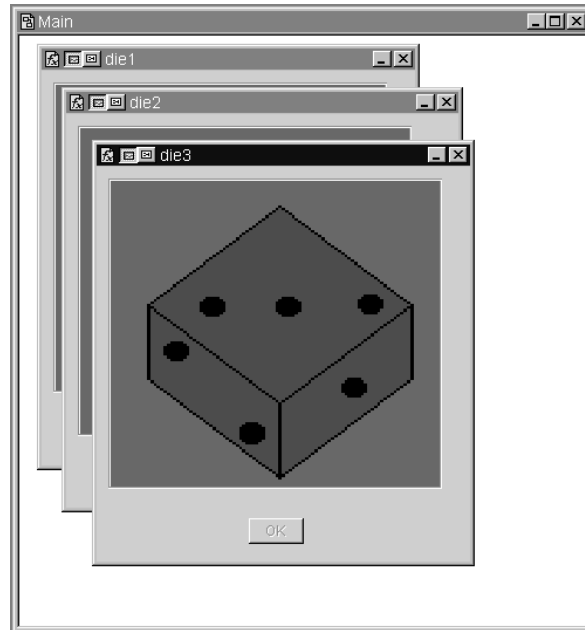
Um Objekte im Panel View (Fensteransicht) zu verschieben, klicken Sie mit der rechten Maustaste auf das Objekt, wählen Sie Move or highlight the object (das Objekt sollte von einem grauen Rahmen umgeben sein) und ziehen Sie es an seine neue Position.

---

Wählen Sie im Fenster Properties **ShowPanelonExecute**. Klicken Sie auf den Ordner Panel, aktivieren Sie die Eigenschaft **PanelGridSize** und ändern Sie die Gittergröße auf 2, um eine präzisere Ausrichtung zu ermöglichen. Klicken Sie anschließend auf **OK**.

- 7 Erstellen Sie zwei weitere UserFunctions, indem Sie im die1-Objektmenü den Befehl Clone wählen. Die neuen UserFunctions erscheinen automatisch als **die2** und **die3**. Ändern Sie die Picture-Objekte in `die2.gif` bzw. `die3.gif`. Überprüfen Sie alle Einstellungen der neuen Funktionen, um sicherzustellen, dass sie mit denen von die1 übereinstimmen

(mit Ausnahme der Namen und Bitmaps). Das Programm sollte jetzt aussehen wie in Abbildung 238. Minimieren Sie die Funktionsfenster.



**Abbildung 238** Eine frühe Phase im Programm Dice

Erstellen Sie ein Menü, über das eine dieser drei Funktionen zum Aufrufen ausgewählt werden kann.

#### 8 Wählen Sie **Data** ⇒ **Selection Control** ⇒ **Radio Buttons**.

Radio Buttons ist ein Objekt, das einen Aufzählungswert (Datentyp Enum; eine Zeichenfolge mit einer zugeordneten Ordnungszahl) aus der benutzerdefinierten Liste an seinen oberen Ausgangsanschluss ausgibt. Wenn Sie die Liste beispielsweise als Montag, Dienstag, Mittwoch, Donnerstag und Freitag definieren, kann der Benutzer in einem Menü den Tag wählen und mit Radio Buttons den entsprechenden Tag dann ausgeben.

Das erste Element in der Liste hat die Ordnungsposition 0; das n-te Element die Ordnungsposition n-1. „Montag“ hat somit in der obigen Liste die Ordnungsposition 0 und „Freitag“ die Ordnungsposition 4. Die Ordnungspositionen erscheinen am unteren Ausgabeanschluss. Eine ausführlichere Erläuterung finden Sie über das Objektmenü und den darin enthaltenen Befehl Help.

- 9 Öffnen Sie das Radio Buttons-Objektmenü und wählen Sie Edit Enum Values...

Geben Sie die Namen der Funktionen **die1**, **die2** und **die3** ein, indem Sie mit der Tabulatortaste zwischen den einzelnen Felder wechseln, *nicht* jedoch nach dem letzten Feld. Klicken Sie auf **OK**.

### HINWEIS

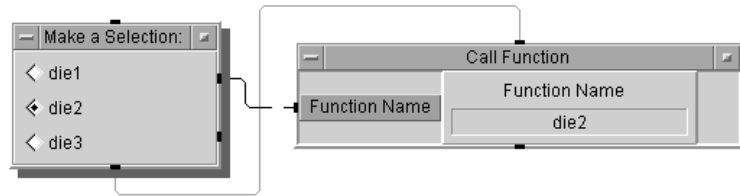
Es gibt sechs Menüformate zur Steuerung der Datenauswahl. Radio Buttons zeigt die Einträge als Schaltflächen an. Die Auswahl des Benutzers wird im Textformat als Datentyp Enum ausgegeben. Cyclic Button blättert nacheinander durch die Aufzählungswerte, wenn der Benutzer auf die Schaltfläche klickt. List zeigt alle Aufzählungswerte in einer Liste an, wobei das ausgewählte Element hervorgehoben ist. Drop-down list, Pop-up list und Slider list sind die weiteren drei Optionen.

- 
- 10 Öffnen Sie das Radio Buttons-Objektmenü, klicken Sie auf **Properties** und wählen Sie **Auto Execute**. Ändern Sie den Titel auf die Eingabeaufforderung: Make a Selection:

Richten Sie ein Call-Objekt ein, sodass der Wert, den der Benutzer im Objekt Radio Buttons wählt, jetzt zum Namen der Funktion wird, die das Objekt Call Function aufruft.

- 11 Klicken Sie auf **Device** ⇒ **Call**. Wählen Sie **Add Terminal** ⇒ **Control Input**, anschließend Function Name und klicken Sie dann auf **OK**. Der Steuer-Pin Function Name akzeptiert einen Wert des Datentyps Enum oder Text als Eingabe. Verbinden Sie den Datenausgangs-Pin Radio Buttons mit dem Eingangsanschluss Function Name des Objekts Call Function. Verbinden Sie den Sequenzausgangsanschluss von Radio Buttons mit dem Sequenzeingangsanschluss von Call Function. Klicken Sie auf **die2** unter **Make a Selection:** und beachten Sie,

dass sich **Call Function Name** in **die2** geändert wird, wie in Abbildung 239 dargestellt.



**Abbildung 239** Das Programm Dice (Detailansicht)

### HINWEIS

Der Eingangsanschluss Call erfordert ein Text Scalar; VEE wandelt daher das Enum Scalar in ein Text Scalar um.

Denken Sie daran, dass die gepunktete Linie einen Steuer-Pin kennzeichnet. Wenn Auto Execute eingeschaltet ist, wird Radio Buttons immer ausgeführt, wenn Sie eine Änderung daran vornehmen, und sendet die Auswahl an Call. Der Steuer-Pin an Call Function ersetzt den Funktionsnamen, sobald der Pin Daten empfängt. Das Objekt Call ruft die angegebene Funktion *nicht* auf, bis sein Sequenzeingangsanschluss ausgelöst wird.

### HINWEIS

Wenn ein Programm Auto Execute und seine Sequenzanschlüsse verwendet, braucht der Benutzer nicht auf die Schaltfläche **Run** zu klicken, um das Programm zu starten.

Fügen Sie eine Benutzerschnittstelle hinzu, die nur die Eingabeaufforderung, das Menü und die Einblendfenster mit den Auswahloptionen anzeigt.

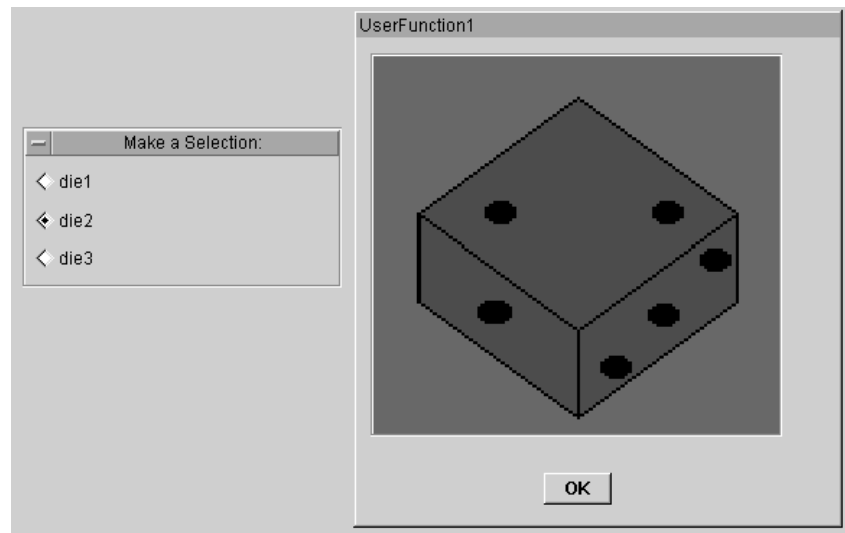
**12** Wählen Sie das Objekt Radio Buttons, indem Sie **Strg** drücken und dann auf das Zielobjekt klicken. Wählen Sie anschließend **Edit** ⇒ **Add To Panel**.

**13** Öffnen Sie das Objektmenü, wählen Sie Properties und passen Sie die Farben und Schriftarten wie gewünscht an.

## 11 Benutzerschnittstellen verwenden

Führen Sie das Programm aus, indem Sie eine Auswahl treffen. (Verwenden Sie nicht die Schaltfläche Run, da hierdurch die bereits in dem Menü getroffene Auswahl verwendet wird.)

Das Programm sollte bei der Ausführung aussehen wie in Abbildung 240 dargestellt.



**Abbildung 240** Das Programm Dice (Fensteransicht)

Vor der nächsten Übung sollten Sie einige Dinge beachten:

- Sie können die in dieser Übung beschriebenen Techniken beim Erstellen von Menüs für jedes beliebige Programm verwenden.
- Radio Buttons können auch zur Auswahl eines Programms in einer kompilierten Sprache verwendet werden; verwenden Sie hierzu das Objekt Execute Program mit einem Steuer-Pin („Command“), der das aufzurufende Programm kennzeichnet. Wenn Sie eine Bibliothek kompilierter Funktionen importiert hatten, können Sie auch das Objekt Call zum Ausführen einer Funktion aus der Bibliothek verwenden.



- Sie können dieses Programm auch mit dem Dateneingangsanschluss File Name am Objekt Picture in einer einzelnen UserFunction optimieren und anschließend die entsprechende Bitmap-Datei an das Objekt senden. Wenn Sie viele verschiedene Bitmaps verwenden, ist diese Vorgehensweise bei der Programmierung effizienter.
- Normalerweise verwenden Sie in komplizierteren Programmen Run statt AutoExecute. Sie können das Programm so gestalten, dass es bei einer Datenkonstante oder einem Auswahl-Steuerobjekt angehalten wird; verwenden Sie hierzu Wait for Input statt AutoExecute. Weitere Informationen hierzu finden Sie unter Help.

## Übung 11-2: Bitmaps für den Fensterhintergrund importieren

Bitmaps sind für den Inhalt Ihrer Programme nicht von Bedeutung; Sie können damit aber Ihre Tests übersichtlicher und professioneller gestalten. Beispielsweise können Sie eine Schemazeichnung importieren, die den Inhalt des Tests darstellt. In dieser Übung importieren Sie Bitmaps für Fensterhintergründe, auf denen die Standard-VEE-Objekte platziert werden.

Bitmaps können importiert werden für Symbole, für das Objekt Picture oder für Fensteransichts-Hintergründe in UserObjects oder UserFunctions. Sie erstellen eine Einblend-UserFunction mit dem Namen Bitmap, die ein Objekt Label und ein Objekt Confirm (OK) enthält.

- 1 Wählen Sie **Device** ⇒ **UserFunction**.
- 2 Wählen Sie **Flow** ⇒ **Confirm (OK)** und **Display** ⇒ **Label** und platzieren Sie diese Objekte im Fenster UserFunction.
- 3 Ändern Sie den Namen der UserFunction in Bitmap.
- 4 Wählen Sie die Objekte OK und Label, um sie zu markieren. Öffnen Sie das Kontextmenü Edit, indem Sie auf den Arbeitsbereich der UserFunction zeigen und mit der rechten Maustaste klicken. Wählen Sie Add to Panel.
- 5 Öffnen Sie das UserFunction-Menü, wählen Sie Properties und anschließend ShowPanelonExecute. (Vergessen Sie nicht, dass Sie auch auf die Titelleiste doppelklicken können,

## 11 Benutzerschnittstellen verwenden

um das Properties-Fenster zu öffnen.) Heben Sie die Auswahl Show Title Bar unter Pop-up Panel auf.

Öffnen Sie den Ordner Panel, ändern Sie die Grid Size in 2, wählen Sie unter Background Picture **default.gif** und **Scaled** und klicken Sie auf **OK**.

- 6 Öffnen Sie das Fenster **Properties** für das Objekt **Label**, und stellen Sie die Werte wie folgt ein:

**Tabelle 46** Eigenschaften des Objekts Label

Label-Eigenschaften	Beschreibung
Title:	Ändern Sie den Eintrag in Bitmap Function.
Justification	Ändern Sie den Eintrag in Center.
BackColor	Wählen Sie Light Gray und klicken Sie auf <b>OK</b> .
Font	Wählen Sie einen größeren Schriftgrad, einen fetten Schriftschnitt und klicken Sie auf <b>OK</b> . Markieren Sie Automatically Resize Object on Font Change.
Border	Klicken Sie auf <b>Raised</b> . Klicken Sie auf <b>OK</b> , um die Änderungen zu bestätigen und das Properties-Fenster zu schließen.

- 7 Positionieren Sie den Titel Bitmap Function und die Schaltfläche OK wie gewünscht. Minimieren Sie die UserFunction Bitmap.
- 8 Wechseln Sie zum Hauptfenster. Klicken Sie auf **Device** ⇒ **Call**, dann im Objektmenü auf Select Function und wählen Sie dann Bitmap. Führen Sie das Programm aus. Das Einblendfenster sollte aussehen wie in Abbildung 241.



Abbildung 241 Die Bitmap-Funktion

### Übung 11-3: Eine auffällige Warnung erstellen

Diese Übung enthält mehrere verschachtelte UserFunctions. Die erste UserFunction ist der Alarm selbst, der ein rotes Quadrat anzeigt und einen Signalton ausgibt. Die zweite UserFunction ruft den Alarm wiederholt auf und generiert somit einen Blinkeffekt und einen pulsierenden Signalton, bis der Benutzer den Alarm ausschaltet.

Beginnen Sie mit der Programmierung der Alarmfunktion.

- 1 Wählen Sie **Device** ⇒ **UserFunction**. Ändern Sie den Namen in `alarm`.
- 2 Wählen Sie **Display** ⇒ **Beep** und platzieren Sie dieses Objekt links oberhalb der UserFunction. Passen Sie die Einstellungen so an, dass ein lauter Signalton mit einer Dauer von 1 Sekunde generiert wird. Ändern Sie den Wert im Feld **Duration (sec)** auf 1. Ändern Sie den Wert im Feld **Volume (0-100)** auf 100.

### HINWEIS

Bei diesen Anleitungen wird davon ausgegangen, dass Ihr Computer mit einer Hardware ausgestattet ist, die den Signalton unterstützt. Einige Windows 98-, Windows 2000-, Windows NT 4.0- und Windows XP-Systeme haben eine geänderte Standard-Systemkonfiguration für den Standard-Systemton.

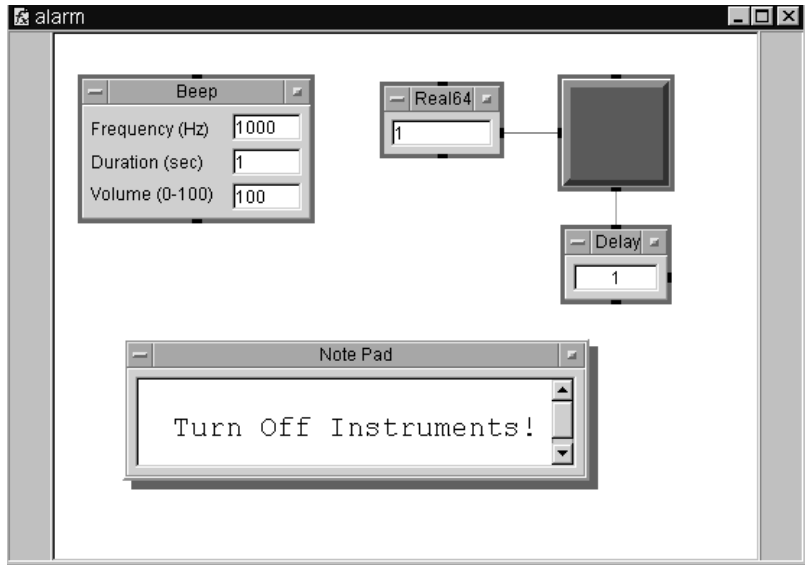
---

### HINWEIS

Sie brauchen das Objekt **Beep** nicht an einer anderen Komponente anzuschließen. Es wird bei der Ausführung der Funktion aktiviert.

---

- 3 Klicken Sie auf **Display** ⇒ **Indicator** ⇒ **Color Alarm** und platzieren Sie dieses Objekt in der UserFunction. Öffnen Sie das „Color Alarm“-Objektmenü, klicken Sie auf Properties und stellen Sie die Felder wie folgt ein: setzen Sie die Eigenschaft ShowTitleBar auf False. setzen Sie die Eigenschaft Shape auf Rectangular. setzen Sie die Eigenschaft HighLimit auf .66 und löschen Sie in der Eigenschaft HighText möglicherweise vorhandenen Text.
- 4 Klicken Sie auf **Data** ⇒ **Constant** ⇒ **Real64**, ändern Sie die Angabe in 1 und schließen Sie das Objekt am Eingangsanschluss von Color Alarm an.
- 5 Verwenden Sie zur Synchronisierung der Anzeige mit dem Beep-Objekt im Sekundentakt ein Objekt Delay (Verzögerung) mit der Einstellung 1 Sekunde.
- 6 Wählen Sie **Flow** ⇒ **Delay**, stellen Sie den Wert auf 1 ein und schließen Sie seinen Sequenzeingangs-Pin am Sequenzausgangs-Pin von Color Alarm an. Der Alarm hält dann eine Sekunde lang an.
- 7 Wählen Sie **Display** ⇒ **Note Pad** und entfernen Sie die Vorlageninformationen. Fügen Sie die folgende Meldung hinzu: TURN OFF INSTRUMENTS! Ändern Sie die Größe des Note Pad wie gewünscht.
- 8 Wechseln Sie zum Hauptfenster. Wählen Sie zunächst **Device** ⇒ **Call**, wählen Sie im Call-Objektmenü dann Select Function und anschließend alarm. Führen Sie das Programm testweise aus. Die Detailansicht des UserFunction-Alarms sollte aussehen wie in Abbildung 242.



**Abbildung 242** Die UserFunction Alarm (Detailansicht)

- 9 Kehren Sie zurück zum Alarmfenster. Wählen Sie die Anzeige Color Alarm und zeigen Sie das Objekt Note Pad an. Öffnen Sie das Menü Edit und wählen Sie Add To Panel. Ordnen Sie in der Fensteransicht die Objekte an und ändern Sie ihre Größe, indem Sie die Ziehmarken der Objekte ziehen, die Funktion `Align` verwenden oder die Objekte verschieben. Setzen Sie die Eigenschaft `PanelDrawGrid` auf `True`, wenn Sie die Funktion `Snap-to-Grid` verwenden möchten. Öffnen Sie das Note Pad-Objektmenü und klicken Sie auf Properties. Legen Sie die Einträge wie folgt fest:

**Tabelle 47** Notepad-Eigenschaften

Eigenschaft	Beschreibung
ShowTitleBar	False
EditEnabled	False
Border	Ändern Sie die Angabe in <code>Raised border</code> .

- 10 Ändern Sie den Color Alarm ebenfalls in die Angabe Raised Border.
- 11 Klicken Sie auf eine leere Stelle innerhalb von UserFunction, um auf das Properties-Fenster zugreifen zu können, und wählen Sie ShowPanelonExecute. Setzen Sie die Eigenschaft ShowTitleBar auf False. Ändern Sie den Panel Title zu alarm. Minimieren Sie alarm.
- 12 Wechseln Sie zum Hauptfenster und löschen Sie das Objekt Call. (VEE hält die Alarmfunktion weiterhin im Speicher.) Wenn Sie die Funktion erneut ändern wollen, wählen Sie **Edit** ⇒ **Edit UserFunction** oder doppelklicken Sie auf das Symbol.)
- 13 Erstellen Sie die Funktion, die die Alarmfunktion wiederholt aufruft.
- 14 Wählen Sie **Device** ⇒ **UserFunction** und ändern Sie den Namen der UserFunction in warning.
- 15 Wählen Sie **Flow** ⇒ **Repeat** ⇒ **Until Break**.
- 16 Wählen Sie **Device** ⇒ **Call**, ändern Sie Function Name in alarm, und verbinden Sie ihren Sequenzeingangs-Pin mit dem Datenausgangs-Pin von Until Break.

Fügen Sie ein „Check Box“-Objekt hinzu, über das der Benutzer gefragt wird, ob der Alarm ausgeschaltet werden soll.

- 17 Wählen Sie **Data** ⇒ **Toggle Control** ⇒ **Check Box**. Öffnen Sie das Properties-Fenster für Check Box, ändern Sie die Eigenschaft Title in Turn off alarm?, wählen Sie die Eigenschaft Scaled, setzen Sie sie auf True, wählen Sie InitializeAtPreRun und vergewissern Sie sich, dass diese Eigenschaft auf **False** gesetzt ist, tragen Sie in der Eigenschaft TitleFont einen größeren Wert für den Schriftgrad des Namens ein. Verbinden Sie den Sequenzausgangsanschluss von Call mit dem Sequenzeingangsanschluss von Check Box.

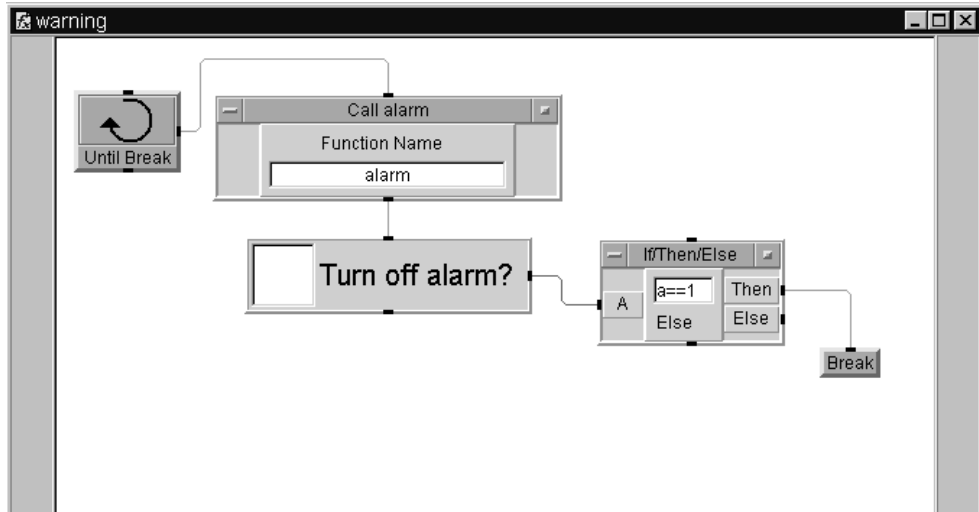
Dadurch wird ein Eingangsobjekt erstellt, das ein Objekt Check Box verwendet. Wenn der Benutzer auf dieses Kontrollkästchen klickt, erscheint ein **X**, und das Objekt gibt 1 aus; andernfalls gibt das Objekt 0 aus. Die Ausgabe kann mit einem If/Then/Else-Objekt getestet werden, um VEE anzuweisen, was als nächstes ausgeführt werden soll.

- 18 Wählen Sie **Flow** ⇒ **If/Then/Else** und platzieren Sie dieses Objekt rechts von Toggle. Verbinden Sie den Datenausgang

von **Toggle** mit dem Dateneingang A des Objekts **If/Then/Else**. Ändern Sie den Ausdruck im **If/Then/Else**-Objekt in:  $a == 1$ . (Das Symbol für „ist gleich“ ist  $==$ , nicht  $=$ .) Wenn der Anschluss **A** eine 1 enthält, löst der **Then**-Ausgang aus; andernfalls löst der **Else**-Ausgang aus.

Verbinden Sie den Ausgang von **Toggle** mit dem Eingang von **If/Then/Else**.

- 19** Wählen Sie **Flow**  $\Rightarrow$  **Repeat**  $\Rightarrow$  **Break** und verbinden Sie dieses Objekt mit dem **Then**-Ausgang des Objekts **If/Then/Else**, wie in Abbildung 243 dargestellt.



**Abbildung 243** Die UserFunction „Warning“ (Detailansicht)

- 20** Wählen Sie das Objekt (**Turn off alarm?**), indem Sie auf die rechte Seite des Objekts klicken. Öffnen Sie das Einblendmenü **Edit** und wählen Sie **Add To Panel**. Ändern Sie die Größe der Fensteransicht, sodass sie die Check Box umschließt.
- 21** Öffnen Sie das Properties-Fenster der Warn-UserFunction, wählen Sie **ShowPanelonExecute** und setzen Sie **ShowPopupPanelTitle** auf **False** (da der Titel für den Benutzer keinen Zweck erfüllt).

## 11 Benutzerschnittstellen verwenden

- 22** Wechseln Sie zum Hauptfenster und klicken Sie auf **Device** ⇒ **Call**, öffnen Sie sein Objektmenü, klicken Sie Select Function und anschließend auf Warning. Verschieben Sie das Call-Objekt nach oben in die Mitte der Anzeige. Minimieren Sie das Hauptfenster.
- 23** Standardmäßig zeigt VEE die Alarm- und Warnungsfenster in der Mitte des Bildschirms an, sodass der Alarm über dem Kontrollkästchen zum Stoppen des Alarms blinkt. Da beide Bildschirmpositionen nicht gesperrt sind, können Sie sie auf dem Bildschirm anders anordnen, indem Sie die Einblendfenster an die gewünschte Position ziehen. Mit dem blinkenden Alarmfenster ist dies allerdings etwas schwierig. Klicken und ziehen Sie statt dessen die Kante des Fensters. Falls erforderlich, stoppen Sie das Programm über die Schaltfläche „Stop“ in der Symbolleiste. Führen Sie das Programm aus.

Wenn Sie die beiden Fenster wie in Abbildung 244 gezeigt positioniert haben, können Sie das Programm durch Klicken auf das Feld neben der Eingabeaufforderung **Turn off alarm?** stoppen.



**Abbildung 244** Das Programm Warning

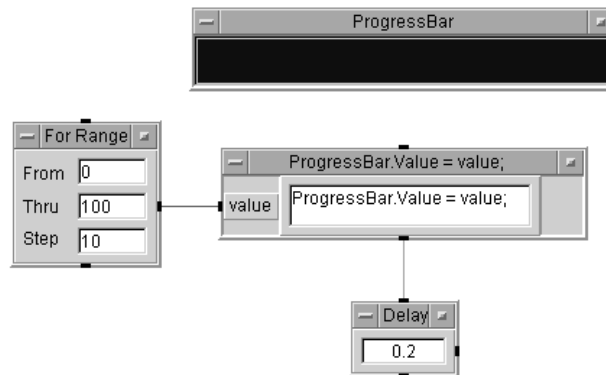


## Übung 11-4: Ein ActiveX-Steuerelement verwenden

Diese Übung zeigt die Verwendung eines ActiveX-Steuerelements in VEE. Sie können ActiveX-Steuerelemente aus anderen Anwendungen in VEE-Programme einbinden. In diesem Fall binden Sie ein Steuerelement ProgressBar (Fortschrittsleiste) ein und verwenden eine Schleife, um den Verlauf in dieser Leiste von 0% bis 100% anzuzeigen. Dieses allgemeine Prinzip gilt auch für andere ActiveX-Steuerelemente.

- 1 Klicken Sie auf **Device** ⇒ **ActiveX Control References...** und wählen Sie Microsoft Windows Common Controls 6.0. Klicken Sie auf **OK**.
- 2 Klicken Sie auf **Device** ⇒ **ActiveX Controls** ⇒ **ProgressBar**. Vergrößern Sie das Objekt ProgressBar. Öffnen Sie sein Objektmenü und beachten Sie, dass der Name ProgressBar lautet. VEE hat automatisch eine deklarierte Variable erstellt, die sich auf das ActiveX-Steuerelement bezieht. Sie können den Namen ProgressBar in Formula-Ausdrücken verwenden, genau wie mit anderen Variablen oder Dateneingängen.
- 3 Klicken Sie auf **Device** ⇒ **Formula & Object Browser**, wählen Sie ActiveX Objects, Library: MSComctlLib, Class: ProgressBar, Members: Value und klicken Sie auf **Create Set Formula**. Platzieren Sie das Objekt oben in der Mitte des Hauptfensters.
- 4 Für die Schleife von 1 bis 100 und zur Anzeige des Verlaufs fügen Sie ein Objekt For Range hinzu. Wählen Sie **Flow** ⇒ **Repeat** ⇒ **For Range**, platzieren Sie das Objekt unter ProgressBar und stellen Sie die Werte wie folgt ein: **From**: 0, **Thru**: 100 und **Step**: 10. Verbinden Sie den Ausgang von For Range mit dem Eingangsanschluss Value von ProgressBar.
- 5 Sie können die Ausführung des Programms verlangsamen, um die Aktualisierung der ProgressBar besser sehen zu können. Wählen Sie hierzu **Flow** ⇒ **Delay** und platzieren Sie das Objekt rechts vom Objekt **For Range**. Setzen Sie es auf den Wert .2. Verbinden Sie den Sequenzausgangs-Pin der ProgressBar mit dem Sequenzeingangs-Pin des Objekts Delay (siehe Abbildung 245) und führen Sie das Programm aus.

## 11 Benutzerschnittstellen verwenden



**Abbildung 245** Das ActiveX-Steuerelement ProgressBar verwenden

### HINWEIS

Objektmenüs von ActiveX-Steuerelementen haben sowohl Properties als auch Control Properties. Über Properties werden die VEE-Eigenschaften des Objekts festgelegt. Die Steuerelement-Eigenschaften werden über das Steuerelement bereitgestellt; sie können sich je nach Typ des ActiveX-Steuerelements unterscheiden.

Untersuchen Sie alle mit VEE ausgelieferten Beispiele für Steuerelemente, um die Arbeitsweise dieser Elemente besser zu verstehen. Sie können auch mit weiteren Steuerelementen und Anzeigen anderer Hersteller die Funktionen der Benutzerschnittstelle in VEE erweitern.

Abbildung 246 zeigt ein weiteres Beispiel von VEE mit einem Steuerelement aus MS Chart. Nach der Auswahl einer Steuerelement-Bibliothek über das Dialogfeld **Device** ⇒ **ActiveX Controls References** können Sie mit dem Function & Object Browser oder dem Objekt Declare Variable die Eigenschaften und Methoden eines Steuerelements verwenden.

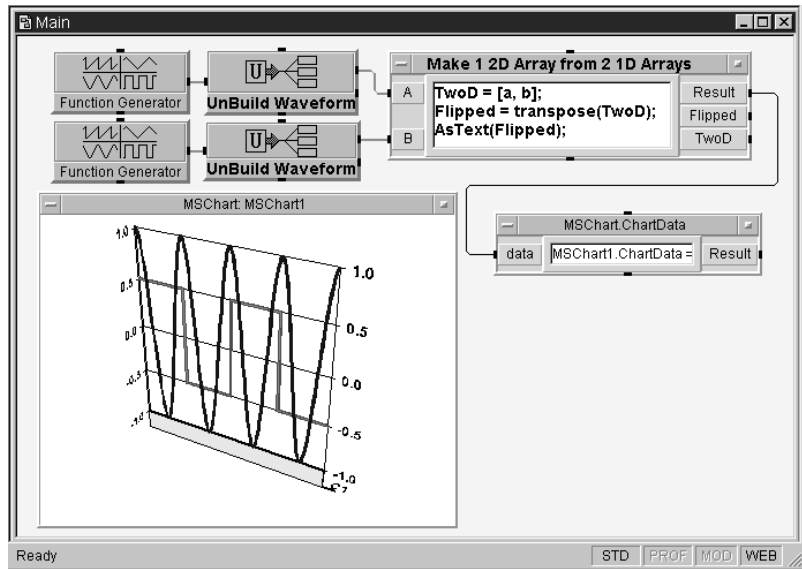


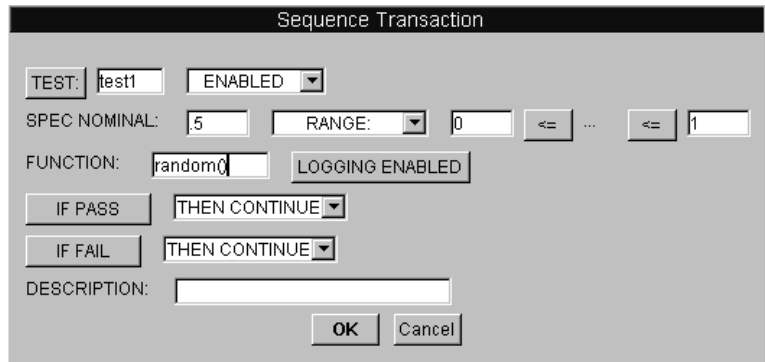
Abbildung 246 Beispiel eines ActiveX-Steuerelements mit MSChart

## Übung 11-5: Ein Statusanzeige erstellen

In dieser Übung lernen Sie, wie Sie die Funktionen aus dem Programm Dice zum Erstellen einer Statusanzeige verwenden können. (Die Übung zum Programm Dice finden Sie unter „Übung 11-1: Menüs verwenden“ auf Seite 439.) Normalerweise wird dieses Programm mit dem Sequencer-Objekt verwendet, wenn eine Reihe von Tests ausgeführt werden sollen und Sie die zurückgegebenen Ergebnisse anzeigen wollen. Sie verwenden die Funktion **random()**, die bei Verwendung der Standardeinstellungen einen Real-Wert zwischen **0** und **1** zurückgibt.

- 1 Klicken Sie auf **Device** ⇒ **Sequencer**. Doppelklicken Sie auf die Transaktionsleiste, und konfigurieren Sie Ihren Test mit dem Standardnamen **test1** und ersetzen Sie das Feld **FUNCTION**: durch **random()**. Siehe Abbildung 247.

## 11 Benutzerschnittstellen verwenden



**Abbildung 247** Test1 konfigurieren

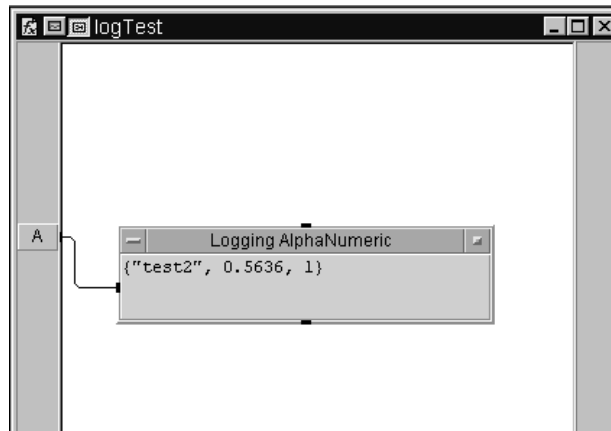
- 2 Konfigurieren Sie einen zweiten Test test2 auf die gleiche Weise.
- 3 Öffnen Sie das Properties-Fenster von Sequencer und wählen Sie das Register Logging. Wählen Sie unter Logging Mode die Option Log Each Transaction To: logTest(thisTest) und klicken Sie auf **OK**.

Bei der Ausführung der einzelnen Transaktionen erstellt der Sequencer für jeden Test einen Record mit dem Namen „thisTest“; die Felder dieses Records können über das gleiche Register konfiguriert werden. Sie können eine UserFunction mit dem Namen „logTest“ (oder einem anderen Namen) erstellen, damit der Sequencer die UserFunction LogTest() mit dem Record thisTest am Ende jeder Transaktion ausführt. Auf diese Weise können Sie die Statusanzeige aktualisieren.

- 4 Wählen Sie **Device** ⇒ **Function & Object Browser** ⇒ **Built-in Functions** ⇒ **Panel** ⇒ **showPanel** und platzieren Sie dieses Objekt über dem Sequencer. Löschen Sie die Eingangs-Pins und ändern Sie die Parameter in "logTest", 420, 180, wobei Sie die beiden letzten Parameter weglassen. Verbinden Sie den Ausgangs-Pin Result von showPanel mit dem Sequenzeingangs-Pin des Sequencer. ShowPanel gibt bei einer erfolgreichen Ausführung eine **1** aus.

LogTest ist der Name der UserFunction. Die beiden anderen Parameter sind die X- und Y-Koordinaten auf dem Bildschirm, ausgehend von der linken oberen Ecke. Diese Angabe weist VEE an, wo das User Function-Fenster bei der Anzeige platziert werden soll. (Die Fensterabmessungen sind in diesem Beispiel nicht enthalten.)

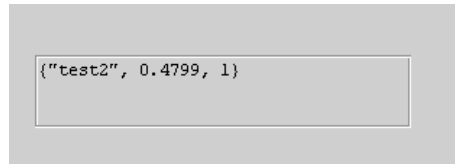
- 5 Erstellen Sie die UserFunction mit dem Namen logTest wie nachfolgend gezeigt. Fügen Sie einen Eingangs-Pin hinzu. (Der Protokoll-Datensatz wird als Eingang verwendet.) Legen Sie das Objekt Logging AlphaNumeric auf dem Fenster ab und verbinden Sie es mit dem Eingangs-Pin. Wählen Sie Logging AlphaNumeric und klicken Sie auf **Edit** ⇒ **Add to Panel**. Passen Sie in der Fensteransicht die Größe und Platzierung wie gewünscht an. Heben Sie die Auswahl der Anzeige und der Fenster-Titelleisten auf.



**Abbildung 248** Die UserFunction „LogTest“ (Detail)

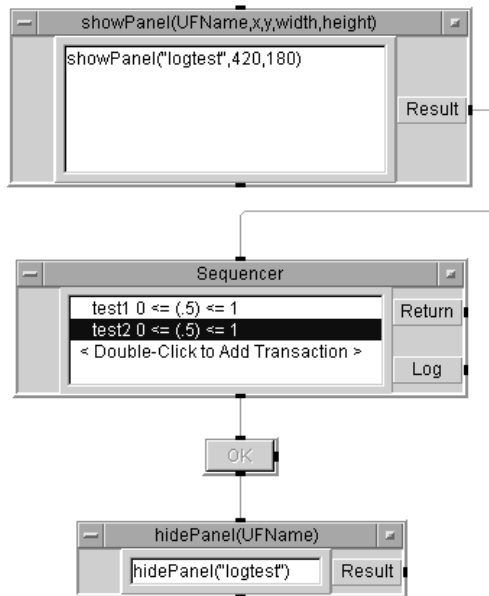
Abbildung 249 zeigt die Fensteransicht nach der Ausführung des Programms. Diese Darstellung gibt Ihnen eine Vorstellung von der Anzeige des Programms.

## 11 Benutzerschnittstellen verwenden



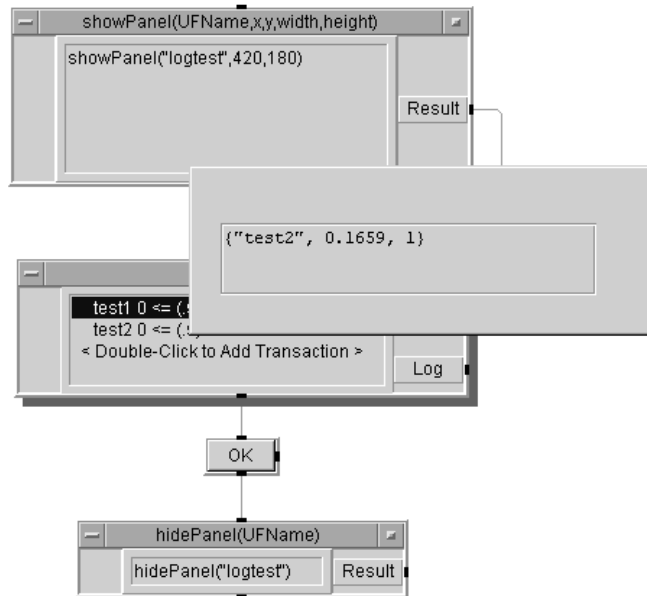
**Abbildung 249** Die UserFunction „LogTest“ (Fensteransicht)

- 6 Wählen Sie `hidePanel` im Fenster `Function & Object Browser` und anschließend `Flow`  $\Rightarrow$  **Confirm (OK)**. Ändern Sie den Parameter `hidePanel()` in `logTest`. Löschen Sie den Eingangs-Pin. Verbinden Sie die Objekte wie in **Abbildung 250** gezeigt.



**Abbildung 250** Statusanzeigeprogramm (vor der Ausführung)

- 7 Führen Sie das Programm aus. Es sollte aussehen wie in **Abbildung 251**.



**Abbildung 251** Das Statusanzeigeprogramm (während der Ausführung)

Das Objekt showPanel zeigt das UserFunction-Fenster an, ohne jedoch die UserFunction aufzurufen. Der Sequencer ruft die UserFunction zwei Mal über die Protokollfunktion auf; bei jedem dieser Aufrufe wird das Fenster aktualisiert. Wenn der Benutzer mit der Ausführung fertig ist, kann er OK drücken, um das Fenster auszublenden. In diesem Beispiel wird ein Objekt OK verwendet, um das Objekt hidePanel auszulösen; Sie können diese Objekt jedoch auch an einer anderen Stelle im Programm platzieren, um das Timing der Ausführung zu steuern. Arbeiten Sie das Programm schrittweise ab, um das Einblenden der Statusanzeige, das Aktualisieren mit den Ergebnissen von Test1 und Test2 und das anschließende Ausblenden genauer zu sehen.

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Die wichtigsten Punkte zu Benutzerschnittstellen zusammenfassen.
- Ein Menü zur Auswahl von Tests mit einer Benutzerschnittstelle verwenden.
- Ein Bitmap-Datei für eine Benutzerschnittstelle importieren.
- Eine Statusanzeige erstellen.
- Einige der von VEE bereitgestellten Benutzerschnittstellen-Funktionen auflisten.
- Ein Programm sichern.
- Eine auffällige Warnung erstellen.
- Ein ActiveX-Steuererelement erstellen und dessen Eigenschaften und Methoden im Function & Object Browser finden.



# 12

## Agilent VEE-Programme optimieren

Überblick 463

Grundlegende Techniken zur Programmoptimierung 464

Überblick über kompilierte Funktionen 471

Dynamic Link Libraries verwenden 475

Agilent VEE-Ausführungsmodi 482

Der Agilent VEE Profiler 489

Kapitel-Checkliste 490

### Agilent VEE-Programme optimieren

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- Grundlegende Techniken zur Programmoptimierung
- Dynamic Link Libraries (DLLs) verwenden
- Optimieren mit kompilierten Funktionen
- Den VEE-Compiler verwenden
- Den VEE Profiler verwenden

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*

## Überblick

In diesem Kapitel lernen Sie, wie Sie die Ausführungsgeschwindigkeit Ihrer VEE-Programme optimieren. Bei der Leistung eines Testprogramms sind drei grundlegende Faktoren zu betrachten: die Geschwindigkeit beim Erfassen des Messwerts, die Geschwindigkeit der Übertragung von Daten an den Computer und die Geschwindigkeit, mit der das Programm die Daten verarbeitet. Durch die Optimierung des VEE-Programms können Sie seine Verarbeitungsgeschwindigkeit erhöhen.

Im ersten Abschnitt lernen Sie die grundlegenden Prinzipien für die Optimierung eines VEE-Programms kennen. Außerdem lernen Sie, wie Sie Dynamic Link Libraries (DLLs) verwenden. Im nächsten Abschnitt wird beschrieben, wie Sie Programme durch die Verwendung kompilierter Funktionen optimieren. Als nächstes finden Sie einen Überblick über den VEE-Compiler. Abschließend finden Sie einen Abschnitt über den VEE-Profiler.

**HINWEIS**

Die Techniken in diesem Kapitel gelten unabhängig davon, ob Sie den Compiler verwenden oder nicht.

---

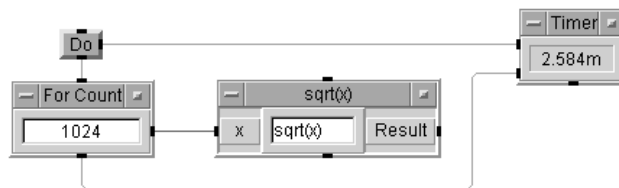
## Grundlegende Techniken zur Programoptimierung

Lesen Sie zum Optimieren von VEE-Programmen die Informationen in diesem Abschnitt. Mit den hier beschriebenen Techniken können Sie sich einen guten Stil bei der Programmierung in VEE angewöhnen.

### Setzen Sie bei der Arbeit mit Arrays, wann immer das möglich ist, mathematische Operationen ein

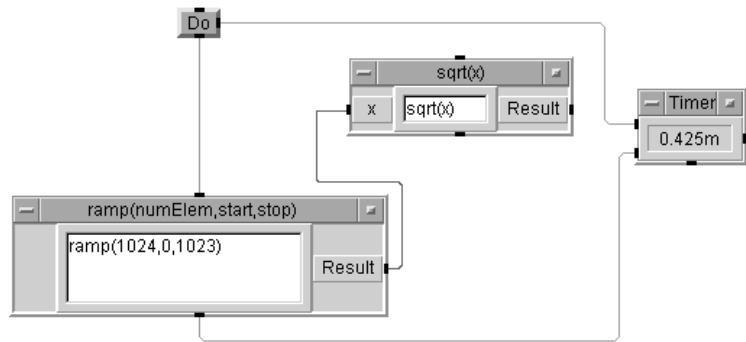
Die Leistung eines Programms kann entscheidend verbessert werden, wenn Sie bei der Arbeit mit Arrays mathematische Operationen anwenden, wann immer dies möglich ist. Angenommen, ein Test soll die Quadratwurzel eines erfassten Messwerts ermitteln. Der herkömmliche Weg für die Programmierung dieser Aufgabe ist das Erfassen des Messwerts und die Berechnung der Wurzel in einer Schleife. In VEE können Sie stattdessen alle Messwerte in einem Array speichern und die Quadratwurzel des Arrays in einem einzigen Arbeitsschritt berechnen.

In Abbildung 252 durchläuft das Programm 1024 Iterationen. Jede Iteration berechnet eine Quadratwurzel.



**Abbildung 252** Quadratwurzeln pro Messung berechnen

In Abbildung 253 erstellt das Programm ein Array aus 1024 Elementen und berechnet die Quadratwurzel der Array-Werte (das Ergebnis ist ein Array aus Quadratwurzeln). Obwohl die beiden Programme die gleichen Ergebnisse liefern, wird das Programm in Abbildung 253 ca. 6 Mal schneller ausgeführt als das in Abbildung 252. (Bei diesem Beispiel wird von einem Pavilion PC mit 300 MHz ausgegangen.)



**Abbildung 253** Quadratwurzeln mit Math-Array berechnen

Der Unterschied bei der Ausführungszeit der beiden Programme ergibt sich aus der erforderlichen Zeit zur Ausführung eines Objekts. Jede Ausführung eines Objekts erfordert einen bestimmten Systemaufwand. Wenn Sie daher mit Hilfe von Arrays statt skalarer Werte die Anzahl der Ausführungen für ein Objekt verringern, wird das Programm schneller ausgeführt.

Die Verwendung eines Do-Objekts empfiehlt sich bei der Verwendung von Timern, um sicherzustellen, dass der Timer in beiden Programmen zuerst auslöst. Die Funktion ramp generiert ein Array aus 1024 Elementen (von 0 bis 1023).

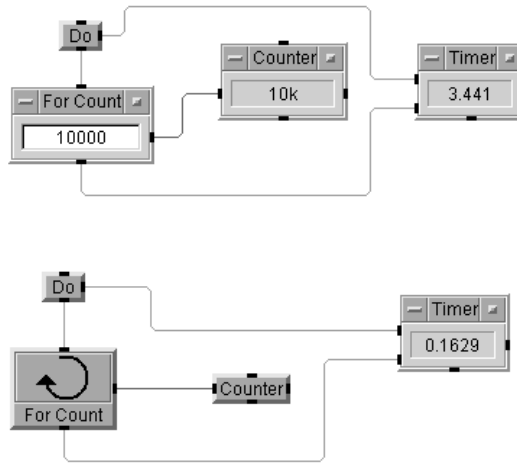
## HINWEIS

Vergewissern Sie sich für eine schnellere Ausführung, dass Sie den modernsten VEE-Ausführungsmodus verwenden. Wählen Sie hierzu **File** ⇒ **Default Preferences** (oder klicken Sie in der Symbolleiste auf die entsprechende Schaltfläche). Wählen Sie unter Execution Mode das Optionsfeld VEE6 & VEE7 und klicken Sie auf **Save**. In der Statusleiste am unteren Rand des VEE-Fensters sollte VEE6&7 angezeigt werden.

## Wenn möglich, Objekte minimiert darstellen

Je mehr Informationen VEE auf dem Bildschirm verwalten muss, desto länger dauert die Ausführung eines Programms. Verwenden Sie zur Optimierung des Programms für Objekte, die ihren Inhalt aktualisieren (z. B. der Counter), die Symbol-

darstellung statt der offenen Ansicht. Das Beispiel in Abbildung 254 arbeitet durch die Symboldarstellung der Objekte For Count und Counter ca. 46 Mal schneller.



**Abbildung 254** Programme durch Verwendung von Symbolen optimieren

### Die Anzahl der Programmobjekte verringern

Je mehr Erfahrung Sie sammeln, desto mehr werden Sie dazu tendieren, weniger Objekte in VEE-Programmen zu verwenden. Es gibt zwei weitere Techniken zur Reduzierung der Anzahl von Objekten und somit zur Optimierung von Programmen:

- 1 Verwenden Sie eine einzelne Gleichung in einem Formula-Objekt, statt separate mathematische Objekte zu verwenden. Legen Sie beispielsweise die Gleichung  $(a + b) * c / d$  in einem **Formula**-Objekt ab, statt separate Objekte für die Addition, Multiplikation und Division zu verwenden. Verwenden Sie außerdem Konstanten in der Formel statt konstanter Objekte, die mit Eingängen verbunden sind. (Legen Sie Konstanten mit **Set Variable** fest.)
- 2 Verschachteln Sie Funktionsaufrufe innerhalb anderer Funktionsparameterlisten. In Abbildung 255 verwendet die Funktion randomize das von der Funktion ramp generierte Array. In Abbildung 256 ist der Funktionsaufruf auf ramp in dem

Aufruf auf randomize verschachtelt; dies führt zu einer etwas schnelleren Programmausführung.

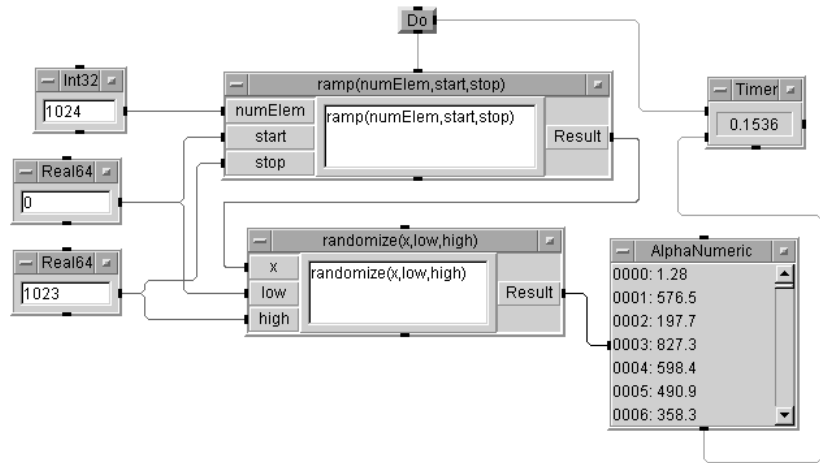


Abbildung 255 Funktionsaufrufe ohne Optimierung

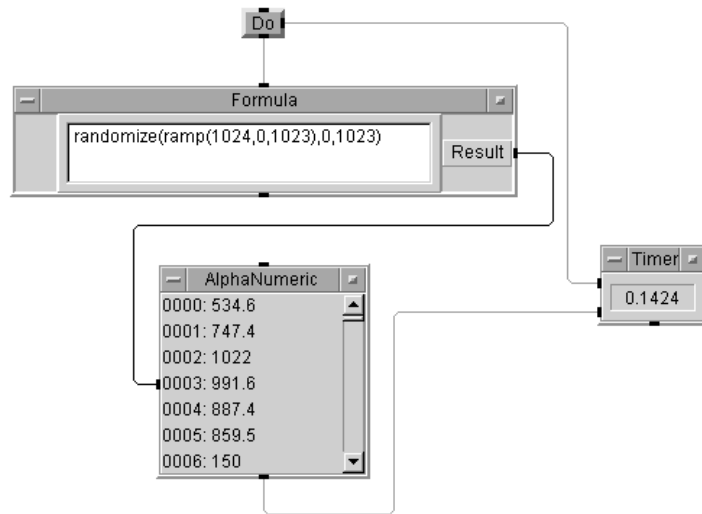


Abbildung 256 Funktionsaufrufe mit Optimierung

### Weitere Möglichkeiten zur Optimierung von Agilent VEE-Programmen

Nachfolgend sind einige weitere Optimierungstechniken beschrieben, die Sie in Ihren Programmen anwenden können:

- Vergewissern Sie sich, dass Sie den VEE-Compiler verwenden, indem Sie Ihre Programme im Ausführungsmodus VEE 4 oder einem höheren Modus ausführen. Weitere Informationen hierzu finden Sie unter „Agilent VEE-Ausführungsmodi“ auf Seite 482.
- Starten Sie das Programm von der Fensteransicht statt von der Detailansicht aus. VEE muss in diesem Fall weniger Objekte auf dem Bildschirm verwalten.
- Verwenden Sie globale Variablen statt der Übergabe von Werten in und aus UserObjects und UserFunctions (insbesondere für große Arrays oder Records). Deklarieren Sie alle globalen Variablen. Auf diese Weise können Sie auch globale Variablen verwenden. Siehe **Data** ⇒ **Variable** ⇒ **Declare Variable**.



- Erfassen Sie Daten für grafische Anzeigen und plotten Sie das gesamte Array in einem Arbeitsschritt statt jeden Skalarpunkt einzeln. Wenn die X-Werte eines Plots gleichmäßige Abstände aufweisen, verwenden Sie eine „XY Trace“-Anzeige statt eines Plots  $X$  gg.  $Y$ .
- Verwenden Sie ein Objekt If/Then/Else mit mehreren Bedingungen statt mehrerer If/Then/Else-Objekte.
- Definieren Sie grafische Anzeigen so einfach wie möglich. Die Einstellungen, die die schnellste Aktualisierung ermöglichen, sind **Grid Type**  $\Rightarrow$  **None** und keine Auswahl im Dialogfeld Properties. Verwenden Sie AutoScale-Steuer-Pins nur, wenn dies unbedingt erforderlich ist, und deaktivieren Sie nach Möglichkeit die Funktion Automatic AutoScaling (im Ordner Scales).
- Verwenden Sie beim Lesen von Daten aus einer Datei die Transaktion **ARRAY 1D TO END: (\*)** statt der Ausführung von READ-Transaktionen mit einzelnen Elementen und der Verwendung des EOF-Pin.
- Verwenden Sie den Sequencer zum Steuern des Ausführungsablaufs mehrerer UserFunctions statt separater Call-Objekte.
- Aktivieren Sie bei Verwendung des Sequencer die Protokollierung für Transaktionen nur, wenn ein Log-Record erforderlich ist.
- Wenn Sie die Anzeigen Strip Charts und Logging Alpha-Numeric verwenden, stellen Sie unter Properties den Wert für Buffer Size auf den kleinsten möglichen Wert für Ihre Anwendung ein.
- Verwenden Sie den triadischen Operator (*bedingung ? ausdruck1 : ausdruck2*) statt des Objekts If/Then/Else mit Gates und einem Junction-Objekt.
- Stellen Sie bei Verwendung von Bitmaps diese auf Actual oder Centered statt auf Scaled, da Scaled ein wenig länger dauert.
- Deaktivieren Sie bei Verwendung von Indikatoren wie Fill Bar oder Thermometer die Option Show Digital Display.

## 12 Agilent VEE-Programme optimieren

- Wenn Sie mit Color Alarms arbeiten und schnell zwischen verschiedenen Farben wechseln müssen, schalten Sie Show 3D Border aus.

Zusätzlich zu den bereits erwähnten Techniken können Sie durch das Verbinden kompilierter Funktionen aus anderen Sprachen mit Ihren VEE-Programm die Ausführungsgeschwindigkeit verbessern. Die Verwendung kompilierter Funktionen auf PCs (als DLLs) ist im nächsten Abschnitt beschrieben.

## Überblick über kompilierte Funktionen

Sie können eine kompilierte Funktion, beispielsweise eine Funktion in einer DLL (Dynamic Link Library), in einem VEE-Programm verwenden. Hierzu müssen Sie sich die kompilierte Funktion besorgen oder auf die folgende Weise erstellen:

- 1 Schreiben Sie Funktionen in C, C++, Fortran oder Pascal und kompilieren Sie sie.
- 2 Schreiben Sie eine Definitionsdatei für die Funktionen.
- 3 Erstellen Sie eine gemeinsam verwendbare Bibliothek, die die kompilierten Funktionen enthält.

### Vorteile der Verwendung kompilierter Funktionen

Die Verwendung kompilierter Funktionen in einem VEE-Programm bietet folgende Vorteile:

- Schnellere Ausführung
- Nutzung aktueller Testprogramme in anderen Sprachen
- Entwicklung von Datenfiltern in anderen Sprachen und Integration in VEE-Programmen
- Sichern eigener Routinen

#### HINWEIS

Das Hinzufügen kompilierter Funktionen macht den Entwicklungsprozess komplexer. Verwenden Sie daher eine kompilierte Funktion *nur*, wenn der erforderliche Funktionsumfang oder die Leistung mit keinem der folgenden Elemente erzielt werden können: einer VEE UserFunction, einem Execute Program-Ausgang zum Betriebssystem oder einem ActiveX Automation-Aufruf zu einem anderen Programm.

---

### Überlegungen zum Design bei der Verwendung kompilierter Funktionen

Wenn Sie vorhaben, kompilierte Funktionen in einem VEE-Programm zu verwenden, beachten Sie die folgenden Informationen:

- Sie können alle über das Betriebssystem verfügbaren Funktionen nutzen einschließlich mathematischer Routinen, Instrumenten-E/A usw. Von dem Programm aus, das Sie verbinden wollen, können Sie jedoch nicht auf die VEE-internen Funktionen zugreifen.
- Sie müssen in Ihrer kompilierten Funktion eine Fehlerprüfung bereitstellen, da VEE Fehler in einer externen Routine nicht erfassen kann.
- Sie müssen die Zuordnung des in der externen Routine zugeordneten Speichers aufheben.
- Wenn Sie Daten an eine externe Routine übergeben, vergewissern Sie sich, dass Sie die Eingangsanschlüsse des Call-Objekts auf den Typ und die Form der für die Routine erforderlichen Daten konfigurieren.
- System-E/A-Ressourcen werden eventuell gesperrt; Ihre externe Routine sollte diese Art von Ereignis verarbeiten können.
- Wenn Ihre externe Routine Arrays akzeptiert, muss sie einen gültigen Zeiger für die Art der untersuchten Daten aufweisen. Außerdem muss die Routine die Größe des Arrays prüfen. Wenn die Routine die Größe ändert, müssen Sie die neue Größe an das VEE-Programm zurückgeben.
- Die kompilierte Funktion muss als letzte Anweisung `return()` verwenden und nicht `exit()`. Wenn die kompilierte Funktion „`exit`“ verwendet, wird auch VEE beendet, da eine kompilierte Funktion mit VEE verbunden ist.

- Wenn Sie die Grenzen eines Array überschreiben, hängt das Ergebnis von der verwendeten Sprache ab. In Pascal, das eine Prüfung der Grenzen durchführt, tritt ein Laufzeitfehler auf, und VEE wird gestoppt. In Sprachen wie C, die keine Prüfung der Grenzen durchführen, ist das Ergebnis nicht vorhersehbar; es können jedoch eine intermittierende Datenfehler auftreten oder ein Absturz von VEE.

## Richtlinien zur Verwendung kompilierter Funktionen

Beachten Sie bei Verwendung kompilierter Funktionen in einem VEE-Programm die folgenden Richtlinien:

- Sie können eine kompilierte Funktion genau wie eine User-Function aufrufen und konfigurieren. Sie können die gewünschte Funktion entweder über das Call-Objektmenü mit Select Function wählen oder ihren Namen eingeben. In beiden Fällen werden die Ein- und Ausgangsanschlüsse des Objekts Call Function automatisch konfiguriert, sofern VEE die Funktion erkennt. Die erforderlichen Informationen werden über die Definitionsdatei bereitgestellt. (VEE erkennt, ob die Bibliothek bereits importiert wurde.)
- Zum Umkonfigurieren der Ein- und Ausgangsanschlüsse von Call wählen Sie im Objektmenü den Eintrag Configure Pinout. Bei beiden Methoden konfiguriert VEE das Call-Objekt mit den für die Funktion erforderlichen Eingangsanschlüssen und mit einem Ret Value-Ausgangsanschluss für den Rückgabewert der Funktion. Darüber hinaus gibt es einen Ausgangsanschluss zu jedem als Referenz übergebenen Eingang.
- Rufen Sie die kompilierte Funktion von einem Ausdruck in einem Formula-Objekt oder von anderen Ausdrücken, die zur Ausführungszeit ausgewertet werden, aus mit ihrem Namen auf. Sie könnten eine kompilierte Funktion auch aufrufen, indem Sie ihren Namen in einem Ausdruck innerhalb einer To File-Transaktion angeben.

### HINWEIS

Lediglich der Rückgabewert der kompilierten Funktion (Ret Value im Call-Objekt) kann von innerhalb eines Ausdrucks abgerufen werden. Wenn Sie weitere von der Funktion zurückgegebene Parameter abrufen wollen, müssen Sie das Objekt Call verwenden.

---

- Löschen Sie eine Bibliothek kompilierter Funktionen mit dem Objekt Delete Library (im Menü Device). Mit den Objekten Import Library, Call und Delete Library können Sie die Ladezeit des Programms verkürzen und Speicherplatz sparen, indem Sie die Objekte importieren und wieder löschen, wenn das Programm den Aufruf dieser Objekte abgeschlossen hat.

## Dynamic Link Libraries verwenden

Auf PCs können Sie die in Dynamic Link Libraries (DLLs) enthaltenen kompilierten Funktionen als Teil eines VEE-Programms verwenden. DLLs können aus kompilierten Funktionen bestehen, die Sie selbst geschrieben haben (Informationen zum Schreiben eigener DLLs können Sie über Microsoft anfordern), oder es handelt sich um gekaufte oder aus dem Web geladene DLLs von anderen Herstellern.

### HINWEIS

VEE unterstützt die beiden Aufrufkonventionen „\_cdecl“ und „\_stdcall“. Die meisten von Kunden selbst geschriebenen DLLs verwenden die Aufrufkonvention \_cdecl. Die meisten Win32 API-Aufrufe verwenden \_stdcall. VEE unterstützt beide Namenskonventionen; Sie können somit die meisten gängigen DLLs wie auch eigene DLLs verwenden.

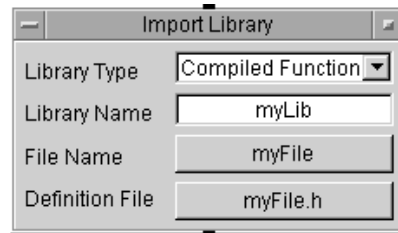
## Eine DLL in ein Agilent VEE-Programm integrieren

In diesem Abschnitt wird beschrieben, wie Sie eine DLL in ein VEE-Programm importieren. Schreiben oder besorgen Sie sich die DLL wie oben beschrieben, und führen Sie die folgenden Schritte durch, um die DLL verwenden zu können:

### 1 Wählen Sie **Device** ⇒ **Import Library**.

Die Dropdown-Liste Library Type bietet drei Auswahlmöglichkeiten: UserFunction, Compiled Function und Remote Function. Ändern Sie den Eintrag in dieser Liste in Compiled Function (der Standardwert lautet UserFunction). Für eine Funktion des Typs Compiled Function enthält das Objekt Import Library eine Schaltfläche, über die Sie eine Definitionsdatei auswählen können (siehe Abbildung 257).

## 12 Agilent VEE-Programme optimieren



**Abbildung 257** Eine Bibliothek kompilierter Funktionen importieren

Die Felder sind nachfolgend beschrieben:

**Tabelle 48** Import-Library-Felder

Feld	Beschreibung
Library Name (Name der Bibliothek)	Der Name, über den VEE die Bibliothek identifiziert. Im allgemeinen wird dieser Name verwendet, wenn Sie die Bibliothek löschen wollen, nachdem Sie im Programm verwendet wurde.
File Name (Dateiname)	Die Datei, die die gemeinsame Bibliothek enthält.
Definition File (Definitionsdatei)	Die Include-Datei mit den Prototypen der Funktionen. Dies ist normalerweise eine *.h-Datei.

### HINWEIS

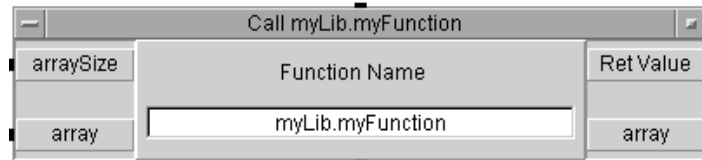
In der Entwicklungsphase können Sie eine Bibliothek auch manuell laden, indem Sie im Objektmenü *Load Lib* wählen.

### 2 Wählen Sie **Device** ⇒ **Call**.

Wenn Sie die Bibliothek mit **Import Library** importiert haben, wählen Sie **Device** ⇒ **Call**, um ein **Call**-Objekt zu erstellen. Anschließend können Sie die kompilierte Funktion aufrufen, indem Sie im Call-Objektmenü den Eintrag **Select Function** und im angezeigten Listenfeld die gewünschte Funktion wählen.



Das in Abbildung 258 dargestellte Call-Objekt ruft beispielsweise in **myLibrary** die kompilierte Funktion mit dem Namen **myFunction** und den Parametern **arraySize** und **array** auf.



**Abbildung 258** Call-Objekte für kompilierte Funktionen verwenden

VEE konfiguriert automatisch das Call-Objekt mit dem Funktionsnamen und der richtigen Anzahl von Ein- und Ausgangs-Pins. Ein zweiter, dritter usw. Pin wird für alle Parameter, die als Referenz an die Funktion übergeben werden, zugeordnet. Wenn Sie den Funktionsnamen eingegeben haben, können Sie das Objekt auch konfigurieren, indem Sie im Objektmenü den Eintrag **Configure Pinout** wählen.

## HINWEIS

Sofern die entsprechende Bibliothek geladen wurde, können Sie eine DLL-Funktion auch von einem Ausdrucksfeld aus aufrufen. Bei einer Verwendung auf diese Weise müssen Sie die Parameter in Klammern nach dem Funktionsnamen angeben; die Funktion gibt nur ihren Rückgabewert zurück. Alle als Referenz übergebenen Parameter können nur über das Objekt **Call** abgerufen werden. Sie können beispielsweise den folgenden Ausdruck in einem Formula-Objekt verwenden:

```
2 * yourFunc (a, b)
```

Die Angaben **a** und **b** beziehen sich auf die beiden Eingangs-Pins am Formula-Objekt, und der Rückgabewert von **yourFunc** wird mit 2 multipliziert und am Ausgangs-Pin angelegt.

### 3 (Optional) Klicken Sie auf **Device** ⇒ **Delete Library**.

Bei der Entwicklung des Programms können Sie im Objektmenü auch **Delete Lib** wählen, um die Bibliothek programmgesteuert zu löschen. Durch das Löschen der Bibliothek nach ihrer Verwendung im Programm wird die Ladezeit verringert und Arbeitsspeicher gespart.

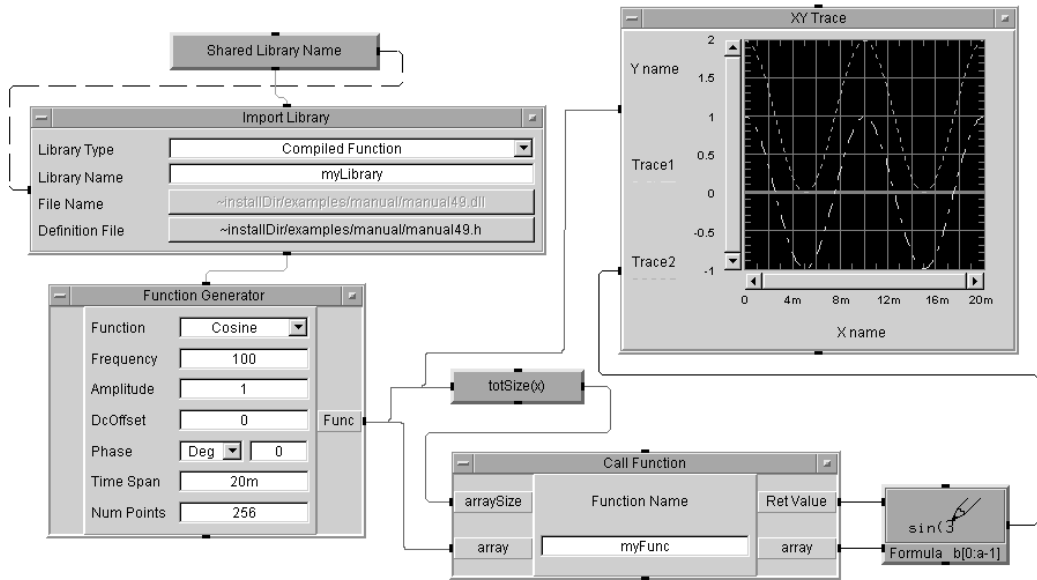
## Ein Beispiel zur Verwendung einer DLL

In dieser Übung importieren Sie eine DLL und rufen eine Funktion von dieser DLL auf. Die verwendete DLL ist im VEE-Produkt enthalten. (Dieses Programm ist für die Verwendung auf allen Plattformen konzipiert.)

Öffnen Sie die Datei **manual49.vee**. Sie befindet sich unter:

*<installation directory>* \EXAMPLES \MANUAL \MANUAL49.

Sehen Sie sich dieses Beispiel genau an. Es sollte aussehen wie in Abbildung 259.



**Abbildung 259** Ein Programm, das eine DLL (MANUAL49) verwendet

**Tabelle 49** Element von Abbildung 259

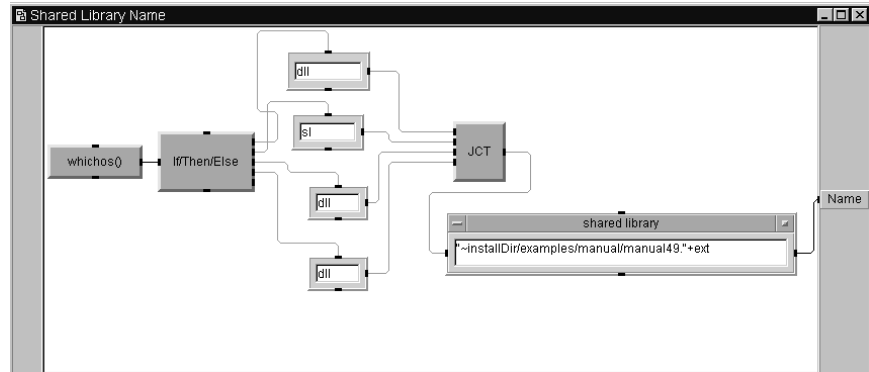
Element	Beschreibung
Import Library	Vor dem ersten Aufruf der kompilierten Funktion Call Function muss die DLL mit dem Objekt Import Library (im Menü Device) geladen werden.
Call Function	<b>MANUAL49</b> ruft eine kompilierte Funktion mit dem Namen myFunc auf. MyFunc erfordert den C-Datentyp long; dieser Typ entspricht dem VEE-Datentyp Int32. Diese Zahl gibt die Größe eines Array an. Der zweite Parameter ist ein Zeiger auf das Array aus Real-Zahlen. Die Definitionsdatei befindet sich in <b>MANUAL49.H</b> , und die Quellendatei für den C-Code befindet sich in <b>MANUAL49.C</b> . MyFunc addiert zu jedem Array-Element den Wert 1.
Function Generator	Der Function Generator wird zum Erstellen einer Wellenform verwendet, die den Ausgang zum Array-Pin am Objekt Call myFunc darstellt.
totSize	Mit dem Objekt <b>totSize</b> (im Fenster <b>Math &amp; Functions</b> ) wird die Größe der Wellenform ermittelt; diese Größe ist der Ausgang zum arraySize-Eingangs-Pin von <b>Call myFunc</b> .
XY Trace	Das Objekt <b>XY Trace</b> zeigt die ursprüngliche und die neue Wellenform an.
Formel	Der Ausgangs-Pin <b>Ret Value</b> des <b>Call</b> -Objekts enthält die Größe des zurückgegebenen Arrays, sodass der Ausdruck <b>B[0:A-1]</b> in dem Formula-Objekt dieses Array für das Anzeigeobjekt richtig angibt.

Führen Sie das Programm aus und beachten Sie, dass der zweite Trace an allen Punkten der Wellenform um eins größer ist als der erste Trace.

Ein weiterer wichtiger Punkt ist die Methode, mit der das Programm für alle VEE-Plattformen portierbar gemacht wird. Windows 98, Windows 2000, Windows NT 4.0 und Windows XP verwenden einen Microsoft 32-Bit Compiler. Alle solchermaßen erstellte DLLs haben die Erweiterung **\*.dll**.

## 12 Agilent VEE-Programme optimieren

Das UserObject mit dem Namen Shared Library Name kennzeichnet das verwendete Betriebssystem und überträgt den richtigen Bibliotheksnamen an das Objekt Import Library, wie in Abbildung 260 gezeigt.



**Abbildung 260** Das UserObject „Shared Library Name“

Die Funktion `whichos()` wurde in einem umbenannten Formula-Objekt zur Kennzeichnung des Betriebssystems verwendet. Ein erweitertes If/Then/Else-Objekt untersucht die Ausgabe der Funktion `whichos()` und löst anschließend die entsprechende Textkonstante aus. Diese Dateierweiterung wird anschließend mit einem umbenannten Formula-Objekt an den Dateinamen **MANUAL49** angehängt. (Der Eingangsanschluss am Formula-Objekt mit dem Namen `shared library` wurde ebenfalls in `ext` umbenannt.)

Ein Steuer-Pin für einen File Name wurde dem Objekt Import Library hinzugefügt; daher wurde eine gepunktete Linie zwischen dem UserObject und der Import Library gezogen.

### HINWEIS

Überprüfen Sie To/From Socket auf die gemeinsame Verwendung von Daten in Mischumgebungen, z. B. beim Senden von Daten an eine Datenbank. Auch die Beispielprogramme „Battleship“ und „Euchre“ machen intensiven Gebrauch von Sockets bei der Kommunikation zwischen verschiedenen VEE-Prozessen.

## Das Objekt „Execute Program“ im Vergleich zu kompilierten Funktionen

Wenn Sie entscheiden, ob Sie ein „Execute Program“-Objekt oder eine kompilierte Funktion zum Integrieren Ihrer kompilierten Programme mit VEE verwenden wollen, beachten Sie die folgenden Informationen.

### Das Objekt "Execute Program"

Das Objekt Execute Program zeichnet sich durch folgende Merkmale aus:

- Einfache Bedienung
- Längere Startzeit
- Geschützter Adressbereich
- Auswahl zwischen synchroner und asynchroner Ausführung
- Service der asynchronen Ereignisse
- Hohe Sicherheit (wenn das aufgerufene Programm abstürzt, erhalten Sie eine Fehlermeldung)
- Besser geeignet für eine laufende Datenerfassung

### Kompilierte Funktionen

Kompilierte Funktionen, die von den Objekten **Import Library** und **Call** verwendet werden, zeichnen sich durch folgende Merkmale aus:

- Kürzere Startzeit
- Kommunikation durch Weitergabe von Stack- und Speicherbereichen, die mit VEE gemeinsam genutzt werden

### Synchrone Ausführung

- Signale und Ausnahmebedingungen werden nicht blockiert oder erfasst (z. B. „Allgemeine Schutzverletzungen“)
- Compiler für Textsprachen erforderlich
- Kompliziertere Bedienung. Höheres Risiko bei der Verwendung. („Out-of-bounds“-Array-Fehler (Bereichsüberschreitung) oder ein Überschreiben des Speichers führt zu einem Absturz von VEE)

### Agilent VEE-Ausführungsmodi

Die Agilent VEE-Ausführungsmodi ermöglichen die Ausführung von Programmen, die mit älteren Versionen von VEE erstellt wurden. Mithilfe der Ausführungsmodi können in einer neueren Version von VEE auch Programme ausgeführt werden, die mit einer älteren Version von VEE geschrieben wurden, und zwar in exakt der gleichen Weise wie mit der älteren Version von VEE. Dadurch ist VEE rückwärtskompatibel und unterstützt bestehende Programme.

#### HINWEIS

Der *Ausführungsmodus* wurde in älteren Versionen von VEE als *Kompatibilitätsmodus* bezeichnet.

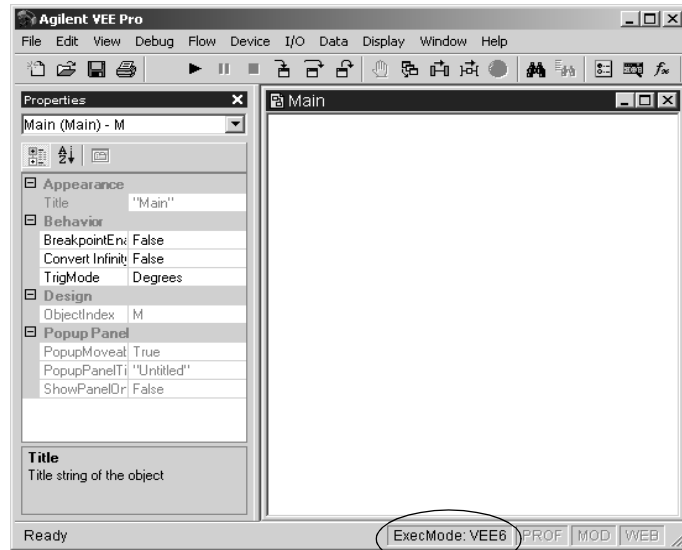
---

VEE kennt vier Ausführungsmodi:

- VEE 6 & VEE 7 (fügt neue Datentypen hinzu)
- VEE 5 (fügt ActiveX hinzu)
- VEE 4 (fügt den Compiler hinzu)
- VEE 3.x

Der Ausführungsmodus des ausgeführten Programms wird in der Statusleiste von VEE angezeigt, wie in Abbildung 261 dargestellt.

Wenn Sie ein bestehendes Programm in VEE ausführen, wird standardmäßig der Ausführungsmodus für die VEE-Version verwendet, in der das Programm erstellt wurde. Ein VEE 5.0-Programm, das Sie in VEE 6.0 oder 7.0 öffnen, wird beispielsweise standardmäßig im VEE 5 Ausführungsmodus ausgeführt.



↑  
Ausführungsmodus

**Abbildung 261** Anzeige des Ausführungsmodus in der VEE-Statusleiste

## Der Agilent VEE-Compiler

Der VEE-Compiler wird in VEE 4 und höheren Ausführungsmodi automatisch aktiviert. Der Compiler ermöglicht eine weitaus schnellere Programmausführung sowie eine besser vorhersehbare Weitergabe von Objekten. Weitere Informationen zum Compiler und zu den Unterschieden zwischen den verschiedenen Ausführungsmodi finden Sie im Handbuch *VEE Pro Advanced Techniques*.

## Den Ausführungsmodus ändern

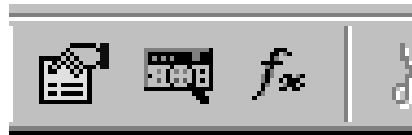
Sie sollten alle neuen Programme im Modus VEE 6 & 7 erstellen. Falls Sie bestehende Programme haben, empfiehlt es sich, den Ausführungsmodus zu ändern, wenn Sie einem solchen Programm neue Funktionen hinzufügen. Wenn Sie beispielsweise ein Programm in VEE 5.0 geschrieben haben und eine neue Funktion aus VEE 6.0 hinzufügen, sollten Sie den Ausführungs-

## 12 Agilent VEE-Programme optimieren

modus in VEE 6 & 7 ändern; andernfalls wird das VEE 5.0-Programm eventuell nicht richtig ausgeführt.

Führen Sie zum Ändern des Ausführungsmodus die folgenden Schritte aus:

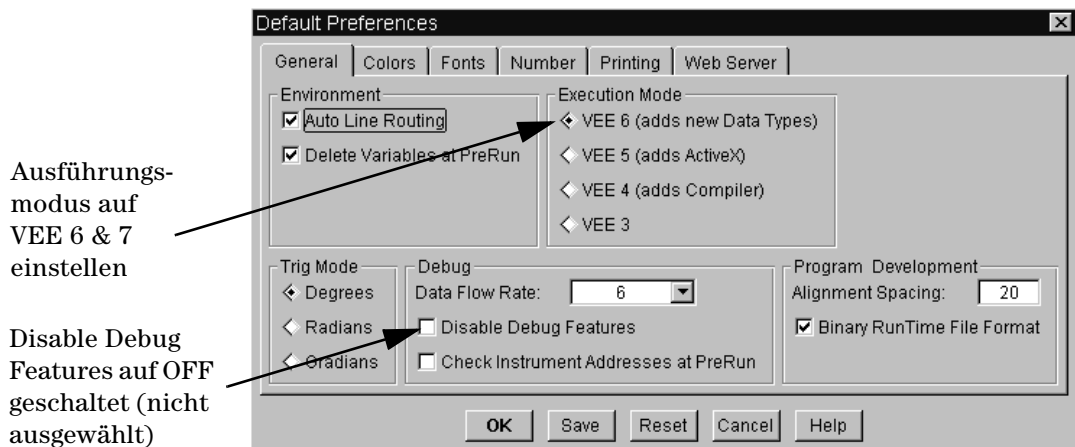
- 1 Wählen Sie im Hauptmenü von VEE **File** ⇒ **Default Preferences** oder klicken Sie in der Symbolleiste auf die Schaltfläche Default Preferences (siehe Abbildung).



↑  
Schaltfläche Default Preferences

### Die Schaltfläche Default Preferences in der Symbolleiste

- 2 Wählen Sie auf der Registerkarte General (wird als oberste Registerkarte bereits angezeigt) unter Execution Mode das Optionsfeld VEE 6 & 7 (siehe Abbildung 262). Vergewissern Sie sich auf der gleichen Registerkarte, dass das Kontrollkästchen Disable Debug Features *nicht* gewählt ist. Klicken Sie auf **OK**.



**Abbildung 262** Den Ausführungsmodus im Dialogfeld Default Preferences ändern



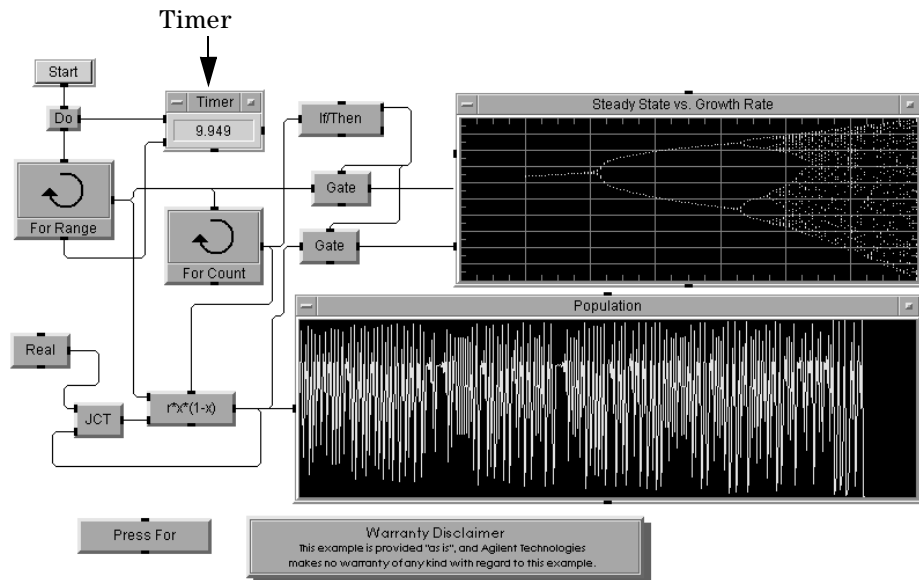
## Die Auswirkung der Änderung des Ausführungsmodus

Das folgende Beispiel zeigt die Steigerung der Geschwindigkeit bei der Aktualisierung eines Programms. Bei diesem Beispiel steht die Geschwindigkeit des Programms ohne die E/A-Geschwindigkeit des Geräts im Vordergrund.

- 1 Öffnen Sie das Programm **chaos.vee** im Unterverzeichnis **examples\Applications**.

Dieses Programm illustriert das explosionsartige Bevölkerungswachstum. Sie können das Programm wie hier gezeigt ändern, indem Sie die Ergebnisse mit einem Timer-Objekt prüfen. Diese Beispiele wurden auf einem 300MHz Pavillion PC unter Windows 98 ausgeführt, wobei zwei weitere große Anwendungen gleichzeitig aktiv waren.

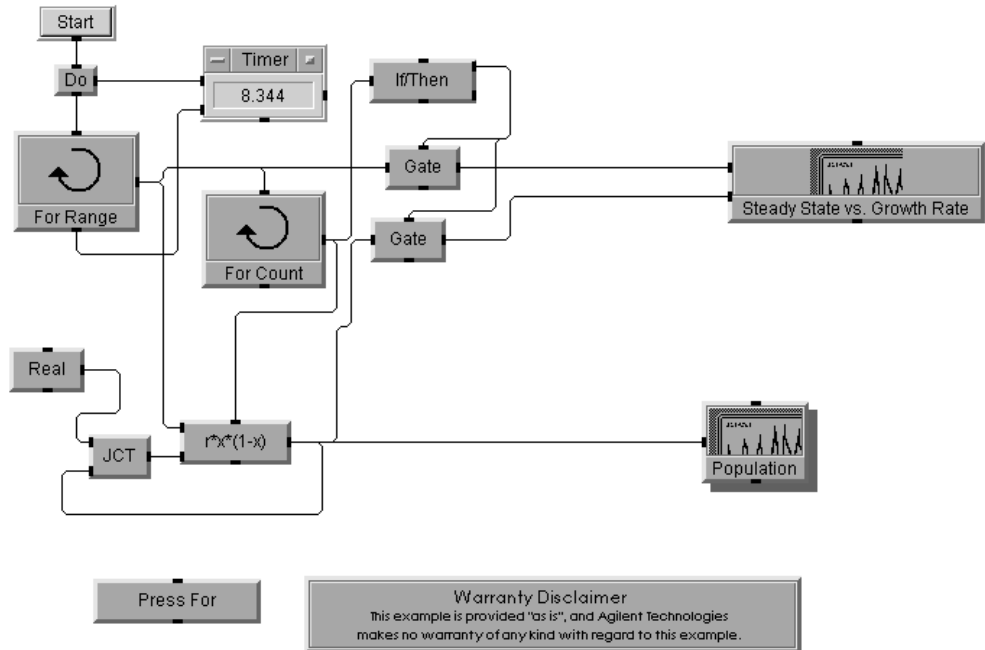
In Abbildung 263 erfolgt das Timing der Programmausführung mit den im VEE 3-Ausführungsmodus geöffneten Anzeigen.



**Abbildung 263** Chaos.vee im VEE 3-Modus mit geöffneten Anzeigen

## 12 Agilent VEE-Programme optimieren

In Abbildung 264 sind die Anzeigen minimiert dargestellt, um die Geschwindigkeit zu verbessern, ohne den Compiler einzuschalten. Die Ausführungszeit wird dadurch um ca.  $\frac{1}{6}$  verkürzt.

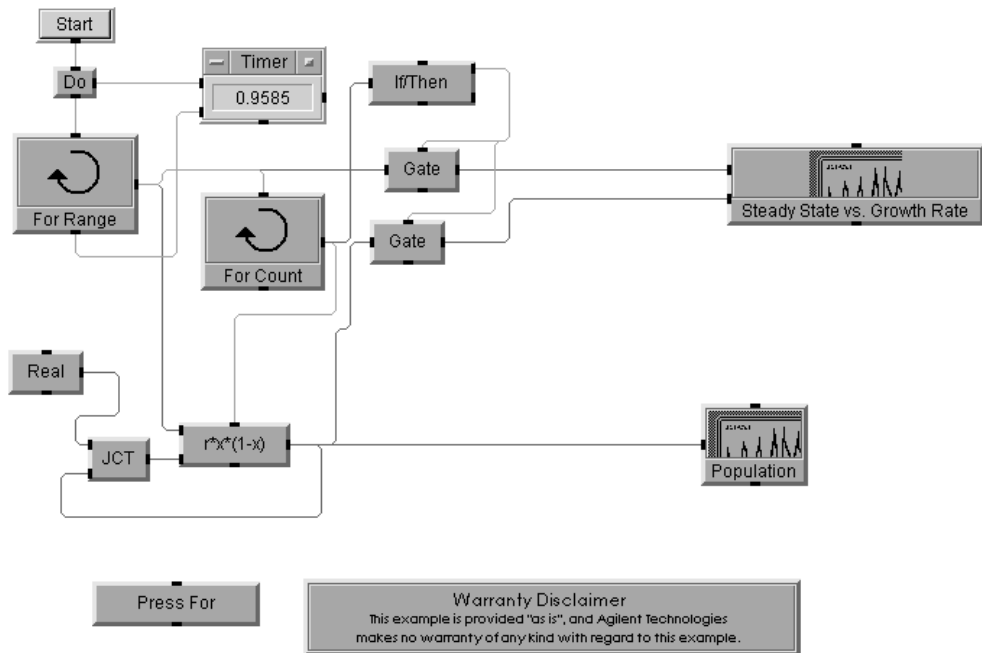


**Abbildung 264** Chaos.vee im VEE 3-Modus mit geschlossenen Anzeigen

In Abbildung 265 schließlich wurden der Compiler eingeschaltet und die Fehlerbehebungsfunktionen („debugging“) ausgeschaltet. Für eine optimale Leistung wählen Sie das Kontrollkästchen **Disable Debug Features** unter **File**  $\Rightarrow$  **Default Preferences**, wenn das Programm bereits bereinigt wurde und einsatzbereit ist.

Die Funktionen zur Fehlerbehebung aktivieren bestimmte Tools einschließlich Show Execution Flow und Activate Breakpoints. Wenn Sie **Disable Debug Features** wählen, werden dadurch die Größe (im Speicher) und die Geschwindigkeit des Programms verbessert. Wie Sie sehen, läuft das Programm jetzt ca. 12 mal

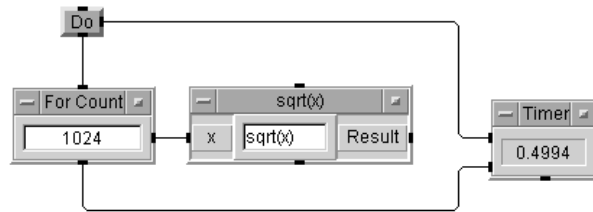
schneller ab. Diese drei Zahlen zeigen, wie Sie durch Kombinieren der Optimierungstechniken mit dem Compiler die beste Ausführungsgeschwindigkeit für Ihr Programm erzielen.



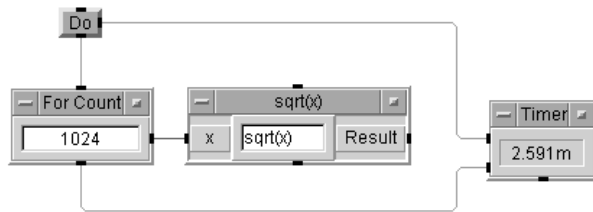
**Abbildung 265** Chaos.vee in VEE 4 oder einem höheren Modus mit ausgeschalteter Fehlerbehebung

In Abbildung 266 und Abbildung 267 verwendet VEE den Compiler zur Verbesserung der Geschwindigkeit in Bereichen des Programms, in denen iterative skalare mathematische Routinen zum Einsatz kommen. Das Beispiel berechnet die Quadratwurzel eines skalaren Werts. (Das Ergebnis wird nicht gespeichert.) Durch die Verwendung des Compilers ist die Geschwindigkeit ca. 107 mal höher als bei Verwendung des VEE 3-Ausführungsmodus.

## 12 Agilent VEE-Programme optimieren



**Abbildung 266** Iteratives mathematisches Beispiel im VEE 3-Modus



**Abbildung 267** Iteratives mathematisches Beispiel mit VEE 4- oder einem höheren Modus

VEE stellt die Ausführungsmodi zur Verfügung, weil es in älteren Versionen von VEE einige Programmieroptionen gab, die in der aktuellen Version nicht zulässig sind (sie erzeugen jetzt eine Fehlermeldung). Außerdem haben die Erweiterungen bei der ActiveX-Automatisierung und den Steuerelementen dazu geführt, dass einige Programme, die in VEE 4 oder VEE 5 ausgeführt werden konnten, für die Ausführung im VEE 6-Modus einige kleine Änderungen erfordern. Ausführliche Informationen zu den Unterschieden zwischen den verschiedenen Ausführungsmodi finden Sie im Handbuch *VEE Pro Advanced Techniques*. Bei allen neuen Programmen sollten Sie im Modus VEE 6 & 7 beginnen.

## Der Agilent VEE Profiler

Der Profiler ist eine Funktion im Professional Development Environment in VEE. Der Profiler erleichtert die Optimierung von Programmen durch die Anzeige der Ausführungsgeschwindigkeiten von UserFunctions oder UserObjects im Programm.

Sie können mit dem Profiler die langsamen Stellen in einem Programm ermitteln und die in diesem Kapitel beschriebenen Techniken anwenden, um die Geschwindigkeit des Programms zu verbessern. Abbildung 268 zeigt das Programm **examples\Applications\mfgtest.vee**.

Zum Einschalten des Profiler wählen Sie **View** ⇒ **Profiler**. Führen Sie anschließend das Programm aus. Sie sehen den Profiler in der unteren Hälfte des Bildschirms. Der Profiler listet vergleichende Informationen zu der erforderlichen Ausführungsdauer der einzelnen UserObjects und UserFunctions auf.

The screenshot shows the VEE Professional Development Environment interface. A flowchart on the left side of the window includes components like 'Until Break', 'Call Login', 'Call selectActions', 'Which tests?', 'Display panels', and 'For Count'. A 'Text' object is also visible. In the bottom right, the 'Profiler' window is open, displaying a table with the following data:

Name	# Executions	% Time	Time(sec)
Login/Login	2	45.28	15.430
Main	2	33.34	11.362
selectActions	1	15.02	5.118
strMatch	1	3.65	1.243
doneYet	3	2.31	0.788
logFunc	3	0.23	0.077
Display panels	1	0.13	0.046
hookUp	1	0.01	0.004
scalarResults	2	0.01	0.004
...	...	...	...

Additional controls in the Profiler window include 'Stop Profiling', 'Refresh', 'Clear', 'Save', 'Options...', and 'Help' buttons.

Abbildung 268 Ein Beispiel zum Profiler

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich ggf. die entsprechenden Themen noch einmal an.

- Drei grundlegende Techniken zur Optimierung von VEE-Programmen mit Beispielen erläutern.
- Mindestens zwei weitere Techniken erläutern.
- Das Grundkonzept einer DLL erläutern.
- Eine DLL importieren, eine darin enthaltene Funktion aufrufen und die DLL löschen.
- Ein Programm im Ausführungsmodus VEE 6 & 7 oder den Ausführungsmodi VEE 5, VEE 4 oder VEE3 schrittweise ausführen und die Gründe zur Verwendung der verschiedenen Modi erläutern.
- Den VEE Profiler verwenden.

# 13

## Plattformspezifische Faktoren und Web-Überwachung

Überblick 493

Der aufrufbare VEE ActiveX-Automatisierungs-Server 494

Technologien zur Web-Aktivierung 495

Web-Überwachung mit Agilent VEE 499

Kapitel-Checkliste 516

### Plattformspezifische Faktoren und Web-Überwachung

*In diesem Kapitel finden Sie Informationen zu folgenden Themen:*

- VEE-Funktionen aus anderen Anwendungen mit dem VEE ActiveX-Automatisierungs-Server aufrufen
- Web-Überwachung

*Erforderliche Zeit für dieses Kapitel: 2 Stunden*



## Überblick

Anschließend erfahren Sie mehr über die wichtigsten Techniken zum Einbinden von VEE-Funktionen in andere Anwendungen und Programme mit dem aufrufbaren VEE ActiveX-Automatisierungs-Server. Zum Abschluss lernen Sie die Konzepte zur Web-Überwachung kennen.

### Der aufrufbare VEE ActiveX-Automatisierungs-Server

In VEE unter Windows 98, Windows NT 4.0, Windows 2000 und Windows XP können Sie VEE-Objekte in andere kommerziell verfügbare oder eigene Testsysteme, die in Standardsprachen geschrieben wurden, integrieren, beispielsweise in MS Visual Basic oder C. Der aufrufbare VEE ActiveX-Automatisierungs-Server kapselt VEE UserFunctions zur Integration in Automatisierungsanwendungen ein, beispielsweise in MS Excel und MS Visual Basic. Der aufrufbare VEE ActiveX-Automatisierungs-Server implementiert eine sprachabhängige Automatisierungsschnittstelle. Sie kann von jeder Anwendung oder Sprache verwendet werden, die den Aufruf eines Automatisierungs-Servers unterstützt.

#### HINWEIS

VEE kann die ActiveX-Automatisierung und -Steuerelemente zum Steuern anderer Anwendungen von VEE aus verwenden. Der aufrufbare VEE ActiveX-Automatisierungs-Server ermöglicht anderen Anwendungen wie MS Visual Basic die Steuerung von VEE.

---

Der aufrufbare VEE ActiveX-Automatisierungs-Server kommuniziert mit seinen Clients über Eigenschaften und Methoden. Die Umgebung, von der aus Sie den aufrufbaren VEE ActiveX-Automatisierungs-Server aufrufen (z. B. Visual Basic oder C) legt fest, wie dieser Aufruf erfolgt. Der aufrufbare VEE ActiveX-Automatisierungs-Server bietet mit der Umgebung, von der aus Sie ihn aufrufen, eine umfangreiche Online-Hilfe. Weitere Informationen finden Sie in der Online-Hilfe in VEE, dem *VEE Pro Advanced Techniques* und den zusammen mit VEE ausgelieferten Beispielen.

#### HINWEIS

Der aufrufbare VEE ActiveX-Automatisierungs-Server ersetzt das aufrufbare VEE ActiveX-Steuerelement, das mit Version 5.0 ausgeliefert wurde.

---

## Technologien zur Web-Aktivierung

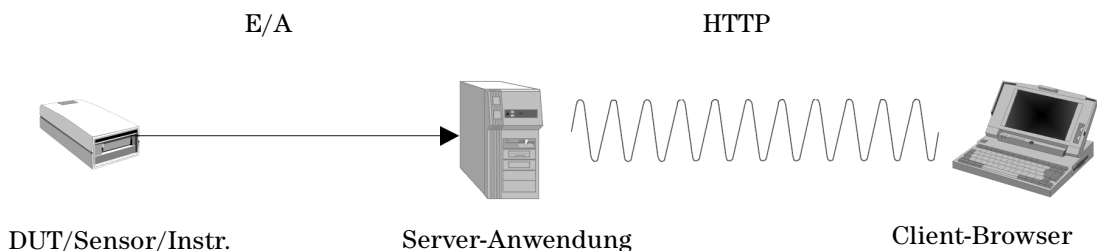
Sie können mit VEE Daten verbreiten, die sie in Programmen erfasst haben, um Testsysteme zu überwachen oder um Testergebnisse von einem entfernten Ort aus zu überprüfen. In diesem Abschnitt werden Technologien zur Web-Aktivierung für Test- und Messanwendungen beschrieben. Außerdem finden Sie hier Informationen dazu, wie VEE diese Technologien unterstützt.

### Überblick über Web-Technologien

Dieses Beispiel beschreibt, wie eine organisationsinterne Web-Site (eine „Intranet-Web-Site“ erstellt wird, die als Server Referenzdaten zur Verfügung stellt. Dabei wird davon ausgegangen, dass die Benutzer, die auf diese Site zugreifen, einen Browser haben. Das Beispiel verwendet eine Microsoft-Umgebung (Windows 98, Windows NT 4.0, Windows 2000 oder Windows XP), MS Office und MS Internet Explorer. Die Kommunikation erfolgt von dem Instrument an den Server und anschließend an den Client-Browser (siehe Abbildung 269).

#### HINWEIS

In diesem Beispiel werden Bildschirmabzüge von einem PC verwendet.



**Abbildung 269** Modell der Web-Messungsanwendung

In Abbildung 269 läuft die Kommunikation wie folgt ab:

- Das *Testgerät* („*Device Under Test, DUT*“) bzw. *der Sensor oder das Instrument* sendet Informationen über die Netzwerk-E/A-Schicht an die Server-Anwendung. Die Netzwerk-E/A-Schicht umfasst die Schnittstelle und die Backplane (GPIB, RS232, VXI, MXI, PC Plug-in) und die E/A-Programmierungsschicht wie Treiber, VISA oder SIDL.
- Die *Server-Anwendung* ist das VEE-Programm, das die Messdaten generiert.
- Das *HTTP* (*HyperText Transfer Protocol*) sendet die Informationen an den Client-Browser.

HTTP ist ein Protokoll innerhalb des vom Web verwendeten TCP-Kommunikationsprotokolls. (TCP/IP steht für Transmission Control Protocol/Internet Protocol.) Niedrigere Stufen des TCP/IP-Protokolls umfassen die Transport-, Netzwerk- und die physische Schicht der Kommunikation. Per Definition ist jede TCP/IP-Anwendung eine Client/Server-Anwendung. Der Client-Browser (z. B. Internet Explorer) kann Informationen anfordern, die von der Server-Anwendung generiert wurden.

Wenn der folgende Universal Resource Locator (URL) in Internet Explorer eingegeben wird, werden Informationen von einem Agilent-Server abgerufen:

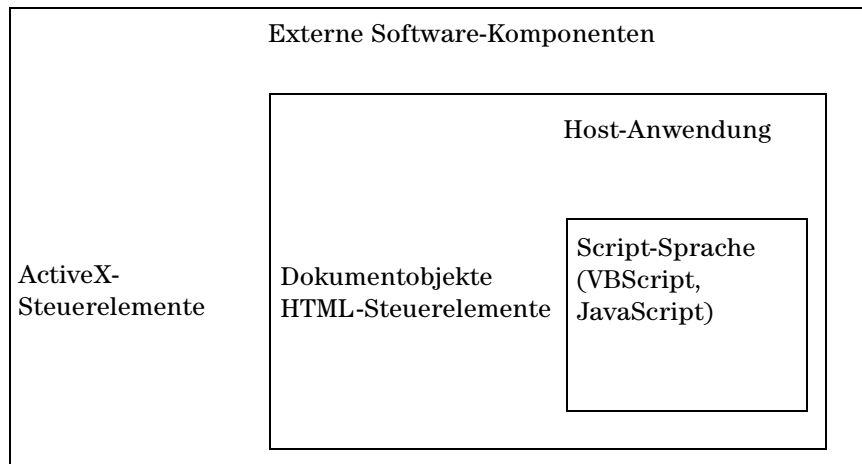
<http://www.agilent.com/find/vee>

- `http` beschreibt den Typ der Ressource, die zum Übertragen der Informationen aufgerufen wurde.
- **`www.agilent.com/find/vee`** ist die URL für die Ressource. Das von HTTP verwendete Hypertext-Format ist ein Script-Format, das als HyperText Markup Language (HTML) bezeichnet wird. HTML bietet eine Möglichkeit, Dokumente miteinander zu verbinden; ursprünglich war dies die einzige Sprache, mit der Programmierer eine Web-Site erstellen konnten. Ursprünglich nur für Textdaten entwickelt, umfasst HTML heute auch Sound-, Video- und Bilddaten, interaktive Bildschirme, ActiveX-Steuerelemente und Java-Applets.

Wenn der Browser eine Anforderung nach Server-Informationen absetzt, wird er nicht automatisch aktualisiert, es sei denn, der Browser ist entsprechend konzipiert. Außerdem ist keine

Interaktion durch den Browser möglich, es sei denn, diese Interaktion ist in der Browser-Seite vorgesehen. Die einfachste Möglichkeit hierzu ist über eine Script-Sprache wie beispielsweise VBScript, JavaScript oder Jscript.

Die Script-Sprache ist eine interpretierte Sprache, die vom Browser unterstützt wird. Die Script-Sprache kann die Einschränkungen von HTML umgehen, um eine Web-Seite mit stärker interaktiven Eigenschaften bereitzustellen. Da sie interpretiert werden, müssen Script-Sprachen in die Web-Seite integriert und vom Browser unterstützt werden. Es handelt sich dabei nicht um unabhängige Programme. Dies wird in Abbildung 270 grafisch dargestellt.



**Abbildung 270** Ein Host-Modell zur Script-Sprache

VBScript, JavaScript und JScript sind Script-Sprachen. VBScript basiert auf MS Visual Basic. JavaScript wurde von (Java) mitentwickelt. JScript basiert auf der Microsoft Version von JavaScript.

Die Script-Sprachen müssen in einer Host-Anwendung vorhanden sein. Die Host-Anwendung ist normalerweise ein Web-Browser wie Internet Explorer oder Netscape Navigator. Innerhalb des Browsers wird das Aussehen und Verhalten der Web-Seite über HTML gesteuert.

## 13 Plattformspezifische Faktoren und Web-Überwachung

Das Component Object Model (COM) von Microsoft definiert kompilierte Software-Komponenten (ActiveX-Steuerelemente), die speziell konzipierte Funktionen einkapseln. Normalerweise wird ein ActiveX-Steuerelement verwendet, um die Funktionalität einer Benutzerschnittstelle bereitzustellen; es wird auf dem Client-Computer ausgeführt. ActiveX-Steuerelemente sind optimierte Software-Komponenten; sie wurden früher auch als OLE-Steuerelemente bezeichnet.

## Web-Überwachung mit Agilent VEE

VEE enthält einen integrierten Web-Server, der über HTTP mit anderen Programmen kommunizieren kann. Sie können entfernten Benutzern Zugriff auf VEE-Programme und Informationen auf Ihrem Computer erteilen. In diesem Abschnitt wird beschrieben, wie Sie VEE-Daten für die gemeinsame Nutzung freigeben und wie ein entfernter Benutzer auf diese Daten zugreift.

### Allgemeine Richtlinien und Tipps

- Das auf Ihrem System ausgeführte VEE-Programm ist das VEE-Programm, auf das der entfernte Benutzer zugreift.
- Auf Ihrem System muss VEE aktiv sein. Der entfernte Benutzer braucht VEE nicht installiert zu haben, um auf ein VEE-Programm auf Ihrem System zugreifen zu können.
- Wenn der entfernte Benutzer eine Anforderung von seinem Netzwerk-Browser an VEE sendet, erstellt VEE ein Bild zur Anzeige im Browser-Fenster des entfernten Benutzers. Dieses Bild ist eine „Momentaufnahme“ des VEE-Programms; es kann nicht geändert werden.
- Durch das Auswählen von Optionen in der VEE Web-Server-Homepage oder durch Eingeben von Befehlszeilenoptionen in der Browser-URL kann ein Benutzer verschiedene Teile Ihres VEE-Programms anzeigen, das VEE-Programm die Browser-Anzeige in regelmäßigen Abständen aktualisieren lassen (zur Überwachung des Programmfortschritts) und Informationen zu Fehlermeldungen anzeigen.

### Einem entfernten Benutzer Agilent VEE-Daten bereit stellen

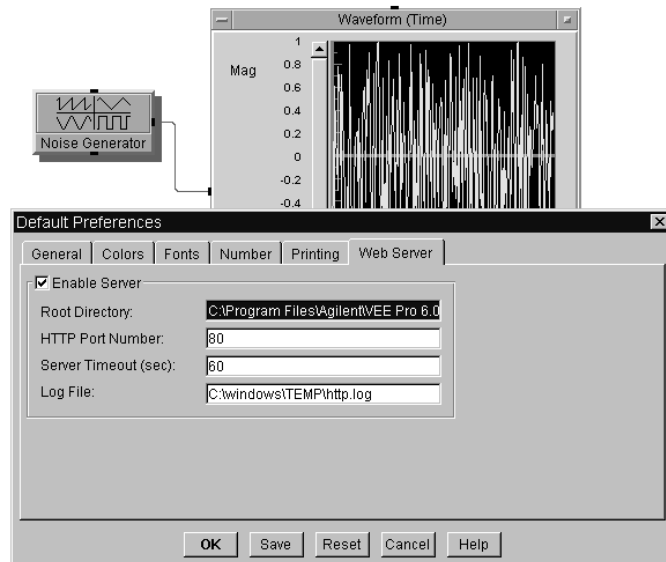
Zum Einrichten des VEE Web-Servers, sodass entfernte Benutzer auf Daten auf ihrem System zugreifen können, führen Sie die folgenden allgemeinen Schritte aus:

- 1 Überprüfen Sie, ob Ihr System mit dem Netzwerk verbunden ist.
- 2 Stellen Sie dem entfernten Benutzer Informationen über die URL zur Verfügung; diese URL muss der Benutzer in seinem Browser eingeben, um auf Ihr System zuzugreifen (dies wird später ausführlich beschrieben).
- 3 Starten Sie VEE und öffnen Sie das Programm, das der entfernte Benutzer aufrufen soll bzw. erstellen Sie alle Dateien, die Sie dem entfernten Benutzer zur Verfügung stellen wollen.
- 4 Aktivieren Sie den Web-Server durch Auswählen der Einstellungen im Dialogfeld **File** ⇒ **Default Preferences** ⇒ **Web Server** ; diese Einstellungen werden weiter unten ausführlich beschrieben.
- 5 Der entfernte Benutzer muss jetzt einen Web-Browser wie Internet Explorer oder Netscape aufrufen.



## Die Registerkarte Web Server

Wenn Sie **File** ⇒ **Default Preferences** ⇒ **Web Server** wählen, wird das Dialogfeld Default Preferences geöffnet und die Registerkarte Web Server angezeigt (siehe Abbildung 271).



**Abbildung 271** Das Dialogfeld Default Preferences-Web Server

## 13 Plattformspezifische Faktoren und Web-Überwachung

Die Felder der Registerkarte Web Server sind:

**Tabelle 50** Die Felder der Registerkarte Web Server

Feldbezeichnung	Beschreibung
Enable Server (Server aktivieren)	<p>Wenn diese Option ausgewählt ist, schaltet Enable Server den integrierten VEE-Web-Server ein. Der Web-Server ermöglicht einem entfernten Benutzer das Anzeigen, Überwachen und Beheben von Fehlern mit VEE-Programmen auf Ihrem System durch Verwenden eines Web-Browser zum Anzeigen Ihres VEE-Programms.</p> <p>Der VEE Web-Server verwendet das Standard-HTTP-Protokoll. Standardmäßig ist Enable Server auf OFF (nicht markiert) eingestellt. Ihr System muss mit einem Netzwerk verbunden sein, VEE muss aktiv sein, und der entfernte Benutzer muss einen Web-Browser aufgerufen haben, damit er auf den VEE Web-Server zugreifen kann.</p>
Root Directory (Root-Verzeichnis)	<p>Gibt die Position der Dateien an, die vom entfernten Benutzer aufgerufen werden können. Der Standardwert lautet <code>~installDir/www</code>. Wenn VEE Pro 7.0 gemäß der Standardvorgaben installiert wurde, enthält das Feld für Windows 98, Windows NT 4.0, Windows 2000 oder Windows XP folgenden Eintrag: <b>C:\Programme\Agilent\VEE Pro 7.0\www.</b></p> <ul style="list-style-type: none"><li>• Legen Sie <i>keine</i> privaten Dateien in diesem Verzeichnis oder seinen Unterverzeichnissen ab, da sie von allen entfernten Benutzern, die mit einem Web-Browser Zugriff auf Ihr System haben, eingesehen werden können.</li><li>• VEE installiert bei der Installation von VEE die Datei <b>index.html</b> im <b>Root Directory</b> des Standard-Web-Servers. Standardmäßig befindet sich die Datei im Verzeichnis <b>C:\Programme\Agilent\VEE Pro 7.0\www\index.html</b>. Dies entspricht der Standard-Homepage des VEE-Web-Servers; sie kann beim Zugriff auf Ihr System durch einen entfernten Benutzer angezeigt werden. (Sie können die Homepage an Ihre Anforderungen anpassen.)</li><li>• Wenn Sie das Root-Verzeichnis für Web-Dateien ändern, verschieben Sie die Datei <b>index.html</b> in das neue Verzeichnis und ändern Sie ihren Namen nicht.</li></ul>

**Tabelle 50** Die Felder der Registerkarte Web Server

Feldbezeichnung	Beschreibung
HTTP Port Number (HTTP-Anschlussnummer)	<p>Gibt die Anschlussnummer für den VEE Web-Server an. Die Standardanschlussnummer lautet 80. Sie brauchen die Anschlussnummer nur zu ändern, wenn ein anderer Web-Server, beispielsweise ein anderes Exemplar von VEE, auf dem gleichen System aktiv ist, oder wenn Sie den Fernzugriff auf Ihren VEE Web-Server beschränken wollen.</p> <ul style="list-style-type: none"> <li>• Durch die Angabe eines HTTP-Anschlusses zwischen 0 und 65535 können Sie begrenzen, wer Zugriff auf Ihr System hat und Daten anzeigen darf.</li> <li>• Wenn Sie eine andere Anschlussnummer eingeben, muss der entfernte Benutzer, der das VEE-Programm anzeigen will, die gleiche Anschlussnummer in seinem Browser angeben. Wenn Sie das Feld beispielsweise auf 85 setzen, muss der entfernte Benutzer den URL als <b>http://hostname:85</b> eingeben. (Weitere Informationen zur Definition des <i>hostname</i> finden Sie weiter hinten in diesem Kapitel.)</li> </ul>
Server Timeout (Server-Zeit-überschreitung)	<p>Gibt die maximale Zeit an, die der VEE Web-Server darauf wartet, dass VEE Befehle verarbeitet. Standardmäßig ist dieses Feld auf 60 Sekunden gesetzt. Wenn das VEE-Programm zur Verarbeitung eines Befehls länger als die angegebene Zeit braucht, sendet VEE eine Timeout-Signal an den Web-Server.</p>
Log File (Protokolldatei)	<p>Gibt die Protokolldatei an, in der alle vom Web-Server verarbeiteten ankommenden HTTP-Anforderungen aufgezeichnet werden. Die Standardposition für diese Datei auf einem PC ist <b>C:\windows\TEMP\http.log</b>. Wenn die Protokolldatei nicht bereits vorhanden ist, wird sie erstellt, wenn Sie im Dialogfeld <b>Default Preferences</b> ⇒ <b>Web Server</b> Änderungen vornehmen.</p>
Save (Speichern)	<p>Diese Werte stellen die Standardwerte für zukünftige VEE Web-Sessions dar, bis Sie die erneut ändern. Wenn Sie die angegebenen Werte <i>nur</i> für die aktuelle VEE-Session verwenden wollen, klicken Sie auf OK.</p>

### Wie ein entfernter Benutzer auf Agilent VEE auf Ihrem System zugreift

Für den Zugriff auf VEE-Dateien auf Ihrem System von einem anderen Standort aus muss der entfernte Benutzer die folgenden allgemeinen Schritte durchführen:

- 1 Das System des entfernten Benutzers muss mit einem Netzwerk verbunden sein.
- 2 Der entfernte Benutzer muss einen Netzwerk-Browser wie beispielsweise Netscape oder Internet Explorer aktivieren.
- 3 Der entfernte Benutzer muss eine URL-Adresse für Ihr System eingeben. (Sie müssen diese Adresse dem entfernten Benutzer mitteilen; dies wird weiter unten ausführlich erläutert.)

#### HINWEIS

Der entfernte Benutzer braucht VEE nicht zu aktivieren und muss VEE auch nicht installiert haben, um auf VEE-Programme auf Ihrem System zugreifen zu können.

Der entfernte Benutzer stellt eine Verbindung zum Netzwerk her, startet den Browser und gibt die URL-Adresse ein. Sie teilen dem entfernten Benutzer die URL-Adresse entsprechend den folgenden Angaben mit. Der entfernte Benutzer kann diese Adresse in einem der folgenden Formate eingeben:

`http://hostname {:port}`

Zeigt die Homepage Ihres VEE Web-Servers an, in der der entfernte Benutzer Optionen zur Anzeige Ihres VEE-Programms auswählen kann.

`http://hostname{:port}/{Befehl}{?Parameter}`

Gibt eine Ansicht oder UserFunction an, die der entfernte Benutzer in dem VEE-Program anzeigen will.

**http://hostname/ViewMainDetail** zeigt beispielsweise die Detailansicht Ihres VEE-Hauptprogramms an.

`http://hostname{:port}/{Datei}`

Gibt eine von Ihnen gespeicherte Datei an, z. B, eine \*.jpeg- oder \*.html-Datei, die dem entfernten Benutzer angezeigt werden soll.

**HINWEIS**

Die Felder in geschweiften Klammern {} sind optionale Felder.

Die Felder in den URL-Adressen haben folgende Bedeutung:

Folgende Befehle und Parameter können in URLs verwendet werden, wenn ein entfernter Benutzer auf ein VEE-Programm auf Ihrem System zugreift.

**Tabelle 51** URL-Befehle und -Parameter

Feldbezeichnung	Beschreibung
Hostname	<p>(Erforderlich) Kennzeichnet Ihr System, auf dem VEE aktiv ist, im Format <code>&lt;computer_name&gt;.domain.com</code>.</p> <p>Sie können festlegen, dass der Benutzer lediglich den <code>&lt;hostname&gt;</code> in einer URL-Adresse anzugeben braucht, z. B. <code>http://&lt;hostname&gt;</code>. Dieser Befehl öffnet die Homepage Ihres VEE-Web-Servers <b>index.html</b> und zeigt sie dem entfernten Benutzer an. Der entfernte Benutzer kann in der Homepage Ihres VEE Web-Servers Optionen auswählen zur Anzeige, zur Überwachung oder zur Fehlerbehebung Ihres VEE-Programms.</p>
Port (Anschluss)	<p>(Optional) Kennzeichnet die Anschlussnummer des Web-Servers, wenn nicht der Standardwert 80 verwendet werden soll. Geben Sie diesen Wert nur an, wenn Sie unter <b>File</b> ⇒ <b>Default Preferences</b> ⇒ <b>Web Server</b> eine andere Nummer als 80 eingegeben haben.</p> <p>Wenn der entfernte Benutzer beispielsweise auf VEE auf Ihrem System zugreift und die Anschlussnummer auf Ihrem System unter <b>File</b> ⇒ <b>Default Preference</b> ⇒ <b>Web Server</b> auf 85 gesetzt ist, muss der entfernte Benutzer <code>http://&lt;hostname&gt;:85</code> eingeben.</p>

## 13 Plattformspezifische Faktoren und Web-Überwachung

**Tabelle 51** URL-Befehle und -Parameter

Feldbezeichnung	Beschreibung
File	<p>(Optional) Kennzeichnet ein Verzeichnis und/oder eine Datei relativ zu dem Root-Verzeichnis, das von dem Browser geöffnet wird. Sie geben nur eine Datei an, wenn Sie eine Datei wie beispielsweise eine *.jpeg- oder *.html-Datei gespeichert haben, die dem Benutzer angezeigt werden soll.</p> <p>Damit die entfernten Benutzer eine Datei von Ihrem System anzeigen können, müssen Sie das Verzeichnis als <b>Root-Verzeichnis</b> im Dialogfeld <b>Web Server</b> unter <b>Default Preferences</b> in VEE angeben.</p>
Commands and Parameters (Befehle und Parameter)	<p>(Optional) Gibt einen Befehl oder einen für einen Befehl erforderlichen Parameter an, der vom VEE Web-Server unterstützt wird. Befehle und Parameter geben einem entfernten Benutzer die Möglichkeit, ein entferntes VEE-Programm über den Browser zu überwachen und Fehler zu beheben. Nachfolgend finden Sie eine Liste der Befehle und Parameter.</p>
Anzeigen des gesamten VEE-Fensters	<p>ViewVEE Beispiel: <code>http://hostname/ViewVEE</code></p>
Fensteransicht des VEE-Hauptprogramms	<p>ViewMainPanel Beispiel: <code>http://hostname/ViewMainPanel</code></p>
Detailansicht des VEE-Hauptprogramms	<p>ViewMainDetail Beispiel: <code>http://hostname/ViewMainDetail</code></p>
VEE-Ausführungsfenster zur Ausführungszeit	<p>ViewExecWindow Beispiel: <code>http://hostname/ViewExecWindow</code></p>

**Tabelle 51** URL-Befehle und -Parameter

Feldbezeichnung	Beschreibung
VEE UserFunction in der Fensteransicht	<p><code>http://hostname/ViewPanel?&lt;UserFunctionName&gt;</code></p> <p>Wenn die <b>UserFunction</b> beispielsweise <b>AddNoise</b> lautet, gibt der entfernte Benutzer ein:  <code>http://hostname/ViewPanel? AddNoise</code></p>
Detailansicht einer UserFunction	<p><code>http://hostname/ViewDetail?UserFunctionName</code></p> <p>Wenn die UserFunction beispielsweise <b>AddNoise</b> lautet, gibt der entfernte Benutzer ein:  <code>http://hostname/ViewPanel? AddNoise</code></p>
Fehlerfenster zum aktuellen Programm	<code>http://hostname/ViewError</code>
Anzeigen einer Liste der verfügbaren Befehls-URLs.	<code>ViewHelp</code>

## Anzeigen der Homepage des Agilent VEE-Web-Servers

Bei der Installation von VEE wird die Standarddatei `index.html` im Verzeichnis `www` erstellt. Diese Datei enthält die Homepage des VEE-Web-Servers. Sie können entfernten Benutzern die Möglichkeit geben, auf dieser Seite Optionen zur Anzeige Ihres VEE-Programms zu wählen. Die Standard-Homepage des VEE-Web-Servers wird in Abbildung 272 dargestellt. Sie können diese Seite auch beispielsweise in MS Word bearbeiten, um sie an Ihre Anforderungen anzupassen.



### Welcome to the Agilent VEE Web Server Home Page!

You can remotely view a VEE program element by selecting one of the **Monitoring Options** below, and then clicking on *View*.

A screenshot of a web browser displaying the "Monitoring Options" menu. The menu is titled "Monitoring Options" and contains several radio button options. The first option, "VEE Workspace snapshot", is selected. Below it are two options for "VEE Workspace with" and "Execution Window with" followed by input fields for "second updates". Further down are "Last Error Message", "Main Panel", and "Main Detail". At the bottom are two options for "Panel View of UserFunction" and "Detail View of UserFunction", each followed by an input field. A "View" button is located at the bottom right of the menu.

**Abbildung 272** Die Standardseite Index.html

### HINWEIS

Zum Anzeigen dieses Menüs auf Ihrem eigenen System können Sie auf Ihr System als localhost verweisen. Wenn Sie beispielsweise VEE ausführen, den Netzwerk-Browser starten und `http://localhosteingegeben`, zeigt der Browser die Homepage des VEE-Web-Servers in Abbildung 272 an. Dies ist eine einfache Möglichkeit für Sie, zu prüfen, welche Anzeigen der entfernte Benutzer mit den verschiedenen Befehlen erhält.

Wenn der entfernte Benutzer das in Abbildung 272 dargestellte Menü öffnet, kann er auf verschiedene Bereiche eines VEE-Programms zugreifen, indem er in dem Menü Optionen wählt. Der Benutzer kann beispielsweise auf Main Detail klicken, um das VEE-Hauptprogramm in der Detailansicht anzuzeigen. Durch die Auswahl im Menü werden die gleichen Informationen ange-



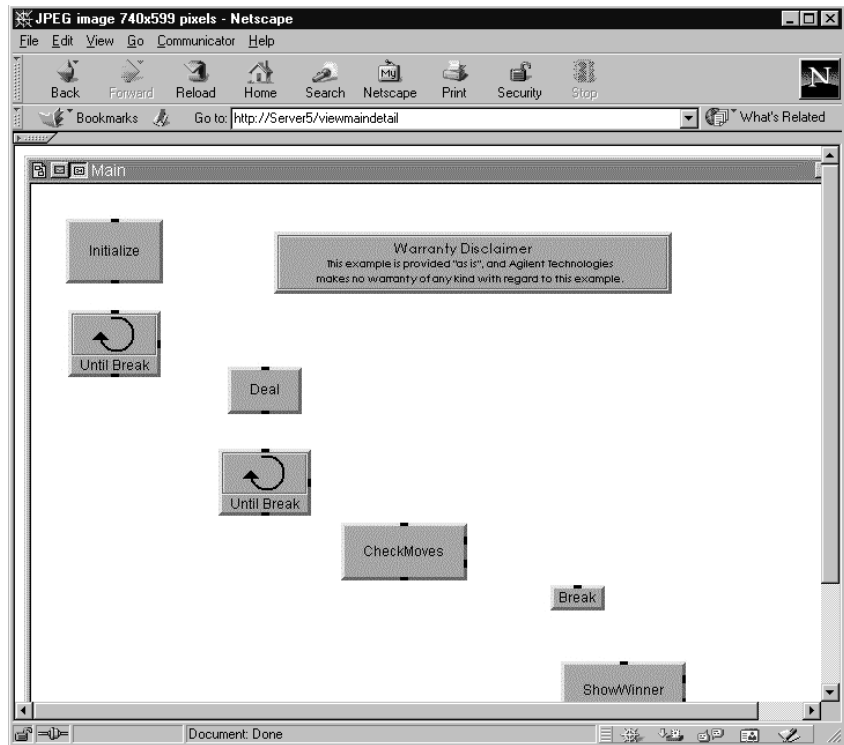
zeigt, die der entfernte Benutzer sieht, wenn er den Befehl `http://hostname/ViewMainDetail` im Netzwerk-Browser eingibt.

### Übung 13-1: Übungs-Session mit Agilent VEE-Web-Browser

Diese Übung simuliert eine Web-Session, in der Sie das Programm **Solitaire.vee** auf Ihrem System für einen entfernten Benutzer bereitstellen. In diesem Fall tritt ein Fehler im Programm auf, und der entfernte Benutzer versucht mit Ihnen zusammen, den Fehler zu beheben.

- 1 Starten Sie VEE. Wählen Sie **File** ⇒ **Default Preferences** ⇒ **Web Server** und klicken Sie auf **Enable Server**. Verwenden Sie die Standardeinstellungen. Öffnen Sie das Programm **Solitaire.vee**, das der entfernte Benutzer anzeigen soll. Starten Sie Ihren Netzwerk-Browser.
- 2 Nehmen Sie Kontakt mit dem entfernten Benutzer auf und teilen Sie ihm mit, dass er Ihr System über das Web mit der Eingabe `http://Server5` erreichen kann. (Statt `http://Server5` würden Sie den Namen Ihres Computers verwenden.)
- 3 Wenn der entfernte Benutzer den URL `http://Server5` eingibt, sieht er die Homepage Ihres VEE-Web-Servers in seinem Browser. (In Abbildung 272 auf Seite 508 ist die Anzeige dargestellt.)
- 4 Der entfernte Benutzer beschließt, zunächst das gesamte VEE-Programm anzuzeigen. In der Homepage Ihres VEE-Web-Servers klickt er auf **Main Detail** ⇒ **View**. Der Browser zeigt die in Abbildung 273 dargestellte Anzeige an.

## 13 Plattformspezifische Faktoren und Web-Überwachung



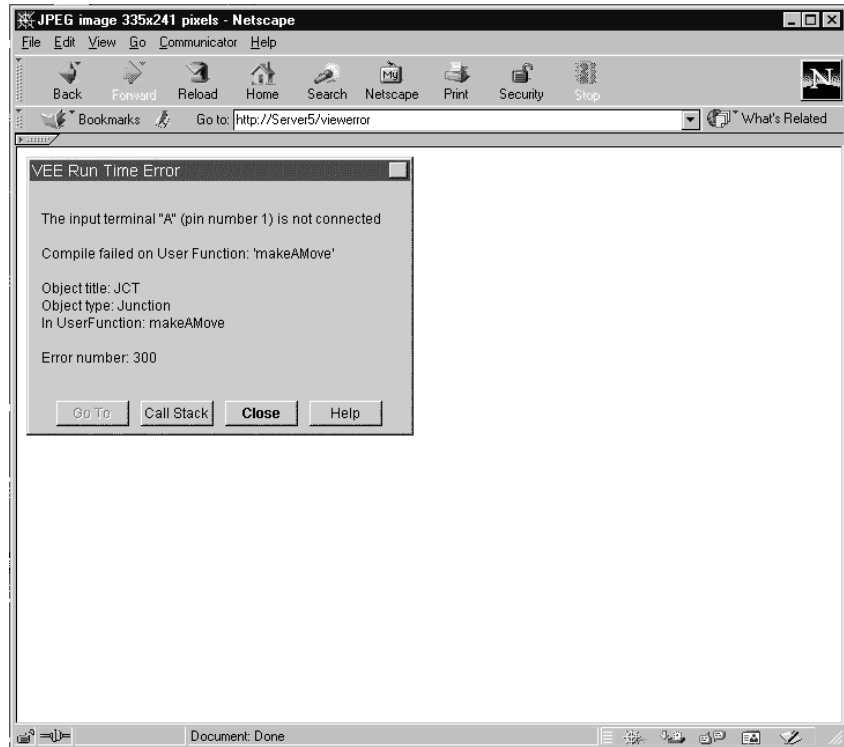
**Abbildung 273** Anzeigen des Hauptprogramms Solitaire.vee im Browser

Abbildung 273 zeigt das Hauptprogramm in VEE an.

### HINWEIS

Für diese Übung enthält das Programm **Solitaire.vee** einen Fehler, der *nicht* im VEE-Programmbeispiel enthalten ist. Falls Sie das Programm anzeigen wollen: es befindet sich in **Help** ⇒ **Open Example...** ⇒ **Games** ⇒ **Solitaire.vee**.

- 5 Der entfernte Benutzer klickt im Browser auf Zurück, um die Homepage des VEE-Web-Servers anzuzeigen, und wählt anschließend Last Error Message. Der Browser zeigt die in Abbildung 274 dargestellte Fehlermeldung an.

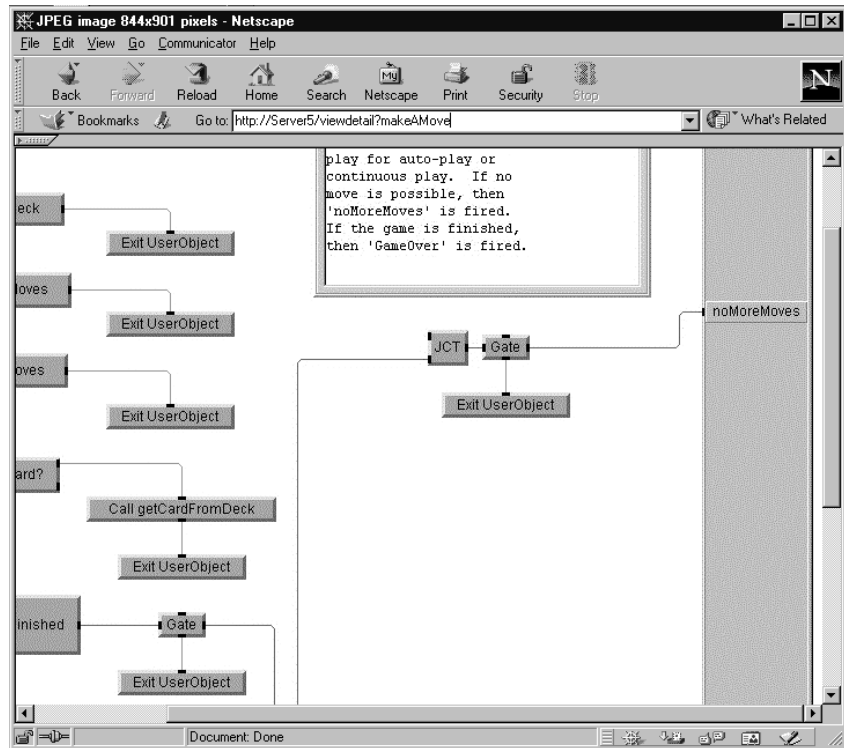


**Abbildung 274** Anzeigen einer VEE-Fehlermeldung mit dem Browser

Beachten Sie, dass die VEE- Fehlermeldung die UserFunction makeAMove angibt.

- 6 Der entfernte Benutzer kehrt wieder zurück zur Homepage des VEE-Web-Servers, klickt auf Detail View of UserFunction und gibt den UserFunction-Namen makeAMove an. Der Browser zeigt die UserFunction makeAMove an (siehe Abbildung 275).

## 13 Plattformspezifische Faktoren und Web-Überwachung



**Abbildung 275** Die Detailansicht einer UserFunction im Browser

Der entfernte Benutzer kann den Fehler im VEE-Programm sehen. Ein Eingangs-Pin am Objekt JCT ist nicht angeschlossen (siehe Abbildung 275). Der entfernte Benutzer könnte Ihnen jetzt bei der Behebung des Fehlers in **Solitaire.vee** helfen und das Problem lösen. Über einen ähnlichen Prozess der Zusammenarbeit über das Web können Sie mit entfernten Benutzern gemeinsam Programme entwickeln.

### Begrenzen des Zugriffs auf Programme über das Web

Wenn Sie ein VEE-Programm über das Web verfügbar machen, wollen Sie eventuell manche Teile des Programms gegen die Blicke der entfernten Benutzer schützen. Wenn die entfernten Benutzer derzeit die URL-Adresse zu Ihrem System kennen,

wollen Sie sicherstellen, dass nur bestimmte entfernte Benutzer auf bestimmte Programme oder Dateien im Web-Verzeichnis zugreifen können.

Wenn Sie verhindern wollen, dass entfernte Benutzer Teile eines VEE-Programms über das Web anzeigen, können Sie das Programm auf drei verschiedene Arten schützen:

Ändern der Anschlussnummer im Ordner **Default Preferences** ⇒ **Web Server**, sodass nur Benutzer mit entsprechender Berechtigung das Programm anzeigen können.

- *ODER* -

Erstellen einer gesicherten Laufzeitversion des VEE-Programms. Auf diese Weise wird sichergestellt, dass der Programmcode nicht angezeigt werden kann. Weitere Informationen hierzu finden Sie unter „Ein Programm sichern (Laufzeitversion erstellen)“ auf Seite 437.

- *ODER* -

Erstellen einer HTML-Datei mit dem Namen des Befehls, den Sie inaktivieren wollen, und Speichern dieses Befehls im VEE-Verzeichnis `www`. Der Browser greift immer auf jede `*.html`-Datei zu, bevor er zu VEE übergeht. Auf diese Weise können Sie Anforderungen von entfernten Benutzern abfangen und eine HTML-Seite mit den entsprechenden Warnungen oder Kommentaren anzeigen.

Sie können beispielsweise verhindern, dass entfernte Benutzer die Detailansicht eines VEE-Programms anzeigen. Sie können eine Datei in einem Programm wie beispielsweise MS Word erstellen und als **ViewMainDetail.html** im Verzeichnis `www` speichern. In der Datei legen Sie die Meldung ab, die der Benutzer sehen soll.

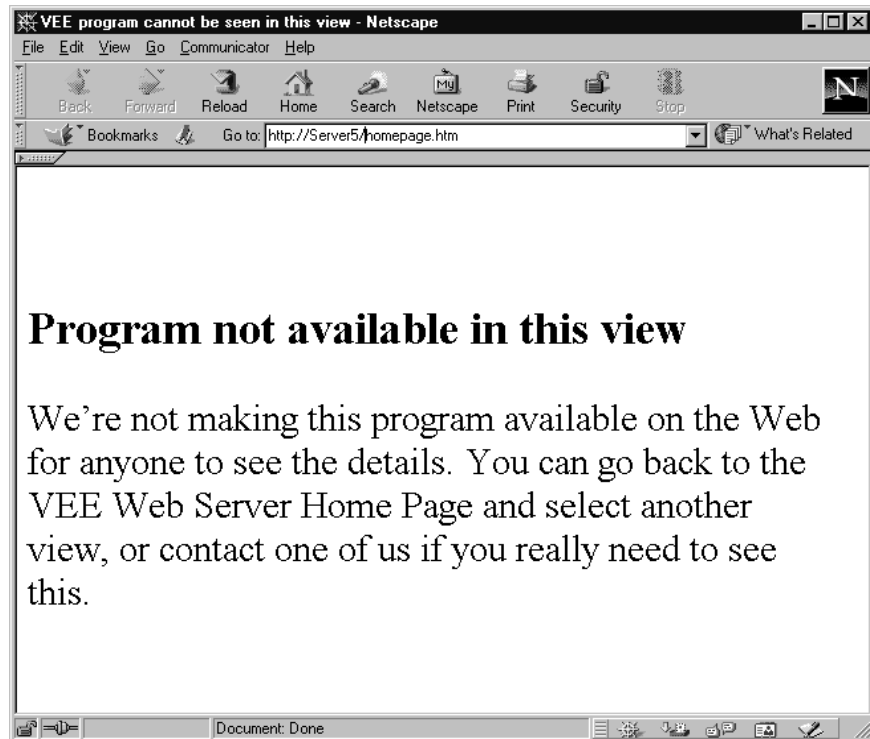
Wenn der entfernte Benutzer Main Detail in der Homepage des VEE-Web-Servers auswählt oder eine URL-Adresse mit der Option ViewMainDetail auswählt, zeigt der Browser nicht die Detailansicht des VEE-Hauptprogramms an. Stattdessen greift der Browser auf die Datei **ViewMainDetail.html** im Verzeichnis

## 13 Plattformspezifische Faktoren und Web-Überwachung

www zu und zeigt die von Ihnen erstellte Datei an. Abbildung 276 zeigt ein Beispiel für eine Meldung, die Sie einem entfernten Benutzer anzeigen könnten.

### HINWEIS

Vergewissern Sie sich, dass der Dateiname der Dateiname eines VEE-Web-Befehls ist und sich in dem Root Directory befindet, das unter Web Server angegeben wurde.



**Abbildung 276** Beispiel für die Anzeige einer HTML-Meldung statt eines VEE-Programms

Sie können auch eine \*.html-Datei für andere Zwecke erstellen, etwa um einen Kennwortschutz auf ein VEE-Programm zu legen, so dass nur Benutzer mit dem entsprechenden Kennwort das Programm anzeigen können. Abbildung 277 zeigt ein Beispiel für den Kennwortschutz.

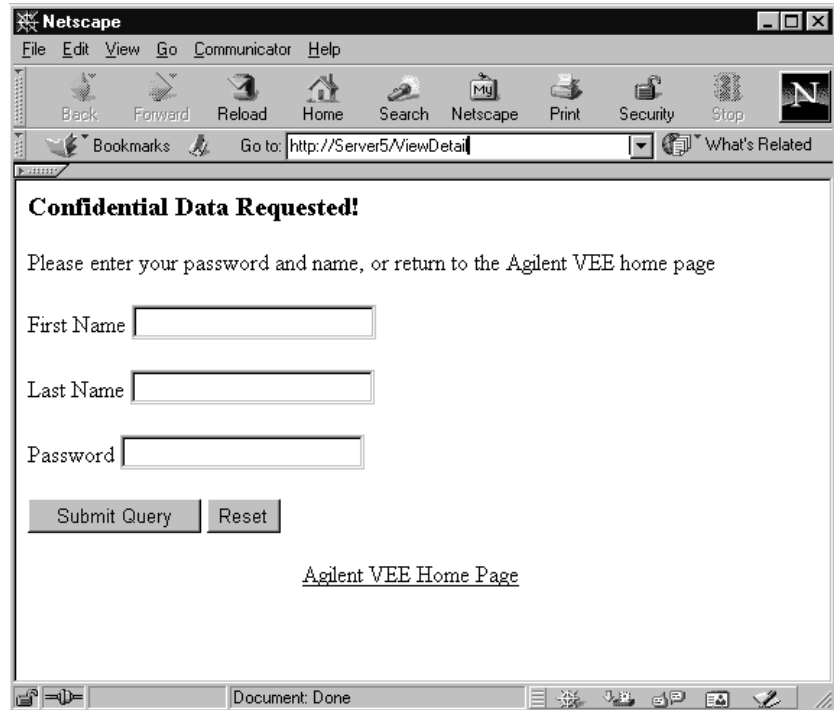


Abbildung 277 Ein Beispiel eines Kennwortfensters

### Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen.

- Die Verwendung des aufrufbaren VEE ActiveX-Automatisierungs-Servers erläutern und wissen, wann er eingesetzt werden sollte.
- Die Integration der VEE-Funktionalität in anderen Anwendungen oder Programmen erläutern.
- Die wichtigsten Konzepte bei der Verwendung des Web bei der Überwachung von VEE-Programmen erläutern.



## Anhang A: Zusätzliche Übungen

Allgemeine Programmier Techniken	519
Zeichenfolgen und globale Variablen verwenden	534
Optimierungstechniken	536
UserObjects	538
Agilent VEE-UserFunctions	541
Benutzerfenster und Einblendfenster erstellen	548
Mit Dateien arbeiten	553
Records	555
Tests sequenzieren	561

## Zusätzliche Übungen

Mit den folgenden Übungen haben Sie die Möglichkeit, weitere praktische Erfahrungen mit den in diesem Handbuch gelernten Konzepten von VEE zu sammeln. Die Übungen sind in vier Kategorien unterteilt.

Um diesen Anhang bestmöglich zu verwenden, sollten Sie eine Lösung entwickeln und diese mit den aufgeführten Antworten vergleichen. Es gibt viele Möglichkeiten, eine vorgegebene Aufgabe zu programmieren. Ihre Lösung ist gültig, wenn Sie die gestellte Aufgabe erfüllt. Programme, die leicht zu bedienen sind und schnell ausgeführt werden, bilden natürlich eine bessere Lösung. Jede Lösung umfasst eine kurze Beschreibung der wichtigsten Punkte.

## Allgemeine Programmiertechniken

### Der Äpfelsammler

Sie wollen wissen, wie viele Äpfel man braucht, um einen Zehn-Pfund-Korb zu füllen. Erstellen Sie ein VEE-Programm, das zählt, wie viele Äpfel benötigt werden, bis der Korb gefüllt ist. Jeder Apfel wiegt zwischen 0 und 1 Pfund.

Vorschläge

Dieses Programm kann mit 10 oder weniger Objekten erstellt werden. Wählen Sie aus den folgenden Objekten:

Start  
Until Break  
random()-Funktion  
Accumulator  
Break  
Real64  
Conditional (A>=B)  
Stop  
Counter  
If/Then/Else  
Alphanumeric

#### HINWEIS

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter **Help** ⇒ **Open Example...** ⇒ **Manual** ⇒ **UsersGuide** enthalten.

---

Lösung 1 – der Äpfelsammler

Abbildung 278 zeigt eine Lösung zur Übung Äpfelzähler (Apple Bagger).

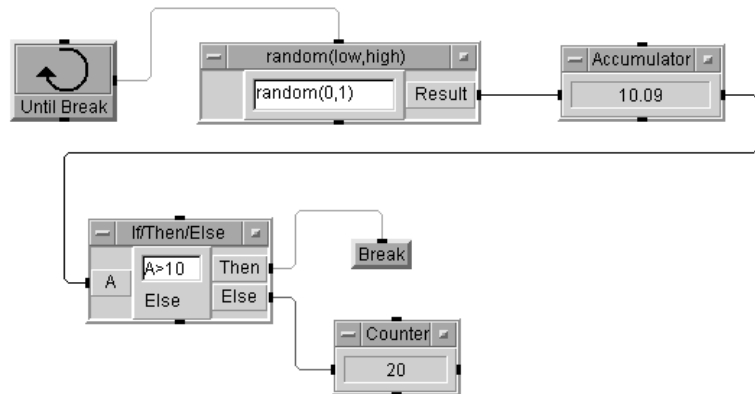


Abbildung 278 Äpfelzähler, Lösung 1

Wichtige Punkte

**Optimale Lösungen:** Verwenden Sie zur Optimierung der Programmleistung möglichst wenige Objekte. Diese Lösung verwendet sechs Objekte. Das Programm kann auch mit 10 Objekten implementiert werden, wie in Abbildung 279 gezeigt.

**Objekte „Until Break“ und „Break“:** Verwenden Sie diese Objekte für Schleifen, die das Testen einer Bedingung erfordern. In diesem Beispiel sollte die Schleife stoppen, wenn das Gesamtgewicht der Äpfel größer als zehn Pfund ist.

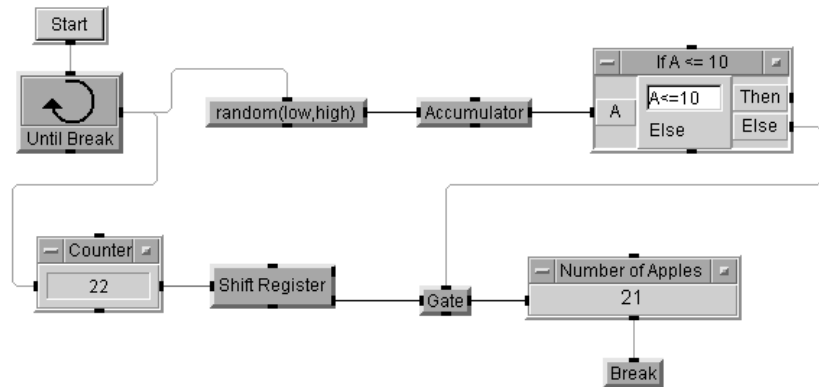
**Accumulator:** Verwenden Sie das Objekt Accumulator zum Erfassen einer laufenden Summe.

**Counter:** Verwenden Sie das Objekt Counter zum Erfassen einer laufenden Anzahl. In diesem Beispiel wird mit dem Objekt Counter die Gesamtzahl der Äpfel im Korb aufgezeichnet. Wenn

das Gesamtgewicht größer als 10 ist, löst nur der Then-Pin am Objekt If/Then/Else aus und liefert die richtige Antwort im Objekt Counter.

### Lösung 2 – der Äpfelsammler

Abbildung 279 bietet eine andere Lösung mit mehr Objekten.



**Abbildung 279** Äpfelsammler, Lösung 2

### Wichtige Punkte

**Start:** Die Verwendung eines Start-Objekts für dieses Programm ist redundant, da Sie die Schaltfläche Run in der Symbolleiste verwenden können. Start wird am besten verwendet, wenn Sie zwei Programme auf dem Bildschirm haben und diese unabhängig voneinander ausführen wollen. Sie können auch ein Programm mit einer Rückmeldungsschleife haben und definieren, wo die Ausführung beginnen soll.

**Shift Register:** Verwenden Sie ein Shift Register für den Zugriff auf die verschiedenen Werte der Ausgabe. In Lösung 2 zeichnet der Counter mit jedem eingesammelten Apfel die aktuelle Anzahl auf, bevor der Apfel gewogen wird. Wenn das Gesamtgewicht 10 überschreitet, muss daher die Anzahl wieder um 1 reduziert werden.

**Gate:** Mit dem Objekt Gate wird die Ausgabe gespeichert, bis eine andere Aktion auftritt und den Sequenz-Pin aktiviert. Wenn die Bedingung  $A \leq 10$  nicht mehr erfüllt ist, aktiviert der Else-Pin am Objekt If/Then/Else das Gate-Objekt.

### Testen von Zahlen

Zahlen testen, Schritt 1

Erstellen Sie ein Programm, bei dem ein Benutzer eine Zahl zwischen 0 und 100 eingeben kann. Wenn die Zahl größer oder gleich 50 ist, soll sie angezeigt werden. Ist die Zahl kleiner als 50, soll ein Fenster mit der Meldung „Bitte eine Zahl zwischen 50 und 100 eingeben“ eingeblendet werden.

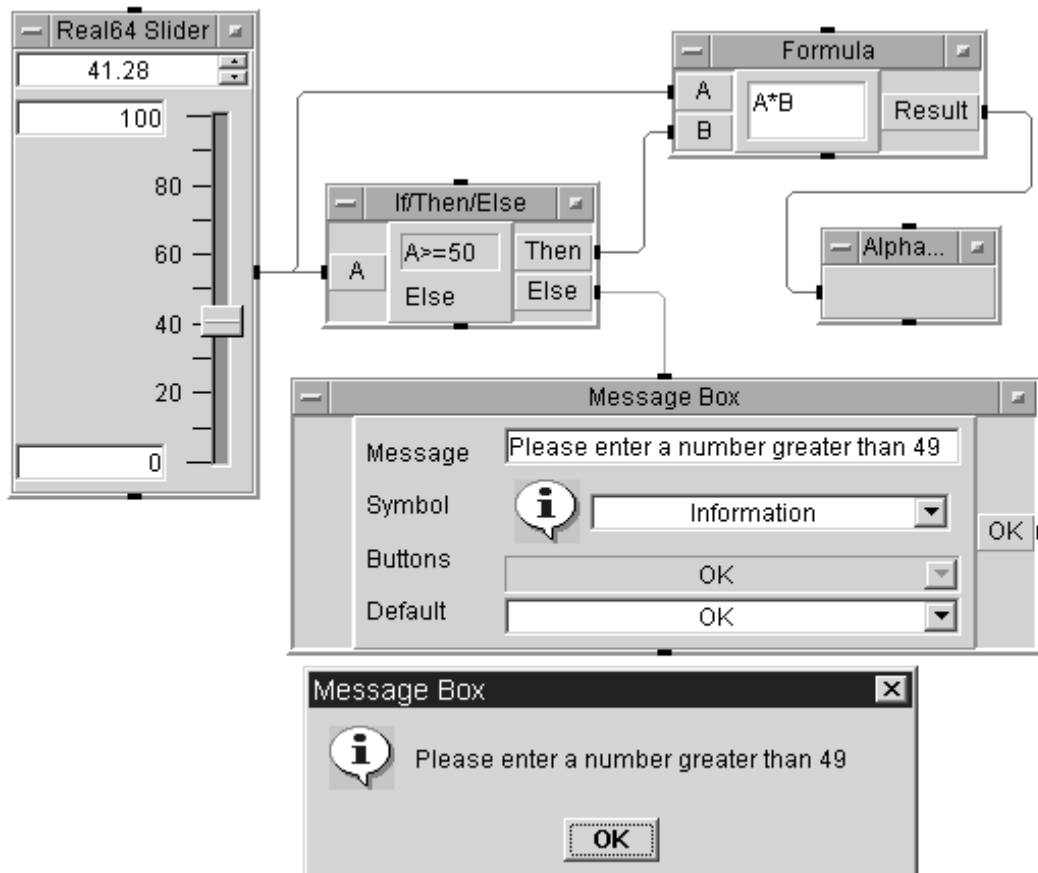
Vorschläge

Dieses Programm kann mit 5 oder weniger Objekten erstellt werden. Wählen Sie aus den folgenden Objekten:

Start  
Int32  
Slider  
Real64  
If/Then/Else  
Formula  
Gate  
Text  
Junction  
Alphanumeric  
Message Box

## Lösung – Zahlen testen, Schritt 1

Abbildung 280 zeigt eine Lösung zur Übung „Zahlen testen“ mit fünf Objekten.



**Abbildung 280** Zahlen testen (mit Einblendfenster)

## Zahlen testen, Schritt 2

Nachdem das Modell mit fünf Objekten funktioniert (die Meldung wird im Einblendfenster angezeigt), versuchen Sie, die Aufgabe mit vier Objekten (ohne das Objekt Gate) zu programmieren.

Lösung – Zahlen testen, Schritt 2

Abbildung 281 zeigt die Lösung zur Übung „Zahlen testen“ mit vier Objekten.

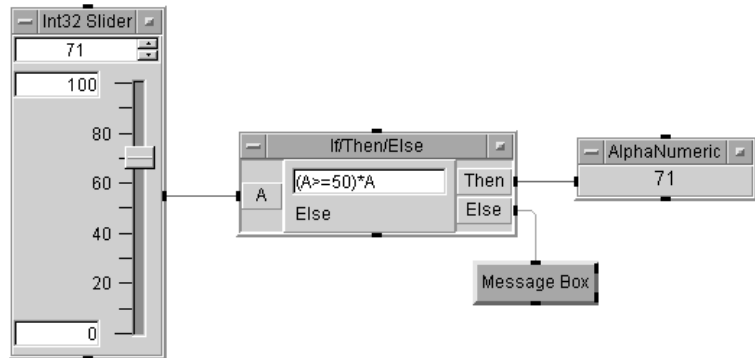


Abbildung 281 Zahlen testen, Schritt 2

Wichtige Punkte

**Auto Execute:** Bei allen Eingangsobjekte wie beispielsweise dem Int32 Slider ist im Properties-Fenster die Eigenschaft Auto Execute gesetzt. Bei dieser Auswahl arbeitet das Objekt immer, wenn sich sein Wert ändert, ohne dass Sie die Schaltfläche Start oder Run drücken müssen.

**Formeln konsolidieren:** Der Ausdruck  $(A \geq 50) * A$  im Objekt If/Then/Else wird zu  $1 * A$  ausgewertet, wenn  $A \geq 50$  wahr ist bzw. zu  $0$ , wenn der Ausdruck falsch ist.  $A$  wird somit am Then-Pin angelegt, wenn der Ausdruck wahr ist, und  $0$  wird am Else-Pin angelegt, wenn der Ausdruck falsch ist. (Jeder beliebige Ausdruck, dessen Auswertung nicht Null ergibt, wird als wahr betrachtet, und der Wert wird an den Then-Pin weitergegeben.)

Zahlen testen, Schritt 3

Erstellen Sie eine Lösung mit nur drei Objekten.

**Tipp:** Verwenden Sie einen triadischen Ausdruck im Formula-Objekt. Das Format lautet:  $\langle \text{Ausdruck} \rangle ? \langle \text{if TRUE, Ausgabe-wert} \rangle : \langle \text{if FALSE, Ausgabewert} \rangle$ . Wenn beispielsweise



$A < 10$  in TRUE (wahr) ausgewertet wird, soll der Wert  $A$  am Result-Pin angelegt werden; andernfalls soll die Zeichenfolge "FALSE" am Result-Pin angelegt werden. Sie verwenden hierzu den folgenden triadischen Ausdruck:  $(A < 10 ? A : \text{"FALSE"})$ .

Lösung – Zahlen testen, Schritt 3

Abbildung 282 zeigt die Lösung zur Übung „Zahlen testen“ mit nur drei Objekten.

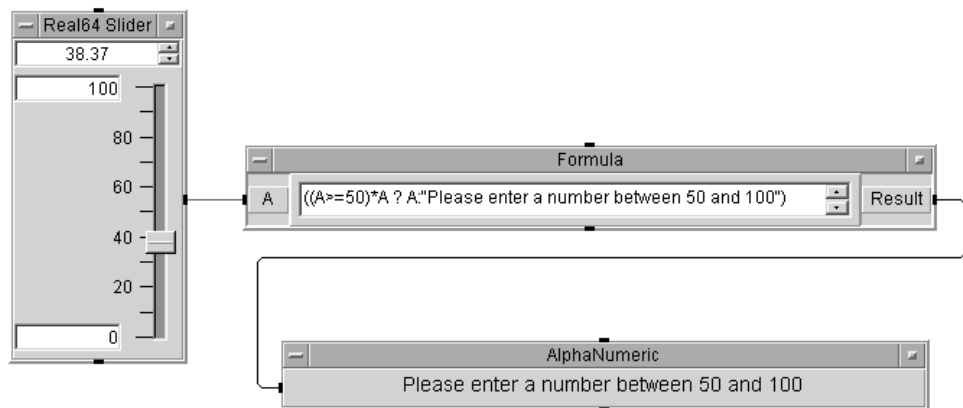


Abbildung 282 Zahlen testen, Schritt 3

## HINWEIS

Dies kann über ein „Real64 Input“-Dialogfeld mit der automatischen Fehlerprüfung implementiert werden. Der Benutzer muss jedoch eine gültige Zahl eingeben, bevor das Programm abgeschlossen werden kann.

## Zufallszahlen erfassen

Erstellen Sie ein Programm, das 100 Zufallszahlen generiert und anzeigt. Zeichnen Sie die erforderliche Gesamtzeit zum Generieren der Werte auf und zeigen Sie die Werte an.

### Vorschläge

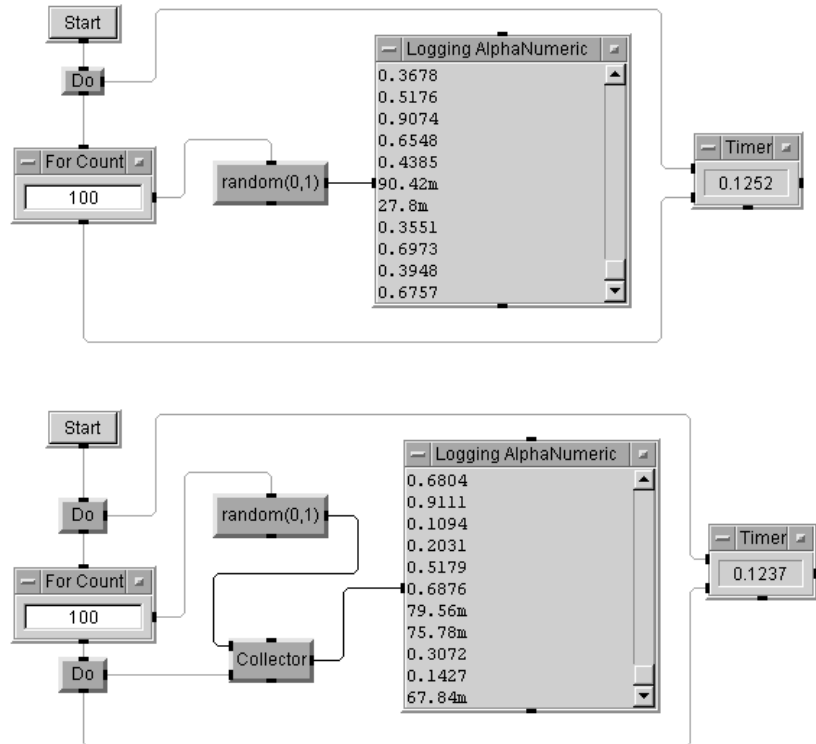
Dieses Programm kann mit sechs oder weniger Objekten erstellt werden. Wählen Sie aus den folgenden Objekten:

```
Start
For Range
Until Break
randomseed()-Funktion
random()-Funktion
Collector
Formula
Set Values
Alloc Int32
Logging AlphaNumeric
Strip Chart
Meter
Date/Time
Timer
Now()
Break
Do
```

**Tipp:** Senden Sie zum Verbessern der Leistung die Daten nur ein Mal an die Anzeige, indem Sie sie mit dem Collector-Objekt zunächst in einem Array erfassen. Beachten Sie die Unterschiede bei der Leistung.

## Lösung – Zufallszahlen erfassen

Abbildung 283 zeigt eine Lösung zur Übung „Zufallszahlen erfassen“.



**Abbildung 283** Zufallszahlen erfassen

## Wichtige Punkte

**Logging AlphaNumeric im Vergleich zu AlphaNumeric:** Verwenden Sie Logging AlphaNumeric zum Anzeigen einer fortlaufenden Eingabe (entweder Scalar oder Array 1D) als eine Historie vorangegangener Werte. Verwenden Sie AlphaNumeric zum Anzeigen von Daten aus einer einzigen Ausführung (der letzten) als einzelnen Wert, als Array 1D oder als Array 2D. Die Anzeige

Logging ist ein Array ohne Indexwerte; die Anzeige AlphaNumeric entspricht dem gleichen Array mit optionalen Indexnummern und Werten.

**Timing Pins:** Das Do-Objekt steuert, welches Objekt zuerst ausgeführt wird. Das Ende des Programms wird vom Sequenzausgangspin des Objekts For Count zeitlich gesteuert, da dieser Pin erst ausgelöst wird, wenn alle Objekte in der Schleife ausgeführt wurden.

## Zufallszahlen-Generator

Zufallszahlen-Generator, Schritt 1

Erstellen Sie einen Zufallszahlen-Generator, der externe Eingänge erfordert. Zeigen Sie die Zahl in einem Streifendiagramm an. Eingänge sollen zulässig sein für:

Zufallszahl Maximum

Zufallszahl Minimum

Anzahl der generierten Zufallszahlen

Lösung – Zufallszahlen-Generator, Schritt 1

Abbildung 284 zeigt eine Lösung für den ersten Schritt der Übung „Zufallszahlen-Generator“.

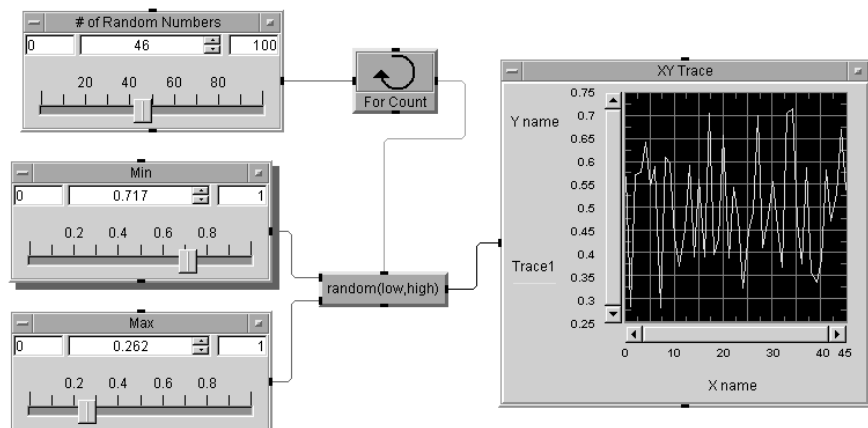


Abbildung 284 Zufallszahlen-Generator, Schritt 1

## Wichtige Punkte

**Layout der Slider-Objekte:** Sie können für die Bildschirm-anzeige der Slider-Objekte ein vertikales oder ein horizontales Format auswählen, indem Sie im Properties-Fenster unter Lay-out den Eintrag Horizontal wählen.

**XY Trace:** Verwenden Sie einen XY Trace zum Anzeigen der letzten Historie der laufend generierten Daten.

## Zufallszahlen-Generator, Schritt 2

Erfassen Sie die Zufallszahlen in einem Array. Ermitteln Sie den laufend wechselnden Durchschnitt und zeigen Sie ihn mit den Zahlen an.

## Lösung – Zufallszahlen-Generator, Schritt 2

Abbildung 285 zeigt eine Lösung für die Übung „Zufallszahlen-Generator“, Schritt 2.

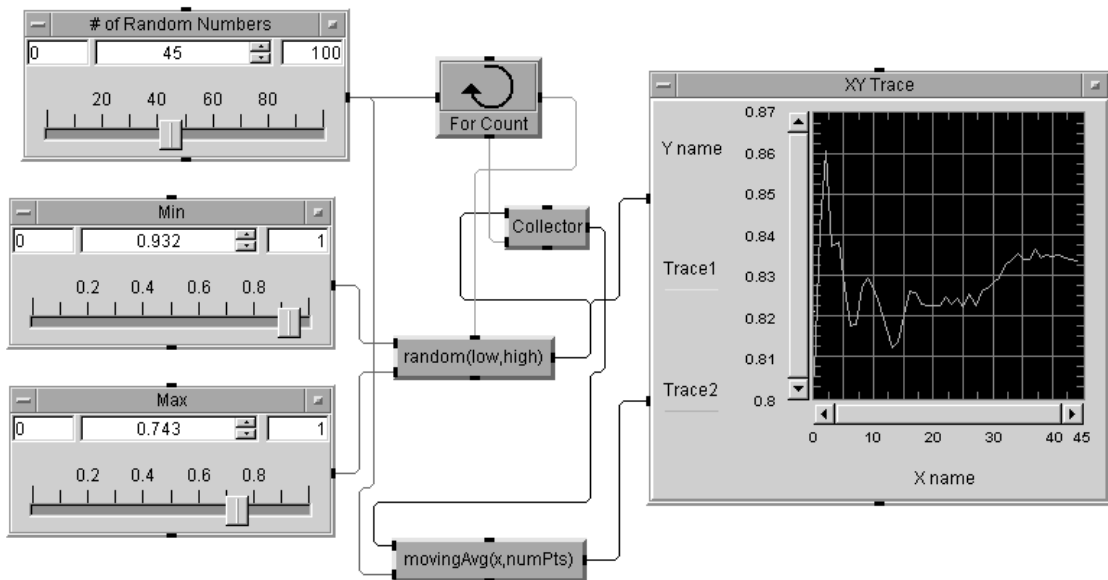


Abbildung 285 Zufallszahlen-Generator, Schritt 2

**MovingAvg(x, numPts):** Verwenden Sie dieses Objekt in der Kategorie Function & Object Browser, Data Filtering zum „Glätten“ der Eingangsdaten mit dem Durchschnitt aus einer angegebenen Anzahl von Datenpunkten vor dem jeweiligen Punkt, um den „geglätteten“ Datenpunkt zu berechnen.

### Masken verwenden

#### Maskentest, Schritt 1

Erstellen Sie eine 50 Hz-Sinuskurve mit einem anpassbaren Störfaktor. Testen Sie die gestörte Sinuskurve, um sicherzustellen, dass die Kurve innerhalb der folgenden Grenzwerte verläuft:

(0,0.5)

(2.2m, 1.2)

(7.2m, 1.2)

(10.2m, 0.5)

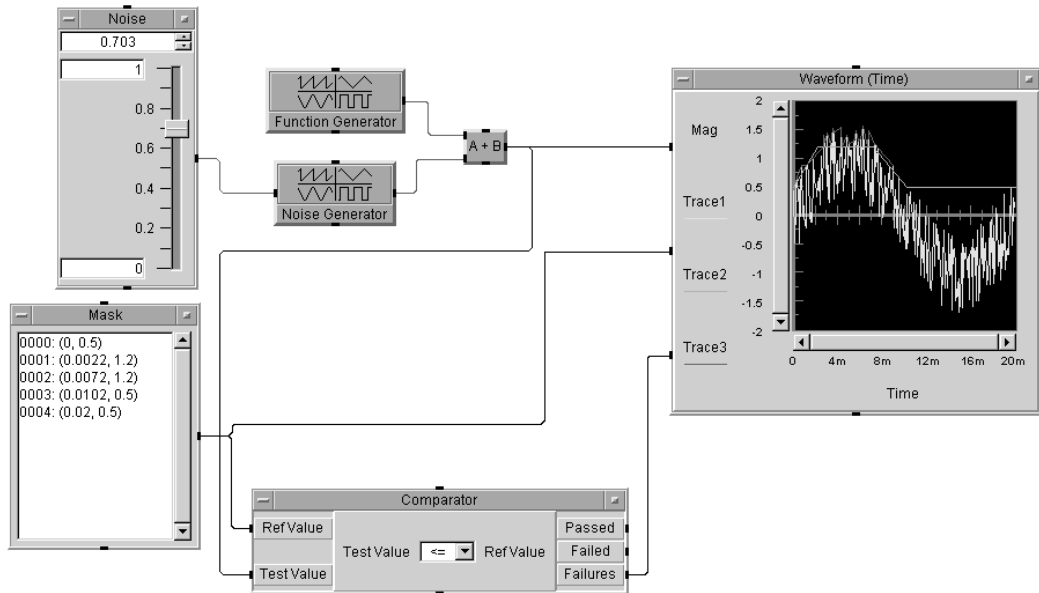
(20m, 0.5)

Wenn die Sinuskurve die Grenzwerte überschreitet, markieren Sie die entsprechenden Punkte mit einem roten Karo.

**Tipps:** Sie können das Format der Anzeige von Linien über Punkte bis zu Karos ändern. (Wählen Sie in Properties das Register Traces für jeden Trace-Eingang; der Linientyp kann durchgezogen, gestrichelt, gepunktet etc. sein. Als Punkttyp können Punkt, Karo, Kästchen oder andere Formen verwendet werden.) Das Objekt Comparator kann hierbei sehr nützlich sein.

## Lösung – Masken verwenden, Schritt 1

Abbildung 286 zeigt eine Lösung für Schritt 1.



**Abbildung 286** Der Maskentest, Schritt 1

## Masken verwenden, Schritt 2

Erweitern Sie das Programm, sodass es den Prozentanteil der Fehler berechnet und anzeigt.

## Lösung – Masken verwenden, Schritt 2

Abbildung 287 zeigt eine Lösung für Schritt 2.

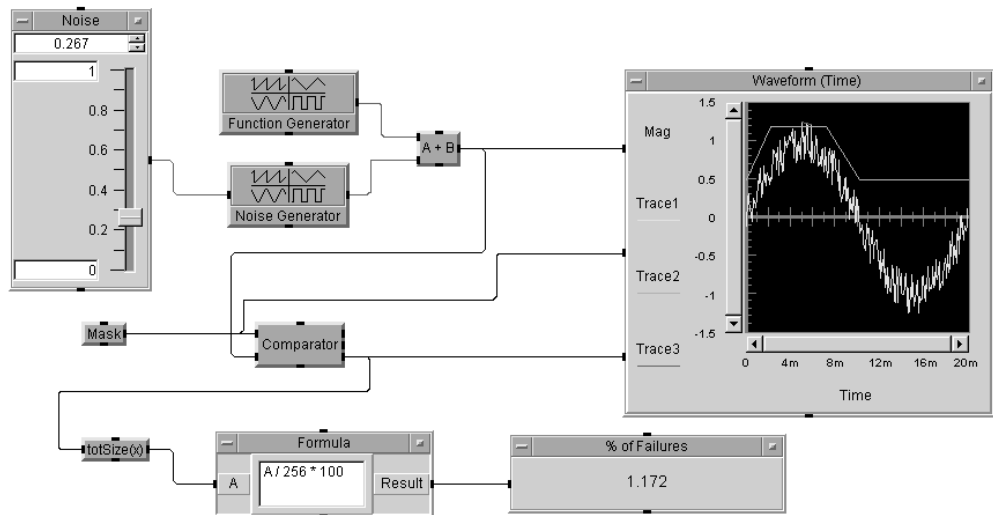


Abbildung 287 Der Maskentest, Schritt 2

## Wichtige Punkte

**Maske:** Die Maske wird mit dem Objekt **Data**  $\Rightarrow$  **Constant**  $\Rightarrow$  **Coord** erstellt und anschließend für fünf Array-Elemente konfiguriert. Sie geben die Koordinatenpaare durch Kommas getrennt ein; VEE fügt die Klammern hinzu. Bei der Auswahl der **x**-Werte war bekannt, dass die Zeitdauer der Wellenform 20 Millisekunden betrug. Beachten Sie auch, dass die Anzeige **Waveform (Time)** den Datentyp **Coord** als Eingang akzeptiert. Sie könnten auch ein Objekt **Data**  $\Rightarrow$  **Build Data**  $\Rightarrow$  **Arb Waveform** verwenden; dieses Objekt wandelt einen Datentyp **Coord** in einen Datentyp **Waveform** um durch Angabe der Anzahl von Punkten in der Waveform.

**Comparator:** Dieses Objekt vergleicht einen Testwert mit einem Referenzwert. Auch hierbei können Sie eine Wellenform mit einem Array aus Koordinatenpaaren vergleichen. Der **Failures**-Pin gibt einen Array der fehlgeschlagenen Datenpunkte



aus; Sie können diesen Array an die Anzeige senden und mit einer anderen Farbe oder einem anderen Linientyp hervorheben.

**TotSize:** Dieses Objekt gibt einfach die Anzahl der Elemente in einem Array aus. Da dieses Array die Anzahl der Fehler enthält, können Sie den Prozentanteil der Fehler ermitteln, indem Sie diesen Wert durch die Anzahl der Elemente in der ursprünglichen Wellenform (256) dividieren und das Ergebnis mit 100 multiplizieren.

**Formula:**  $A/256*100$  ist die zur Berechnung des Fehler-Prozentanteils verwendete Formel, da der Function Generator und der Noise Generator auf 256 Punkte eingestellt sind.

## Zeichenfolgen und globale Variablen verwenden

### Zeichenfolgen und globale Variablen manipulieren

Erstellen Sie mit Objekten oder Funktionen ein Programm, das einen Benutzernamen im folgenden Format akzeptiert: *<Leerzeichen> <Vorname> <Leerzeichen> <Nachname>*. Nachdem der Benutzer einen Namen eingegeben hat, soll das Programm den Vornamen weglassen und nur den Nachnamen ausgeben. Speichern Sie die Zeichenfolge in einer globalen Variablen. Rufen Sie die Zeichenfolge mit dem Formula-Objekt ab.

Lösung – Zeichenfolgen und globale Variablen manipulieren

Abbildung 288 zeigt eine Lösung zur Übung „Zeichenfolgen und globale Variablen manipulieren“.

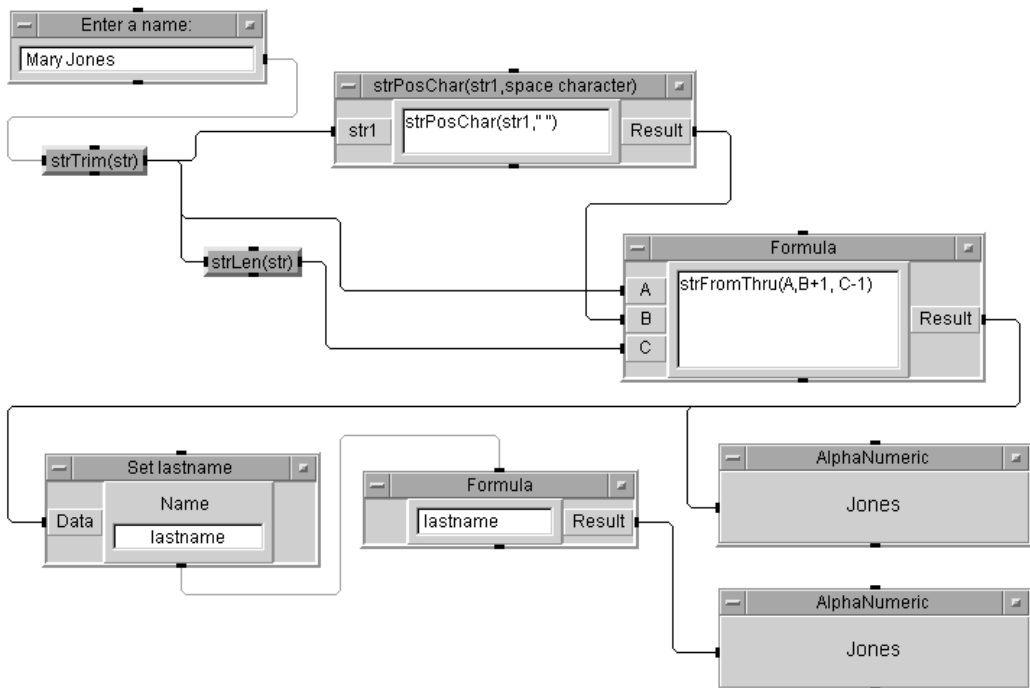


Abbildung 288 Zeichenfolgen und globale Variablen manipulieren

## Wichtige Punkte

**Zeichenfolgeobjekte und Funktionen:** `StrTrim(str)` entfernt zunächst alle Leerzeichen und Tabulatorzeichen vor oder hinter dem Namen. `StrPosChar(str1, „ “)` liefert den Index des Leerzeichens zwischen dem `firstname` und dem `lastname`. `StrLen(str)` liefert die Länge der Zeichenfolge. Alle diese Funktionen wurden mit den Zeichenfolgeobjekten ausgeführt; Sie könnten jedoch auch mit Zeichenfolgefunktionen innerhalb eines Formula-Objekts ausgeführt werden.

**Formula-Objekt:** `StrFromThru(A,B+1,C-1)` verwendet den Zeichenfolgeeingang **A**, addiert 1 zu dem Index des Leerzeichens von Eingang **B** und subtrahiert 1 von der Länge der Zeichenfolge am Eingang **C**. (Die gesamte Indexierung erfolgt zur Basis Null.)

**Set Variable:** Die globale Variable `lastname` kann sehr einfach gesetzt und anschließend in einem beliebigen Ausdrucksfeld referenziert werden; in diesem Beispiel etwa im Formula-Objekt.

**Optimizing:** Die drei Formeln können zu einer einzigen Formel kombiniert werden. Es empfiehlt sich, `strTrim()` als separates Element beizubehalten, da sein Ausgang mehrmals verwendet wird; die anderen Elemente können jedoch kombiniert werden, um die Geschwindigkeit zu optimieren. (Das Programm ist dadurch allerdings etwas weniger leicht lesbar.)

## Optimierungstechniken

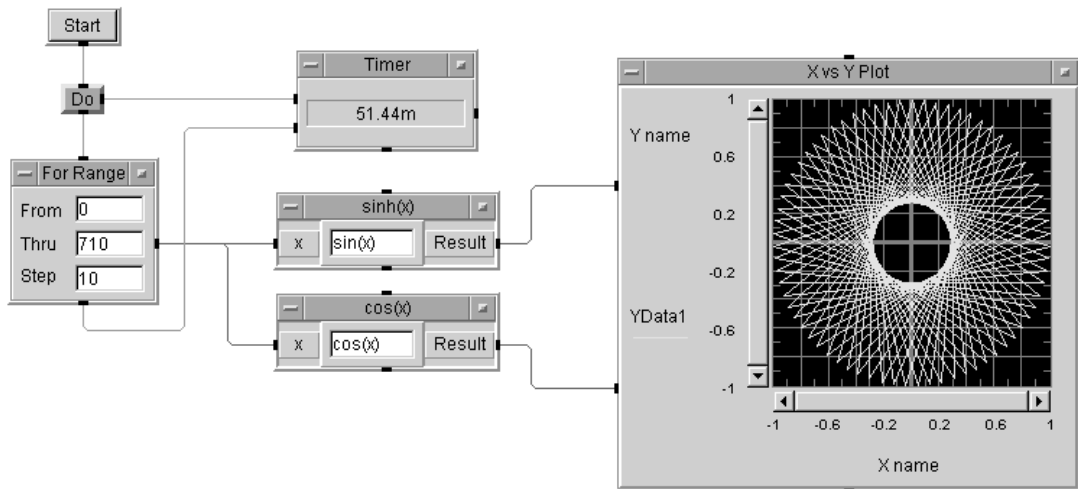
Für diese Übung erstellen Sie ein VEE-Programm auf zwei verschiedene Arten. Beachten Sie den Unterschied bei der Ausführungsgeschwindigkeit.

### Optimierungstechniken, Schritt 1

Erstellen Sie ein Programm, das den Bereich **0 to 710 step 10** über eine Sinus- und eine Kosinusfunktion sendet. Leiten Sie das Ergebnis dieser Funktionen an eine Anzeige **X vs.Y**. Verwenden Sie das Objekt Timer, um zu messen, wie lang die Ausführung des Programms dauert. (Legen Sie die Standardeinstellungen für Trig Mode auf Radians fest.)

### Lösung – Optimierungstechniken, Schritt 1

Abbildung 289 zeigt eine Lösung für Schritt 1.



**Abbildung 289** VEE-Programme optimieren, Schritt 1

## Optimierungstechniken, Schritt 2

Klonen Sie alle Objekte aus dem ersten Programm. Ändern Sie die neue Gruppe so, dass sie den Bereich in einem Array erfasst. Die Funktionen sine und cosine werden jetzt mit einem Array aus Punkten ausgeführt und nur ein einziges Mal geplottet. Beachten Sie die Zeitersparnis.

Lösung – Optimierungstechniken, Schritt 2

Abbildung 290 zeigt eine Lösung für Schritt 2.

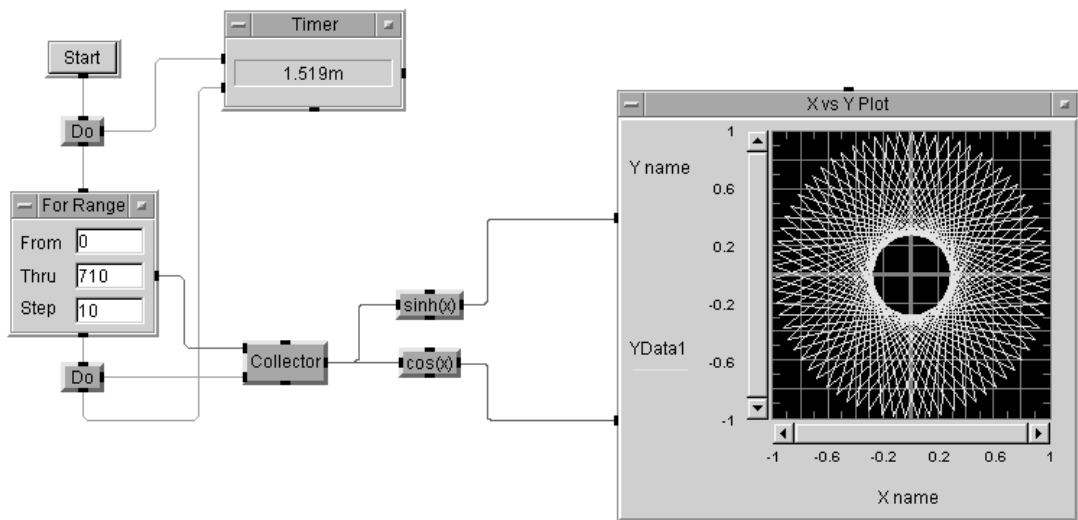


Abbildung 290 VEE-Programme optimieren, Schritt 2

## Wichtige Punkte

**Optimieren mit Arrays:** Beachten Sie die Verbesserung der Leistung von Schritt 1 auf Schritt 2 durch die Verwendung von Arrays. Führen Sie die Analyse oder Anzeige von Ergebnissen nach Möglichkeit immer mit Arrays statt mit Skalarwerten durch.

**X gg. Y-Anzeige:** Dieses Beispiel verwendet diese Anzeige statt der Anzeigen Waveform oder XY, da separate Daten für X und Y vorliegen.

## UserObjects

### UserObject „Random Noise“

UserObject „Random Noise“, Schritt 1

Erstellen Sie ein UserObject, das eine „Random Noise“-Waveform (Wellenform mit Zufallsstörung) generiert. Zeigen Sie die gestörte Wellenform und das Störungsspektrum außerhalb des UserObject an. Ermöglichen Sie die Steuerung folgender Komponenten außerhalb des UserObject: amplitude, number of points (Anzahl der Punkte), interval (time span) (Intervall), DC offset (Gleichstromversatz).

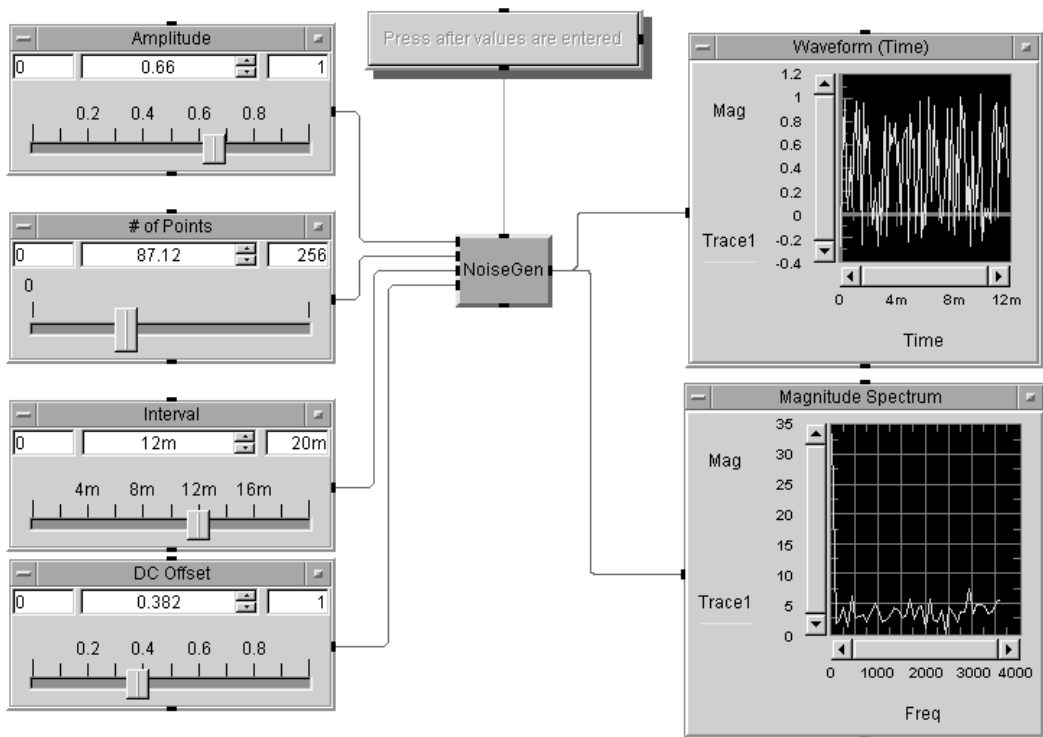
#### **HINWEIS**

Verwenden Sie keine virtuelle Quelle innerhalb des UserObject. Verwenden Sie Objekte wie beispielsweise Build Waveform und Random zum Erstellen des UserObject.

---

## Lösung – UserObject „Random Noise“

Abbildung 291 zeigt eine Lösung für das UserObject „Random Noise“.



**Abbildung 291** Ein UserObject „Random Noise“

Lösung – NoiseGen-Objekt in Random Noise

Abbildung 292 zeigt eine Lösung für das UserObject NoiseGen.

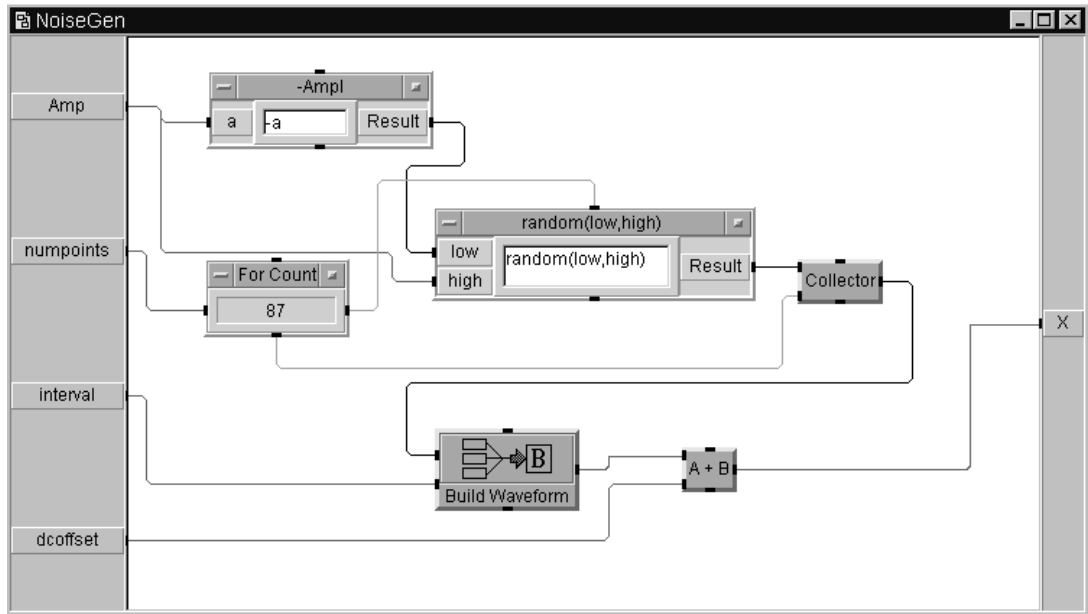


Abbildung 292 Das UserObject „NoiseGen“

Wichtige Punkte

**UserObject:** Beachten Sie, dass die von Ihnen erstellten UserObjects im Grund angepasste Objekte sind, die Sie VEE hinzufügen.

**Build Waveform:** Dieses Objekt erstellt einen Datentyp Waveform aus einem Real-Array von Amplitudenwerten und einer time span (der Zeitdauer in Sekunden, über die die y-Daten gesampelt wurden).



## Agilent VEE-UserFunctions

### UserFunctions verwenden

UserFunctions, Schritt 1

Erstellen Sie eine Funktion NoiseGen, die einen Amplitudenwert (0-1) aus einem Slider-Objekt akzeptiert und eine gestörte Wellenform zurückgibt.

**Verwenden Sie folgende Komponenten NICHT:**

Virtual Source  
For Count  
For Range

**Verwenden Sie:**

Formula  
Ramp  
Build Waveform

**Tipp:** Verwenden Sie **randomize(array, -a,a)**, wobei des Array 256 Punkte haben muss und **a** die Amplitude ist. Erstellen Sie ein einfaches Hauptprogramm, das diese Funktion aufruft; auf diese Weise können Sie sicherstellen, dass die Funktion richtig funktioniert.

Lösung – UserFunctions, Schritt 1

Abbildung 293 zeigt eine Lösung für Schritt 1.

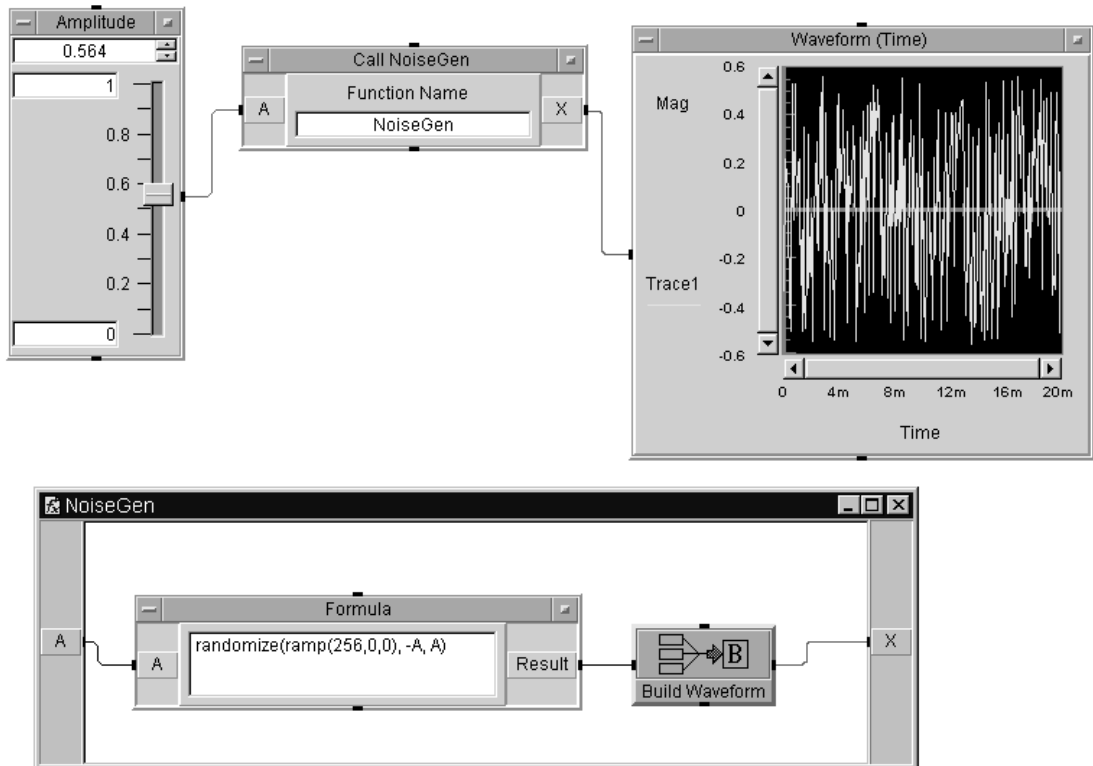


Abbildung 293 UserFunctions, Schritt 1

Wichtige Punkte

**Ramp():** Beachten Sie, dass mit der Funktion `ramp()` innerhalb der Parameterliste für `randomize()` ein Array aus 256 Punkten generiert wird.

**Build Waveform:** Beachten Sie, dass die Standard-Zeitdauer („default time span“) 20 Millisekunden beträgt; Sie brauchen daher nur ein Array an dieses Objekt zu senden, um eine Wellenform zu generieren.

## UserFunctions, Schritt 2

Erstellen Sie in dem gleichen Programm eine weitere Funktion AddNoise, die die erste Funktion NoiseGen aufruft. AddNoise sollte die gestörte Wellenform von der Funktion NoiseGen zu einer Sinuskurve addieren. AddNoise sollte zwei Eingänge haben, einen für die NoiseGen-Amplitude und einen für die Sinuskurve. Sie sollte einen Ausgang für das Ergebnis haben.

Erstellen Sie ein einfaches Hauptprogramm mit einem Slider für die Noise-Amplitude und die **Virtual Source**  $\Rightarrow$  **Function Generator (sine wave, Freq = 100 Hz)** für die gute Wellenform, zu der die Störung addiert werden soll. Zeigen Sie die resultierende Wellenform an.

## Lösung – UserFunctions, Schritt 2

Abbildung 294 zeigt eine Lösung für Schritt 2.

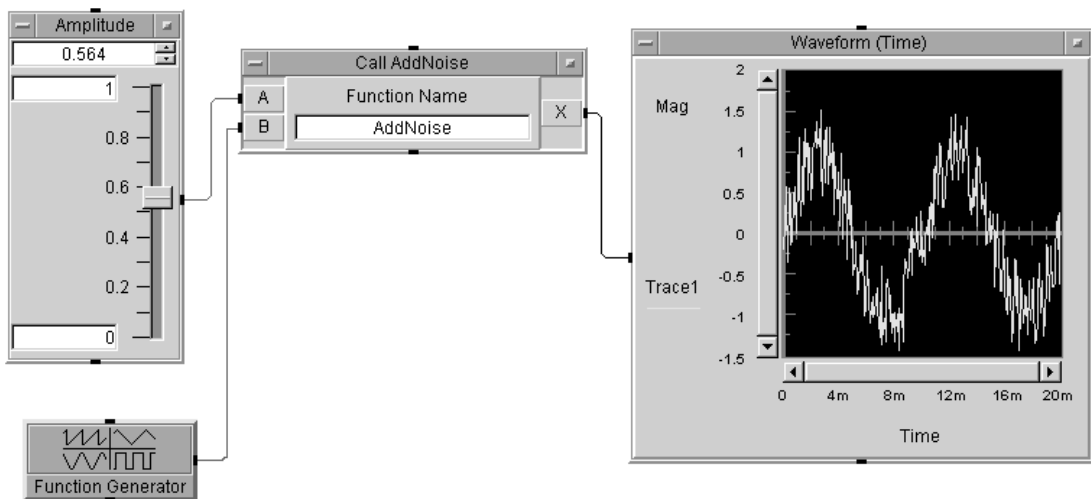


Abbildung 294 UserFunctions, Schritt 2

## UserFunctions, Schritt 3

Rufen Sie im gleichen Programm die Funktion AddNoise erneut auf; dieses Mal von einem Formula-Objekt aus, und verwenden Sie den absoluten Wert des Ergebnisses. Zeigen Sie die Wellen-

## Anhang

form für den absoluten Wert in der gleichen Anzeige an. Bereiten Sie als nächstes die Bearbeitung der Funktion AddNoise vor. Schalten Sie **Debug**  $\Rightarrow$  **Show Data Flow** ein. Lassen Sie das Fenster AddNoise geöffnet und führen Sie das Programm aus. Beachten Sie, wie nützlich diese Funktion für die Fehlerbehebung ist.

Lösung – UserFunctions, Schritt 3

Abbildung 295 zeigt eine Lösung für Schritt 3.

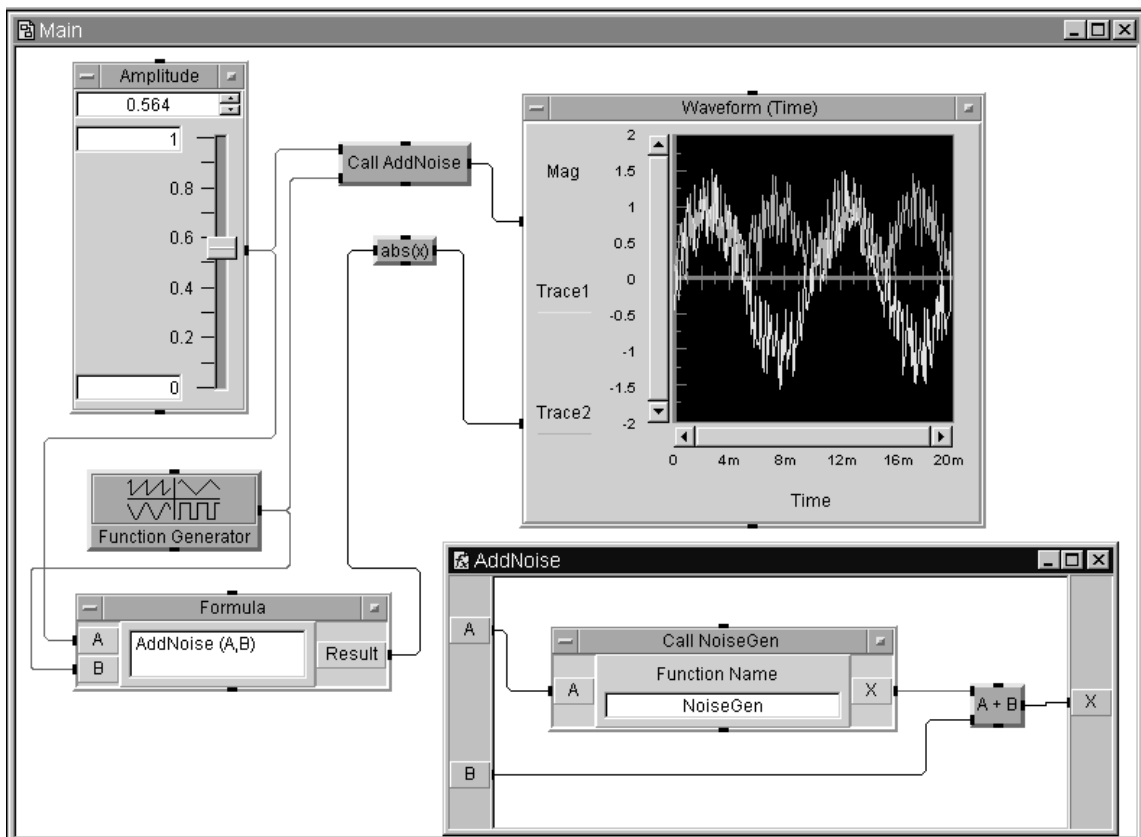


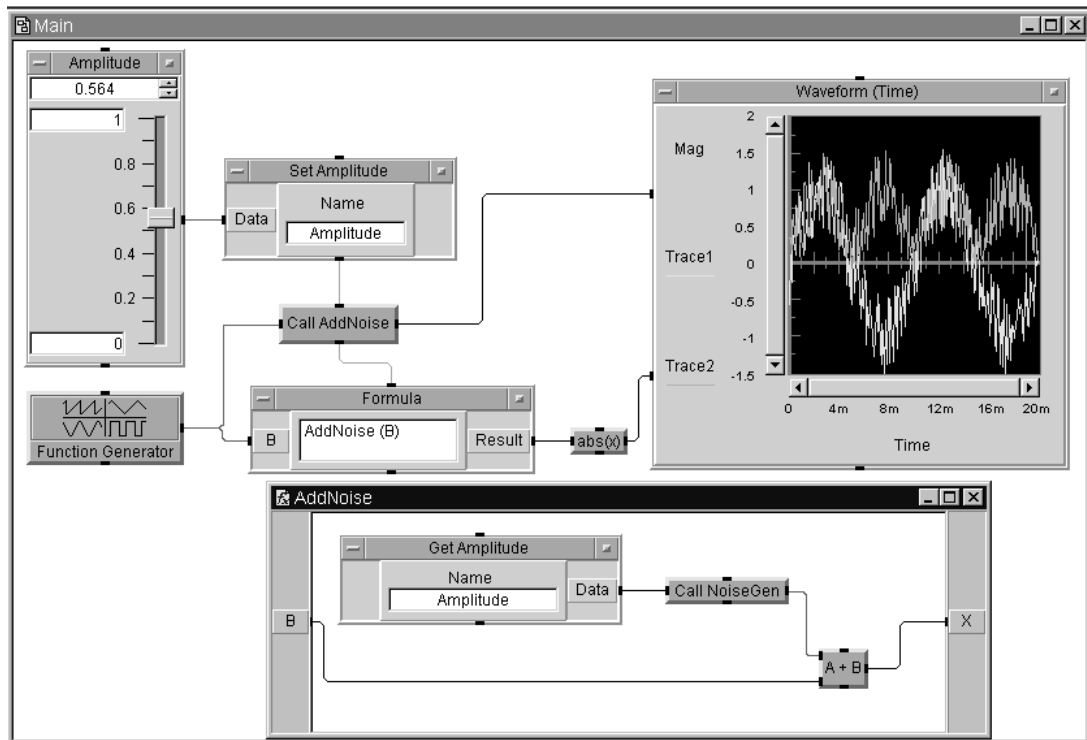
Abbildung 295 UserFunctions, Schritt 3

## UserFunctions, Schritt 4

Ändern Sie jetzt das Programm so, dass der Slider eine globale Variable mit dem Namen Amplitude festlegt. Sorgen Sie dafür, dass die Funktion NoiseGen diese globale Variable verwendet (damit NoiseGen keinen Eingangs-Pin mehr erfordert). Stellen Sie sicher, dass das Programm richtig ausgeführt wird. Speichern Sie diese Datei als **uflab.vee**.

## Lösung – UserFunctions verwenden, Schritt 4

Abbildung 296 zeigt eine Lösung für Schritt 4.



**Abbildung 296** UserFunctions, Schritt 4

**Tipp:** Beachten Sie, dass die Objekte Call AddNoise und Formula die globale Variable Amplitude verwenden; daher müssen beide Objekte nach Ausführung des Objekts Set Amplitude aus-

geführt werden. Durch das Verbinden der Sequenz-Pins von Set Amplitude mit Call AddNoise und von Call AddNoise mit Formula wird sichergestellt, dass die Objekte in der richtigen Reihenfolge ausgeführt werden.

## UserFunctions-Bibliotheken importieren und löschen

Erstellen Sie ein einfaches Programm zum Importieren der Funktionen **uflab.vee** aus der vorigen Übung. Rufen Sie die Funktion auf, die die Störung hinzufügt, und löschen Sie anschließend die Bibliothek über das Programm. Wählen Sie im Objektmenü des Objekts Call den Befehl Select Function.

**Tipp:** Wählen Sie Load Lib im Import Library-Objektmenü, um die angegebene Bibliothek manuell zu laden, damit Sie die Funktion Select Function in Call verwenden können.

Lösung – Bibliotheken programmgesteuert importieren und löschen

Abbildung 297 zeigt eine Lösung zum programmgesteuerten Löschen der Bibliothek.

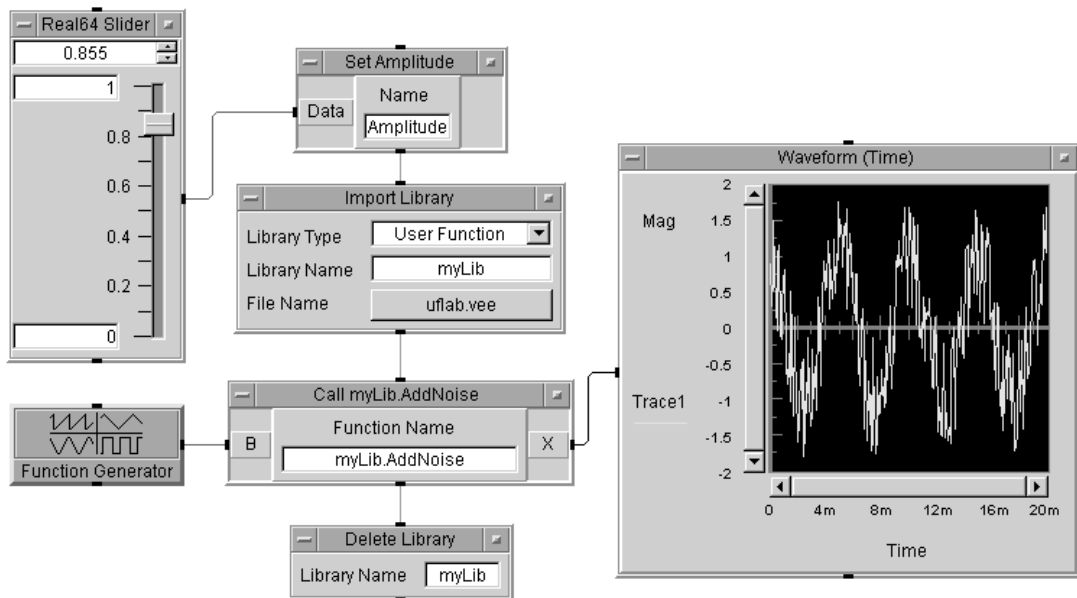


Abbildung 297 Bibliotheken importieren und löschen

### Wichtige Punkte

**Select Function:** Beachten Sie, dass diese Auswahl die richtigen Ein- und Ausgangs-Pins für die ausgewählte Funktion konfiguriert.

**UserFunctions bearbeiten:** Wenn Sie eine Bibliothek von UserFunctions über ein Programm importieren, können Sie die Funktionen nicht bearbeiten. Sie können die Funktionen anzeigen und Unterbrechungspunkte für die Fehlerbehebung setzen. Wenn Sie die importierten UserFunctions bearbeiten wollen, verwenden Sie den Befehl Merge Library.

**Variable „Caution“ setzen:** Bei Verwendung einer globalen Variablen in einer Funktion müssen Sie daran denken, diese globale Variable zu erstellen, wenn Sie die Funktion in anderen Programmen verwenden wollen. Einer der Vorteile beim expliziten Erstellen von Eingängen und Ausgängen ist, dass diese einfacher nachzuvollziehen sind.

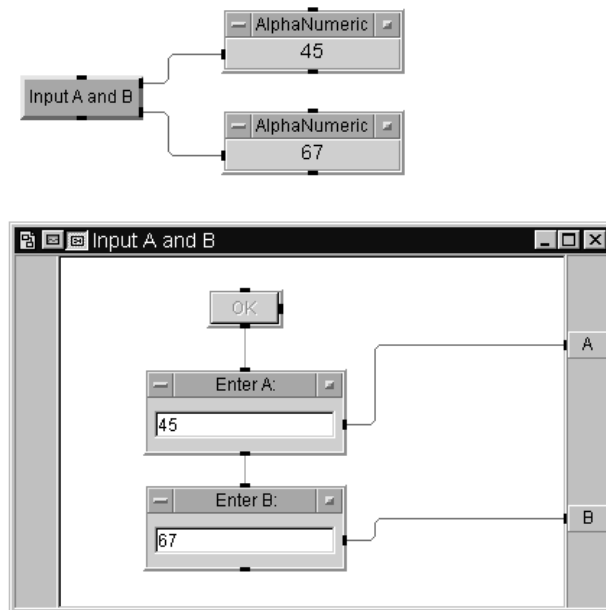
## Benutzerfenster und Einblendfenster erstellen

Benutzerfenster und Einblendfenster erstellen, Schritt 1

Erstellen Sie ein Fenster, über das der Benutzer aufgefordert wird, Zahlen einzugeben. Erstellen Sie ein UserObject für die Interaktion mit einem Benutzer. Fordern Sie den Benutzer zu 2 Eingaben auf, **A** und **B**. Senden Sie beide Eingaben an eine Anzeige. Verwenden Sie ein UserObject mit ausgewählter Option Show On Execute zum Anzeigen des Fensters.

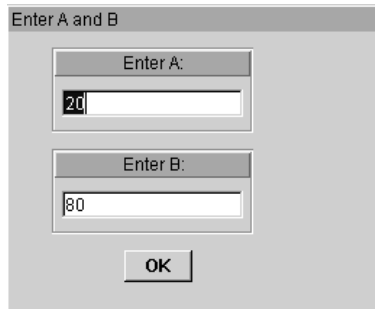
Lösung – Benutzerfenster und Einblendfenster erstellen, Schritt 1

Abbildung 298 zeigt eine Lösung in der Detailansicht. Abbildung 299 zeigt das Fenster, das bei der Ausführung des Programms erscheint.



**Abbildung 298** UserObject, das den Benutzer zur Eingabe von A und B auffordert





**Abbildung 299** Fenster zur Eingabe von A und B durch den Benutzer

#### Wichtige Punkte

**UserObject Properties:** Wählen Sie im Dialogfeld UserObject Properties die Option Pop-Up Panel und klicken Sie, um Show Panel On Execute einzuschalten. Ändern Sie den Namen **Pop-Up Panel** ⇒ **Panel Title** in `""Enter A or B"`.

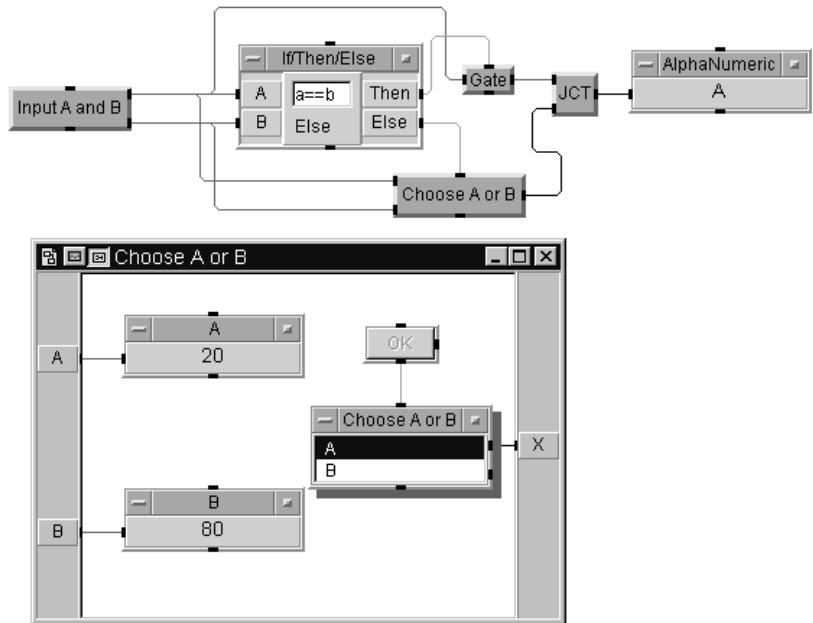
#### Benutzerfenster und Einblendfenster erstellen, Schritt 2

Statt sowohl **A** als auch **B** anzuzeigen, fragen Sie den Benutzer, ob **A** oder **B** angezeigt werden soll, wenn die beiden Zahlen unterschiedlich sind. Wenn nach der Abfrage die beiden Werte **A** und **B** gleich sind, zeigen Sie den Wert an. Sind die beiden Werte **A** und **B** unterschiedlich, fordern Sie den Benutzer auf, einen Wert für die Anzeige auszuwählen. Zeigen Sie je nach der Auswahl des Benutzers **A** oder **B** an.

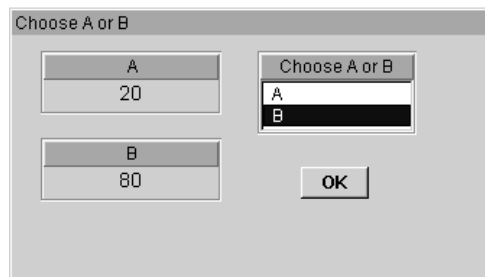
**Tipp:** Fügen Sie ein weiteres **UserObject** mit einem Einblendfenster mit der Einstellung **Show Panel on Execute** hinzu, und fordern Sie den Benutzer auf, den Wert hier einzugeben.

#### Lösung – Benutzerfenster und Einblendfenster erstellen, Schritt 2

Abbildung 300 zeigt das UserObject, das den Benutzer auffordert, eine Auswahl zu treffen, wenn **A** und **B** unterschiedliche Zahlen sind. Abbildung 301 zeigt das zweite Einblendfenster, über das der Benutzer gefragt wird, ob **A** oder **B** angezeigt werden soll.



**Abbildung 300** UserObject, das den Benutzer fragt, ob A oder B angezeigt werden soll



**Abbildung 301** Fenster für Benutzer zur Auswahl, ob A oder B angezeigt werden soll

### Wichtige Punkte

**Gate:** Das Objekt Gate sendet nur einen Wert, wenn die beiden Zahlen gleich sind.

**Junction:** Das Objekt JCT ermöglicht mehrfache Eingänge zum Objekt Alphanumeric. Das Objekt JCT ist ein „Verdrahtetes ODER“-Objekt.

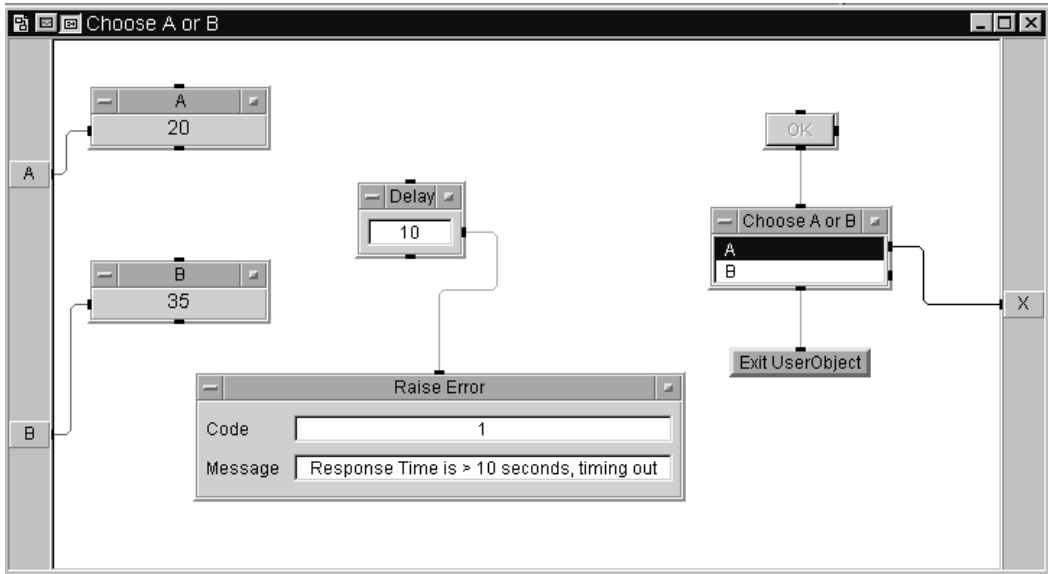
**Ein List-Objekt als Menü:** Beachten Sie die Verwendung des Objekts **Data** ⇒ **Selection Controls** ⇒ **List** für zwei Auswahloptionen und die Formatierung für eine Liste. Diese Konfiguration gibt einen Text **A** oder **B** aus. Wenn Sie den Ordnungswert (0 oder 1) benötigen, verwenden Sie stattdessen den Ordnungs-Datenausgang des Objekts List.

### Benutzerfenster und Einblendfenster erstellen, Schritt 3

Wenn der Benutzer keine Zahlen eingibt, soll eine Fehlermeldung generiert werden. Fügen Sie beim zweiten UserObject, das den Benutzer bei zwei unterschiedlichen Zahlen zur Auswahl zwischen **A** und **B** für die die Anzeige auffordert, einen Fehler hinzu. Wenn der Benutzer innerhalb von 10 Sekunden weder **A** noch **B** auswählt, soll der Fehler generiert werden.

### Lösung – Benutzerfenster und Einblendfenster erstellen, Schritt 3

Abbildung 302 zeigt das UserObject mit der Änderung, sodass es einen Fehler generiert, wenn der Benutzer nicht innerhalb von 10 Sekunden **A** oder **B** auswählt.



**Abbildung 302** Fehler generieren, wenn der Benutzer keine Auswahl eingibt

### Wichtige Punkte

**UserObject beenden:** Wenn der Benutzer vor Ablauf von 10 Sekunden reagiert, beendet dieses Objekt das UserObject, auch wenn die Ausführung des Objekts Delay noch nicht abgeschlossen ist.

**Verzögerung und Fehler auslösen:** Nach 10 Sekunden führt das Objekt Delay einen „Ping“ auf das Objekt Raise Error aus; dadurch wird die Ausführung des Programms angehalten und die eingegebene Fehlermeldung angezeigt. Außerdem wird ein roter Umriss um das Objekt angezeigt, das den Fehler verursacht hat; diese rote Umrisslinie verschwindet, wenn auf eine der Schaltflächen **Stop** oder **Run** in der Symbolleiste geklickt wird.

**OK und Delay:** Beachten Sie, dass die beiden Threads in **AnotB** getrennt sind, sodass **OK** und **Delay** gleichzeitig ablaufen.

## Mit Dateien arbeiten

### Daten in und aus Dateien bewegen

Erstellen Sie ein VEE-Programm, das die Uhrzeit in eine Datei schreibt. Generieren Sie 100 Zufallspunkte und schreiben Sie sie in die Datei. Berechnen Sie den Mittelwert und die Standardabweichung der Zahlen und hängen Sie sie im folgenden Format an die Datei an:

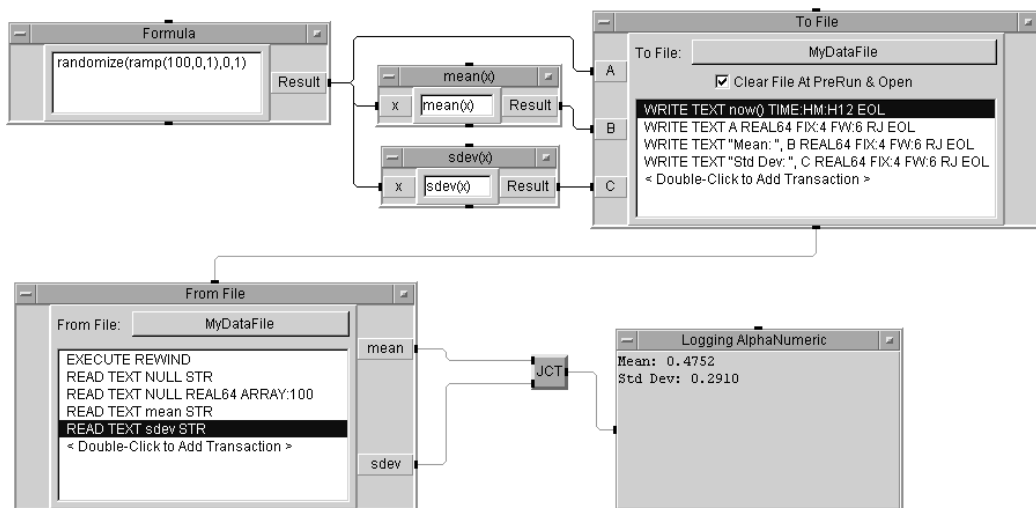
Mittelwert: xxxxxx

Std.Abweichung: yyyyyy

Lesen Sie als nächstes nur den Mittelwert und die Standardabweichung aus der Datei aus. Abbildung 303 zeigt das Bewegen von Daten in und aus Dateien.

Lösung – Bewegen von Daten in und aus Dateien

Abbildung 303 zeigt eine Lösung zum Bewegen von Daten in und aus Dateien..



**Abbildung 303** Bewegen von Daten in und aus Dateien

### Wichtige Punkte

**Ein Array generieren:** Verwenden Sie **randomize(ramp(100,0,1), 0, 1)** im Formula-Objekt zum Erstellen eines Array aus 100 Zufallszahlen. Die Funktion **ramp()** generiert ein geordnetes Array und liefert es an die Funktion **randomize()**, die dann Zufallswerte zwischen 0 und 1 generiert.

**Zeitstempel:** Die Funktion **now()** wird im Ausdrucksfeld des Dialogfelds **I/O Transaction** für die Transaktion **Eins** des Objekts **To File** verwendet. Wenn Sie das Format in **TIME STAMP FORMAT** ändern, werden zusätzlichen Schaltflächen in dem Dialogfeld angezeigt, um anzugeben, wie die Uhrzeit gespeichert werden soll.

**Zwei Werte in einer Zeile speichern:** In der dritten und vierten Transaktion des Objekts **To File** wird eine konstante Textzeichenfolge gespeichert, gefolgt von einem Real-Wert. In der dritten Transaktion geben Sie beispielsweise **"Mean: ", B** im Ausdrucksfeld des Fensters **I/O Transaction** ein (sofern der Mittelwert am Eingangs-Pin **B** anliegt).

**Einen Wert aus einer Datei lesen:** Zum Ermitteln des Mittelwerts und der Standardabweichung senden Sie zunächst einen Befehl **EXECUTE REWIND**, um den Lesezeiger an den Anfang zu positionieren. Verwenden Sie anschließend **NULL** mit dem richtigen Format, um eine **READ**-Transaktion über den Zeitstempel und das Real-Array hinaus auszuführen. (Dadurch werden die eingelesenen Werte gelöscht, statt sie in einem Ausgangsanschluss abzulegen.) Schließlich lesen Sie die beiden letzten Zeilen in der Datei als Zeichenfolgen ein.

**Junction:** Verwenden Sie das Objekt **Flow**  $\Rightarrow$  **Junction**, um mehrere Ausgänge zu einem einzigen Eingang zu verbinden, beispielsweise zum Verbinden der Ausgänge **mean** und **sdev** zur Anzeige **Logging AlphaNumeric**.

## Records

### Records manipulieren

#### Records manipulieren, Schritt 1

Erstellen Sie einen Record mit drei Feldern, die eine Ganzzahl enthalten, die aktuelle Uhrzeit als Zeichenfolge und ein Array aus vier Real-Elementen. Die Felder sollen die Namen int, daytime und rarry haben. Fügen Sie diesen Datensatz mit einem anderen, der eine Zufallszahl zwischen 0 und 1 enthält, und einer Wellenform zusammen. Benennen Sie diese Felder rand und wave.

#### Lösung – Records manipulieren, Schritt 1

Der resultierende Record sollte fünf Felder enthalten, wie in Abbildung 304 gezeigt.

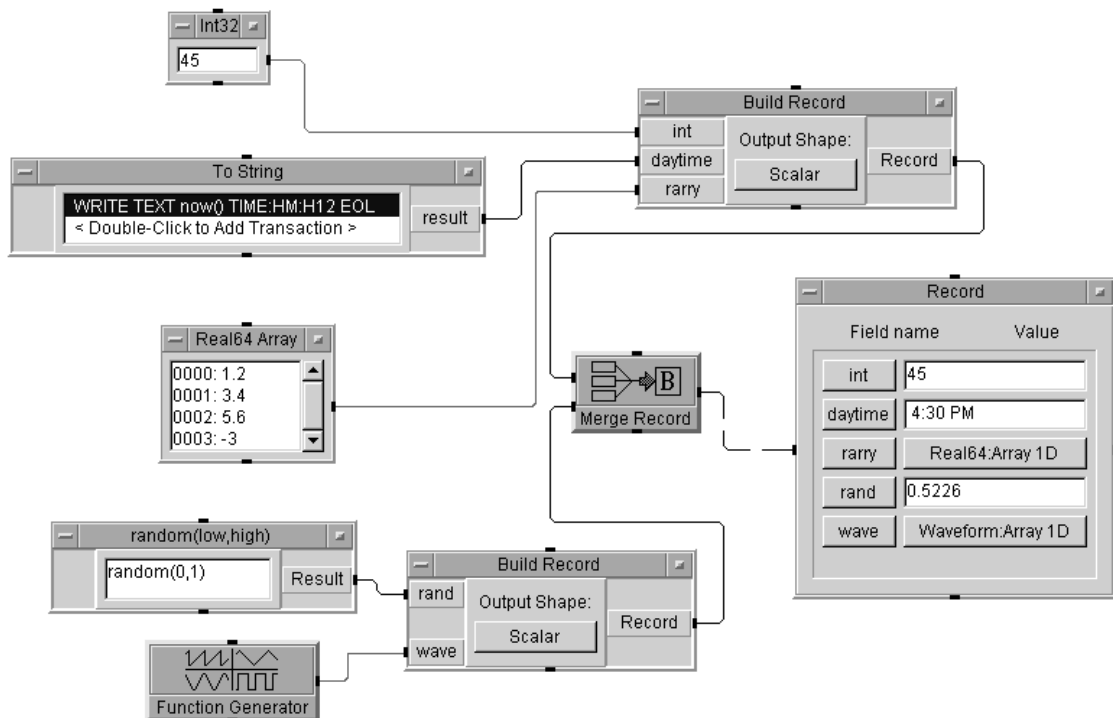


Abbildung 304 Records manipulieren, Schritt 1

### Wichtige Punkte

**Zeitstempel:** Verwenden Sie die Funktion `now()` im Objekt `To String`, um Ihren Zeitstempel für dieses Programm zu erstellen. Anschließend können Sie das Format angeben.

**Konfigurieren einer Datenkonstanten als Array:** Ein beliebiger Datentyp im Menü **Data**  $\Rightarrow$  **Constant** kann zu einem Array werden, indem Sie **Properties** und anschließend unter **Konfiguration** die Option **1D Array** auswählen. Die Größe kann hier eingegeben werden, oder Sie können bei der Eingabe von Werten die Eingabetaste drücken, um weitere Werte anzuhängen.

**Felder benennen:** Durch Umbenennen der Eingangsanschlüsse am Objekt **Build Record** können Sie Ihrem Record spezifische Feldnamen zuweisen wie beispielsweise `int`, `rand` und `wave`.

**Der Eingang „Default Value Control“:** Eine Record-Konstante bildet ein hervorragendes interaktives Anzeigeobjekt, wenn Sie einen Default Value Control-Pin für die Steuerung der Standardwerte hinzufügen. Die Record-Konstante konfiguriert sich automatisch selbst für den Datensatz, den sie empfängt.

### Records manipulieren, Schritt 2

Verwenden Sie einen bedingten Ausdruck in einem Formula-Objekt, um den Zufallswert in dem Record zu testen, und zeigen Sie entweder den Wert oder eine Textzeichenfolge an. Wenn der nächste Wert niedriger als 0.5 ist, zeigen Sie den Zufallswert an; geben Sie andernfalls eine Textzeichenfolge **„More than 0.5“** aus. Extrahieren Sie als nächstes nur die Uhrzeit und die Wellenform.

**Tipp:** Verwenden Sie kein Formula-Objekt zum Extrahieren der Uhrzeit und der Wellenform. Zeigen Sie diesen Record mit einem `AlphaNumeric`-Objekt an.



## Lösung – Records manipulieren, Schritt 2

Abbildung 305 zeigt das Manipulieren von Records, Schritt 2.

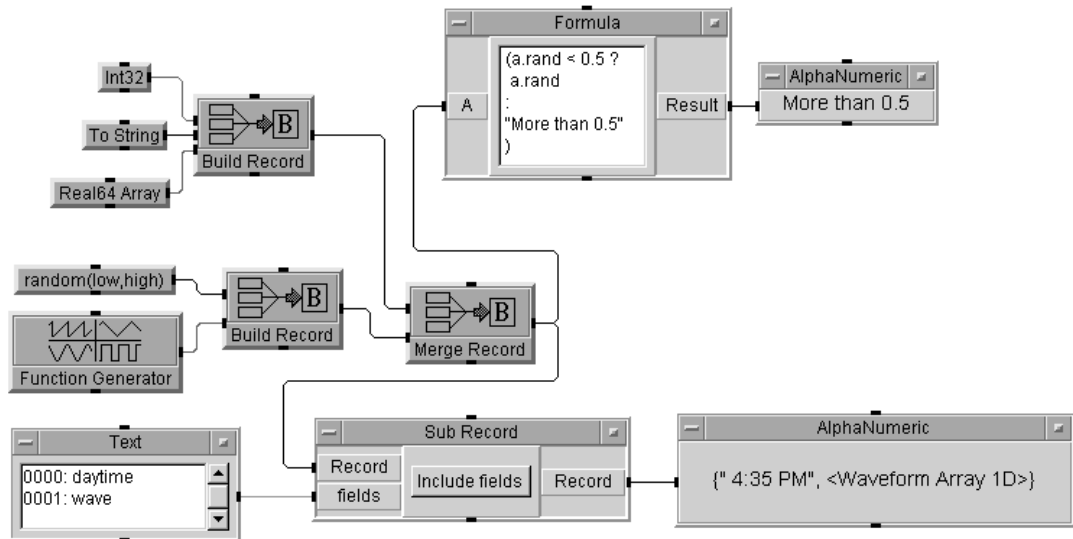


Abbildung 305 Records manipulieren, Schritt 2

## Wichtige Punkte

**Verwenden eines bedingten Ausdrucks:** VEE unterstützt einen bedingten Ausdruck; dies bietet eine effiziente Möglichkeit, eine If-Then-Else-Aktion zu implementieren. Der bedingte Ausdruck in diesem Formula-Objekt wird als „triadischer Ausdruck“ bezeichnet. Er lautet `(a.rand < 0.5 ? a.rand : "More than 0.5")`. Beachten Sie, dass es sich dabei um einen einzigen vollständigen Ausdruck handelt; Sie können diesen Ausdruck mit Zeilenendezeichen im Formula-Objekt wie gezeigt schreiben. Wenn das Formula-Objekt mehrere Ausdrücke enthält, werden die Ausdrücke durch Semikolons (;) getrennt.

**Das Objekt „Sub Record“ (Teildatensatz):** Beachten Sie das Text-Array der Felder an den Eingangs-Pins des Sub Record. Wenn Sie das Objekt Sub Record so konfigurieren, dass es Felder enthält, gibt es einen Record aus, der nur die angegebenen Felder enthält.

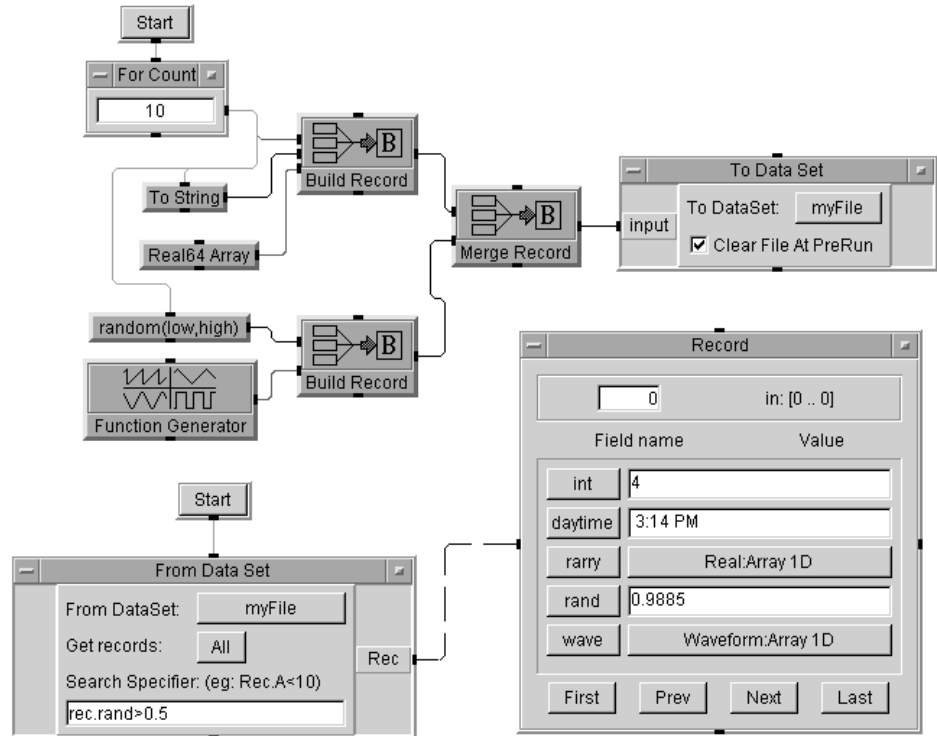
Records manipulieren, Schritt 3

Ersetzen Sie den ganzzahligen Eingang für das erste Feld durch ein Objekt For Count und durchlaufen Sie 10 Iterationen. Vergewissern Sie sich, dass Sie bei jeder Iteration einen „ping“ auf den Zufallszahlen-Generator und auf die Uhrzeitfunktion ausführen. Senden Sie den vollständigen Record an ein Objekt To DataSet. Rufen Sie in einem separaten Thread alle Records aus dem Dataset auf, in denen der Zufallswert größer als 0.5 ist. Legen Sie die resultierenden Records in einer Record-Konstanten ab.

**Tipp:** Sie benötigen einen Steuer-Pin für einen Default Value (Standardwert ) am Objekt Record Constant.

## Lösung – Records manipulieren, Schritt 3

Abbildung 306 zeigt eine Lösung zum Manipulieren von Records, Schritt 3.



**Abbildung 306** Records manipulieren, Schritt 3

### Wichtige Punkte

**Das Objekt „To DataSet“:** Die Option Clear File at PreRun löscht die Datei nur, bevor zum ersten Mal Daten gesendet werden. Beachten Sie, dass das Programm 10 verschiedene Records sequenziell an die gleiche Datei sendet, die an die Datei angehängt werden.

**Das Objekt „From DataSet“:** Dieses Objekt wird so konfiguriert, dass es alle Records abrufen, bei denen das Feld rand größer als 0.5 ist. In diesem Fall erfüllen fünf der zehn Records dieses Kriterium, und der erste Record wird mit der Indexnummer 0 angezeigt.

## Tests sequenzieren

Den Test Sequencer verwenden, Schritt 1

Erstellen Sie eine einfache UserFunction mit dem Namen UpperLimit als Einblendfenster mit einem Real64 Slider und einem Objekt Confirm (OK). Senden Sie die Ausgabe des Slider an eine globale Variable mit dem Namen UpLimit und auch an einen Ausgangsanschluss. Erstellen Sie ein Sequencer-Objekt und konfigurieren Sie **test1** im Sequencer als EXEC-Transaktion, die UpperLimit aufruft.

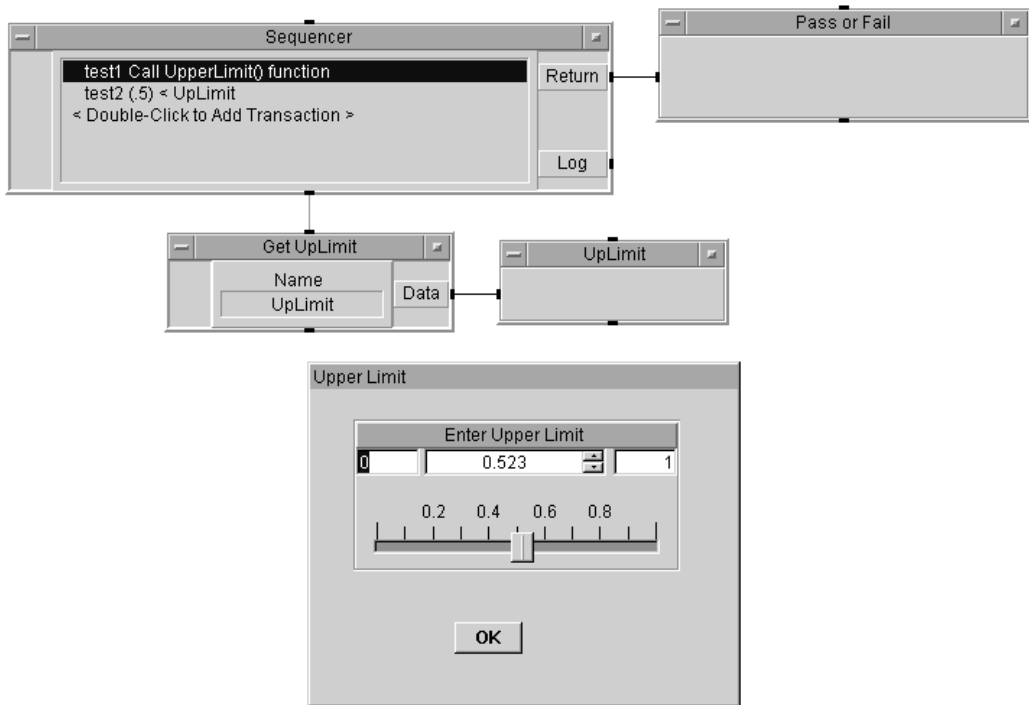
Erstellen Sie eine weitere Funktion mit dem Namen AddRand, die den Test simuliert, den Sie aufrufen können. Diese Funktion sollte einen Eingangswert zu einem Zufallswert (0 bis 1) addieren. Sie hat einen Eingangs-Pin und einen Ausgangs-Pin.

Erstellen Sie vom Sequencer aus **test2**, um AddRand aufzurufen und eine Null zu senden. Testen Sie den Rückgabewert und vergleichen Sie ihn mit einem Grenzwert, der kleiner ist als der globale UpLimit-Wert. Falls der Test erfolgreich ist, geben Sie "PASS" + test2.result + test2.result zurück. Schlägt der Test fehl, geben Sie "FAILED" + test2.result zurück. Legen Sie eine Alphanumeric-Anzeige am Return-Pin des Sequencer ab.

Führen Sie nach dem Sequencer-Objekt einen „Ping“ auf das Objekt Get Variable (UpLimit) und eine weitere Alphanumeric-Anzeige aus. Führen Sie das Programm mehrmals aus.

Lösung – Den Test Sequencer verwenden, Schritt 1

Abbildung 307 zeigt eine Lösung für den ersten Schritt zur Verwendung des Sequencer.



**Abbildung 307** Den Test Sequencer verwenden, Schritt 1

Wichtige Punkte

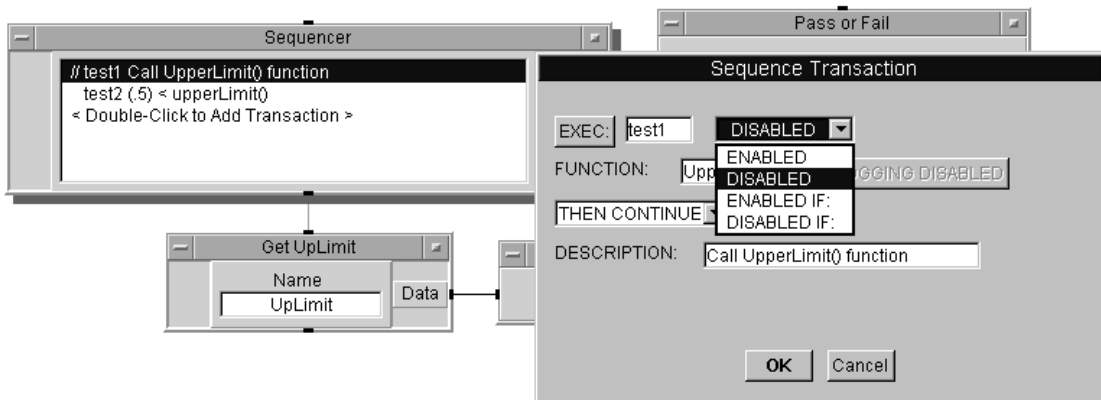
**Einstellen von globalen Variablen mit einer UserFunction:** Eine typische Verwendung der ersten Sequencer-Transaktion ist der Aufruf einer UserFunction, die die Globalen Variablen festlegt wie in diesem Fall. Anschließend können Sie diese Variablen in allen nachfolgenden Tests verwenden, wie hier gezeigt.

**Der Return-Pin des Sequencer:** Der Return-Pin in diesem Beispiel liefert eine Meldung PASS oder FAIL (erfolgreich oder fehlgeschlagen) sowie den Testwert. Sie können mit diesem Pin eine beliebige Meldung und einen Wert aus einem bestimmten Test liefern.

Den Test Sequencer verwenden, Schritt 2

Deaktivieren Sie den ersten Test. Wenn Sie die globale Variable nicht auch an einer anderen Stelle benötigen, können Sie die Funktion UpperLimit direkt aufrufen. Ändern Sie test2, sodass der Rückgabewert von AddRand(0) mit dem Ergebnis der Funktion UpperLimit verglichen wird.

**Tipp:** Verwenden Sie zum Deaktivieren des ersten Tests das Fenster Sequencer Transaction, wie in Abbildung 308 gezeigt.

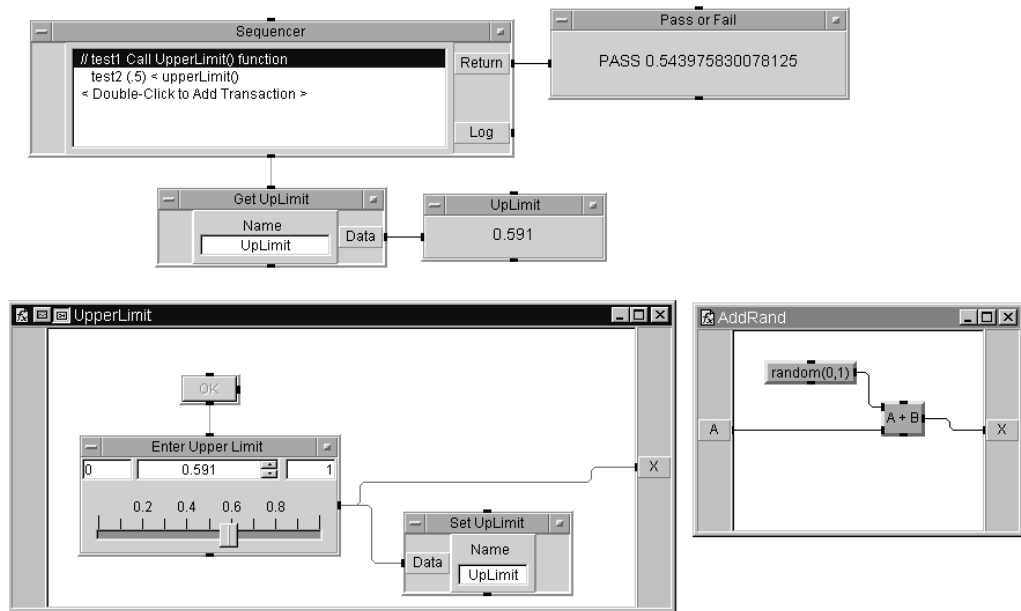


**Abbildung 308** Deaktivieren des ersten Tests in der Sequenz

Beachten Sie, dass in Abbildung 308 der erste Test im Sequencer mit den beiden Schrägstrichen „auskommentiert“ und damit deaktiviert ist.

Lösung – Den Test Sequencer verwenden, Schritt 2

Abbildung 309 zeigt eine Lösung zur Verwenden des Test-Sequencer, Schritt 2.



**Abbildung 309** Den Test Sequencer verwenden, Schritt 2

Wichtige Punkte

**Die UserFunction in einem Ausdrucksfeld:** In diesem Beispiel können Sie, statt ein Testergebnis mit der Variablen UpLimit zu vergleichen, den Funktionsnamen UpperLimit() in dem Ausdrucksfeld eingeben, in das die Variable geschrieben wird.

Den Test Sequencer verwenden, Schritt 3

Bearbeiten Sie die Transaktion **test2 Sequencer**, die die VEE-Funktion **random (0,1)** aufruft. Vergleichen Sie das Ergebnis mit einem Grenzwert kleiner als 0 . 5. Schneiden Sie die Transaktion test1 aus und fügen Sie sie ein, bis Sie insgesamt vier Tests haben.



Erstellen Sie ein Programm, mit dem der Sequencer fünf Mal ausgeführt wird. Zeichnen Sie die Daten in einem Dataset von Records auf, und erfassen Sie die Daten in einem Array. Ermitteln Sie mit diesem Array den kleinsten und größten Wert, den Mittelwert und die Standardabweichung der Ergebnisse des zweiten Tests.

Lösung – Den Test Sequencer verwenden, Schritt 3

Abbildung 310 zeigt eine Lösung für Schritt 3.

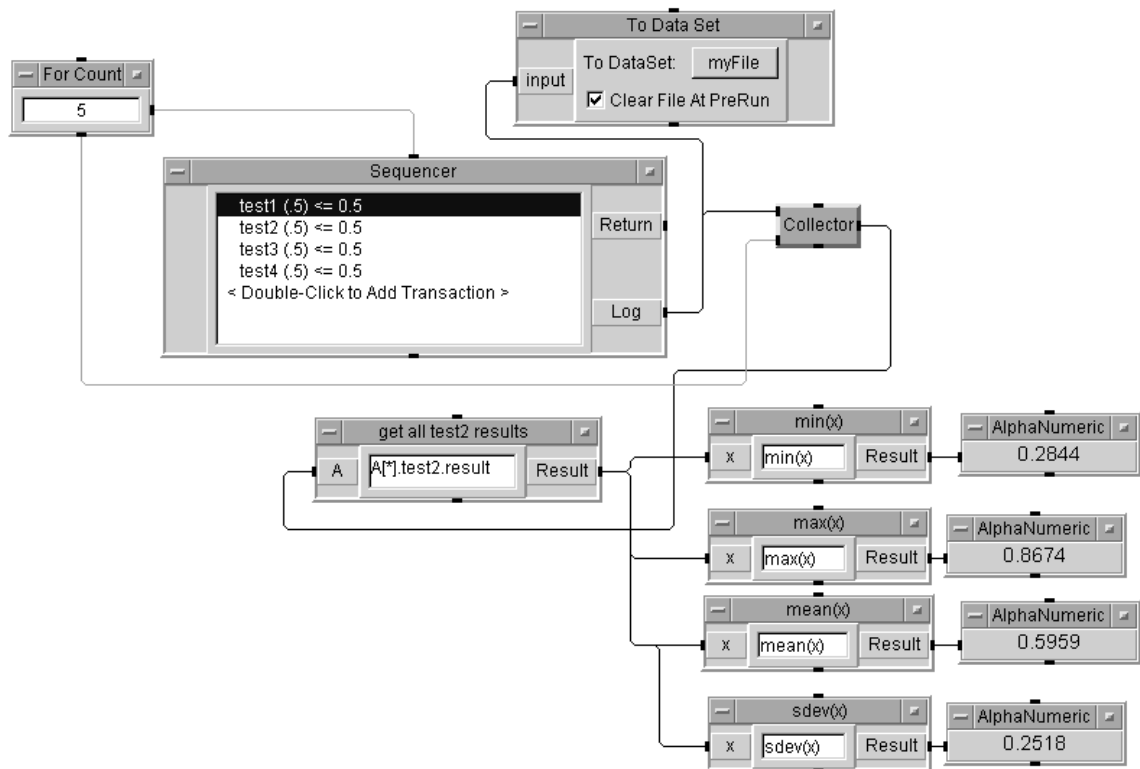


Abbildung 310 Den Test Sequencer verwenden, Schritt 3

### Wichtige Punkte

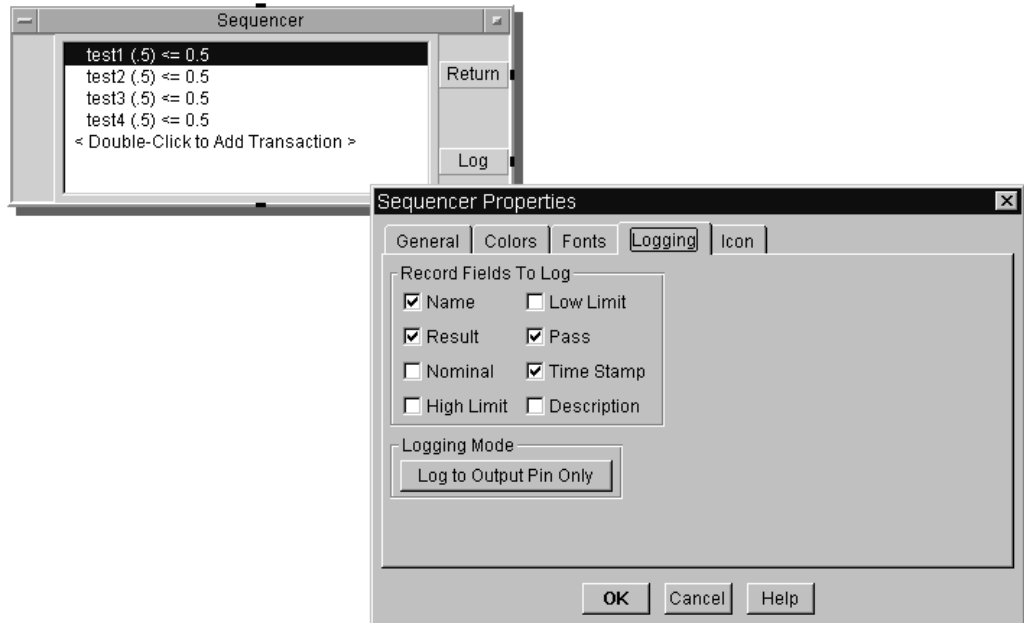
**Das Datenformat für mehrere Durchläufe des Sequencer (erster Thread):** Wenn der Sequencer ein Mal ausgeführt wird, gibt er einen Record of Records (Datensatz aus Datensätzen) aus. Der erste Record enthält Feldnamen, die den Testnamen entsprechen, anschließend enthält jedes Feld einen Record mit den verschiedenen Datenteilen für diesen bestimmten Test. Wenn der Sequencer mehrmals ausgeführt wird, kann jeder Record of Records einem Array hinzugefügt werden; diese Arrays können anschließend untersucht werden. Wenn Sie das Format `<record>[*].<record>.<field>` im Formula-Objekt verwenden, erhalten Sie ein Array aus Daten. In diesem Fall erhalten Sie ein Array aus Real-Werten mit den Testergebnissen aus fünf Durchläufen von test2. Sie können anschließend den kleinsten und größten Wert, den Mittelwert und die Standardabweichung aus diesem Array ermitteln. Sie können auch einen einzelnen Durchlauf von test2 angeben, indem Sie ein bestimmtes Element im Array der Records of Records angeben. Wenn Sie beispielsweise das Ergebnis des ersten Durchlaufs von test2 erhalten wollen, verwenden Sie den Ausdruck: `A[0].test2.result`.

Den Test Sequencer verwenden, Schritt 4

Fügen Sie dem Aufzeichnungs-Record ein Zeitstempelfeld hinzu. Fügen Sie eine Verzögerung hinzu, sodass jeder Schritt mit einer Sekunde Abstand erfolgt. Rufen Sie in einem separaten Thread alle Ergebnisse von test2 ab und senden Sie sie an eine Record-Konstante.

**Das Objekt „Delay“ (Erster Thread):** Dieses Objekt enthält den Ausführungsablauf für die angegebene Anzahl von Sekunden. Hier wird es verwendet, um sicherzustellen, dass sich die Werte der Zeitstempel zwischen den einzelnen Durchläufen des Sequencer unterscheiden.

**Einen Zeitstempel hinzufügen:** Öffnen Sie zum Hinzufügen eines Zeitstempels das Sequencer-Objektmenü und wählen Sie das **Properties** ⇒ **Logging**, um **Record Fields to Log** ⇒ **Time Stamp** zu wählen. Abbildung 311 zeigt das Register **Properties** ⇒ **Logging**.



**Abbildung 311** Einen Zeitstempel zum Aufzeichnungs-Record hinzufügen

Lösung – Den Test Sequencer verwenden, Schritt 4

Abbildung 312 zeigt eine Lösung für Schritt 4 von „Den Test Sequencer verwenden“.

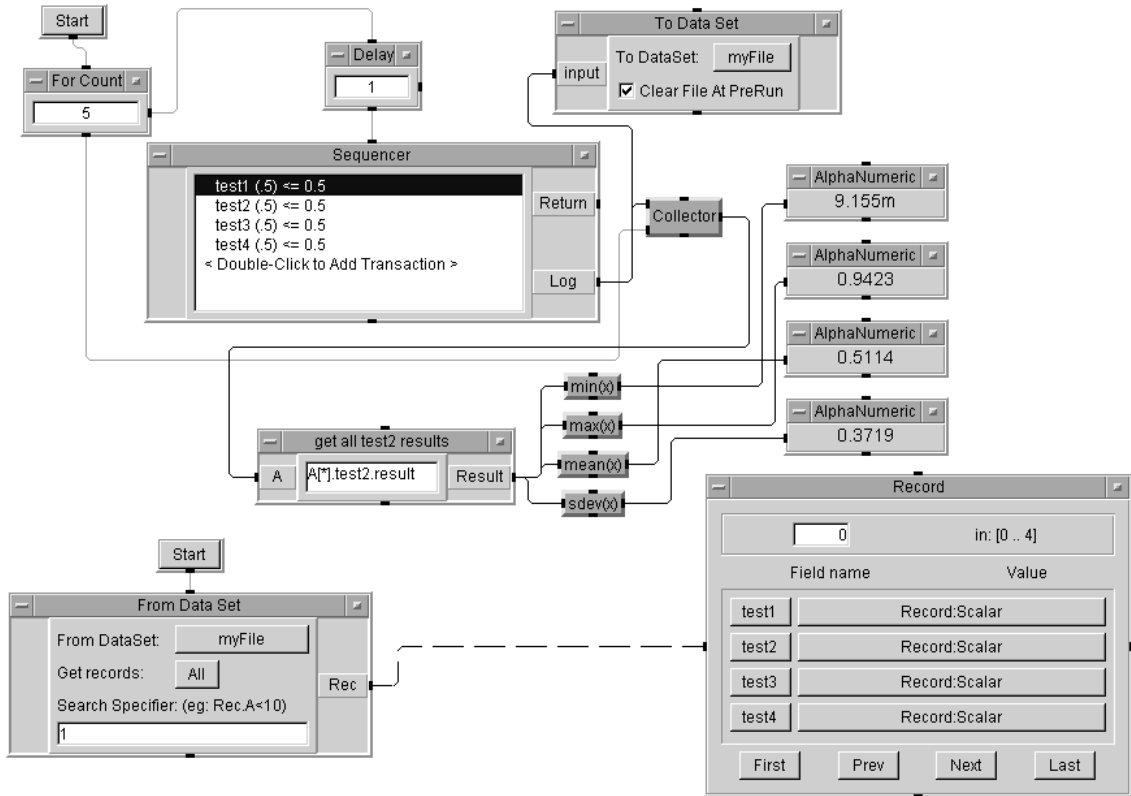
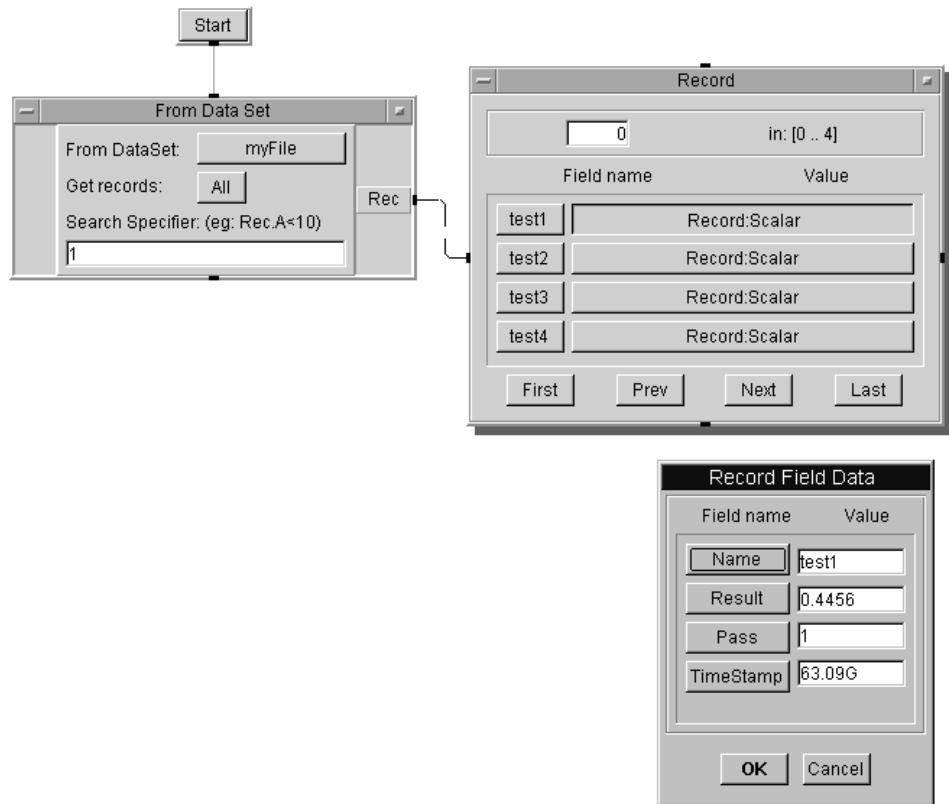


Abbildung 312 Den Test Sequencer verwenden, Schritt 4

**Tip:** Klicken Sie zum Anzeigen eines Records auf das Feld **Record** ⇒ **Record: Scalar** für einen der Tests, um das Dialogfeld Record Field Data zu öffnen. Abbildung 313 zeigt das Dialogfeld Record Field Data.



**Abbildung 313** Einen Record prüfen

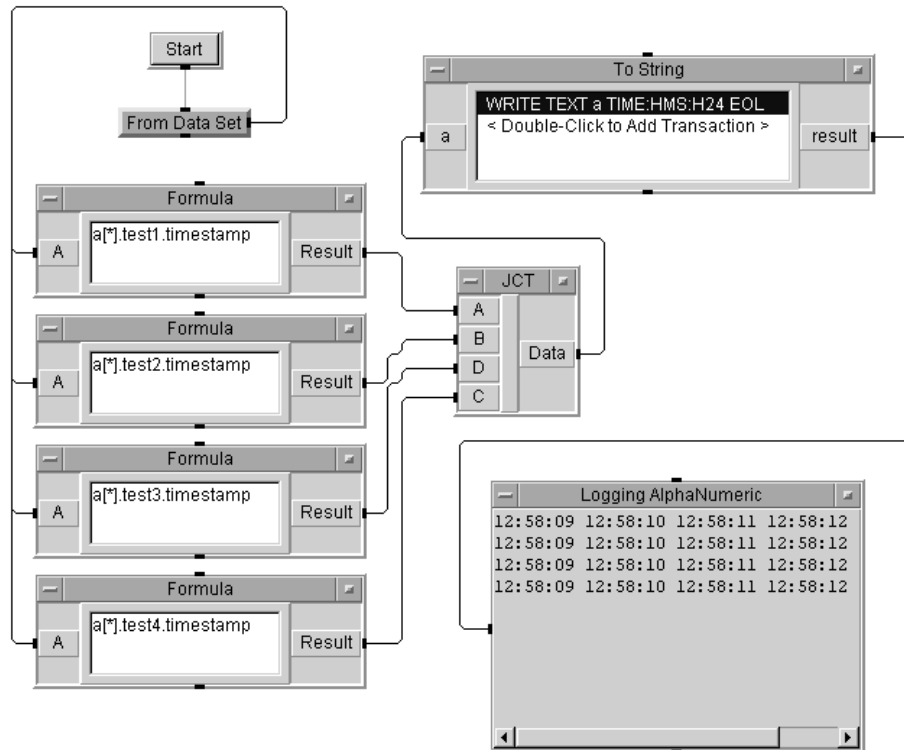
Den Test Sequencer verwenden, Schritt 5

Geben Sie die Zeitstempelfelder aus den Record-Feldern an eine Logging Alphanumeric-Anzeige aus.

**Tipp:** Verwenden Sie vier Formula-Objekte (einen für jeden Test). Zum Anzeigen aller vier Formula-Ergebnisse in einer Logging Alphanumeric-Anzeige fügen Sie ein Junction-Objekt hinzu. Verwenden Sie ein Objekt To String zum Formatieren des 63G-Zeitstempelwerts in eine leichter lesbare Zeichenfolge.

Lösung – Den Test Sequencer verwenden, Schritt 5

Abbildung 314 zeigt den Programm-Thread zum Ausgeben der Zeitstempel an eine Anzeige, Schritt 5 von „Den Test Sequencer verwenden“.



**Abbildung 314** Den Test Sequencer verwenden, Schritt 5

Wichtige Punkte

**Zeitstempelformate umwandeln:** Das Objekt To String vor Logging AlphaNumeric wandelt die Zeitstempel aus einem Real-Format in ein Time Stamp-Format um, sodass sie besser lesbar sind.

Den Test Sequencer verwenden, Schritt 6

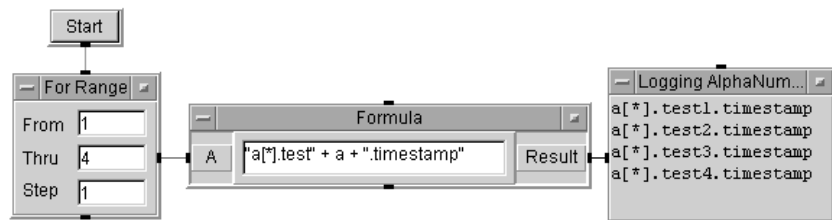
Wenn der Sequencer viele Tests umfasst, kann es sehr aufwändig sein, viele einzelne Formula-Objekte mit Verbindung zu einem Junction-Objekt zu verwenden. Sie können stattdessen eine Formula verwenden, die einen Ausdruck enthält, zur Ausführungszeit einen Ausdruck generieren und eine Schleife durch die möglichen Ausdrücke durchlaufen.

Das Beispiel generiert zunächst die Ausdrucks-Zeichenfolgen.

Verwenden Sie in einem separaten Thread eine Schleife („Loop“) und eine Formula, um eine Testausdrucks-Zeichenfolge zu generieren. Geben Sie die Informationen als Zeichenfolge in einer Logging Alphanumeric-Anzeige aus. Die in der Formula generierte Zeichenfolge sollte „a[\*].test<x>.timestamp“ lauten, wobei <x> von 1 bis 4 reicht.

Lösung – Den Test Sequencer verwenden, Schritt 6

Abbildung 315 zeigt eine Lösung für Schritt 6.



**Abbildung 315** Den Test Sequencer verwenden, Schritt 6

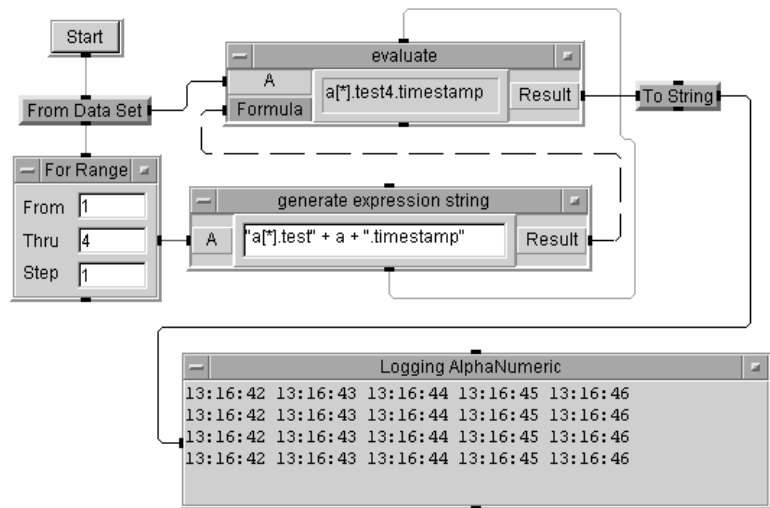
Den Test Sequencer verwenden, Schritt 7

Verwenden Sie jetzt die in Schritt 6 erstellte Schleife und die the Formula, und ersetzen Sie die vier Formulas und die Junction im vorigen Schritt durch die Schleife und Formula. Außerdem soll die erstellte Zeichenfolge ausgewertet werden. Senden Sie die generierte Zeichenfolge (den Ausdruck "a[\*].test<x>.timestamp") an eine Formula, die zur Ausführungszeit ausgewertet werden soll.

**Formula-Steuer-Pin am Formula-Objekt:** Die auszuwertende Formula wird von der Formula in der Schleife generiert. Sie können ein zweites Formula-Fenster mit einem Steuereingang für seinen Formula-Ausdruck erstellen. Der Ausdruck, den die zweite Formula auswertet, wird zur Ausführungszeit generiert.

Lösung – Den Test Sequencer verwenden, Schritt 7

Abbildung 316 zeigt eine Lösung für Schritt 7.



**Abbildung 316** Den Test Sequencer verwenden, Schritt 7

Wichtige Punkte

- Das Objekt To String wird noch immer verwendet, um den Real64-Wert im Zeitstempelformat zu formatieren.
- Beachten Sie die Sequenzlinie zwischen der ersten Formula („Generierung“) und der zweiten Formula („Auswertung“). Auf diese Weise wird sichergestellt, dass die zweite Formula nicht ausgeführt wird, bis sie die neue auszuwertende Zeichenfolge erhält.

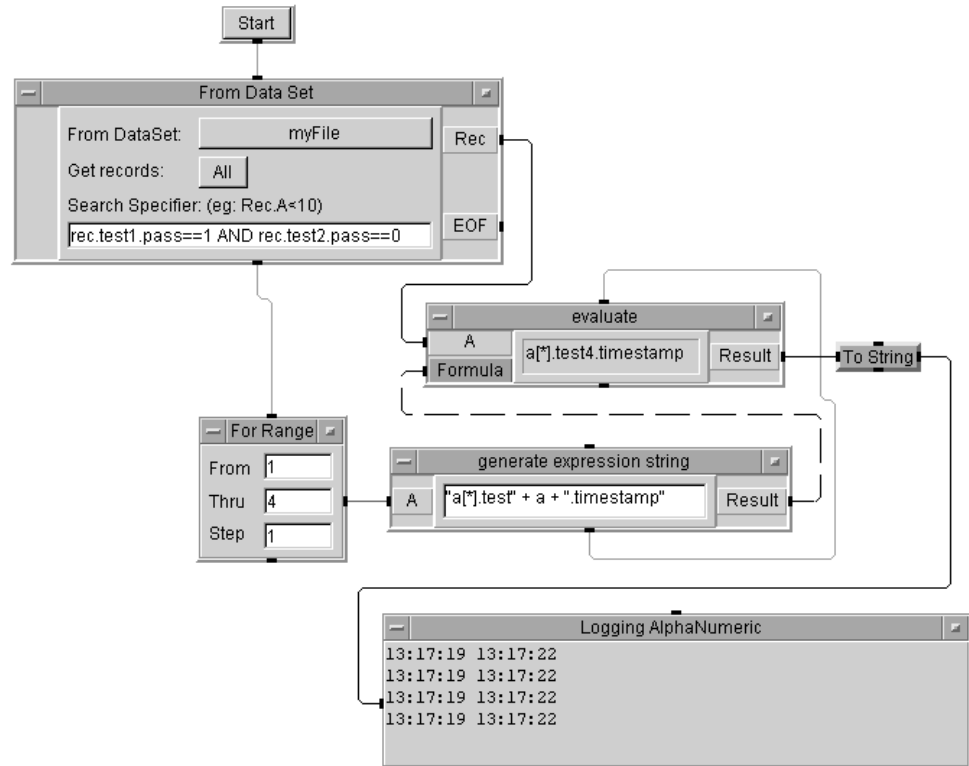
Den Test Sequencer verwenden, Schritt 8

Zeigen Sie nur die Records an, in denen Test 1 erfolgreich war und Test 2 fehlgeschlagen ist.



## Lösung – Den Test Sequencer verwenden, Schritt 8

Abbildung 317 zeigt eine Lösung für den letzten Schritt beim Verwenden des Test Sequencer.



**Abbildung 317** Den Test Sequencer verwenden, Schritt 8

## Wichtige Punkte

**Der EOF-Pin am Objekt „From Data Set“ (zweiter Thread):** Der EOF-Pin wird hinzugefügt, falls keine Records vorliegen, die die Kriterien erfüllen. In diesem Fall löst der EOF-Pin aus, statt dass VEE das Programm mit einer Fehlermeldung anhält.

**Der bedingte Ausdruck im Objekt „From Data Set“ (Zweiter Thread):** Der Ausdruck lautet *(Rec.test1.pass==1) OR (Rec.test2.pass==0)* mit jeweils dem gleichen Format *<Record>.<Record>.<Feld>*. Rec ist der Name der einzelnen Records in dem Dataset beim Lesen und Testen. Test1 und test2 geben an, welche Tests untersucht werden sollen, und der übergebene Feldname ist der Standardname für den Pass-Fail-Indikator (1 oder 0), der von VEE zugeordnet wird. (Sie aktivieren oder deaktivieren verschiedene Felder für alle Tests durch Auswählen des Registers Logging im Fenster Sequencer Properties.)

# Glossar

In diesem Glossar werden die im vorliegenden Handbuch verwendeten Begriffe erläutert. Ein vollständiges Glossar der VEE-Begriffe finden Sie in der Online-Hilfe unter **Help** ⇒ **Contents and Index**. Wählen Sie dann **Reference** und anschließend **Glossary**. Klicken Sie in diesem Glossar auf einen Begriff, um seine Definition anzuzeigen. Wenn Sie die Definition gelesen haben, klicken Sie auf eine beliebige Stelle, um die Definition auszublenden.

**Abgeblendete Funktion** Ein Menübefehl, der grau statt schwarz dargestellt wird; dies weist darauf hin, dass der Befehl bzw. die betreffende Funktion nicht aktiv oder nicht verfügbar ist. Dialogelemente wie Schaltflächen, Kontrollkästchen oder Optionsfelder können ebenfalls abgeblendet sein.

**Anschluss** Die interne Darstellung eines Pins, die Informationen zu dem Pin und dem an diesem Pin anliegenden Datencontainer anzeigt. Doppelklicken Sie auf einen Anschluss, um die Containerinformationen anzuzeigen.

**Ansichten** VEE zeigt ein Programm in einer von zwei Ansichten an: Die Fensteransicht bietet eine Benutzerschnittstelle zu einem VEE-Programm, und die Detailansicht bietet ein Fenster zur Entwicklung eines VEE-Programms.

**Arbeitsbereich** Ein Bereich im Hauptfenster (oder den Fenstern `UserObject` und `UserFunction`), in dem Sie VEE-Objekte platzieren und verbinden, um ein VEE-Programm zu erstellen.

**Arbeitsbereich** Ein Bereich im VEE-Fenster, das die Programmier- oder Bearbeitungsfenster wie Hauptfenster, `UserObject` und `UserFunction` enthält. Diese Fenster enthalten Arbeitsbereiche, in denen Sie VEE-Objekte platzieren und verbinden.

**Ausdruck** Eine Gleichung in einem Eingabefeld, die Namen von Eingangsanschlüssen, globalen Variablen, mathematischen Funktionen und benutzerdefinierte Funktionen enthalten kann. Ein Ausdruck wird zur Laufzeit eines Programms ausgewertet. Ausdrücke sind zulässig in den Objekten **Formula, If/Then/Else, Get Values, Get Field, Set Field, Sequencer** sowie in **Dialog Box-** und in E/A-Transaktionsobjekten.

**Bildlaufleiste** Ein rechteckiger Balken, mit deren Hilfe im aktiven Fenster oder in der aktiven Liste ein vertikaler oder horizontaler Bildlauf durchgeführt werden kann, beispielsweise ein Bildlauf durch den Arbeitsbereich oder durch eine Liste von Datendateien oder anderen Auswahloptionen in einem Dialogfeld.

**Bildlaufpfeil** Durch einen Klick auf den Bildlaufpfeil wird im aktiven Fenster oder in der aktiven Liste ein vertikaler oder horizontaler Bildlauf durchgeführt, beispielsweise ein Bildlauf durch den Arbeitsbereich oder durch eine Liste von Datendateien oder anderen Auswahloptionen in einem Dialogfeld.

**Clone** Ein Menübefehl in den VEE-Objektmenüs, mit dem Objekte und ihre Verbindungen dupliziert werden. Beim Klonen wird eine Kopie der Objekte sowie ihrer Verbindungen in die Zwischenablage übertragen. Der Befehl Clone kopiert sämtliche Attribute der Original-Objekte einschließlich Pins, Parameter und Größe.

**Container** Siehe **Datencontainer**.

**Datenausgangs-Pin** Ein Verbindungspunkt an der rechten Seite eines Objekts, der den Datenfluss an das nächste Objekt weiterleitet und die Ergebnisse der Operation des ersten Objekts an das nächste Objekt weiterleitet.

**Dateneingangs-Pin** Ein Verbindungspunkt an der linken Seite eines Objekts, der den Datenfluss in das betreffende Objekt ermöglicht.

**Datencontainer** Das Datenpaket, das über die Leitungen übertragen und von den Objekten verarbeitet wird. Jeder Datencontainer enthält Daten und den Datentyp, die Datenform und die Zuordnungen (sofern vorhanden).

**Datenfluss** Der Verlauf der Daten durch und zwischen VEE-Objekten. Daten fließen von links nach rechts durch die Objekte; ein Objekt wird jedoch erst dann ausgeführt, wenn an allen seinen Dateneingangs-Pins Daten anliegen. Daten werden vom Datenausgangs-Pin eines Objekts an den Dateneingangs-Pin des nächsten Objekts weitergegeben. Der Datenfluss ist der wichtigste Faktor, der die Ausführung eines VEE-Programms festlegt.

**Datenform** Jeder Datencontainer hat eine Form und einen Typ. Die Datenform kann entweder ein Skalar oder ein ein- oder mehrdimensionales Array sein. In VEE wird ein eindimensionales Array als „Array 1D“ bezeichnet, ein zweidimensionales Array als „Array 2D“ usw.

**Datentyp** Jeder Datencontainer hat einen Typ und eine Form. VEE unterstützt eine Vielzahl von Datentypen einschließlich Text, Real64, Real32 und Int32.

**Detailansicht** Die Ansicht eines VEE-Programms, die alle Objekte und die Verbindungslinien zeigt.

**Direct I/O (Objekt)** Ein Objekt zur Instrumentensteuerung, das VEE eine direkte Steuerung eines Instruments ohne einen Instrumententreiber ermöglicht.

**Doppelklicken** Zweimaliges Drücken und Loslassen der Maus-taste in schneller Folge. Das Doppelklicken ist meist eine Abkürzung zum Auswählen und Ausführen einer Aktion. Wenn Sie beispielsweise in dem Dialogfeld, das über **File** ⇒ **Open** geöffnet wird, auf einen Dateinamen doppelklicken, wird die Datei gewählt und geöffnet.

**Dropdown-Liste** Eine Liste von Auswahloptionen, die durch Klicken auf den Pfeil auf der rechten Seite eines Dropdown-Listenfelds geöffnet wird.

**Dropdown-Listefeld** Ein Feld in einem Objekt oder Dialogfeld, das eine Auswahl in einer Dropdown-Liste ermöglicht.

**Eigenschaften** Objekteigenschaften sind Attribute von VEE-Objekten, die Sie im **Properties**-Fenster (über das Objektmenü) ändern können, beispielsweise Farben, Schriftarten und Titel.

**Einfügemarke** Eine Marke (Zirkumflex) in einen Eingabefeld, die anzeigt, an welcher Stelle alphanumerische Daten eingefügt werden, wenn Sie Informationen über die Tastatur eingeben.

**Eingabefeld** Ein Eingabefeld ist meist Teil eines Dialogfelds oder eines bearbeitbaren Objekts und wird zur Eingabe von Daten verwendet. Der Inhalt eines Eingabefelds kann geändert werden, wenn sein Hintergrund weiß ist.

**Einstellungen** Einstellungen sind Attribute der VEE-Umgebung, die Sie durch einen Klick auf die Schaltfläche **Default Preferences** (in der Symbolleiste) oder durch **File** ⇒ **Default Preferences** ändern können. Sie können beispielsweise die Standardfarben und -schriftarten sowie das Zahlenformat ändern.

**Fensteransicht** Die Ansicht eines VEE-Programms oder eines UserObject bzw. einer UserFunction, in der nur die Objekte angezeigt werden, die der Benutzer zur Ausführung des Programms und zum Anzeigen der resultierenden Daten benötigt. Sie können mithilfe von Fensteransichten eine Benutzerschnittstelle für Ihr Programm erstellen.

**Gruppenfenster** Ein Gruppenfenster in Microsoft Windows ist ein Fenster, das Symbole für eine Gruppe von Anwendungen enthält. Jedes Symbol startet eine Anwendung in der Gruppe.

**Hauptfenster** Ein Fenster, das den primären Arbeitsbereich enthält, in dem Sie ein VEE-Programm entwickeln. Der Arbeitsbereich dieses Fensters befindet sich im Bereich des VEE-Fensters.

**Hauptmenü** Ein Menü, das aus der Menüleiste aufgeklappt wird, wenn Sie mit dem Mauszeiger auf den Namen eines Menüs zeigen und mit der linken Maustaste klicken.

**Hypertext** Ein System zum Verbinden von Themen, über das Sie ein verwandtes Thema anzeigen können, wenn Sie weitere Informationen erhalten möchten. In Online-Hilfesystemen werden Hypertext-Verbindungen normalerweise durch unterstrichenen Text gekennzeichnet. Wenn Sie auf einen solchen Text klicken, werden die zugehörigen Informationen angezeigt.

**Kaskadierendes Menü** Ein Untermenü eines Haupt- oder Kontextmenüs, das weitere Auswahlmöglichkeiten bietet.

**Klicken** Drücken und Loslassen der Maustaste. Durch Klicken wird normalerweise ein Menübefehl oder ein Objekt im VEE-Fenster gewählt. Siehe auch **Doppelklicken** und **Ziehen**.

**Komponente** Eine einzelne Instrumentenfunktion oder ein Messwert in einem VEE-Instrumentenfenster oder einem Komponententreiber. Ein Treiber für einen Voltmeter enthält beispielsweise Komponenten, die den Bereich, die Auslöserquelle und den letzten Lesevorgang aufzeichnen.

**Komponententreiber** Ein Instrumenten-Steuerobjekt, das Werte aus speziell ausgewählten Komponenten schreibt oder in diese Komponenten schreibt. Verwenden Sie Komponententreiber, um ein Instrument mit einem Treiber zu steuern, indem Sie die Werte nur weniger Komponenten zu einem Zeitpunkt setzen. (Komponententreiber unterstützen keine Kopplung.)

**Kontext** Eine Ebene des Arbeitsbereichs, die weitere Ebenen von Arbeitsbereichen (beispielsweise verschachtelte User-Objects) enthalten kann, von diesen jedoch unabhängig ist.

**Kontextmenü** Ein Menü, das geöffnet wird, wenn Sie mit der rechten Maustaste klicken. Sie können beispielsweise das Menü **Edit** öffnen, indem Sie mit der rechten Maustaste auf einen leeren Bereich im Arbeitsbereich klicken. Oder Sie öffnen das Objektmenü, indem Sie mit der rechten Maustaste auf einen nicht aktiven Bereich des Objekts klicken.

**Kontrollkästchen** Ein leicht zurückgesetztes Rechteck in VEE-Dialogfeldern, mit dem Sie eine Einstellung aktivieren können. Wenn Sie zum Aktivieren auf das Kästchen klicken, wird darin ein Markierungszeichen eingeblendet, das darauf hinweist, dass die betreffende Einstellung aktiviert wurde. Zum Aufheben der Auswahl klicken Sie einfach erneut auf das Kontrollkästchen.

**Mauszeiger** Das grafische Abbild, das der Bewegung der Maus zugeordnet ist. Der Mauszeiger ermöglicht das Auswählen und bietet Rückmeldungen zu einem bestimmten aktiven Prozess. VEE verwendet für die verschiedenen Verarbeitungsmodi Mauszeiger in verschiedenen Formen, z. B. Pfeil, Fadenkreuz oder Sanduhr.

**Maximieren, Schaltfläche** Eine Schaltfläche in einem User-Object-, einem UserFunction- oder im Hauptfenster, mit der das jeweilige Fenster auf den gesamten verfügbaren Arbeitsbereich ausgedehnt werden kann.

**Menüleiste** Die Leiste am oberen Rand des VEE-Fensters, das die Namen der Hauptmenüs enthält, in denen Sie Befehle und Objekte wählen können.

**Minimieren, Schaltfläche** Eine Schaltfläche in einem Objekt oder im VEE-Fenster, über die das Objekt bzw. das VEE-Fenster auf minimale Größe (Symbolgröße) verkleinert wird.

### Objekt:

- 1 Ein grafische Darstellung eines Elements in einem Programm, z.B. eines Instruments, eines Steuerelements, einer Anzeige oder eines mathematischen Operators. Ein Objekt wird im Arbeitsbereich platziert und mit anderen Objekten verbunden, um ein Programm zu erstellen.
- 2 Ein Datentyp für die ActiveX-Automatisierung und -Steuerelemente.

**Objektmenü** Das einem Objekt zugeordnete Menü, das Befehle enthält, die speziell auf das jeweilige Objekt zugeschnitten sind (z. B. Verschieben, Größe ändern, Kopieren und Löschen des Objekts). Klicken Sie zum Öffnen des Objektmenüs in der linken



oberen Ecke des Objekts auf die Schaltfläche „Objektmenü“, oder klicken Sie mit der rechten Maustaste, während der Mauszeiger auf das Objekt zeigt.

**Objektmenü, Schaltfläche** Die Schaltfläche in der linken oberen Ecke eines Objekts, das in der offenen Ansicht dargestellt wird; beim Klicken auf diese Schaltfläche wird das Objektmenü geöffnet.

**Offene Ansicht** Die Darstellung eines VEE-Objekts, die mehr Details als die Symbolansicht anzeigt. Die meisten Objekte zeigen in der offenen Ansicht Felder an, über die die Arbeitsweise des Objekts geändert werden kann.

**Panel-Treiber** Ein Objekt zur Steuerung, das eine Abstimmung aller Funktionseinstellungen im entsprechenden physischen Instrument mit den Einstellungen des Steuerfensters in der offenen Ansicht des Objekts erzwingt.

**Pin (oder Pins)** Ein externer Verbindungspunkt an einem Objekt, an dem eine Linie angeschlossen werden kann.

**Program Explorer** Eine Einrichtung im VEE-Fenster, die das Untersuchen eines Programms ermöglicht, insbesondere der Teile eines Programms, die auf dem Bildschirm nicht sichtbar sind.

**Programm** In VEE ein grafisches Programm, das aus einer Reihe von Objekten besteht, die mit Linien verbunden sind. Das Programm stellt normalerweise eine Lösung zu einem technischen Problem dar.

**Schaltfläche** Ein grafisches Objekt in VEE, das einen beweglichen Schalter oder eine Taste simuliert. Schaltflächen werden in der Anzeige als leicht erhöhte Tasten dargestellt. Wenn Sie VEE eine Schaltfläche „drücken“ (mit der Maus auf sie klicken), wird eine Aktion ausgeführt. (Dies kann auch auf die linke oder rechte Maustaste verweisen.)

**Schriftart** VEE ermöglicht das Ändern der Schriftart, des Schriftgrads sowie des Schriftschnitts für die Anzeige verschiedener VEE-Objekte, Titel etc.

**Sequenzausgangs-Pin** Der *untere* Pin an einem Objekt. Wenn dieser Pin angeschlossen ist, wird er aktiviert, sobald die Ausführung des Objekts und die gesamte Datenweitergabe dieses Objekts abgeschlossen sind.

**Sequenzeingangs-Pin** Der *obere* Pin an einem Objekt. Wenn dieser Pin angeschlossen ist, wird die Ausführung des Objekts so lange angehalten, bis der Pin einen Container empfängt (bis ein „Ping“ ausgeführt wird).

**Statusfeld** Ein Feld mit Informationen, die nicht geändert werden können. Ein Statusfeld sieht aus wie ein Eingabefeld, hat jedoch einen grauen Hintergrund.

**Statusleiste** Eine Zeile am unteren Rand des VEE-Fensters, in der Informationen zum aktuellen Status und zu VEE angezeigt werden.

### Symbol:

- 1 Ein kleine, grafische Darstellung eines VEE-Objekts, beispielsweise eines Instruments, eines Steuerelementes oder einer Anzeige.
- 2 Ein kleine, grafische Darstellung einer Anwendung, einer Datei oder eines Ordners im Betriebssystem Microsoft Windows.

**Symbolleiste** Die rechteckige Leiste am oberen Rand des VEE-Fensters, die Schaltflächen für den schnellen Zugriff auf häufig verwendete Befehle enthält. Über die Schaltflächen werden Menübefehle, beispielsweise in den Menüs **File**, **Edit**, **View**, **Device** und **Debug** ausgeführt.

**Titelleiste** Die rechteckige Leiste am oberen Rand der offenen Ansicht eines Objekts oder Fensters, die den Titel des Objekts oder Fensters enthält. Sie können die Titelleiste eines Objekts im Fenster **Properties** (über das Objektmenü erreichbar) ausblenden.

**Transaktion** Die Spezifikationen für die von bestimmten Objekten in VEE verwendete Ein- und Ausgabe (I/O). Beispiele hierzu sind die Objekte `To File`, `From File`, `Sequencer`, und `Direct I/O`. Transaktionen erscheinen in der offenen Ansicht dieser Objekte als Texte.

**UserObject** Ein Objekt, das eine Gruppe von Objekten zur Ausführung eines bestimmten Zwecks in einem Programm ein-kapseln kann. Ein UserObject ermöglicht die Anwendung von Top-Down-Design-Techniken beim Erstellen eines Programms und das Erstellen von benutzerdefinierten Objekten, die in einer Bibliothek gespeichert und wiederverwendet werden können.

**Wählen** Wählen eines Objekts, einer auszuführenden Aktion oder eines Menübefehls. Normalerweise verwenden Sie dazu die Maus.

**Weitergabe** Die Regeln, nach denen Objekte und Programme arbeiten bzw. ausgeführt werden. Siehe auch **Datenfluss**.

**Windows 98, Windows NT 4.0, Windows 2000, Windows XP** Von Microsoft Corporation entwickelte Betriebssysteme, unter denen VEE ausgeführt werden kann.

**Ziehen** Drücken und *gedrückt Halten* einer Maustaste, während die Maus bewegt wird. Durch Ziehen können Komponenten (z. B. ein Objekt oder eine Bildlaufleiste) bewegt werden.

**Zwischenablage** Der Speicherpuffer, der die Objekte enthält, die Sie ausgeschnitten oder kopiert haben. Sie können den Inhalt der Zwischenablage wieder in den Arbeitsbereich einfügen, wenn Sie in der Symbolleiste auf die Schaltfläche **Paste** klicken (**Edit** ⇒ **Paste**).

## **Glossar**

## Symbole

- \*.dll-Dateierweiterung 479
- \*.h-Dateierweiterung 476
- \*.vxe.Dateien 437, 447
- .NET- und IVI-Treiber 324
- .NET zum Abrufen von Dateinformationen 322
- .NET zum Ausführen von DateTime-Operationen 317
- .NET zur Wahl von Dateien 314
- .NET-Arraydatentypen in VEE-Datentypen konvertieren 306
- .NET-Datentypmodifizierer 311
- .NET-Programmierertipps 313
- .NET-Skalardatentypen in VEE-Datentypen konvertieren 302
- \_cdecl 475
- \_stdcall 475
- 24 Hour Zeitstempelformat 220

## A

- Abmessungen des Array 157
- Abrufen von Daten 413
- Abrufen von Daten mit dem Objekt „From File“ 223
- Access Array => Get Values 213
- ActiveX
  - Datentyp Variant 178
- Adresse, Schnittstelle 138
- Agilent VEE
  - Ausführungsfluss anzeigen 73
  - beenden 63
  - Compiler 482
  - Datenfluss anzeigen 73
  - Datenfluss im Programm 73
  - Eingangs-Pin-Verbindungen 82
  - Farb- und Schriftinformationen speichern 65
  - Fehlerbehebung 105
  - Go To 112
  - grafische Programme im Vergleich zu Textprogrammen 4
  - Objekte 30
  - Objekt-Pins und Anschlüsse 49
  - speichern der E/A-Konfiguration 65
  - Profiler 489
  - Programm schließen 65
  - Programme ausführen 58
  - Speichern von Programmen 63
  - Speichern von Testergebnissen 210
  - Starten von VEE 66
  - stoppen 66
  - Überblick 3

- Alarm, Benutzerschnittstelle erstellen 447
- Alphanumeric
  - Anzeigen 198
- Alphanumerische Anzeigen
  - zur Fehlerbehebung 109
- Anschlüsse 49
  - Anzeigen von Anschlussbeschriftungen 50
  - hinzufügen 51
  - Informationen abrufen 52
  - löschen 54
  - überprüfen 109
- ändern
  - Eigenschaften 50
  - Einstellungen 46
  - Größe eines Objekts 39
  - Objektansichten 33
- Ansicht
  - Detail 6, 421
  - offene Ansicht eines Objekts 33
  - Panel 95, 421
  - Symbolansicht eines Objekts 33
- Ansicht wechseln
  - Detail 95
- Anpassen
  - Testdatenanzeigen 200
- Anzeigen
  - Anschlüsse 50
  - Ausführungsfluss anzeigen 73
  - Datensatz mit Record Constant 354
  - Datenfluss anzeigen 73
  - Detailansicht 95
  - Fensteransicht 95
  - Noise Generator 201
  - Program Explorer 67
  - Programmverbindungen (Detailansicht) 6
  - Verbindungen zwischen Objekten 6
  - Wellenform 201
- Anzeigen von Testdaten 198
- Arbeitsbereich 24
  - alle Objekte verschieben 45
  - löschen 46
  - verschieben 46
  - verwalten 67
- Array
  - Abmessungen einstellen 157
  - Collector 212
  - Collector-Objekt 211
  - Dialogfeld I/O Transaction 157
  - Extrahieren von Array-Elementen mit Ausdrücken 213

- Extrahieren von Werten aus Testergebnissen 213
- Optimieren von Programmen 464
- Scalar-Menü 157
- Speichern von Testergebnissen 210
- UserFunction Arraystats 356
- Ausschneiden eines Objekts 37
- Ausdrücke
  - Aufrufen von UserFunctions 355
  - Formula, Objekt 187
  - Senden einer Ausdrucksliste an ein Instrument 153
- Ausdrucksfeld
  - DLLs aufrufen 477
- Aufrufen einer freigegebenen/statischen Methode 312
- Aufrufen einer Instanzmethode 311
- Aufrufen von UserFunctions 348
- aufrufen, UserFunction über Ausdruck 355
- Ausführen eines Programms 58
- Ausführung
  - Ausführungsfluss anzeigen 73
  - Datenfluss anzeigen 73
  - Datenfluss im VEE-Programm 73
  - Einblendfenster anzeigen 438
  - Execute Program 481
  - Modi 482
  - Reihenfolge im Programm 117
- Ausführungsmodi
  - Optimieren von Programmen 465
- Auswahl von Objekten aufheben 40
- Auswählen
  - Menüs 22
- Auswerten von Ausdrücken im Formelobjekt 185

## B

- Balkendiagramm 376
- Bearbeiten
  - Edit-Menü 43
  - Linien bereinigen 58
  - Objekte 43
  - UserFunction 348
- Bedienerschnittstelle
  - Fensteransicht erstellen 94
- Bedienerschnittstelle erstellen 94
- Beenden von VEE 65
- Beep
  - Anzeigen 198
- Beep-Objekt 448

- Benutzerschnittstelle 22
  - Fensteransicht erstellen 94
- Benutzerschnittstellen
  - Bitmap-Dateien importieren 425
  - Einblendfenster anzeigen 438
  - Farben und Schriftarten wählen 424
  - farbige Alarmmeldungen 424
  - Fensteransicht von Programmen 420
  - Füllanzeigen 424
  - für eine Suchoperation 244
  - Messelemente 424
  - Radio Buttons 441
  - Slider, Real64 73
  - Softkeys und Funktionstasten 434
  - Steuerelemente (Umschalter) 431
  - Tankanzeigen 424
  - Thermometer 424
- Benutzereingabe
  - Dialogfeld erstellen 87
- Bibliotheken
  - Befehl „Merge Library“ 361
  - Delete Library-Objekt 368
  - DLLs (Dynamic Link Libraries) 475
  - Import Library object 368
  - UserFunction 350
  - UserFunctions 361, 368
  - Zusammenführen von UserFunctions 361
- Bildschirmfarben 435
- Bildlaufleiste 45
- Bitmap-Dateien 425
- Byte Ordering 139

## C

- Call
  - Device, Call, Select Function 351
- Call Stack 113
- Call-Objekt, wann sind Klammern erforderlich 363
- Caution-Fenster in VEE-Programmen
  - Agilent VEE
    - Fehlermeldungen in VEE 105
- Collector 212
- Collector-Objekt 211
- Compiler 482
- Complex plane
  - Anzeigen 198
- Complex, Datentyp 177
- Confirm (OK)-Objekt 437

Coord, Datentyp 178  
C-Programmbeispiel 4

## D

Dataset  
Such- und Sortieroperationen 243  
DataSets zum Speichern und Abrufen von Records 238 bis 242  
date & time, Zeitstempelformat 220  
Datei  
Senden eines Real Array 221  
To/From File-Objekte 214  
Zeitstempel senden an 220  
Dateien  
Programm 63  
Daten  
Abrufen eines Records 231  
Abrufen mit dem Objekt „From File“ 223  
Anzeigen von Testdaten 198  
Ausgang hinzufügen 148  
Build Data, Record-Objekt 353  
Constant, Record 355  
DataSets und Datentypen 238  
Datenfluss 75  
Datenfluss anzeigen 73  
Datenlinien zwischen Objekten 44  
Datentypen 177  
Eingang hinzufügen 148  
einlesen von einem Instrument 154  
Erstellen von Datenlinien 44  
Form, Definition 176  
From File-Objekt dem Programm hinzufügen 93  
Löschen von Datenlinien 44  
mathematische Verarbeitung 96  
mit MATLAB unterstützte Datentypen 195  
Objekt 38  
Objekt „To File“ in Programm 90  
Pins und Objekte 49  
Records 228  
Speichern gemischter Datentypen 227  
Testdaten protokollieren 390  
Typ, Definition 176  
Verwenden von Datenformen im Programm 98  
Verwenden von Datentypen im Programm 96  
Weitergabe und Datenfluss 70  
weitergeben im Sequencer 397  
Zugreifen auf protokollierte Daten 395

Daten einlesen von einem Instrument 154  
Datensatz  
Data, Build Data, Record-Objekt 353  
Datenausgangs-Pin 49  
Dateneingangs-Pin 49  
Datenfluss  
Ausführungsfluss anzeigen 73  
Datenfluss anzeigen 73  
Description, Dialogfeld 125  
Delta-Markierungen 202  
Detailansicht  
anzeigen 95  
Definition 6  
kein Zugriff bei gesicherter Fensteransicht 437  
Schaltfläche in der Symbolleiste 421  
wann ist keine Änderung möglich 447  
Device  
Call, Function 351  
Import Library 361  
Device => Import Library 475  
Dialogfeld 22  
erstellen für die Benutzereingabe 87  
Dialogfeld I/O Transaction  
Array-Abmessung wählen 157  
Display  
Anzeiger 424  
Direct I/O 150 bis 159  
konfiguriert zum Einlesen vom Instrument 157  
Objekt 151  
Transaktion 153  
Direktaufrufe  
Textbeschreibungen anzeigen 24  
DLLs  
(Dynamic Link Libraries) 475  
Aufruf von einem Ausdrucksfeld aus 477  
PC-Zusatzkarten 135  
Dokumentieren  
Description, Dialogfeld 125  
Programm mit Save Documentation 125  
doppelt klicken 22  
Download des Instrumentenstatus 158  
Download String 159  
Drucken der Anzeige 62  
Drucker, mit VEE verwenden 62  
Duplizieren eines Objekts 36  
Dynamic Link Libraries (DLLs) 475  
Dynamic Link Library  
Aufruf von einem Ausdrucksfeld aus 477

## E

- E/A
  - Arbeitsweise der E/A-Transaktionen 215
  - To File-Objekt 215
  - Transaktions-Dialogfeld 215
  - Transaktionsformat (Syntax) 216
- E/A-Bibliotheken 133
- E/A-Konfiguration speichern 65
- E/A-Transaktionsfenster
  - Format 216
- E/A-Transaktions-Timeout 139
- Edit-Menü
  - Find 373
- Eigenschaften
  - ändern 50
- Ein UserObject erstellen ?? bis 87
- Einblendfenster 437
- Eine Assembly aktualisieren 326
- Eine neuen Assembly installieren 326
- Einfügen
  - UserObject in Programm 80
- Einfügen eines Objekts 37
- Eingangs-Pins
  - Ausgabe 49
  - Daten 49
  - Fehler, über 82
  - Sequenz 49
- Einstellungen
  - ändern 46
- Elemente
  - Array extrahieren 213
- Enabled
  - Sequencer-Transaktionsfeld 388
- Enum, Datentyp 178
- Enumeration 295
- EOF, Fehler vermeiden in „From DataSet“ 242
- serielle Schnittstelle 138
- Erneut Starten von VEE 66
- Erstellen eines Records 228
- Erstellen eines UserObject 80 bis ??
- Erstellen einer UserFunction 350
- EXECUTE-E/A-Transaktion 216

## F

- Farben
  - Ändern in der Wellenform-Anzeige 203
  - Speichern mit dem Programm 65
- Fehler
  - Aufrufstapel anzeigen 113
  - Eingangs-Pins nicht angeschlossen 82
  - Fehler-Ausgangs-Pins hinzufügen 116
  - Fehlerbehebungsprogramme 105
  - Go To 112
  - View => Last Error 112
- Fehler beheben 112
- Fehlerbehebung
  - alphanumerische Anzeigen hinzufügen 109
  - Anschlüsse überprüfen 109
  - Ausführungsfluss anzeigen 106
  - Daten an einer Linie prüfen 107
  - Datenfluss anzeigen 105
  - Line Probe 107
  - Programme in VEE 105
  - Schrittfunktionen 118
  - Unterbrechungspunkte 110
- Feld
  - Abrufen eines Felds aus einem Record 231
- Feld setzen in einem Record 233
- Fenster
  - Hauptfenster 24
- Fensteransicht
  - anzeigen 95
  - Ausrichten von Objekten 432
  - Bedienerschnittstelle erstellen 94
  - Beep-Objekt 448
  - sichern 437
  - snap-to-grid 432
  - Objekt hinzufügen 420
  - Objekte hinzufügen 440
  - Objekte verschieben 440
  - Radio Buttons 441
  - Schaltfläche in der Symbolleiste 421
  - Softkeys und Funktionstasten 434
  - Wechseln zur Detailansicht 95
- Fensteransicht erstellen 94
- File, Menü
  - Save Documentation... 125
- File-Menü
  - Default Preferences 424
  - Merge 375



- Save As... 63
- Zusammenfügen
  - Library 361
- Find-Funktion 373
- Flow => Confirm (OK) 437
- Fluss, Daten 75
- Format
  - E/A-Transaktion 216
- Formelobjekt 99 bis ??
- Formelobjekte
  - Zeilenumbrüche 188
- Formula, Objekt 184
  - Ausdruck auswerten 185
  - Ausdrücke erstellen 184
  - Auswerten einfacher Ausdrücke 185
  - mehrere Ausdrücke 187
- Formula-Objekt ?? bis 101
  - Verwenden vordefinierter Funktionen 99
- Frequency
  - Anzeigen 199
- From File
  - Objekt dem Programm hinzufügen 93
- From File-Objekt 225
- Füllanzeigen 424
- Function & Object Browser 180
- Funktion
  - Funktionen auf entfernten Computern 349
  - gleiche Namen 372
  - kompilierte Funktion 348, 471
  - Menü 146
  - Select Function in Device Call 351
  - Sequencer-Transaktionsfeld 389
- Funktion wählen, Beispiel 352
- Funktionsaufrufe verschachteln 467
- Funktionen auf entfernten Computern 349
- Funktionstasten, in Programmen verwenden 434

## G

- Gateway 138
- Geschwindigkeit, Ausführung 489
- Gemischte Datentypen speichern 227
- Get field-Objekt 231
- gewählten
  - Objekte 40
- Globale Variablen
  - setzen und abrufen 122
  - Optimieren von Programmen 468

- Setzen vor dem Verwenden 124
- Weitergeben von Daten im Sequencer 400
- Go To 112
- GPIO 138
- GPIO 138
- Groß- und Kleinschreibung
  - VEE und MATLAB 194
- Größe
  - Objektgröße ändern 39
  - Objektgröße in der Fensteransicht 422
- Größe von Objekten ändern 39

## H

- Hauptfenster
  - Anzeigen in VEE 68
  - Beschreibung 24
- HH, Zeitstempelformat 220
- Hierarchie des Programms 113
- Hilfe
  - Menüposition für Objekt suchen 104
  - Objektmenü 34
  - Online 22, 26
  - Online-Lernprogramme 102
  - System 29
- Hinzufügen
  - Anschluss 51
  - Objekte 30
  - to panel 440

## I

- If Pass
  - Sequencer-Transaktionsfeld 390
- Importieren
  - UserFunctions 361
- Indicator
  - Anzeigen 198
- Int16, Datentyp 177
- Int32, Datentyp 177
- integrierte Operatoren 180
- Instrument Manager 136
- Instrumente
  - Ausdrucksliste senden an 153
  - Daten einlesen aus 154
  - für die Verwendung im Programm wählen 142
  - Hinzufügen physischer Instrumente 143

- Konfiguration 136
  - lokal oder fern steuern 138
  - Senden von Textbefehlen 151
- Instrumente konfigurieren 136

## K

- Klammern im Call-Objekt 363
- klicken 22
- Klonen eines Objekts 36
- Klonen oder Kopieren 37
- Kompatibilitätsmodus 482
- Kompilierte Funktionen 471
  - erstellen, verbinden, aufrufen 348
- Konfiguration
  - E/A-Konfiguration mit dem Programm speichern 65
- Konfigurieren
  - Tests 384
  - VXIPlug&Play-Treiber 166
- Kontextmenü 34
- Kontextmenüs
  - Edit 43
- Konstruktor 295
- Kopieren eines Objekts 37
- Kopieren mehrerer Objekte 42
- Kopieren oder Klonen 37

## L

- Label
  - Anzeigen 198
- Laufzeitversion 11
  - Definition 13
- Leiste, Bildlaufleiste 45
- Lernzeichenfolge 159
- Line Probe 107
- Linien
  - Edit => Clean Up Lines 58
  - Erstellen von Datenlinien zwischen Objekten 44
  - Löschen von Linien zwischen Objekten 44
- Linien bereinigen 58
- Live Mode 139
- Löschen
  - „Rückgängig machen“ eines Löschvorgangs 38
- Löschen des Arbeitsbereichs 46

- Logging Alphanumeric
  - Anzeigen 198
- Logging Enabled
  - Sequencer-Transaktionsfeld 390
- Lokale Funktionen benennen 372

## M

- Markieren von Objekten 40
- Mathematisch
  - Ausführen mathematischer Operationen auf Arrays 464
  - Device => Function & Object Browser 180
- Mathematische Verarbeitung von Daten 96
- MATLAB 190 bis 195
  - Funktion 179
  - Grafik 193
  - Groß- und Kleinschreibung 194
  - in Function & Object Browser 181
  - MATLAB Script-Objekt verwenden 191
  - Objekt in VEE-Programm 192
  - Script-Objekt in VEE-Programm einbeziehen 194
  - Signalverarbeitungs-Toolbox 15
  - Support-Informationen 18
  - Überblick 15
  - unterstützte Datentypen 195
- Maustaste 22
- Messelemente 424
- Menü
  - Auswählen 22
  - Kontextmenü 34
  - Leiste 24
  - Objektmenü 33
- Menüs
  - Device => Import Library 475
  - Display => Indicator 424
  - Display => Sequencer 384
  - Eigenschaften, Title 39
  - File => Default Preferences (Auswahl von Farben und Schriftarten) 424
  - File => Merge 375
  - File => Merge Library 361
  - File => Save As... 63
  - File => Save Documentation 125
  - Flow => Confirm (OK) 437
  - Function & Object Browser 180
  - I/O => Instrument Manager... 136

- Objektmenü 34
- Position suchen in Online-Hilfe 104
- Properties => Icon 425

Merge Library 361

Microsoft Windows 22

Mit Datentypen arbeiten 195

Modi

- Ausführung 482
- Kompatibilität 482

Move

- Daten zwischen Objekten 49
- ein Objekt 35
- gesamten Arbeitsbereich verschieben 45
- Objekte in der Fensteransicht 440

## N

Namen

- Ändern des Objektnamens 39
- zusammengefügte Funktionen und lokale Funktionen 372

snap to grid 432

Noise Generator

- Anzeigen einer Wellenform 201
- Objekt hinzufügen 71

Note Pad

- Anzeigen 198

Nummern

- Real64 Slider 73

## O

Object-Datentyp 178

Objekt minimieren 33

Objekte

- „Rückgängig machen“ oder Einfügen eines gelöschten Objekts 38

- Abrufen von Daten mit dem Objekt "From File" 225

- alles verschieben 45

- Anschlüsse 51

- Ändern von Parametern 59, 62

- Ansichten ändern 33

- ausschneiden 37, 38

- Ausführungsreihenfolge im Programm 117

- Ausrichten in der Fensteransicht 432

- Auswahl aufheben 40

- bearbeiten 43

- Beep 448

- benennen, Namen ändern 39

- Call-Objekt 363

- Confirm (OK) 437

- Data, Build Data, Record 353

- Data, Constant, Record 355

- Delete Library 368

- Device => Function & Object Browser 99

- Device, Import Library 361

- duplizieren 36

- einfügen 37

- Eingangs- und Ausgangs-Pins 49

- Erstellen einer UserFunction 350

- Erstellen von Datenlinien 44

- Execute Program 481

- Formel 184

- Get Field 231

- gewählten 40

- Größe ändern 39

- Hilfe anzeigen 103

- Hilfemenü 34

- Import Library 368

- klonen 36

- kopieren 37

- löschen 37, 38

- Löschen von Datenlinien 44

- MATLAB 193

- mehrere Objekte kopieren 42

- Menü 33

- Menüposition suchen in Online-Hilfe 104

- Minimieren 33

- minimiert darstellen für bessere Leistung 465

- Namen ändern 39

- Object, Datentyp 178

- Objekte To/From File 413

- Objektmenü öffnen 34

- offene Ansicht eines Objekts 33

- Pins und Anschlüsse 49

- platzieren 30

- positionieren im Fenster 83

- Positionsinformationen 36

- Radio Buttons 441

- Record auflösen 236

- Reduzieren der Anzahl der Objekte in Programmen 466

- Reihenfolge der Pin-Operationen 115

- Sequencer 383

- Show Title Bar deaktiviert 439

- Symbole 33

- Titel ändern 39

- To File 222

- To/From DataSet-Objekte 414
- UserFunction 348
- Variable abrufen 122
- Variable setzen 122
- verbinden 56
- verschieben 35
- verschieben in der Fensteransicht 440
- ziehen 35
- zum Fenster hinzufügen 440
- Objekte To/From File 413
- Objektmenü
  - öffnen 34
  - öffnen, wenn die Titelleiste verdeckt ist 439
- offen
  - Objektansicht 33
- Öffnen
  - VEE 66
- öffnen
  - Objektmenü 34
- Online
  - Lernprogramm 102
- Online-Hilfe 22, 26
- Operatoren
  - integriert 180

## P

- Pass
  - Sequencer 390
- Panel-Treiber 133, 143, 145 bis 149
- Parameter
  - ändern 59, 62
- Pause 59
- PComplex, Datentyp 177
- PC-Zusatzkarten 160, 163
- Physisches Instrument
  - hinzufügen zur Konfiguration 143
- Pictures
  - Anzeigen 198
- Pins
  - einen Anschluss bearbeiten 52
  - Eingang und Ausgang 49
  - löschen eines Anschlusses 54
  - Reihenfolge der Operation im Objekt 115
  - Steuer-Pins 116
  - Terminals hinzufügen 51
- Pixel, Objekte exakt positionieren 36
- Platzierung
  - Objekte in der Fensteransicht verschieben 422
- Pos1-Taste zum Positionieren von Objekten 83
- Polar Plot
  - Anzeigen 198
- Produkt-Support-Informationen 17
- Profiler 489
- Program Explorer 24, 359
  - Anzeigen von UserFunctions 359
  - Program Explorer öffnen 68
- Programme
  - Alarm-Übung 447
  - ausführen 58
  - Ausführungsgeschwindigkeit 489
  - Beenden von VEE 65
  - schrittweise Ausführung 118
  - Dateien 63
  - Datenfluss 73
  - Einblendfenster anzeigen 438
  - erstellen 56, 58
  - Farben und Schriftarten speichern 65
  - Fehlerbehebung 105
  - Go To 112
  - Hierarchie 113
  - sichern 437
  - Instrumente wählen 142
  - offene Ansicht von Objekten 33
  - speichern 63
  - speichern der E/A-Konfiguration 65
  - Signalton mit Beep-Objekt 448
  - Speichern von Testergebnissen 210
  - Starten von VEE 66
  - Symbolansicht von Objekten 33
  - UserFunctions 368
  - Unterbrechungspunkte einsetzen 110
  - Unterprogramme 348
  - VEE 75
  - Weitergabe und Datenfluss 70
- Programme optimieren 464
- Properties
  - Icon 425
- Protokollieren
  - Datenzugriff 395
  - Objekte To/From File 413
  - To/From DataSet-Objekte 414

## R

- Radio Buttons 441
- Range
  - Sequencer-Transaktionsfeld 389
- READ-E/A-Transaktion 216
- Real Array senden an Datei 221
- Real32, Datentyp 177
- Real64 Slider 73
- Real64, Datentyp 177
- Record
  - Abrufen eines Felds 231
  - auflösen 236
  - erstellen 228
  - Feld setzen 233
  - Sequencer 396
  - Sortieroperation mit einem Feld 250
  - Speichern und einlesen in/von einem DataSet 238
  - Übereinstimmungsfehler vermeiden mit EOF 242
  - Verwenden von DataSets zum Speichern und Abrufen 238
  - verwenden zum Speichern gemischter Datentypen 227
- Record auflösen 236
- Record Constant 354
- Record, Datentyp 178
- Resume (Fortsetzen) 59
- Rückgängig
  - gelöschtes Objekt 38
- Rückwärtskompatibilität 482
- Run, Schaltfläche in der Symbolleiste 66

## S

- Schalterelemente 431
- Schaltfläche
  - Maus 22
  - Symbolgröße 33
- Schaltflächen
  - auf Symbolleiste, Textbeschreibung anzeigen 24
  - Run, Schaltfläche 66
- Schatten an markierten Objekten 40
- Schließen von VEE 65
- Schnittstelle
  - GPIB 138
  - GPIO 138
  - Seriell 138
  - VXI 138
- Schriftarten
  - Speichern mit dem Programm 65

- Sequencer
  - Daten weitergeben 397
  - Definition 381
  - Objekte To/From File 413
  - Records 396
  - Speichern und Abrufen von Daten 413
  - To/From DataSet-Objekte 414
  - Transaktions-Dialogfeld 386
- Sequenzieren von Tests 380
- Sequenz-Pins 49, 115
- Sichern eines Programms 437, 447
- Signalton im Programm
  - Beep-Objekt 448
- Signalverarbeitungs-Toolbox, MATLAB 15
- Skalarwerte, Definition 210
- Slider
  - Real64 Slider 73
- Softkeys in der Fensteransicht verwenden 434
- Sortieroperation mit einem Record-Feld 250
- Spec Nominal
  - Sequencer-Transaktionsfeld 388
- Spectrum
  - Anzeigen 199
- Spectrum, Datentyp 177
- Speichern
  - ein Programm 63
  - Gesicherte Laufzeitversion 437, 447
  - Save Documentation, Menü 125
- Speichern gemischter Datentypen 227
- Speichern von Daten 413
- Standardabweichung, Übung 182
- Standardwerte
  - Einstellungen ändern 46
- Statische Member 294
- Statusleiste
  - Anzeige 24
  - Objekte exakt positionieren 36
- Step Into 118
- Step Into (Schritt in) 59
- Step Out 118
- Step Over 118
- Steuern von Instrumenten 133
  - Live Mode 139
- Steuer-Pins 116
- Stoppen von VEE 66
- Strip Chart
  - Anzeigen 199
- Suchen und Sortieren mit DataSet 243

## Support

- Agilent VEE-Support 17
- MATLAB 18

## Symbole

- ändern 425
- Ausführungszeit verbessern 465
- Run, Schaltfläche in der Symbolleiste 66
- Schaltfläche Minimieren bei Objekten 33
- Symbolansicht 33
- Symbolansicht eines Objekts 33
- Textbeschreibungen anzeigen 24

## Symbolleiste 22, 24

- Tooltips anzeigen 24

## Systeme

- unterstützte 22

## T

### Task beenden (VEE beenden) 65

### Tankanzeigen 424

### Tasten

- Pos1 83

### Tastenkürzel

- Anschluss hinzufügen 51
- Pause 59
- Resume (Fortsetzen) 59
- Run (Ausführen) 59
- Step Into (Schritt in) 59

### Test

- Angabe des auszuführenden Tests 389
- Protokollieren von Daten 390
- Sequenz-Transaktionsfeld 388
- Speichern und Abrufen von Daten 413
- Verzweigungsanleitungen 390

### Testsequenzen 380

### Testergebnisse

- Extrahieren von Werten aus einem Array 213

### Testergebnisse in Arrays speichern ?? bis 213

### Testergebnisse speichern 210 bis 213

### Testergebnisse, speichern in Arrays 210 bis ??

### Testserien ausführen 380

### Text

- Datentyp 178
- Textzeichenfolge an eine Datei senden 218

### Textbefehl senden an Instrument 151

### Thermometer 424

### Threads 117

### Titel

### Ändern des Objektstitels 39

### Leiste 24

### To File-Objekt 222

### zum Programm hinzufügen 90

### To/From DataSet-Objekte 414

### To/From File-Objekte 214 bis 226

### Transaktion, Direct I/O 153

### Treiber

### Panel 133

### VXIPlug&Play 133

### Triadischer Operator 469

## U

### Übungsprogramm

- Verwenden von Records 228

### Übungsprogramme

- Alarm 447
- Anzeige von Datenfluss und Weitergabe 70
- Anzeigen einer Wellenform 56
- Dialogfeld erstellen 87
- Erstellen eines Array für Testergebnisse 211
- Fensteransicht erstellen 94
- Generieren einer Zufallszahl 121
- globale Variablen setzen und abrufen 122
- Hinzufügen eines Amplituden-Eingangs 73
- Hinzufügen eines Noise Generator 71
- Real64 Slider 73
- Standardabweichung 182
- Suchen und Sortieren mit DataSets 243
- Verwenden von DataSets 238

### UserFunction

- ArrayStats 356
- erstellen, aufrufen, ändern, übertragen 348

### UserFunctions

- ändern 361
- Aufrufen über Ausdruck 355
- bearbeiten 370
- Befehl „Merge Library“ 361
- Bibliotheken wiederverwenden 361
- Import Library 361
- Merge Library 361
- Objekte Import Library und Delete Library 368
- speichern als Programm 368
- Profiler 489
- suchen mit Find 373
- Unterschiede zu UserObjects 349
- zusammenfügen 376

## UserObject

- Erstellen 80 bis 87
- Minimiert 67
- offene Ansicht 67
- Profiler 489
- Symbolansicht 67
- suchen mit Find 373
- Unterschiede zu UserFunction 349
- zusammenfügen 376

## UInt8, Datentyp 177

## Unterbrechungspunkte 110

## Unterprogramme

- UserObjects und UserFunctions 348

## Unterstützte Systeme 22

## Upload des Instrumentenstatus 158

## Upload String 159

## URLs

- Web-Adressen für MATLAB 18
- Web-Adressen für VEE 17

## V

## Variablenobjekt abrufen 122

## Variablenobjekt setzen 122

## Variant, Datentyp 178

## VEE

- Arbeitsbereich 24
- Ausführungsfluss anzeigen 73
- Compiler 482
- Datenfluss anzeigen 73
- drucken 62
- Eingangs-Pin-Verbindungen 82
- Farb- und Schriftinformationen speichern 65
- Fehlerbehebung 105
- Fehlermeldungen in Programmen 105
- Go To 112
- Interaktion mit 22
- Online-Hilfe 29
- speichern der E/A-Konfiguration 65
- Profiler 489
- Program Explorer 24
- Programme ausführen 58
- Programmieren 75
- Speichern von Testergebnissen 210
- starten 23

## VEE starten 23

## VEE- und .NET-Sicherheit 328

## VEE-Arraydatentypen in .NET-Datentypen konvertieren 309

## VEE-Datentypen in .NET-Skalardatentypen konvertieren 304

## VEE-Laufzeitversion weitergeben 327

## Verbinden von Objekten 56

## Verbindungen zwischen Objekten, anzeigen 6

## Verwalten des Arbeitsbereichs 67

## Verwenden von .NET mit VEE 288

## Verzweigungstests 390

## VXI 138

## VXIPlug&Play-Treiber 133, 166 bis 170

## W

## Was ist .NET? 289

## WAIT-E/A-Transaktion 216

## Waveform

- Datentyp 177

## Web-URLs

- Agilent VEE 17
- MATLAB 18

## Weitergabe und Datenfluss 70

## Welcome-Menü in der Online-Hilfe 102

## Wellenform

- Anzeige, Ändern der Farbe des Trace 203
- Anzeige, Delta-Markierungen 202
- Anzeigen 199, 201
- Anzeigen, Ändern der X- und Y-Skala 201
- Anzeigen, zoomen 201
- Display Waveform, Programm 56

## WRITE-E/A-Transaktion 216

## X

## X gg. Y plot

- Anzeigen 199

## XY Trace

- Anzeigen 199

## Z

## Zeichenfolge

- Download String 159
- Lernzeichenfolge 159
- Upload String 159

Zeilenumbrüche in Formelobjekten 188  
Zeitstempel an Datei senden 220  
ziehen 22  
Ziehen eines Objekts 35  
Zoomen der Wellenform 201

Zusammenfügen  
Benennen von Funktionen 372  
File 375  
UserFunctions 361  
VEE-Programme 375  
Zufallszahl  
Generieren in einem Übungsprogramm 121  
Zugreifen auf protokollierte Daten 394, 395