

# **TestPoint<sup>ä</sup> für Windows**

**„Schnell erlernt“**

Program and documentation copyright © 1994 by Capital Equipment Corporation. No parts of this book may be reproduced or transmitted in any form or by any means, electronic, optical, or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission in writing from Capital Equipment Corporation.

The software accompanying this manual is licensed to the user by Capital Equipment Corporation. The software is copyrighted © 1994 by Capital Equipment Corporation. Details of the license agreement appear on the software media packaging.

#### Limited Warranty

Capital Equipment Corporation (CEC) warrants the physical diskettes and documentation enclosed herein to be free of defects in materials and workmanship for a period of sixty days from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be limited to replacement of defective diskettes or documentation, and shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data, or use of the software, or special, incidental, or consequential damages or other similar claims. In no event shall CEC's liability exceed the purchase price of the product.

TestPoint is a trademark of Capital Equipment Corporation.

Windows, Excel, and Microsoft are trademarks of Microsoft Corporation.

Quattro, Quattro Pro, and Turbo Pascal are trademarks of Borland International.

123 is a trademark of Lotus Development Corporation.

Überarbeitete deutsche Übersetzung: ME Keithley Instruments GmbH 1995

Learning TestPoint  
Part number 04000-90100, vol. 1

Capital Equipment Corporation  
76 Blanchard Rd.

Second edition

99 98 97 96 95 94

10 9 8 7 6 5 4 3 2 1

Burlington, Massachusetts 01803

(617) 273-1818

# Inhalt

---

<b>INSTALLATION.....</b>	<b>4</b>
<b>EINLEITUNG.....</b>	<b>7</b>
WAS IST TESTPOINT ? .....	8
WAS TESTPOINT FÜR SIE TUN KANN .....	10
WIE SIE MIT TESTPOINT ARBEITEN.....	12
WIE TESTPOINT FÜR SIE ARBEITET .....	16
<b>ÜBUNGEN.....</b>	<b>17</b>
STARTEN SIE EINIGE ANWENDUNGSBEISPIELE.....	18
EREIGNISGESTEUERTE (EVENT-DRIVEN) PROGRAMMIERUNG.....	21
WÄHLEN SIE IHRE ÜBUNG.....	23
ÜBUNG 'AUTOMATISIERTER TEST'.....	25
ÜBUNG 'MEB DATENERFASSUNG'.....	44
ÜBUNG 'ANALYSE'.....	66
<b>TESTPOINT-KONZEPTE.....</b>	<b>83</b>
OBJEKTE.....	84
ACTION LISTS.....	87
DATEN .....	88
AUSFÜHRUNG.....	89
<b>WEITERE BEISPIELE.....</b>	<b>91</b>
BEISPIEL FREQUENZANTWORT .....	92
BEISPIEL ANALYSE.....	95
BEISPIEL CUSTOM CONTROL (VBX).....	98
MEHR ÜBER DIE TESTPOINT-OBJEKTE .....	102
WEITERE INFORMATIONEN/BESTELLUNG .....	103



# ***Installation***

---

## ***Einrichten***

- Legen Sie die TestPoint Disk #1 ins Diskettenlaufwerk A oder B.
- Starten Sie Microsoft Windows.
- Im Windows Programm-Manager wählen Sie aus dem Menü den Befehl: File Run (Datei Ausführen).
- Geben Sie ein: "A:SETUP" oder "B:SETUP" und drücken Sie die RETURN-(ENTER-, EINGABE-) Taste.
- Folgen Sie den Instruktionen auf dem Bildschirm. Sie können ein Verzeichnis wählen, in dem das TestPoint-System eingerichtet werden soll. Die Voreinstellung ist "C:\TESTPT".

Das SETUP-Programm kopiert alle notwendigen Dateien in das TestPoint-Verzeichnis und fügt die entsprechenden Symbole Ihrer Windows-Oberfläche hinzu.

## ***Einrichten der Gerätebibliotheken***

GPIB und andere 'Libraries' sind optional. Sie werden vom SETUP-Programm eingerichtet, genauso wie das übrige TestPoint-System. Sie können wählen, ob Sie nur bestimmte oder alle 'Libraries' einrichten wollen.

## ***Execution Control Key***

Mit Ihrem TestPoint-System erhalten Sie einen sogenannten "Execution Control Key", der für die parallele Schnittstelle (Druckeranschluß) bestimmt ist. Dieser Stecker wird benötigt, um alle Möglichkeiten von TestPoint zu nutzen. Er wird nicht benötigt, um Programme zu starten, die mit TestPoint erstellt wurden. Sie können diesen Stecker auf die parallele Schnittstelle stecken und daran Ihren Drucker wie gewohnt betreiben.

## ***Systemvoraussetzungen***

- Microsoft Windows 3.1 oder höher
- IBM-PC-kompatibler Computer mit mindestens 80286 Prozessor (80386 oder 80486 empfohlen)
- Mindestens 4 MegaByte Speicher (8 MB empfohlen)
- Mindestens 7 MegaByte frei auf der Festplatte
- MS-Windows kompatible Maus

## ***Löschen von TestPoint (Uninstall)***

Um die Installation rückgängig zu machen und die TestPoint-Dateien wieder von der Festplatte zu entfernen, gehen Sie wie folgt vor:

Erster Schritt:

Gehen Sie in den Programm-Manager von Windows und verkleinern Sie das TestPoint-Gruppenfenster (klicken Sie auf den Abwärtspfeil in der rechten oberen Ecke des TestPoint-Gruppenfensters, so daß die gesamte TestPoint-Gruppe nur noch als ein Symbol zu sehen ist). Klicken Sie dann mit der linken Maustaste einmal auf dieses TestPoint Gruppen-Symbol und drücken die Entf-Taste (Del). Bestätigen Sie Windows, daß Sie die Gruppe löschen wollen.

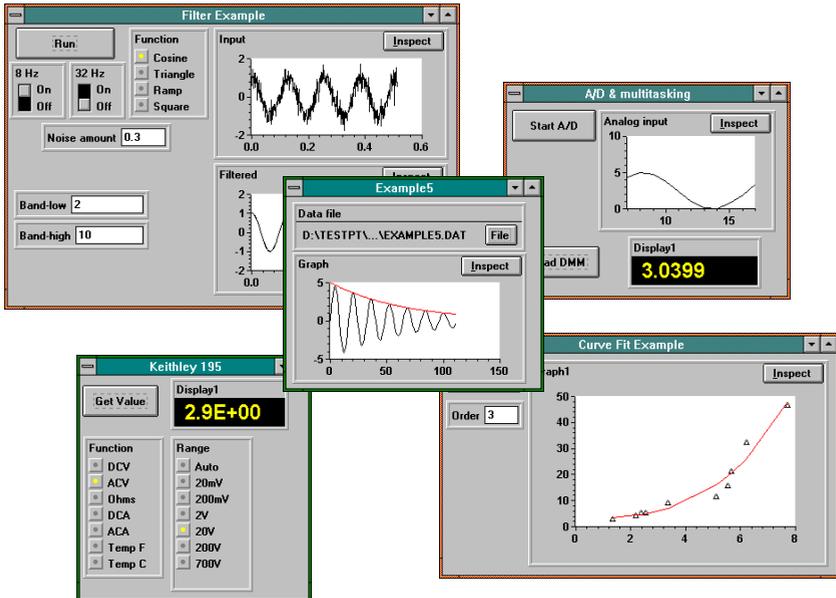
Zweiter Schritt:

Gehen Sie zurück zu DOS und starten die BATCH-Datei  
UNINSTALL:

```
C:> UNINSTALL
```

Damit werden alle TestPoint Dateien und Verzeichnisse gelöscht.

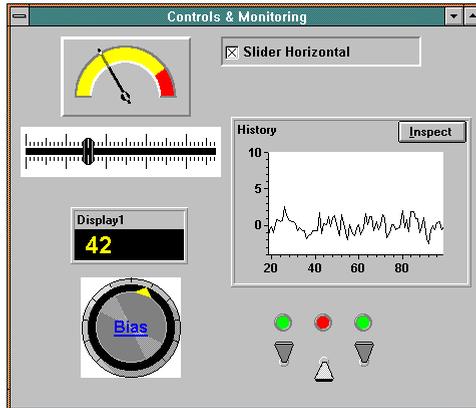
# Einleitung



# Was ist TestPoint ?

---

TestPoint ist eine Entwicklungsumgebung unter Windows, die es Ihnen erlaubt, schnell und einfach Anwendungen im Bereich der Meßdatenerfassung zu erstellen. TestPoint unterstützt GPIB-488.2, RS-232, RS-485, sowie Datenerfassungskarten verschiedener Hersteller.



TestPoint beinhaltet Funktionen, um Hardware anzusteuern (GPIB, RS-232, A/D Karten, ...), benutzerfreundliche Oberflächen zu generieren, Daten zu verarbeiten, aufzubereiten und darzustellen, Meßprotokolle zu erstellen, sowie Daten und Informationen mit anderen Windows-Programmen auszutauschen.

TestPoint ist eine neue und einzigartige Alternative zu herkömmlichen Programmiersprachen. In TestPoint wählen Sie nur die Objekte aus, die Sie für Ihre Anwendung benötigen. Sie kombinieren die möglichen Aktionen dieser Objekte miteinander, indem Sie einfach mit der Maus Beziehungen zwischen diesen Objekten herstellen, um Ihre Anwendung in der gewünschten Form ablaufen zu lassen.

TestPoint hat all die Fähigkeiten und Flexibilität einer Programmiersprache. Es gibt nicht die üblichen Einschränkungen von menügeführten Programmpaketen, denn TestPoint beinhaltet alles, was Sie brauchen, um den Programmfluß zu steuern, benutzerspezifische Hardware anzusteuern, das System durch eigene Routinen zu erweitern sowie Daten mit anderen Programmen auszutauschen.

# ***Was TestPoint für Sie tun kann***

---

## ***Gerätesteuerung***

- Gerätebibliotheken (Libraries) für Hunderte von gängigen Meßgeräten
- Unterstützung von GPIB (IEEE-488), RS-232, und RS-485.
- VXI-Unterstützung via GPIB Controller

## ***Meßdatenerfassung***

- Unterstützung von analogen Ein- und Ausgängen (A/D, D/A), digitalen Ein- und Ausgängen, benutzerspezifischer Hardware durch direkten Zugriff auf I/O Ports
- Schnelle Hintergrund-Datenerfassung (A/D mit DMA)
- Linienschreiber, Balkengrafiken, numerische Darstellung
- Visualisierung von Alarm-Grenzen (Alarm Limits)
- Speichern der Daten auf Disk (Disk Logging)

## ***Analyse***

- FFT, digitale Filter, Kurvenglättung
- Kurvenanpassung, Polynome, Interpolation
- Statistik, inkl. Mittelwert, Standardabweichung, Histogramme, ...
- Vektor- und Matrixoperationen, inkl. Invertierung, Determinante

## ***Darstellung***

- Linien-, Symbol-, Balkengrafik, XY-Darstellung, Linienschreiber
- Numerische und Balkengraph-Darstellung eines Wertes
- Indikatoren und Grafiken
- Druckfähige Versuchsberichte

## ***Benutzer-Schnittstelle***

- Knöpfe, Schalter, Selektoren, Schieberegler, Dateneingabefelder
- Beliebige Anzahl von Fenstern
- Industrie-Standard: Visual Basic Custom Controls (VBX-Dateien)

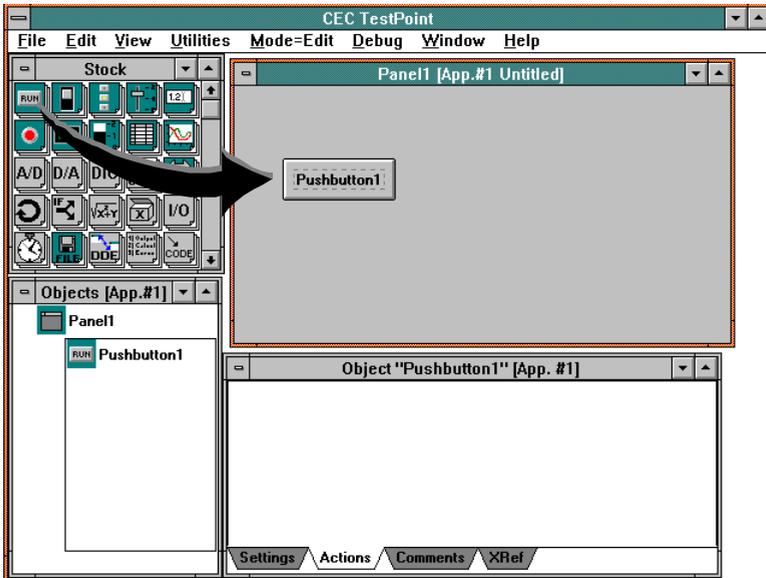
## ***Datenaustausch unter Windows: DDE / OLE***

- Senden und Empfangen von Daten aus Tabellenkalkulations-, Textverarbeitungs-, Datenbank- und Mathematikprogrammen
- Automatische Generierung von formatierten Berichten
- Anzeige von Grafiken und Bildern aus anderen Anwendungen

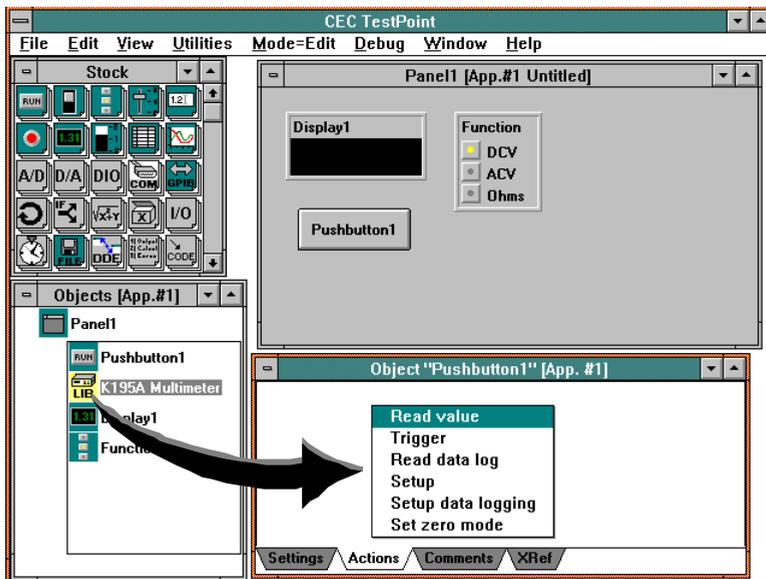
# Wie Sie mit TestPoint arbeiten

---

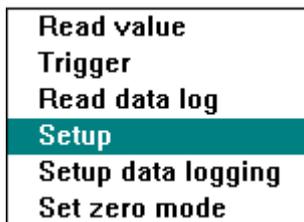
Sie beginnen ganz einfach damit, TestPoint-Objekte wie Knöpfe, Schalter, Numerische Anzeigen, Grafiken, GPIB-Instrumente, A/D-Karten, Mathematische Formeln, Dateien usw. vom 'Stock Window' mit der Maus in das 'Panel Window' (die spätere Benutzerschnittstelle) zu ziehen:



Dann ziehen Sie die entsprechenden Objekte einfach in die 'Action List':



und wählen aus einem Menü eine von mehreren Möglichkeiten. Damit sagen Sie TestPoint, wie Ihr Anwendungsprogramm ablaufen soll:



So erstellt TestPoint in einem klar und einfach zu lesenden Format 'Action Lines', die deutlich beschreiben, was geschieht:

- 1) Setup                    K195A Multimeter    Function="DCV"
- 2) Read value            K195A Multimeter
- 3) Set                     Display1                to K195A Multimeter

Sie brauchen nicht all die Dinge tun, die bei Verwendung von herkömmlichen Programmiersprachen anfallen. Sie müssen keine Variablen deklarieren, keinen langen Code eingeben, keine Rücksicht auf 'reservierte Wörter' nehmen, und nicht auf Syntaxregeln wie Punkt, Komma, Strichpunkt usw. achtgeben. Anstatt sich darauf zu konzentrieren, alles so zu machen, wie der Computer es verlangt, erlaubt Ihnen TestPoint, sich auf Ihre eigentliche Anwendung zu konzentrieren.

Obwohl TestPoint das Erstellen von Anwendungsprogrammen sehr einfach macht, geht nichts von der Mächtigkeit und Flexibilität einer Programmiersprache verloren. TestPoint erreicht und übertrifft sogar die Möglichkeiten von Werkzeugen wie C, Pascal, Visual Basic oder auch sogenannten "Wiring-Diagramm"-Paketen.

TestPoint erlaubt strukturiertes Programmieren, z. B. durch Gruppieren von Aktionen und Daten in Objekten, oder durch Mechanismen wie Schleifen, IF/THEN/ELSE, usw. Objekte unterstützen alle Datentypen wie numerische Daten (ASCII, binär), Strings (Zeichenketten), Vektoren, Felder und Listen. Trotzdem brauchen Sie sich nicht um Datentypen zu kümmern. TestPoint wählt in den meisten Fällen automatisch den am besten geeigneten Datentyp und führt notfalls die Konvertierung durch. Falls ein spezifisches Datenformat benötigt wird, können die Daten trotzdem manuell formatiert werden.

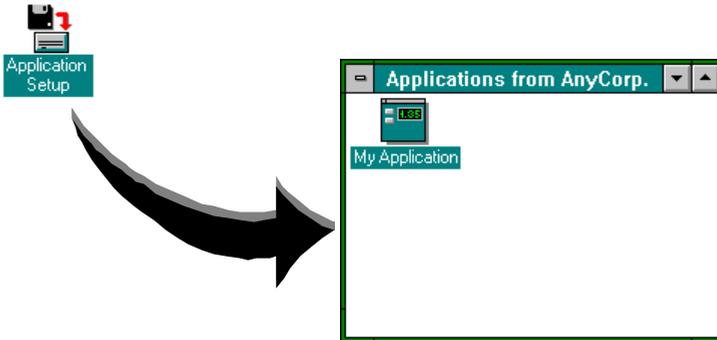
Die 'Action Lists' in TestPoint werden sequentiell abgearbeitet. Es gibt Schleifen und IF/THEN/ELSE-Abfragen. Außerdem kann auf Ereignisse und Interrupts reagiert werden. TestPoint ist eine Multitaskingumgebung unter Windows. Jede TestPoint 'Action List' kann andere 'Action Lists' aktivieren, was ein vollkommen modulares Programmieren ermöglicht.

## Wie TestPoint für Sie arbeitet

---

Wenn Sie Ihre TestPoint Anwendung erstellt haben, können Sie ganz einfach eine Runtime-Version daraus generieren.

Ein TestPoint Runtime-Programm ist ein unter Windows ausführbares Programm, komplett inklusive eines professionellen Installations-Programms.



Sie können es selbst benutzen, Ihren Kollegen geben, in der Produktion einsetzen oder als kommerzielles Produkt verkaufen. Sie benötigen keinerlei TestPoint-Lizenz oder Hardware wie z. B. Stecker für die parallele Schnittstelle.

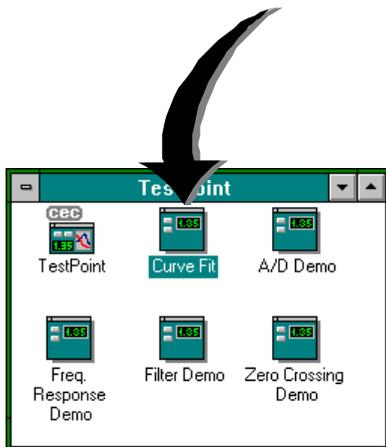
# *Übungen*

---

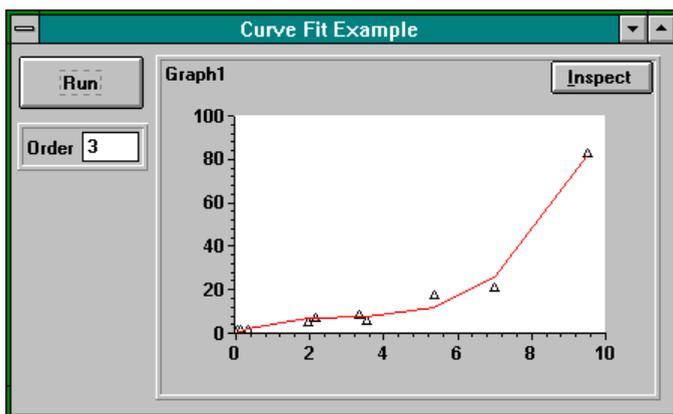
## Starten Sie einige Anwendungsbeispiele

Wenn Sie TestPoint noch nicht installiert haben, tun Sie es jetzt.  
Instruktionen hierzu finden Sie am Anfang dieses Handbuchs.

Führen Sie mit der Maus einen Doppelklick auf das "Curve Fit"  
Symbol der TestPoint-Gruppe im Windows Programm-Manager  
aus:

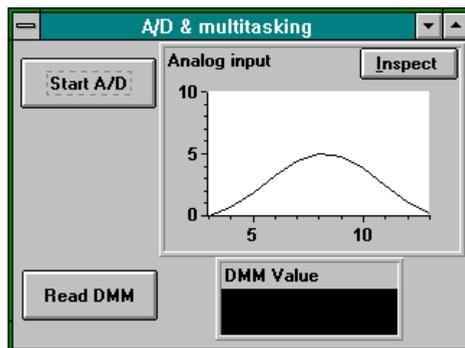


Klicken Sie nun auf den "Run" Pushbutton (Drucktaste):



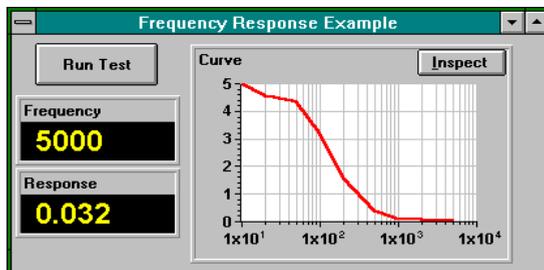
Dieses Beispiel führt eine Kurvenanpassung (Polynomfunktionen) mit einigen Datenpunkten durch und stellt das Ergebnis grafisch dar. Den Grad des Polynoms können Sie bestimmen, indem Sie im Feld "Order" eine Zahl eingeben.

Nun starten Sie das "A/D demo" Programm und klicken auf dessen "Start" Pushbutton:



Starten Sie nun das "Freq. Response" Programm.

Beachten Sie, daß, nachdem Sie einen Doppelklick auf das Symbol ausgeführt haben, das Programm automatisch startet und schrittweise die Frequenz erhöht, zu jeder Frequenz die dazugehörige resultierende Spannung mißt (natürlich im Demo-Mode), und dann die Ergebnisse darstellt:



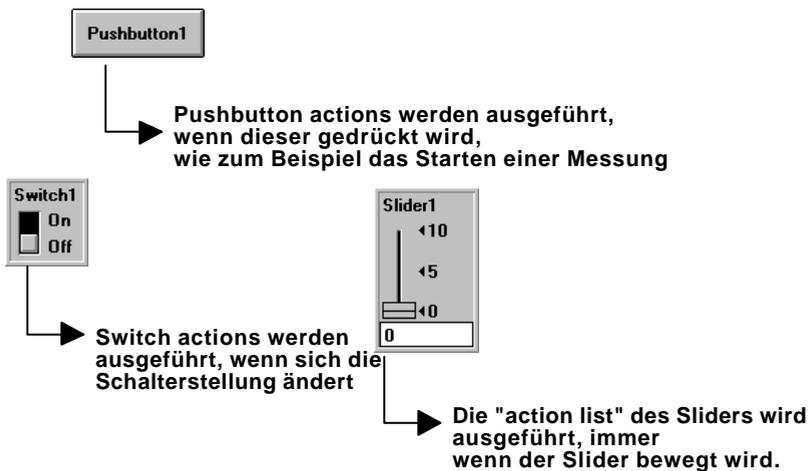
Diese Beispiele zeigen einige wenige der unzähligen Möglichkeiten von TestPoint. Sie zeigen auch, wie eine TestPoint Anwendung optisch aussehen kann.

# Ereignisgesteuerte (Event-Driven) Programmierung

---

TestPoint-Anwendungen laufen unter Microsoft Windows und reagieren auf Maus oder Tastatur. Der Benutzer kann jederzeit Daten in den dafür vorgesehenen Objekten eingeben. Die TestPoint-Anwendung reagiert dann so, wie es bei der Entwicklung der Anwendung festgelegt wurde.

Dies bezeichnet man als event-driven programming (Ereignisgesteuerte Programmierung), denn das Programm reagiert auf Aktionen des Anwenders, anstatt vom Anwender nur unter bestimmten Umständen eine Eingabe zuzulassen oder zu erwarten. Es gibt auch kein 'Hauptprogramm'. Wann immer ein Ereignis generiert wird, werden die entsprechenden Aktionen ausgeführt.



Die Ereignisgesteuerte Programmierung ist eines der Konzepte, die Sie verstehen müssen, um TestPoint effektiv einzusetzen. Deshalb sollten Sie sich ein wenig Zeit nehmen, um zu verstehen, wie eine TestPoint-Anwendung abläuft.

Wie die meisten anderen Windows-Programme verbringt TestPoint die meiste Zeit im 'idle' Zustand, um auf irgendwelche Ereignisse zu warten. Ein Ereignis kann ein Tastendruck sein, eine Mausbewegung, ein Druck auf eine Maustaste, ein Zeitintervall oder auch ein Hardwaresignal, wie z. B. der Taktgenerator von einer A/D-Karte.

Jedem TestPoint-Objekt ist ein Ereignis zugeordnet, auf das es reagiert, indem es eine Folge von Aktionen ausführt, die von Ihnen, dem Programmierer, bestimmt wurden.

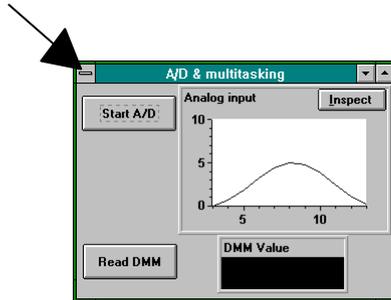
Z. B. reagiert das 'Dateneingabe'-Objekt immer dann, wenn ein neuer Wert eingegeben wird. Das 'Slider'-Objekt (Schieberegler) dagegen reagiert, wenn sein Wert verändert wird. Das 'A/D'-Objekt reagiert, wenn Werte von der A/D-Karte im Computer empfangen werden (DMA oder Interrupt).

Die Schritte, die ausgeführt werden, wenn ein Ereignis eintrifft, bestimmen alleine Sie. Jedes Objekt stellt Ihnen eine Auswahl von Aktionen zur Verfügung, die es Ihnen ermöglichen, Hardware anzusteuern, mathematische Berechnungen auszuführen, Grafiken zu erzeugen, usw. Die Reihenfolge, wie Sie diese Aktionen ereignisgesteuert verbinden, bestimmt den Ablauf Ihrer Anwendung.

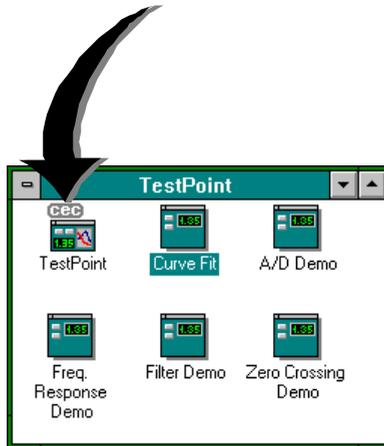
# Wählen Sie Ihre Übung

---

Wenn momentan noch TestPoint-Anwendungen geöffnet sind, schließen Sie diese bitte jetzt, indem Sie mit der linken Maustaste einen Doppelklick in die linke obere Ecke des jeweiligen Fensters ausführen.



Nun starten Sie die TestPoint-Entwicklungsumgebung mit einem Doppelklick auf das "TestPoint"-Symbol:



(Sie müssen alle TestPoint-Anwendungen geschlossen haben, und Sie sollten nun auch den 'Dongle' auf die parallele Schnittstelle stecken, bevor Sie TestPoint starten).

**Dies gilt nicht für die Demo-Version oder den Demo-Modus!**

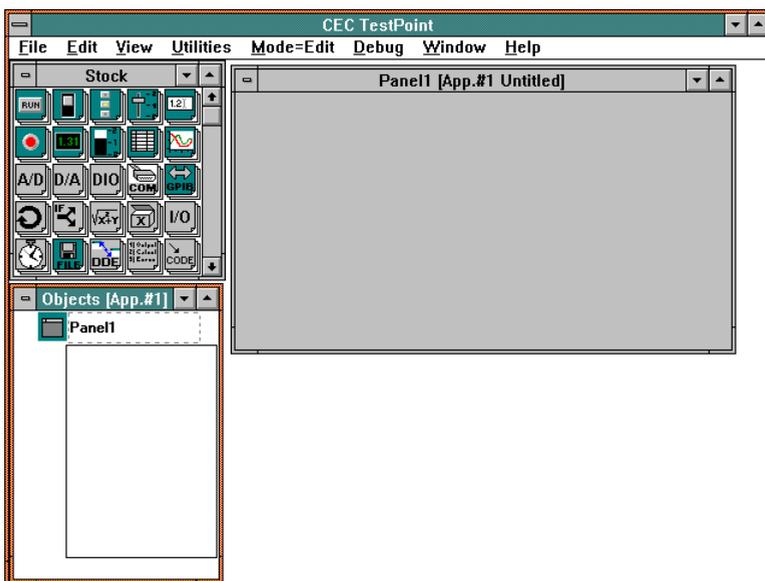
Der Übungsabschnitt ist in 3 Teile gegliedert. Jeder zeigt Ihnen die Grundlagen zum Erstellen von TestPoint-Anwendungen. Wählen Sie nun das Anwendungsgebiet, das für Sie am wichtigsten ist:

- **Automatisierter Test**      **Seite 21**
- **Meßdatenerfassung**    **Seite 38**
- **Analyse**                      **Seite 57**

und beginnen Sie mit dem entsprechenden Teil.

## Übung 'Automatisierter Test'

Wenn Sie die Testpoint-Entwicklungsumgebung starten, wird der Editor aktiviert, und es erscheint ein neues leeres Fenster (Panel), das Sie nun mit den gewünschten Objekten füllen können:

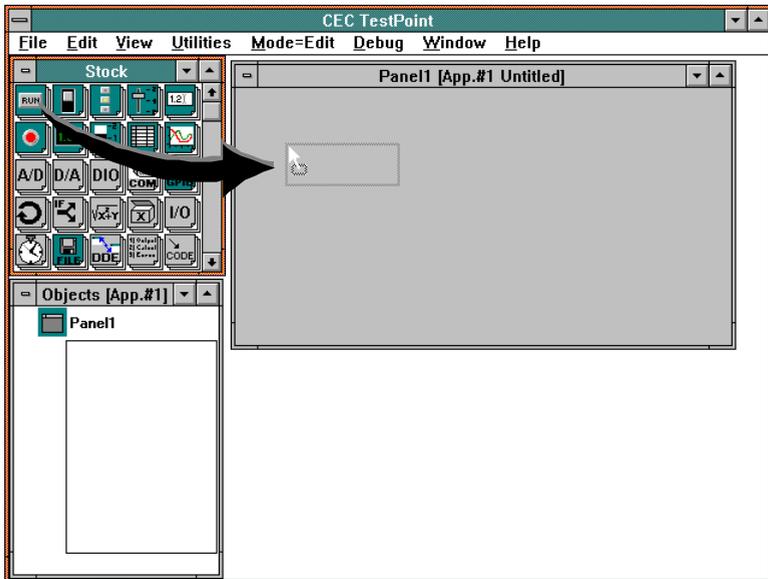


Der TestPoint-Editor beinhaltet vier Fenster: Das 'Panel Window', das 'Stock Window', das 'Object Window' und das 'Action List Window'. In diesen vier Fenstern spielt sich die gesamte Programm-erstellung in TestPoint ab. Im folgenden wird deren Verwendung genauer erklärt. Trotzdem werden wir gleich ein Beispiel durchführen, das Ihnen zeigt, wie eine TestPoint-Anwendung erstellt wird, und wie die vier Fenster zu verwenden sind. Diese Anwendung soll ganz einfach einen Wert von einem Voltmeter einlesen und auf dem Bildschirm darstellen.

## Hinzufügen eines 'Pushbuttons'

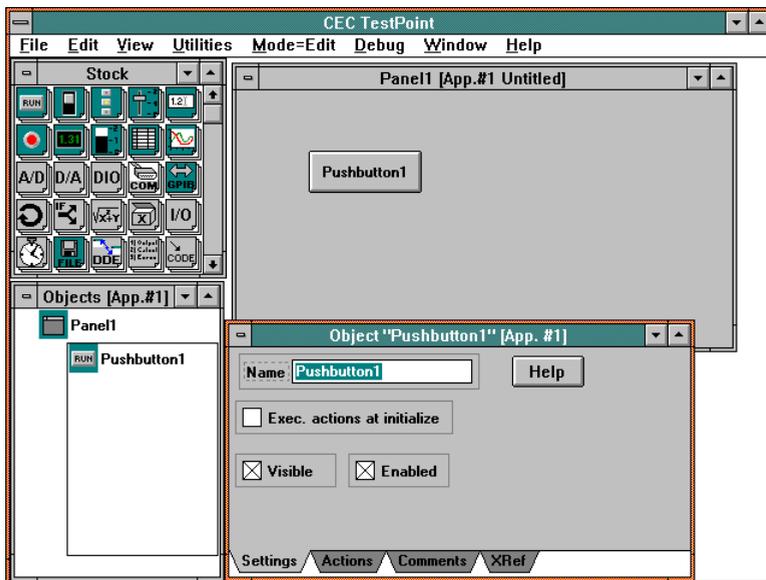
Ziehen Sie einen 'Pushbutton'  vom 'Stock Window' in das 'Panel'.

('Ziehen' heißt: Klicken Sie mit der Maus das 'Pushbutton'-Symbol an und lassen Sie die linke Maustaste gedrückt. Nun bewegen Sie die Maus mit dem Objekt einfach an die Stelle im 'Panel', wo Sie das Objekt positionieren wollen, und lassen die Maustaste wieder los)



- + Neue Objekte werden TestPoint-Anwendungen hinzugefügt, indem man sie einfach mit der Maus vom 'Stock Window' in das 'Panel' zieht ('drag and drop').

Wenn Sie die Maustaste loslassen, erscheint im 'Panel' das 'Pushbutton'-Objekt sowie das dazugehörige 'Settings Window'. Dieses Fenster erlaubt Ihnen, Informationen und Eigenschaften des Objektes zu bestimmen, zu definieren oder zu modifizieren.



Als erstes ist der Name des Objektes aktiviert (blau unterlegt), so daß Sie dem Objekt sofort jeden beliebigen Namen zuweisen können.

**Geben Sie als neuen Namen "Run Test" ein.**

Die anderen Voreinstellungen lassen Sie unverändert. Drücken Sie nun die Taste ENTER oder klicken Sie mit der Maus auf OK.

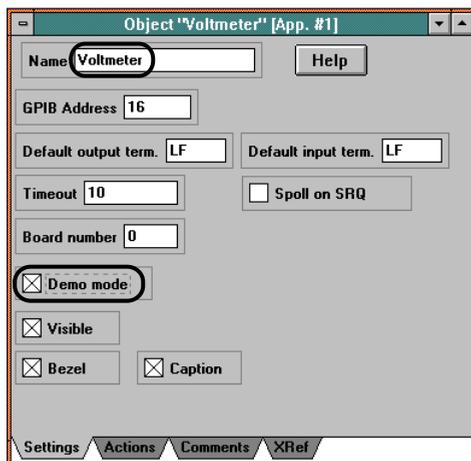
- + **Sie können jederzeit auf das Objekt oder auf sein Symbol einen Doppelklick ausführen (die linke Maustaste zweimal in kurzem Abstand drücken). Das 'Settings Window' des Objektes erscheint wieder, dessen Voreinstellungen Sie beliebig ändern können.**

Beachten Sie, daß das neue Objekt nicht nur im Panel grafisch dargestellt wird, sondern auch im 'Object Window', und daß die 'Action List' dieses Objektes im 'Action List Window' erscheint.

## Hinzufügen des GPIB-Voltmeters

Ziehen Sie nun das GPIB-Objekt  vom 'Stock Window' in das 'Panel'.

Im 'Settings Window' geben Sie den Namen "Voltmeter" ein, und markieren das Kästchen für "Demo mode".



Object "Voltmeter" [App. #1]

Name

GPIB Address

Default output term.  Default input term.

Timeout   Poll on SRQ

Board number

Demo mode

Visible

Bezel  Caption

Settings Actions Comments XRef

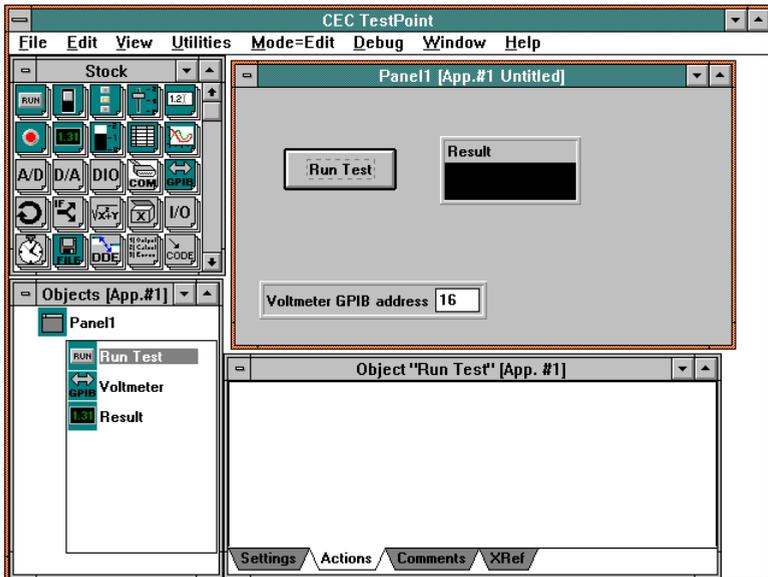
'Demo Mode' kann als Einstellung immer dann gewählt werden, wenn das dazugehörige Objekt auf eine Hardware zugreift (A/D-Karten, IEC-Bus-Karten usw.). Dies erlaubt Ihnen, ein Anwendungsprogramm zu entwickeln, auch wenn die dazu erforderliche Hardware im Moment nicht verfügbar oder installiert ist.

## Hinzufügen eines "Numeric Display"

Ziehen Sie das Objekt  vom 'Stock Window' in das 'Panel'.

Im 'Settings Window' geben Sie als neuen Namen "Result" ein, die anderen Voreinstellungen lassen Sie unverändert.

Der Bildschirm sollte nun folgendermaßen aussehen:



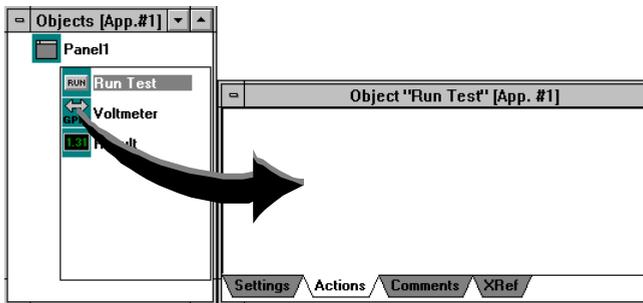
**Hinweis:** Um ein "Settings-Window" zu schließen oder zu verlassen, klicken Sie mit der linken Maustaste an eine beliebige Stelle des Bildschirms!

## Erstellen der "Action List"

Das 'Action List-Window' sollte nun die 'Action List' des 'Pushbutton'-Objektes 'Run Test' zeigen, welche im Moment natürlich eine leere Liste ist.

- + Die 'Action List' eines jeden beliebigen Objektes kann jederzeit durch einen Klick mit der rechten Maustaste auf das entsprechende Objekt im 'Object Window' sichtbar gemacht werden.

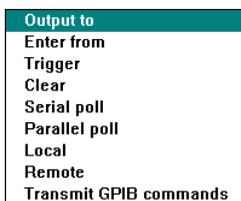
Eine Action-List ist eine Folge von Instruktionen oder Kommandos, die sequentiell ausgeführt werden, immer wenn für das dazugehörige Objekt ein Ereignis generiert wird. Es stellt also im Prinzip den Programmablauf Ihrer Anwendung dar. Im Falle des 'Pushbutton'-Objektes wird ein Ereignis generiert, wenn dieses vom Benutzer mit der Maus angeklickt wird. Sobald das Ereignis generiert wurde, wird die 'Action List' des 'Pushbutton' Objektes ausgeführt. In unserem Beispiel müssen zwei Aktionen durchgeführt werden: Einen Wert vom Instrument lesen und diesen dann auf dem Bildschirm darstellen.



Einen Wert vom Voltmeter lesen:  
Ziehen Sie das Voltmeter-Objekt vom 'Object Window' in die 'Action List' des Pushbuttons 'Run Test'.

- + 'Action lines' werden ganz einfach dadurch erzeugt, indem Sie das entsprechende Objekt vom 'Objects Window' in das 'Action List-Window' ziehen.

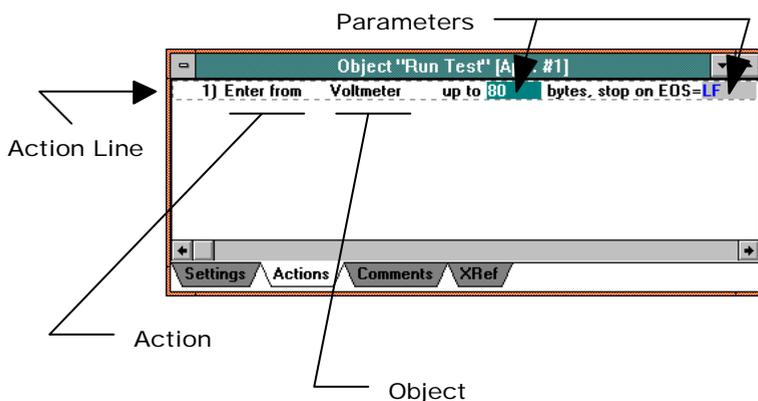
Wenn Sie die Maustaste loslassen, werden Sie sehen, daß eine Auswahl von mehreren Möglichkeiten erscheint, denn das GPIB- Objekt erlaubt mehr als nur eine Aktion.



**In unserem Fall wählen Sie die "Enter from" Aktion, welche Daten vom Gerät einliest.**

Eine 'Action Line', die deutlich beschreibt, was Sie ausführt, erscheint automatisch. Beachten Sie, daß eine solche Zeile Felder enthalten kann, in die Parameter eingefügt werden müssen/sollen. Im Falle der 'GPIB Enter From' Aktion, spezifiziert der Parameter die Anzahl der maximal einzulesenden Bytes, oder 'end-of-string' Zeichen (EOS), oder ob die Übertragung durch Aktivieren des EOI-Signals mit dem letzten DatenByte beendet werden soll.

**Klicken Sie auf den ersten Parameter (256) und geben Sie dort 80 ein.**

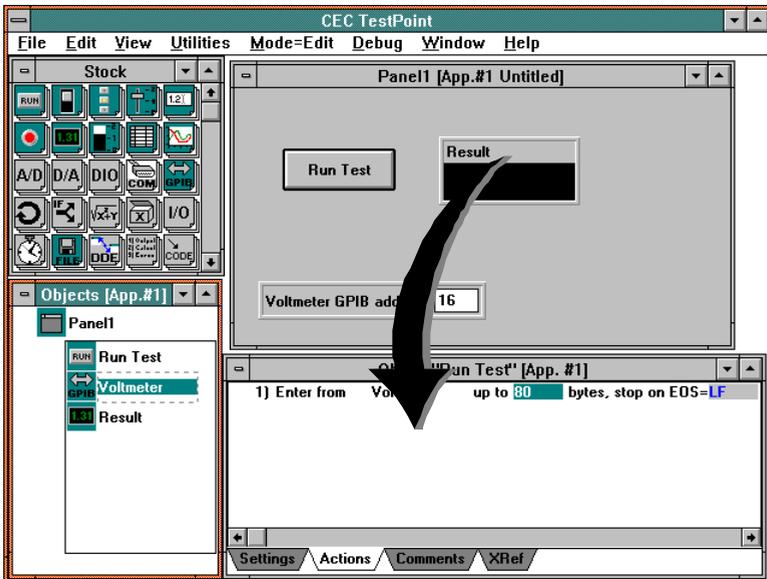


**+ Konstanten werden eingegeben, indem auf den gewünschten Parameter geklickt und der Wert über die Tastatur**

**eingeben oder ein Variablen Fenster (Data Entry, Slider, Selektor usw.) eingefügt wird.**

Das Ergebnis darstellen:

Ziehen Sie das 'Display Object' vom 'Panel' in die 'Action List'.



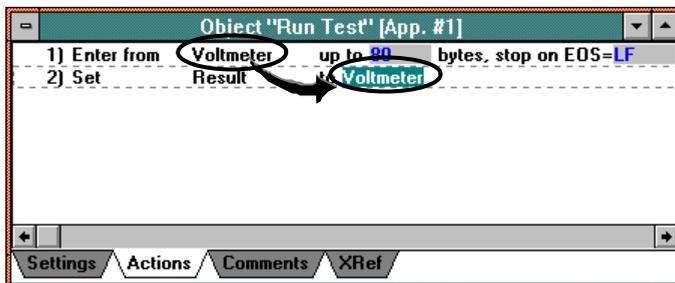
Beachten Sie, daß Sie das Objekt sowohl vom 'Panel' als auch vom 'Objects Window' in die 'Action List' ziehen können, um Zeilen in der 'Action List' zu generieren - das Resultat ist das Gleiche.

Die neue Zeile hat ein leeres Parameter-Feld, in dem Sie dem 'Display Object' mitteilen, was es darstellen soll. In diesem Beispiel möchten wir den Wert darstellen, den wir gerade vom Voltmeter gelesen haben. Dieser Wert ist im Voltmeter-Objekt gespeichert. Um diesen Wert nun im 'Display' Objekt darzustellen, ziehen Sie nun einfach das Voltmeter-Objekt in das leere Parameter-Feld in Zeile 2 der 'Action List'.



**Datenwerte von Objekten werden durch das Ausführen der entsprechenden Aktion verändert.**

Ziehen Sie den Namen 'Voltmeter' des GPIB-Objektes aus Zeile 1 in das leere Parameter Feld der Zeile 2.



Beachten Sie, daß Sie, um das leere Parameter-Feld in Zeile 2 zu füllen, nur den Namen 'Voltmeter' aus Zeile 1 ziehen müssen. Sie können aber auch das GPIB-Objekt 'Voltmeter' aus dem 'Objects Window' oder aus dem 'Panel' in dieses Feld ziehen, das Resultat ist das Gleiche.



**Objekte können sich wie Variablen verhalten - sie beinhalten Datenwerte. Sie können Datenwerte von Objekten verwenden, indem Sie die entsprechenden Objekte in die Parameterfelder ziehen.**

## ***Starten des Programms***

**Benutzen Sie die Menüzeile am oberen Bildschirmrand, um vom EDIT-Modus in den RUN-Modus zu wechseln.**

Der 'Run mode' ändert das Verhalten des 'Panels'. Im Edit-Modus können Sie mit der Maus die Größe und Position der Objekte verändern. Im Run-Modus verhalten sich die Objekte so wie im fertigen Anwendungsprogramm, d. h. wenn Sie auf den "Run Test" Pushbutton drücken, wird dessen Action List auch ausgeführt.

**Drücken Sie nun den "Run Test" Pushbutton.**

Ein "Meß-Wert" wird im "Numeric Display" erscheinen. Da wir Demo Mode im 'Settings Window' des Voltmeter-GPIB-Objektes gewählt haben, wird natürlich nur ein simulierter Wert dargestellt, da nicht auf die Hardware zugegriffen wurde. Wann immer Sie nun auf den 'Pushbutton' drücken, wird ein neuer Wert im 'Display' erscheinen.

**Benutzen Sie das Menükommando 'Mode', um wieder in den EDIT-Modus zu wechseln.**

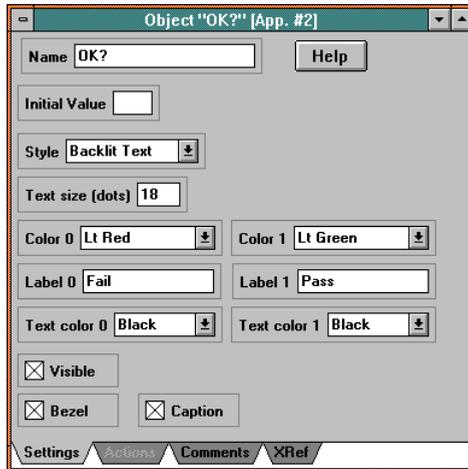
**Dieses Beispiel ist für Sie bereits vorgefertigt. Sie finden es im Verzeichnis \TUTORIAL unter dem Namen TEST1.TST.**

## Hinzufügen einer gut/schlecht-Prüfung

Nachdem eine Messung durchgeführt wurde, ist es häufig erforderlich, zu prüfen, ob der Wert über oder unter einer bestimmten Grenze liegt, und dies zu visualisieren. Für diese Aufgabe benötigen wir einen Indikator (z. B. eine optische Anzeige) in unserem Panel, sowie einen Mechanismus, der diese Prüfung durchführt.

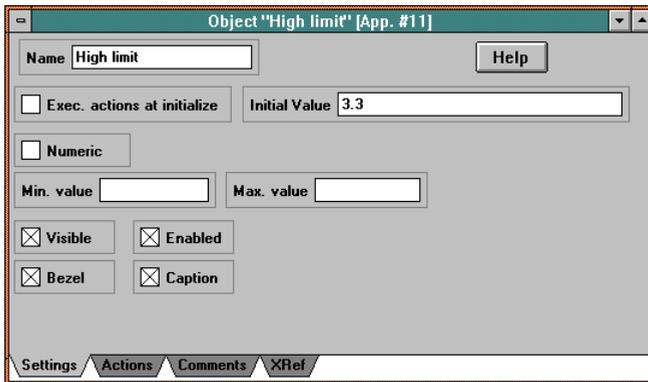
### Indikator

Ziehen Sie ein 'Indicator Object'  vom Stock in das Panel. Geben Sie ihm den Namen "OK?", und wählen Sie "Backlit text" als 'style'.



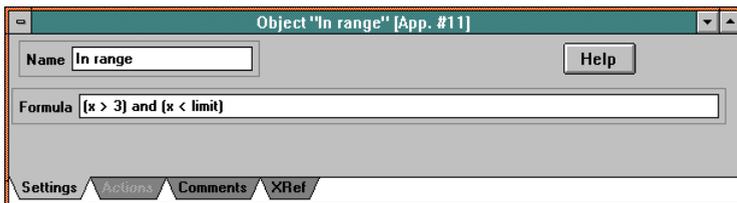
## Data-Entry

Ziehen Sie ein Data-Entry Objekt  vom Stock in das Panel und nennen Sie es 'High Limit'. Setzen Sie die Voreinstellungen in 'Initial Value' auf '3.3'.



## Mathematik

Ziehen Sie ein 'Math Object'  vom Stock in das Object Window. Nennen Sie es "In range", und geben Sie die Formel "(x > 3) and (x < limit)" ein:



Ein 'Math Object' kann jede beliebige Funktion mit beliebig vielen Variablen unter Verwendung einer Fülle von bereits vordefinierten Operationen durchführen. In diesem Beispiel möchten wir nur prüfen, ob ein einzelner Wert sich in gewissen Grenzen befindet. Das Ergebnis ist ein logischer Wert (True/False): 1 oder 0!

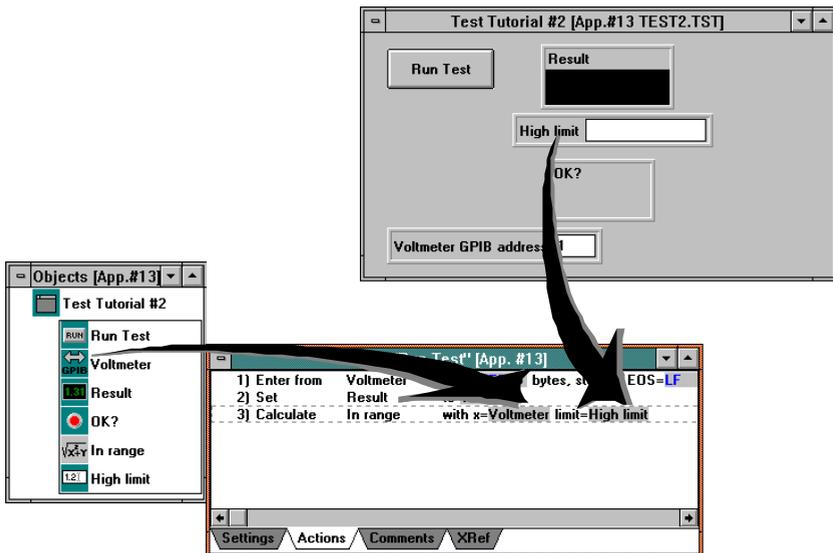
Nun fügen wir diese beiden neuen Schritte in unserer Action List hinzu.

**Klicken Sie mit der rechten Maustaste auf den "Run Test" Pushbutton im Panel-Fenster, um dessen Action List zu sehen. Ziehen Sie nun das "In range" Math Object in die Action List. Wählen Sie die "Calculate" action.**

Beachten Sie, daß die automatisch generierte Action Line ein leeres Parameterfeld für die Variable 'x' und 'limit' aus Ihrer Formel aufweist.

**Ziehen Sie nun das 'Voltmeter Object' vom 'Object Window' in dieses leere Feld hinter "x=" in der Action List.**

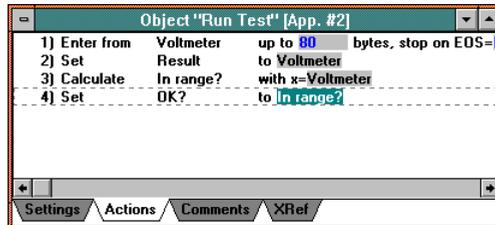
**Ziehen Sie das 'High Limit' Dateneingabe-Objekt vom Panel in das leere Feld hinter 'limit=' in der Action List.**



Diese neue Zeile führt nun die Formel aus mit dem Wert, der zuvor vom Voltmeter gelesen wurde, und der nun dem Voltmeter-Objekt zugeordnet ist. Das Ergebnis dieser Berechnung ist nun als logischer Wert dem Objekt "In range" zugeordnet.

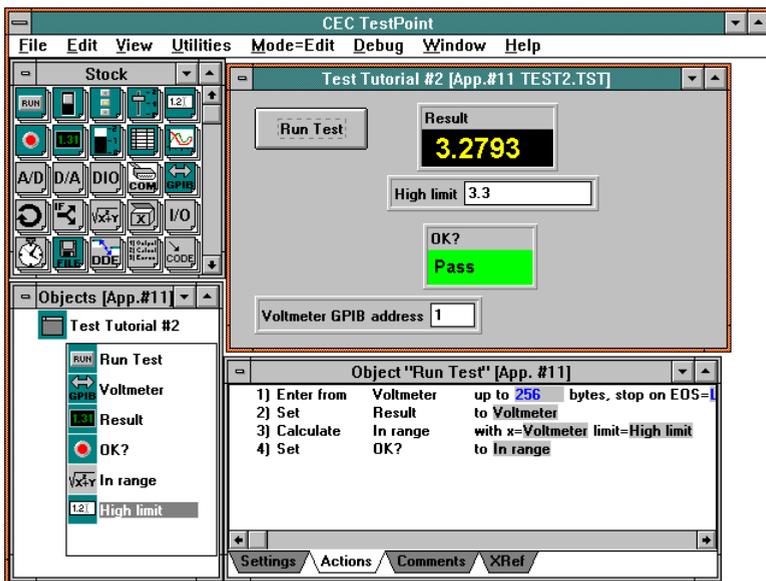
Ziehen Sie nun das "OK?" Indicator Object in die Action List.  
Wählen Sie die "Set" Action.

Ziehen Sie das "In range" 'Math Object' in das leere Feld der neuen Action Line.



Versuchen Sie nun, das Programm zu starten. Verwenden Sie das 'Mode'- Menükommando, um in den 'Run mode' zu gelangen. Drücken Sie den "Run Test"-Pushbutton mehrmals. Abhängig vom Voltmeter-Wert wird im Indicator entweder 'Pass' oder 'Fail' angezeigt. Verändern Sie die Werte im 'High Limit' Dateneingabefeld und klicken Sie nochmals auf den 'Run Test' Pushbutton.

**Wechseln Sie zurück in den 'Edit Mode'.**



Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *TEST2.TST*.

## **Hinzufügen eines weiteren Bedienelementes und einer neuen Action List**

Bis jetzt enthält unsere Anwendung nur eine einzige Action List, die sequentiell abgearbeitet wird. Wir werden nun ein weiteres Objekt hinzufügen, das uns erlaubt, das zu testende Bauteil zu wählen, sowie ein weiteres Gerät, eine Spannungsquelle, die das zu testende Bauteil mit einer bestimmten Spannung versorgt.

### **Der Selektor**

**Ziehen Sie ein 'Selector Object'  vom Stock in das Panel.**

Ein 'Selector Object' gibt dem Anwender die Möglichkeit, zwischen verschiedenen Alternativen zu wählen. Die 'Settings' für den Selektor beinhalten die Anzahl der Alternativen, die Bezeichnungen sowie die Werte, die den Bezeichnungen zugeordnet sind.

**Geben Sie "Component type" als Objektnamen ein.**

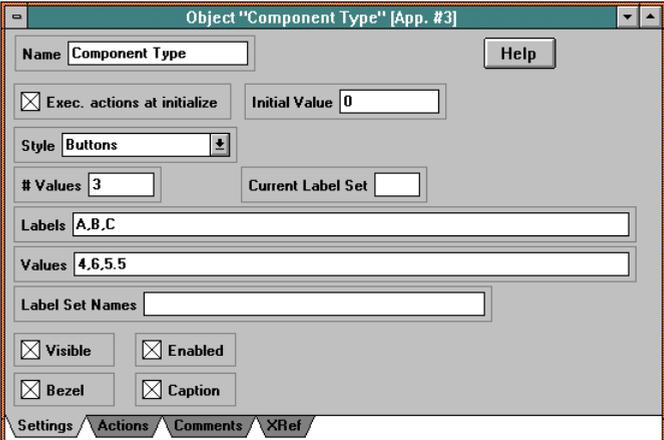
**Geben Sie "3" für "# values" ein.**

**Geben Sie "A,B,C" für die "labels" (Bezeichnungen) ein.**

**Geben Sie "4,6,5.5" für die dazugehörigen Werte (values) ein.**

**Markieren Sie das "Exec. actions at initialize" Feld.**

**Geben Sie "0" als "initial value" (Voreinstellung) ein.**



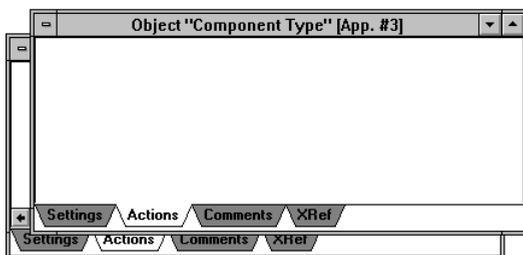
The screenshot shows a dialog box titled "Object 'Component Type' [App. #3]". It contains the following fields and controls:

- Name: Component Type
- Help button
- Exec. actions at initialize
- Initial Value: 0
- Style: Buttons
- # Values: 3
- Current Label Set: (empty)
- Labels: A,B,C
- Values: 4,6,5.5
- Label Set Names: (empty)
- Visible
- Enabled
- Bezel
- Caption
- Settings | Actions | Comments | XRef

**Ziehen Sie ein weiteres 'GPIB Object' vom Stock in das Panel, und geben Sie ihm den Namen "Voltage source".  
Markieren Sie das "Demo mode" Feld.**

Wann immer wir den Bauteiltyp wechseln, soll nun auch automatisch die entsprechende Spannung von der Spannungsquelle ausgegeben werden. Die Action List des 'Component type'-Selectors wird immer dann ausgeführt, wenn Sie den Selector betätigen.

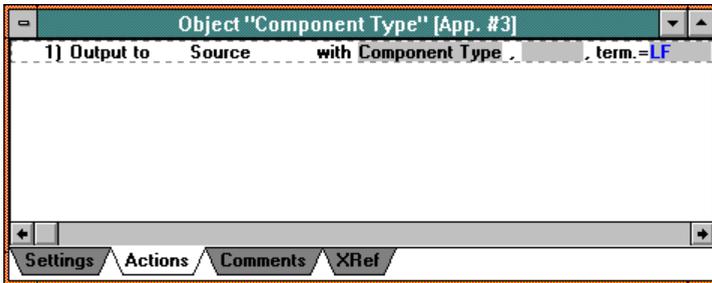
**Klicken Sie mit einer Maustaste auf den "Component Type"-Selector, um seine Action List sichtbar zu machen.**



**+ In einer Anwendung kann es viele verschiedene Action Lists geben. Jede wird dann ausgeführt, wenn das zugeordnete Objekt ein Ereignis empfängt.**

Beachten Sie, daß die Action List zunächst noch leer ist. Die Action List des "Run Test"-Pushbutton und die des "Component Type"-Selectors sind voneinander völlig unabhängig.

Ziehen Sie das "Voltage Source"-Objekt in die Action List.  
Wählen Sie die "Output to"-Action.  
Ziehen Sie den "Component Type"-Selector in das erste leere  
Parameterfeld der neuen Action Line.

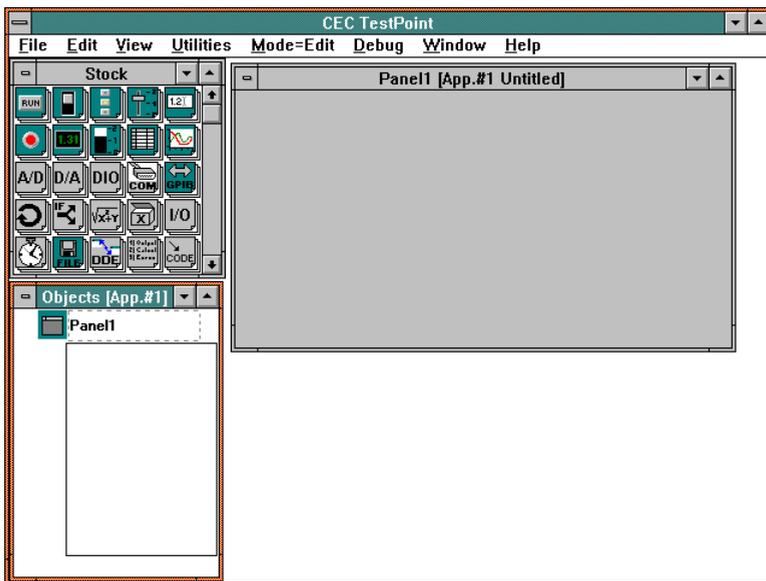


Der Wert des "Component Type"-Selektors ist 4, 6, oder 5.5 - die gewünschte Spannung, welche an die Spannungsquelle gesendet wird, in Abhängigkeit von dem gewählten Bauteiltyp.

**Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *TEST3.TST*.**

# Übung 'Meßdatenerfassung'

Wenn Sie die Testpoint-Entwicklungsumgebung starten, wird der Editor aktiviert, und es erscheint ein neues leeres Fenster (Panel), das Sie nun mit den gewünschten Objekten füllen können:

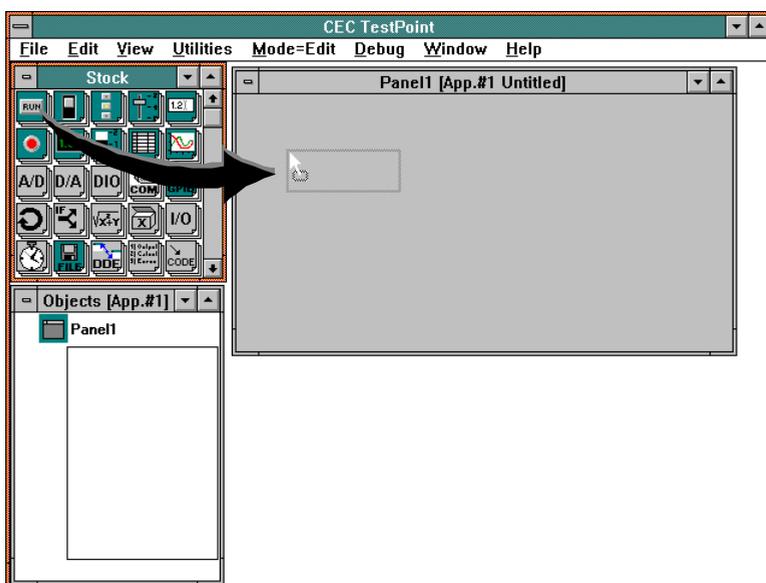


Der TestPoint-Editor beinhaltet vier Fenster: Das 'Panel Window', das 'Stock Window', das 'Object Window' und das 'Action List Window'. In diesen vier Fenstern spielt sich die gesamte Programmerstellung in TestPoint ab. Im folgenden wird deren Verwendung genauer erklärt. Trotzdem werden wir vorher ein Beispiel durchführen, das Ihnen zeigt, wie eine TestPoint Anwendung erstellt wird, und wie die vier Fenster zu verwenden sind.

Diese Anwendung soll ganz einfach einige Werte mit einer Meßdatenerfassungskarte aufnehmen und diese grafisch darstellen, wenn ein 'Pushbutton' gedrückt wird. Hinzufügen eines 'Pushbuttons'.

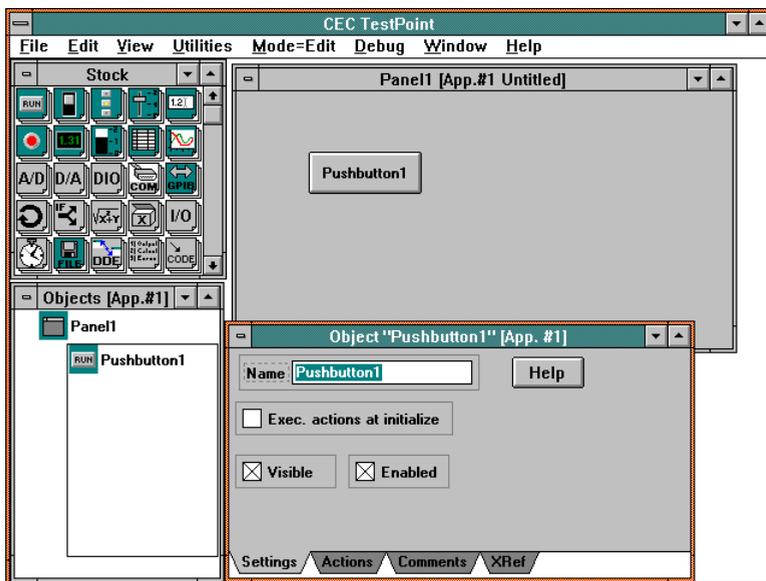
Ziehen Sie einen 'Pushbutton'  vom 'Stock Window' in das 'Panel'.

(Ziehen heißt: Klicken Sie mit der Maus das 'Pushbutton'-Symbol an und lassen Sie die linke Maustaste gedrückt. Nun bewegen Sie die Maus mit dem Objekt einfach an die Stelle im 'Panel', wo Sie das Objekt positionieren wollen, und lassen die Maustaste wieder los).



† **Neue Objekte werden TestPoint Anwendungen hinzugefügt, indem man sie einfach mit der Maus vom 'Stock Window' in das 'Panel' zieht ('drag and drop').**

Wenn Sie die Maustaste loslassen, erscheint im 'Panel' das 'Pushbutton Object' sowie das dazugehörige 'Settings Window'. Dieses Fenster erlaubt Ihnen, Informationen und Eigenschaften des Objektes zu bestimmen, zu definieren oder zu modifizieren.



Als erstes ist der Name des Objektes aktiviert (blau unterlegt), so daß Sie dem Objekt sofort jeden beliebigen Namen zuweisen können.

**Geben Sie als neuen Namen "Acquire" ein.**

Die anderen Voreinstellungen lassen Sie unverändert. Drücken Sie nun die Taste ENTER oder klicken Sie mit der Maus auf eine beliebige Stelle im TestPoint-Fenster.

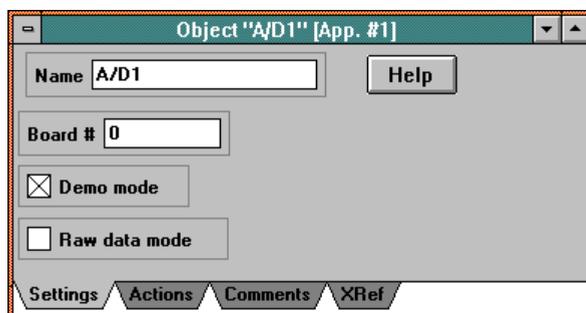
- ✦ **Sie können jederzeit auf das Objekt oder auf sein Symbol einen Doppelklick ausführen (die linke Maustaste zweimal in kurzem Abstand drücken). Das 'Settings Window' des Objektes erscheint wieder, dessen Voreinstellungen Sie beliebig ändern können.**

Beachten Sie, daß das neue Objekt nicht nur im Panel erscheint, sondern auch im 'Object Window', und daß die 'Action List' dieses Objektes im 'Action List Window' erscheint.

### ***Hinzufügen eines 'A/D Board' Objektes***

Ziehen Sie ein 'A/D Object'  vom 'Stock Window' in das 'Panel'.

Markieren Sie im 'Settings Window' das 'Demo mode' Feld.



Demo mode kann als Setting immer dann gewählt werden, wenn das dazugehörige Objekt auf eine Hardware zugreift (A/D-Karten, IEC-Bus- Boards usw.). Das erlaubt Ihnen, ein Anwendungsprogramm zu entwickeln, auch wenn die dazu erforderliche Hardware im Moment nicht verfügbar oder installiert ist.

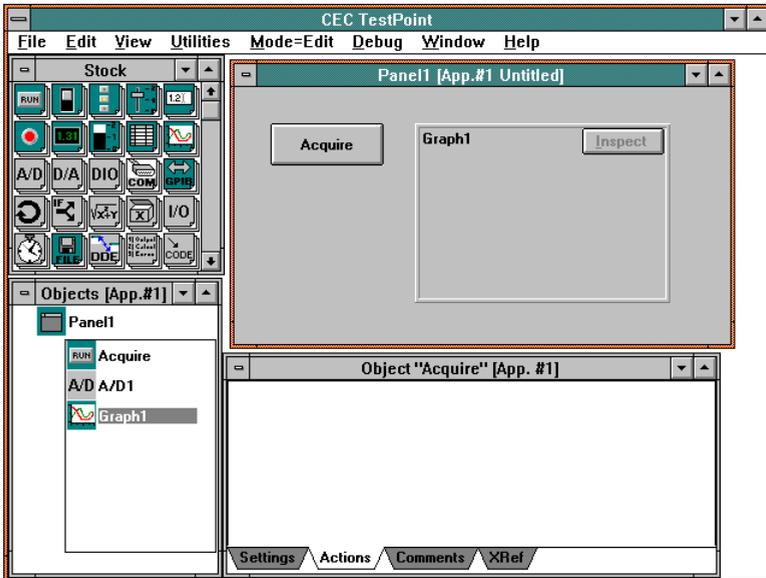
## Hinzufügen eines 'Graph Objects'

Ziehen Sie ein 'Graph Object'  vom 'Stock' in das 'Panel'.

Lassen Sie die Voreinstellungen unverändert.

**Hinweis:** Sie können später die Einstellungen im Grafik-Fenster beliebig verändern, um einige Darstellungsmöglichkeiten auszuprobieren.

Der Bildschirm sollte nun folgendermaßen aussehen:



## ***Erstellen der "Action List"***

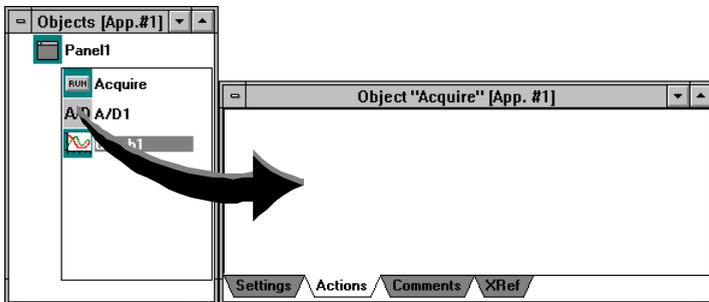
Das Action List-Window sollte nun den Inhalt der Action List des Pushbutton-Objektes "Acquire" zeigen, die im Moment natürlich eine leere Liste darstellt.

- ✦ **Die Action List eines jeden beliebigen Objektes kann jederzeit durch einen Klick mit der rechten Maustaste auf das entsprechende Objekt im 'Object Window' sichtbar gemacht werden.**

Eine Action List ist eine Folge von Instruktionen oder Kommandos, die sequentiell ausgeführt wird, immer wenn für das dazugehörige Objekt ein Ereignis generiert wird. Es stellt also im Prinzip den Programmablauf Ihrer Anwendung dar. Im Falle des Pushbutton-Objects wird ein Ereignis generiert, wenn der Pushbutton vom Benutzer mit der Maus angeklickt wird (oder wenn in der Action List eines beliebigen anderen Objektes dieser Pushbutton gedrückt wird). Sobald das Ereignis generiert wurde, wird die Action List des Pushbutton Objects ausgeführt. In unserem Beispiel müssen zwei Aktionen durchgeführt werden: Werte von der A/D-Karte lesen und diese dann grafisch darstellen.

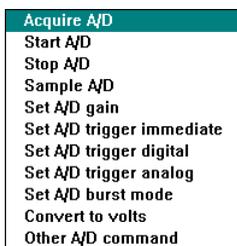
## ***Daten aufnehmen***

**Ziehen Sie das 'A/D Object' vom 'Object Window' in die Action List.**



- + **Action Lines werden erzeugt, indem man einfach Objekte in die Action List zieht.**

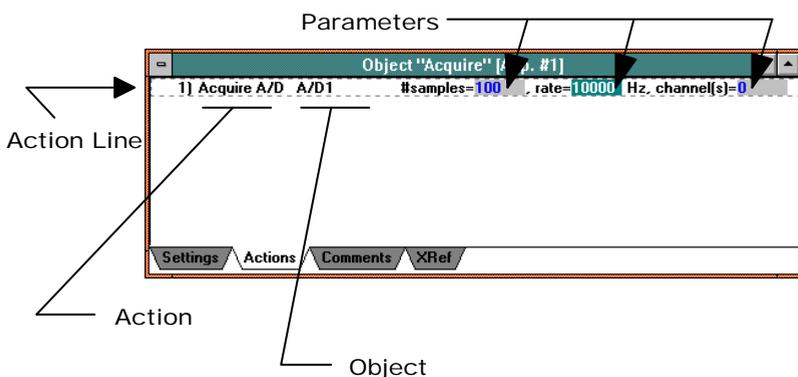
Wenn Sie die Maustaste loslassen, erscheint ein Auswahlmü, das es Ihnen erlaubt, zwischen verschiedenen 'Actions' zu wählen, denn ein A/D-Objekt erlaubt verschiedene Aktionen.



**Wählen Sie die "Acquire A/D"-Action, die analoge Daten mit der A/D-Karte aufnimmt.**

Eine Textzeile (Action Line), die deutlich beschreibt, welche Aufgabe sie erledigt, erscheint automatisch. Beachten Sie, daß sie leere Parameterfelder für die auszuführende Aktion aufweist. Im Falle der Acquire A/D-Action stehen die Parameter für die Anzahl der aufzunehmenden Werte, die gewünschte Datenerfassungsrate, sowie die zu verwendenden Kanäle.

**Geben Sie 100 für '# samples', 10000 für 'rate' und 0 für 'channels' ein.**



- +** Konstanten werden eingegeben, indem auf den gewünschten Parameter geklickt und der Wert über die Tastatur eingegeben oder ein Variablen Fenster (Data Entry, Slider, Selektor usw.) eingefügt wird.

## Darstellen der Daten

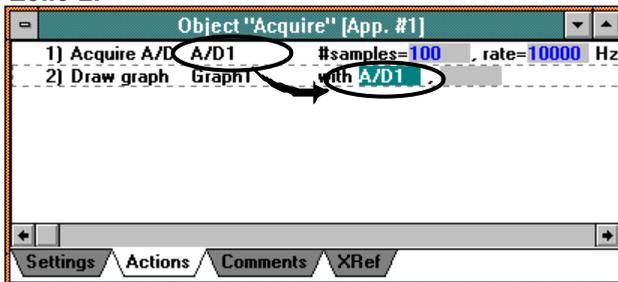
Ziehen Sie das 'Graph Object' vom Panel in die Action List.  
Wählen Sie die "Draw graph"-Funktion aus.

Beachten Sie, daß Sie die Objekte sowohl vom Panel als auch vom 'Object Window' in die Action List ziehen können - das Resultat ist das Gleiche.

Die neue Zeile weist ein leeres Parameterfeld für die darzustellenden Daten auf. In diesem Beispiel wollen wir die Daten darstellen, die wir gerade mit unserem A/D-Objekt aufgenommen haben. Diese Daten sind nun im A/D-Objekt gespeichert. Um die Daten darzustellen, ziehen wir deshalb das A/D Objekt in das leere Parameterfeld in Zeile 2.

† **Datenwerte von Objekten verändern sich als Ergebnis ausgeführter Aktionen.**

Ziehen Sie den Namen des A/D-Objektes von Zeile 1 in das leere Feld von Zeile 2.



Ein zweites leeres Feld erscheint in der Action Line, denn es können mehrere Datensätze von einem Graph-Objekt gleichzeitig dargestellt werden. In diesem Beispiel stellen wir nur einen Vektor dar und lassen das zweite Feld leer.

Beachten Sie, daß Sie eine Datenreferenz zwischen den Objekten herstellen können, indem Sie das Objekt aus dem 'Object Window' in das Parameterfeld ziehen, oder den Namen des Objektes aus einer anderen Action-Line ziehen.

† **Objekte können sich wie Variablen verhalten - sie beinhalten Datenwerte. Sie können Datenwerte von**

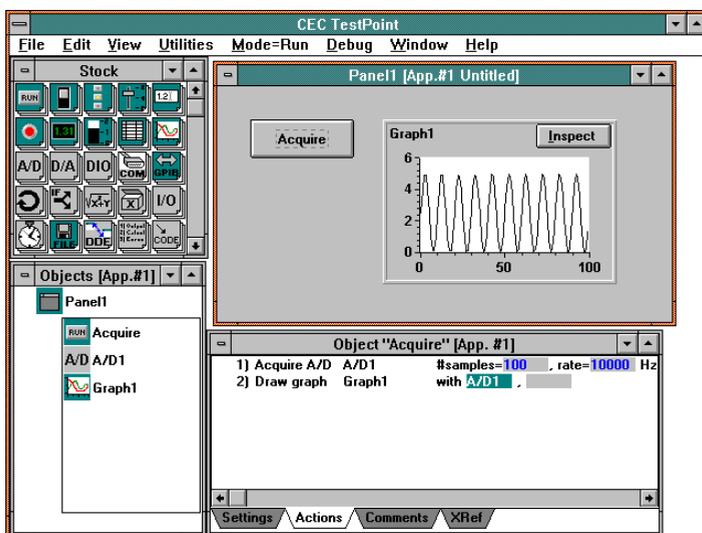
**Objekten verwenden, indem Sie die entsprechenden Objekte in die Parameterfelder ziehen.**

## Starten Sie das Programm

Verwenden Sie das 'Mode'-Menükommando, um in den 'Run Mode' zu gelangen.

Der 'Run Mode' ändert das Verhalten des 'Panel Windows'. Im 'Edit Mode' können Sie mit der Maus die Größe und die Position der Objekte verändern. Im 'Run Mode' verhält sich das Panel wie im fertigen Anwendungsprogramm - das Drücken des Pushbuttons erzeugt ein Ereignis, und seine Action List wird ausgeführt.

Drücken Sie den "Acquire" Pushbutton.



Daten werden in der Grafik erscheinen. Da wir für das A/D-Objekt den 'Demo Mode' gewählt haben, wird nicht auf Hardware zugegriffen, sondern die Daten werden nur simuliert.

Verwenden Sie das 'Mode'-Menükommando, um wieder in den 'Edit Mode' zu gelangen.

Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *DATAACQ1.TST*.

## ***Versuchen wir es mit einem Linienschreiber***

Das erste Beispiel, das wir durchgeführt haben, nimmt Daten auf und stellt diese grafisch dar. Das ist sicher ausreichend, wenn es um kleinere Datenmengen geht, die mit einer relativ hohen Geschwindigkeit aufgenommen wurden. Es gibt aber auch eine große Anzahl von Anwendungen, die Daten über einen großen Zeitraum zu erfassen und diese gleichzeitig auf dem Bildschirm darzustellen. Für diesen Zweck benötigen wir eine Datenerfassung im Hintergrund (Background) und eine Art Linienschreiber zur Darstellung der erfaßten Meßwerte.

## ***Einstellen des 'Graph' als Linienschreiber 'Strip Chart'***

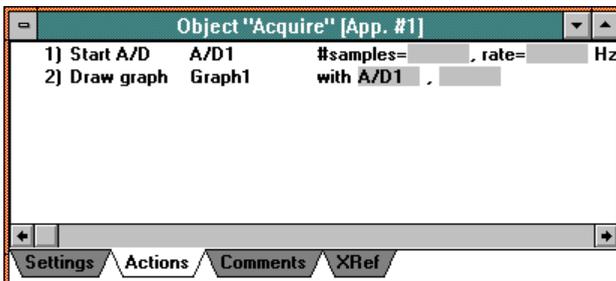
Führen Sie einen Doppelklick auf das 'Graph Object'-Symbol im 'Object Window' aus, um dessen 'Settings-Window' zu öffnen. Ändern Sie den "Mode" im 'Setting-Window' nach 'Strip Chart'.

## ***Hintergrunddatenerfassung***

Klicken Sie auf die Bezeichnung "Acquire A/D" in Zeile 1 der Action List. Verwenden Sie die 'Entf' ('Del')-Taste auf Ihrer Tastatur, um diese Action-Line zu löschen.

- † Objekte können an jeden beliebigen Ort einer Action List gezogen werden, um dort eine neue 'Action Line' zu erzeugen.

Ziehen Sie nun das A/D- Objekt abermals in die Action List vor die Action-Line 1. Wählen Sie die "Start A/D"-Aktion aus.



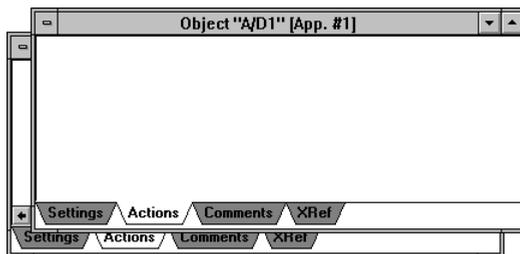
**Geben Sie "continuous" ein für '# samples', 1.01 für 'rate', und lassen Sie die anderen Voreinstellungen unverändert.**

(Bemerkung: Wir verwenden 1.01 Hz als Datenerfassungsrate, denn die simulierte Datenquelle liefert uns Daten wie von einem Frequenzgenerator mit genau 1 KHz Signalfrequenz. Wenn wir Daten mit 1 Hz von einer Quelle mit 1 KHz aufnehmen würden, würden wir im Linienschreiber nur eine Gerade sehen [das ist ein Sonderfall des Aliasing Effektes]).

Die 'Start A/D'-Aktion beginnt sofort, Daten aufzunehmen, aber sie wartet nicht, bis Daten auch wirklich ankommen, um diese darzustellen, sondern Action-Line 2 wird sofort ausgeführt, obwohl noch keine Daten erfaßt wurden. Deshalb können wir die Action-Line 2 nicht mehr verwenden, um unsere Daten sichtbar zu machen.

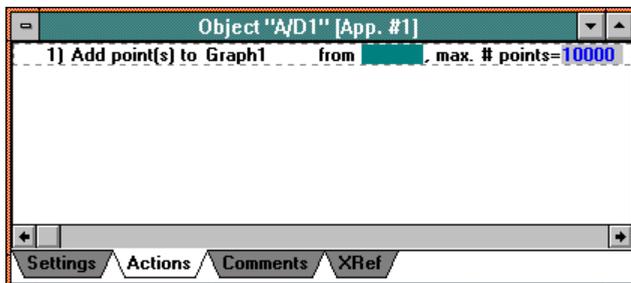
### **Grafik in der 'A/D' Action List**

**Klicken Sie auf den Begriff "Draw graph" in Zeile 2. Drücken Sie die 'Entf' ('Del')-Taste, um diese Zeile zu löschen. Klicken Sie mit der rechten Maustaste auf das 'A/D Object'-Symbol, um dessen Action List sichtbar zu machen (welche noch leer ist). Hierfür bitte die rechte Maustaste verwenden.**

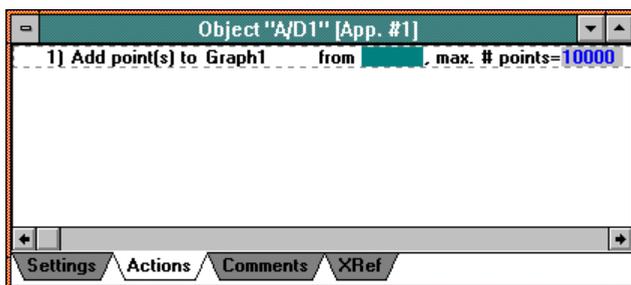


- + In einer Anwendung kann es viele verschiedene Action Listen geben. Jede wird dann ausgeführt, wenn das zugeordnete Objekt ein Ereignis empfängt.**

Ziehen Sie das 'Graph Object' in die 'A/D'-Action List.  
Wählen Sie die "Add point(s) to" Action aus.



Ziehen Sie das 'A/D Object'-Symbol in das leere Feld in Zeile 1.



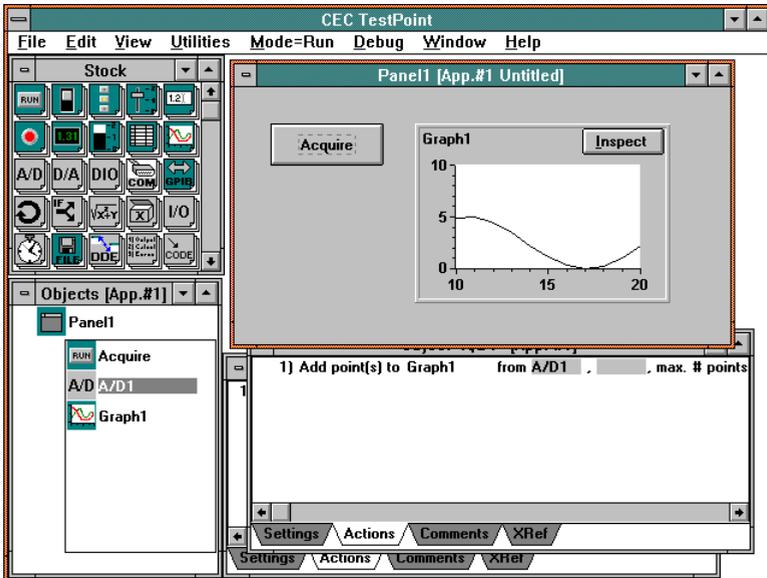
Die Action List des A/D-Objektes wird ausgeführt, immer wenn ein Meßwert durch die "Start A/D"-Aktion aufgenommen wurde. Diesen Meßwert wollen wir der Grafik hinzufügen.

Sie sollten nun zwei Action Lists haben: Die des Pushbutton, welcher die Datenerfassung startet, und die des A/D-Objektes, welche einen Wert, sobald er eintrifft, zur Grafik hinzufügt.

**Hinweis:** Action-Lists für Objekte können durch Klicken der rechten Maustaste auf das gewünschte Objekt sichtbar gemacht werden.

Starten Sie die Anwendung

Wechseln Sie in "Mode=Run" mit dem Menükommando.  
Drücken Sie den "Acquire" Pushbutton.



Die Meßwerte erscheinen, und der Graph wandert (scrollt) nach links.

Sie können nun auf den "Inspect"-Pushbutton innerhalb des Grafikensters drücken. Dadurch wird eine Kopie der aktuellen Grafik in einem neuen Fenster dargestellt. Sie können nun auch die Daten betrachten, die bereits nach links aus der Grafik geschoben wurden. Beachten Sie, daß der Linienschreiber währenddessen weiterhin aktualisiert wird.

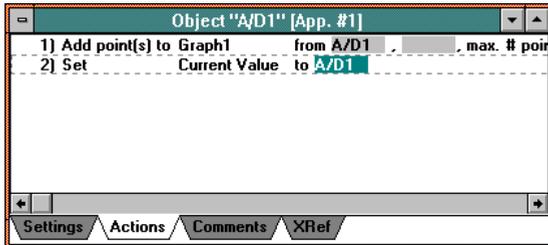
Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis **TUTORIAL** unter dem Namen **DATAACQ2.TST**.

## Meßwertdarstellung mit einem 'Bar Graph Indicator'

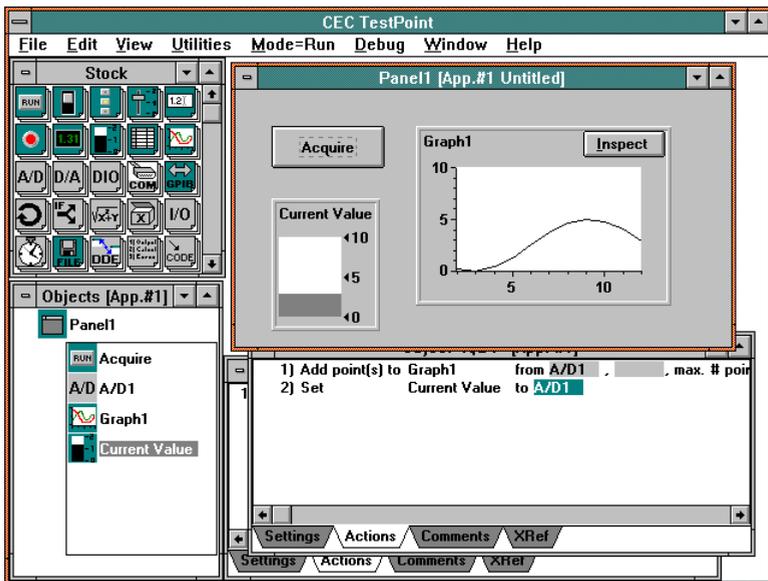
Ziehen Sie ein 'Bar Object' vom Stock in das Panel.  
Geben Sie ihm den Namen "Current value".

Ziehen Sie dieses Objekt in die 'A/D Action' List.

Ziehen Sie das A/D-Objekt in das neue leere Parameterfeld.



Starten Sie das Programm.



Wechseln Sie wieder zurück in den Edit Mode.

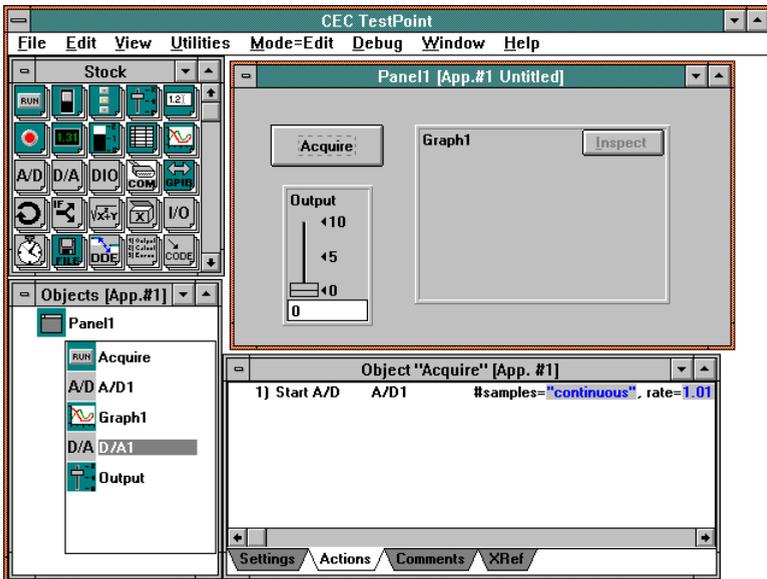
## Hinzufügen eines analogen Ausgangs (D/A)

Als nächstes wollen wir einen analogen Ausgang mit einem Schieberegler (Slider) ansteuern.

Ziehen Sie ein 'D/A Object'  vom Stock in das Panel.  
Markieren Sie das 'Demo mode'- Feld im Settings Fenster.

## Hinzufügen eines 'Sliders'

Ziehen Sie ein 'Slider Object' vom Stock in das Panel.  
Geben Sie ihm den Namen "Output" (siehe Bild unten).



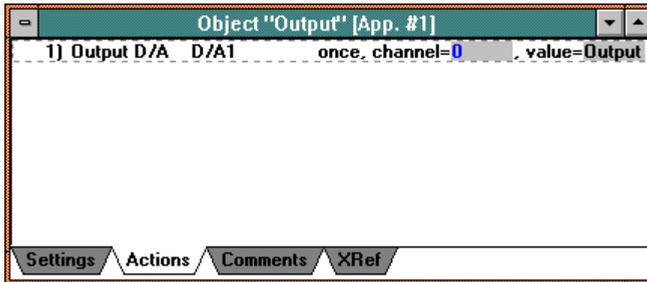
## Erstellen der Action List

Klicken Sie mit der rechten Maustaste auf das 'Output'-Slider-Symbol im 'Object Window', um seine Action List zu sehen.

Ziehen Sie das D/A-Objekt in die Action List.

Wählen Sie die "Output D/A"-Aktion aus.

Ziehen Sie das 'Slider Object' in das 'value'-Parameterfeld (siehe Bild unten).



Wann immer sich der Wert des Sliders ändert, wird seine Action List ausgeführt, und der entsprechende Wert an den analogen Ausgang geschickt.

**Starten Sie diese Anwendung.**

**Drücken Sie den 'Acquire'-Pushbutton.**

**Bewegen Sie den Slider (Schieberegler) mit der Maus.**

**Kehren Sie zum 'Edit Mode' zurück.**

- + **Multitasking:** Sie können den "Acquire"-Pushbutton drücken, um den Linienschreiber zu starten. Sie können nun mit dem Schieberegler den analogen Ausgang ansteuern, während der Linienschreiber weiterläuft.

Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *DATAACQ3.TST*.

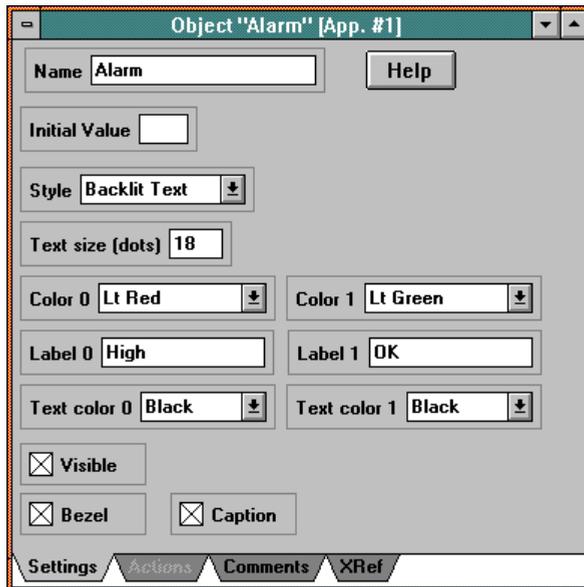
## Hinzufügen von Alarmgrenzen

Eine typische Anwendung ist es, darzustellen, ob sich ein Wert in einem definierten erlaubten Bereich, also innerhalb bestimmter "Grenzen", befindet.

Um dies zu prüfen, benötigen wir ein 'Math Object' und ein 'Indicator Object'.

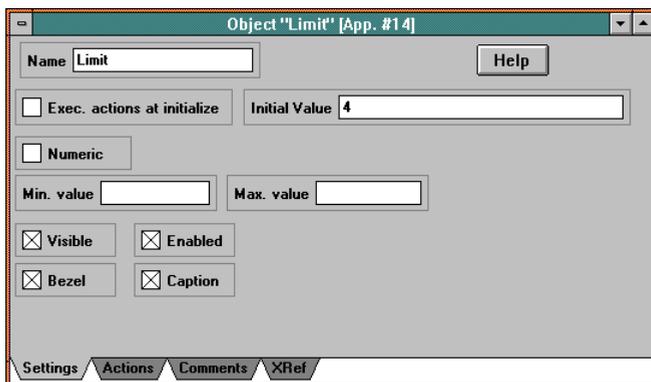
## Hinzufügen eines 'Indicators'

Ziehen Sie ein 'Indicator Object' vom Stock in das Panel. Nennen Sie es 'Alarm'. Wählen Sie 'Backlit Text' als 'Style', und 'Label 0' als 'High', 'Label 1' als 'OK' (siehe Bild unten).



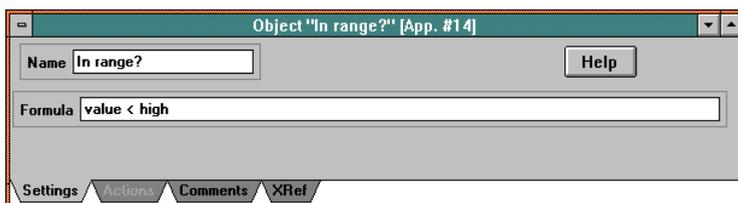
Hinzufügen eines 'Data Entry'-Objektes (zur Daten- oder Wert-eingabe)

Ziehen Sie das 'Data Entry'-Objekt  vom Stock ins Panel.  
Benennen Sie es 'Limit' und setzen Sie den 'Initial Value' auf 4.



Hinzufügen des 'Math'-Objektes (Mathematik-Objektes)

Ziehen Sie ein 'Math Object'  vom Stock in das Panel.  
Nennen Sie es "In range?"  
Geben Sie als Formel "value < high" ein.



## Hinzufügen von 'Actions'

Klicken Sie mit der rechten Maustaste auf das 'A/D'-Objekt-symbol im 'Object Window', um die Action List auszuwählen.

Ziehen Sie das "In range?"-Objekt in die Action List.

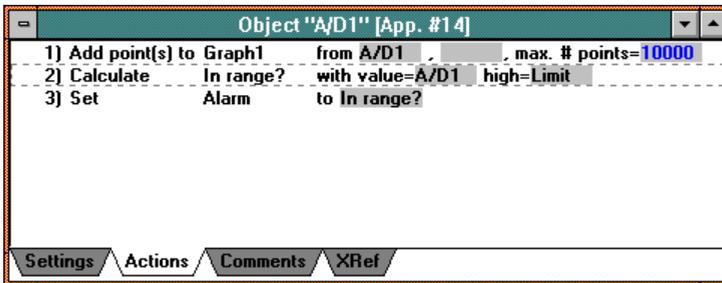
Wählen Sie die "Calculate"-Aktion aus.

Ziehen Sie das 'A/D'-Objekt in das "value"-Parameterfeld und 'Limit' in das Feld für 'high'.

Ziehen Sie das 'Alarm'-Objekt in die Action List.

Wählen Sie die 'Set'-Aktion aus.

Ziehen Sie das 'In range?'-Objekt in das Parameterfeld.



Wann immer ein Wert aufgenommen wird, wird die Action List ausgeführt. Zuzüglich zum Hinzufügen eines Punktes zum Graph wird nun auch überprüft, ob der Wert kleiner als 4 ist, und der Indicator entsprechend gesetzt.

**Starten Sie das Programm.**

**Drücken Sie den 'Acquire'-Pushbutton.**

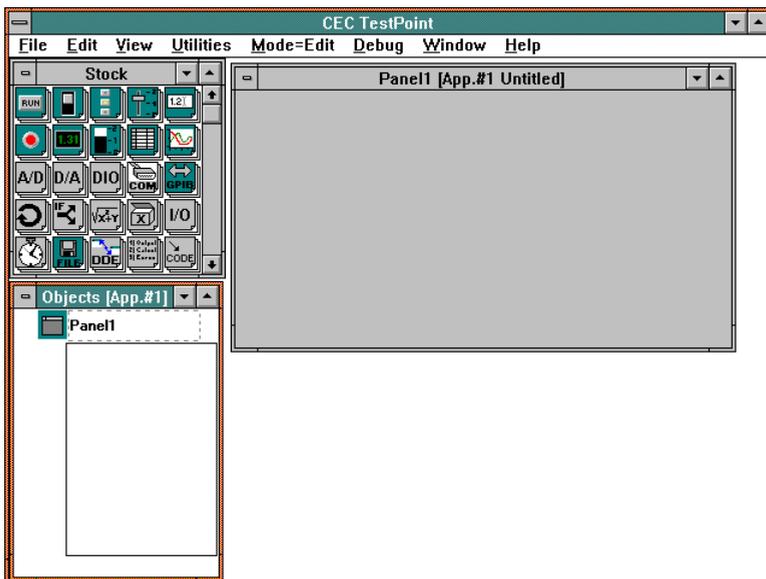
Beobachten Sie, wie der Indicator den Status des Wertes veranschaulicht.

**Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *DATAACQ4.TST*.**

# Übung 'Analyse'

---

Wenn Sie die Testpoint-Entwicklungsumgebung starten, wird der Editor aktiviert und es erscheint ein neues leeres Fenster (Panel), das Sie mit den gewünschten Objekten füllen können:

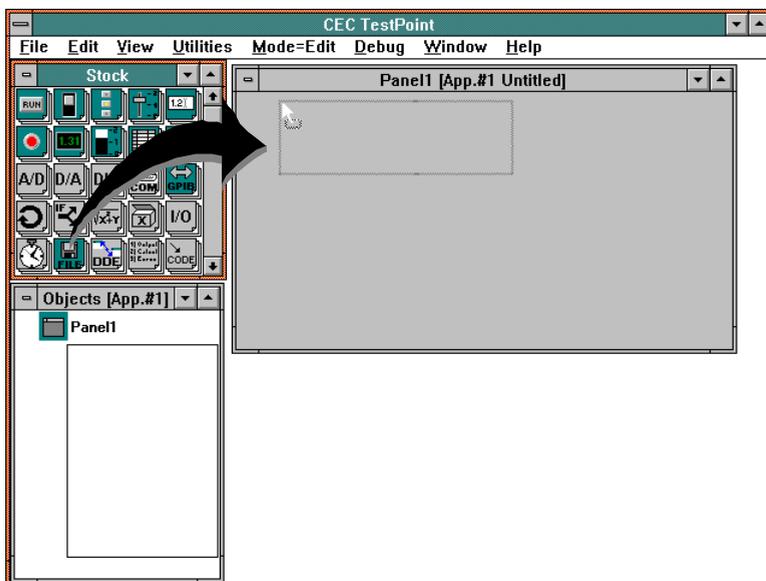


Der TestPoint Editor beinhaltet vier Fenster: Das 'Panel Window', das 'Stock Window', das 'Object Window' und das 'Action List Window'. In diesen vier Fenstern spielt sich die gesamte Programmiererstellung innerhalb von TestPoint ab. Im folgenden wird deren Verwendung genauer erklärt. Trotzdem werden wir gleich ein Beispiel durchführen, das Ihnen zeigt, wie eine TestPoint Anwendung erstellt wird, und wie die vier Fenster zu verwenden sind.

Diese Anwendung soll Daten aus einer Datei lesen und diese grafisch darstellen.  
Hinzufügen eines 'File objects'.

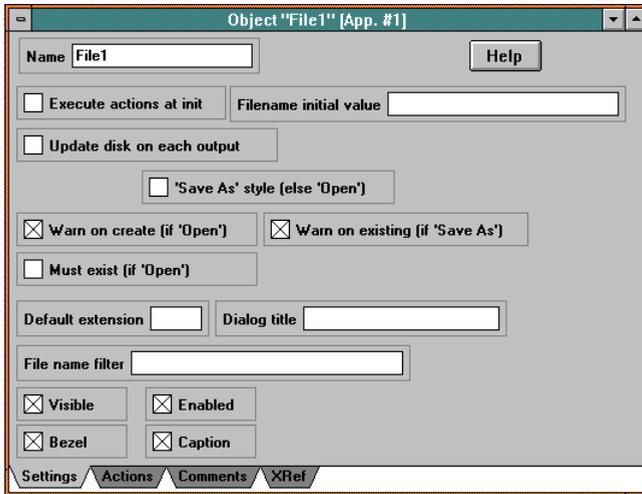
Ziehen Sie ein 'File object'  vom Stock in das 'Panel Window'.

(Ziehen heißt: Klicken Sie mit der Maus das 'file'-Icon an und halten Sie die linke Maustaste gedrückt. Nun bewegen Sie die Maus mit dem Objekt einfach an die Stelle im 'Panel', wo Sie das Objekt positionieren wollen, und lassen die Maustaste wieder los).



+ Neue Objekte werden einer TestPoint Anwendung hinzugefügt, indem man sie mit der Maus vom Stock holt und im 'Panel Window', 'Object Window' oder 'Action List Window' ablegt (Drag and Drop!!!).

Wenn Sie die Maustaste wieder loslassen, erscheint das 'File Object' im 'Panel Window'. Ebenso erscheint ein 'Settings Window', in das Sie Informationen über das Objekt eintragen können.



Als erstes ist der Name des Objektes aktiviert (blau hinterlegt) so daß Sie ihn sofort eingeben können.

**Geben Sie als Namen für das Objekt: "Data file" ein.  
Geben Sie folgenden String in die 'File name filter'-Zeile ein:  
*Data files/\*.DAT*  
(Die senkrechten Linien werden als Separator verwendet).**

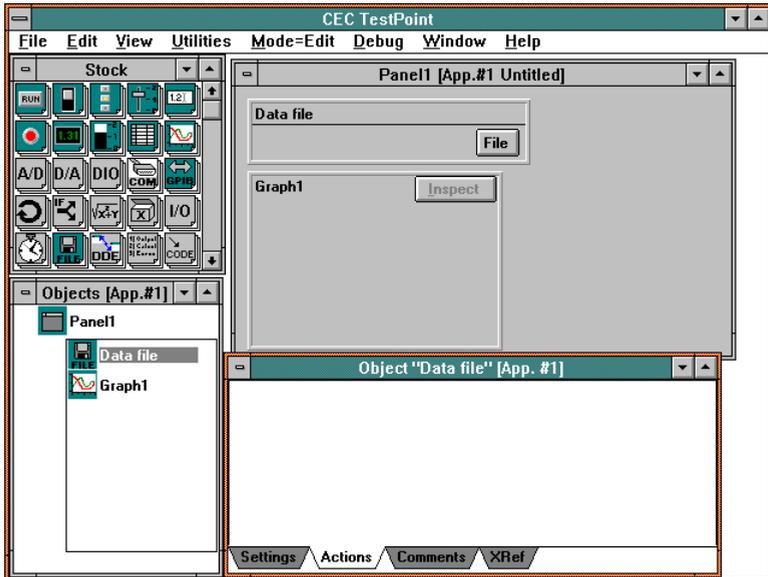
Die anderen Voreinstellungen lassen Sie unverändert. Drücken Sie nun einfach die Taste ENTER (RETURN, EINGABE), um dieses 'Settings Window' zu schließen.

**+ Sie können jederzeit mit einem Doppelklick auf das Objekt oder sein Symbol im 'Object Window' das 'Settings Window' wieder öffnen, um es zu editieren.**

## Hinzufügen eines 'Graph Objects'

Ziehen Sie ein 'Graph Object'  vom Stock in das Panel. Lassen Sie die Voreinstellungen (Settings) unverändert.

Der Bildschirm sollte nun folgendermaßen aussehen:



## ***Erstellen der 'Action List'***

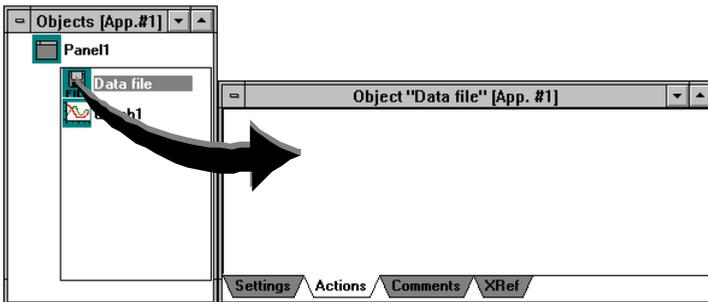
Das 'Action List Window' sollte nun die Action List des 'File'-Objektes zeigen, welche momentan noch leer ist.

- + **Die Action List jedes Objektes im 'Object Window' kann durch Klicken mit der rechten Maustaste auf das Symbol im 'Objects Window' sichtbar gemacht werden.**

Eine 'Action List' ist eine Folge von Aktionen, die ausgeführt wird, sobald für das Objekt ein Ereignis (Event) generiert wird. Im Falle des 'File'-Objektes geschieht dies durch die Auswahl eines neuen Filenamens (oder durch eine Aktion in einer anderen Action List). In unserem Beispiel müssen zwei Aktionen ausgeführt werden, wenn der Anwender einen Filenamens auswählt: Die Daten aus dem File einlesen und diese dann grafisch darstellen.

## ***Daten einlesen***

Ziehen Sie das 'File'-Objekt vom 'Object Window' in die 'Action List'.



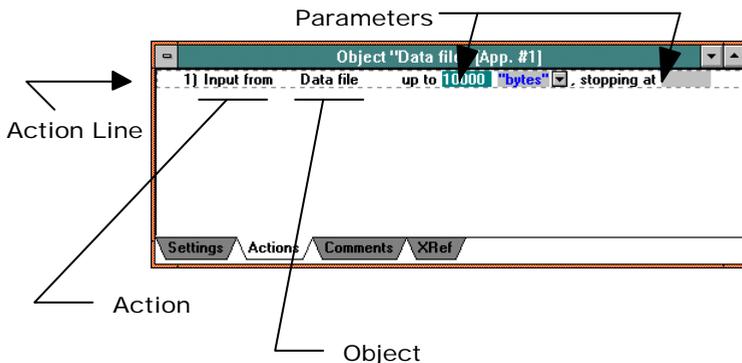
- + **'Action Lines' werden erzeugt, indem man Objekte in die Action List zieht (Drag and Drop).**

Wenn Sie die Maustaste wieder loslassen, erscheint ein Auswahlmnü (Pop up Menu), denn das 'File'-Objekt bietet mehr als nur eine Aktion...

- Open
- Output to
- Input from
- Close
- Erase
- Set filename of
- Get filename of
- Does file exist
- Set file position
- Get file size
- Get file number

**Wählen Sie die "Input from"-Aktion, welche die Daten aus dem File (Datei) einliest.**

Der Text der 'Action Line' erscheint automatisch und beschreibt die auszuführende Aktion. Beachten Sie, daß diese Zeile Parameterfelder beinhaltet, die die Aktion betreffen. Im Falle der "Input from"-Aktion, beschreiben die Parameter die maximale Anzahl der einzulesenden Bytes und optional einen Terminator, der die Operation beenden kann. In unserem Beispiel verwenden Sie einfach die vorgegebenen Einstellungen.

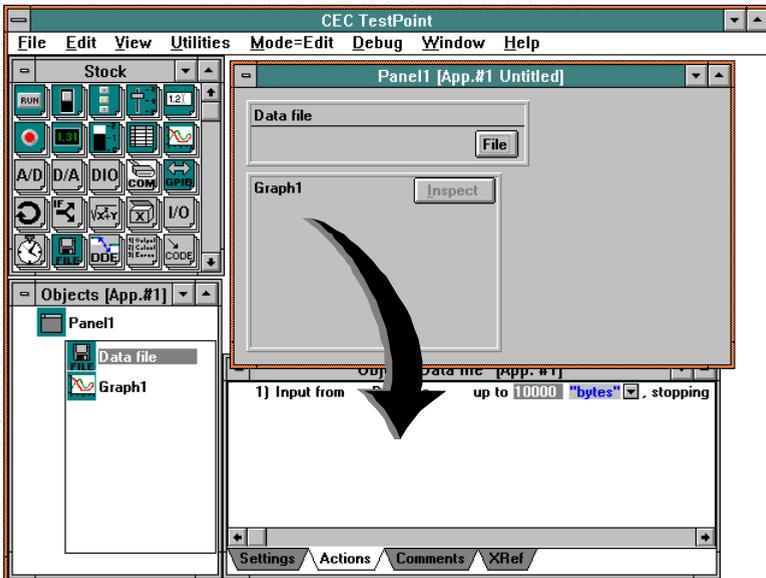


- + Konstanten werden eingegeben, indem Sie auf den gewünschten Parameter klicken und sie über die Tastatur eingeben.

## ***Daten grafisch darstellen***

Ziehen Sie das 'Graph Object' vom 'Objects Window' in die 'Action List'.

Wählen Sie die "Draw graph"-Aktion aus.

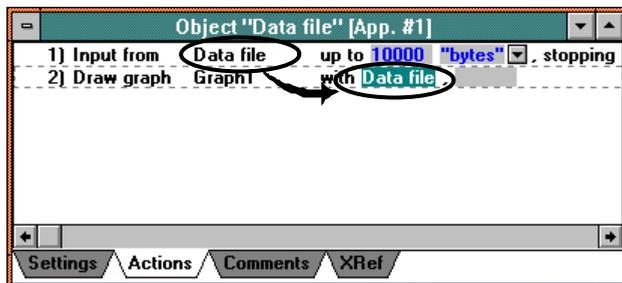


Beachten Sie, daß Sie das Objekt vom 'Objects Window' oder vom Panel in die Action List ziehen können - das Resultat ist das Gleiche.

Die neue Zeile beinhaltet ein leeres Parameterfeld für die Daten, die dargestellt werden sollen. In diesem Beispiel wollen wir die Daten darstellen, die wir gerade aus dem File (Datei) gelesen haben. Diese Daten sind nun dem 'File'-Objekt als Wert zugeordnet. Deshalb ziehen wir nun einfach das 'File'-Objekt in das leere, graue Parameterfeld in Zeile 2.

- † **Datenwerte von Objekten verändern sich als Ergebnis ausgeführter Aktionen.**

Ziehen Sie den Namen des 'File'-Objektes von Zeile 1 in das leere Feld in Zeile 2.



Ein zweites leeres Feld erscheint, denn das 'Graph'-Objekt kann mehrere Datensätze gleichzeitig darstellen. In diesem Beispiel stellen wir aber nur die Daten aus dem File dar.

Beachten Sie, daß Sie sowohl das Symbol des 'File'-Objektes aus dem 'Objects Window' als auch den Namen des Objektes aus einer anderen Action-Line in das Parameterfeld ziehen können.

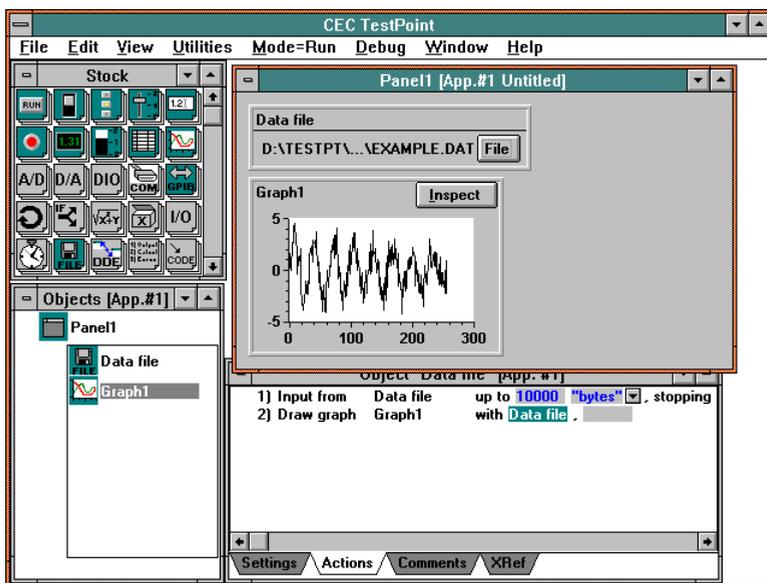
- + **Objekte können sich wie Variablen verhalten - sie beinhalten Datenwerte. Sie können Datenwerte von Objekten verwenden, indem Sie die entsprechenden Objekte in die Parameterfelder ziehen.**

## Starten Sie das Programm

Verwenden Sie das 'Mode'-Menükommando, um in den 'RUN'-Modus zu wechseln.

Im 'Run'-Modus ändert sich das Verhalten des Panels. Im 'Edit'-Modus können Sie mit der Maus z. B. die Position und die Größe eines Objektes verändern. Im 'Run'-Modus dagegen verhält sich das Panel wie in der endgültigen Anwendung, d. h. die Maus aktiviert das Objekt mit einem Ereignis (Event).

Drücken Sie den "File"-Pushbutton im "Data file"-Objekt. Wählen Sie die "example.dat"-Datei (file) im 'Tutorial'-Unterverzeichnis aus.



Verwenden Sie das 'Mode'-Menükommando, um in den 'Edit'-Modus zurückzukehren.

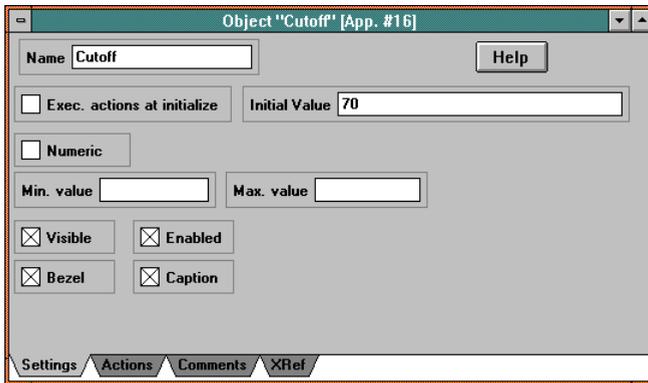
Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *ANALYZE1.TST*.

## Filtern der Daten

Die Beispieldaten aus dem verwendeten File sind Daten, wie sie in einer Anwendung typischerweise anfallen können. Sie sind mit Rauschen überlagert, das die eigentliche Information verfälscht. TestPoint beinhaltet digitale Filterfunktionen für derartige Situationen.

## Hinzufügen eines Dateneingabefeldes

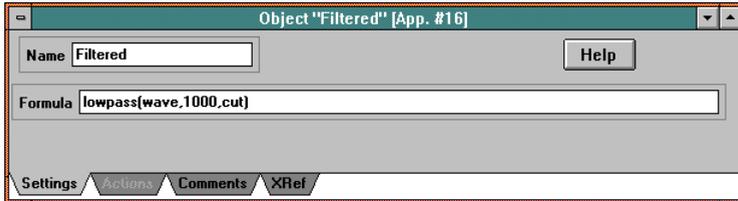
Ziehen Sie ein 'Data-entry'-Objekt  vom Stock ins Panel. Benennen Sie es 'Cutoff' und setzen Sie den 'Initial value' auf 70.



## Hinzufügen eines 'Math Objects' (Mathematik Objektes)

Ziehen Sie ein 'Math Object'  vom Stock in das 'Object Window'.

Wählen Sie als Namen "Filtered",  
und geben Sie als seine Formel "lowpass(wave,1000,35)" ein.



Die 'Lowpass' Filterfunktion ist eine von vielen mathematischen Funktionen, die innerhalb von TestPoint für die Datenanalyse zur Verfügung stehen. Als Parameter stehen das zu filternde Signal, die Datenerfassungsgeschwindigkeit und die Eckfrequenz ('cutoff frequency') zur Verfügung.

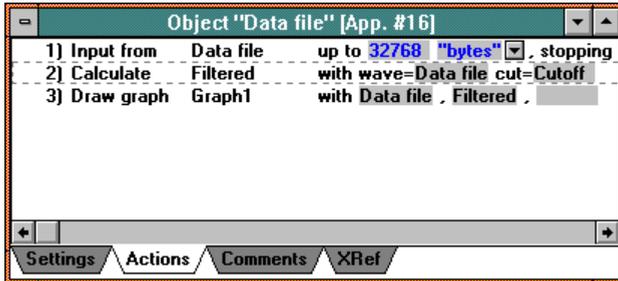
## Modifizieren der Action List

Ziehen Sie das 'Math'-Objekt in die Action List zwischen Zeile 1 und 2.

Wählen Sie die "Calculate"-Aktion aus.

Ziehen Sie das 'File'-Objekt in das "wave"-Parameterfeld (Data File).

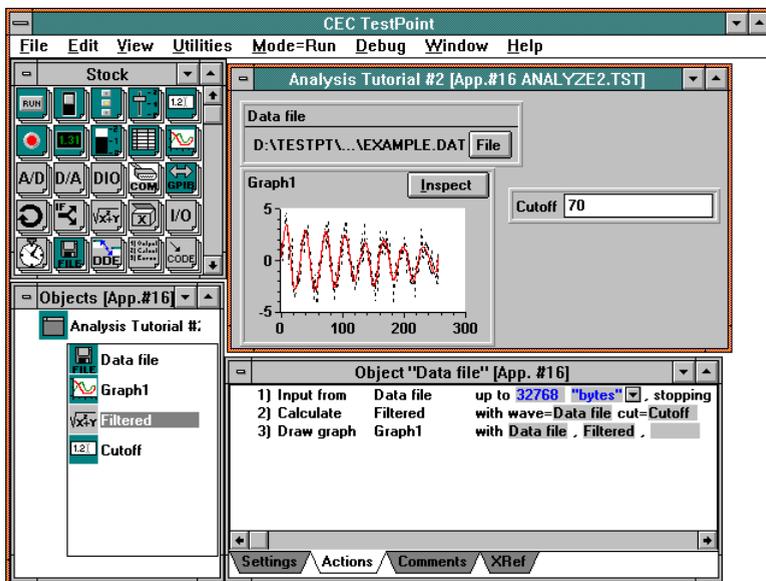
Ziehen Sie das 'Math'-Objekt in das zweite (leere) Parameterfeld (Filtered) in der "Draw graph" Action-Line.



✦ **Objekte können an jeden beliebigen Ort einer Action List gezogen werden, um dort eine neue 'Action Line' zu erzeugen.**

In Zeile 3 der Action List werden nun für die Grafik zwei Datensätze angegeben. Diese werden in der Grafik als zwei Linien dargestellt.

Starten Sie das Programm erneut und wählen Sie wieder die Datei "example.dat".



Wenn Sie nur die gefilterten Daten sehen wollen, klicken Sie auf den "Data file" Parameter in Zeile 3 und löschen diesen mit der 'Entf' ('Del') -Taste und wählen dann erneut die Datei.

Gehen Sie zurück in den 'Edit'-Modus.

Dieses Beispiel ist ebenfalls für Sie vorbereitet. Sie finden es im Unterverzeichnis *TUTORIAL* unter dem Namen *ANALYZE2.TST*.

## ***Bestimmen der Frequenz***

Die Daten zeigen eine Sinuskurve, deren Amplitude mit der Zeit abnimmt. Um ihre Frequenz zu bestimmen, können wir eine Fourier Transformation anwenden, um den höchsten Punkt des Frequenzspektrums zu bestimmen.

**Hinzufügen eines weiteren 'Math'-Objektes:**

**Ziehen Sie ein weiteres 'Math'-Objekt vom Stock in das 'Objects Window'.**

**Nennen Sie es "Freq".**

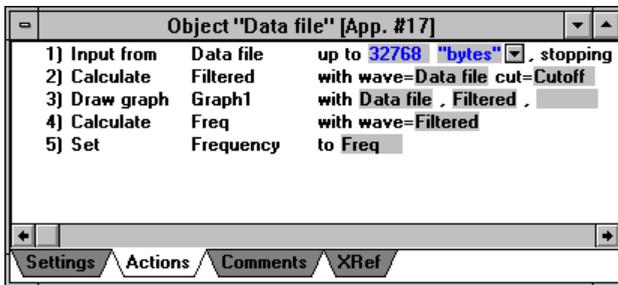
**Geben Sie folgende Formel ein:**

**' $\text{maxindex}(\text{select}(\text{FFT}(1000,\text{wave}),1)) * 1000/\text{dim}(\text{wave})$ '**

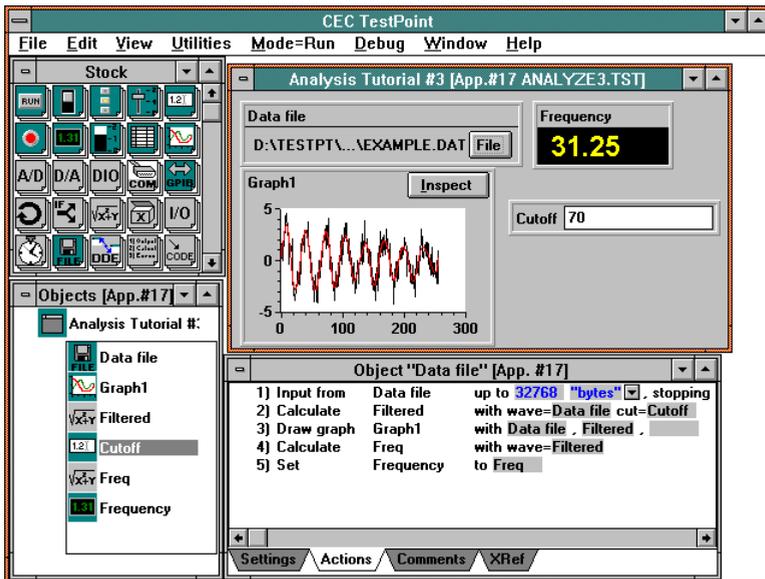
Die **FFT** Funktion führt eine 'Fast Fourier Transformation' mit gegebenem Signal und gegebener Erfassungsrate durch. Das Ergebnis ist eine 'Liste', bestehend aus drei Datenfeldern (Vektoren): Die Frequenzkomponenten, deren Amplituden und deren Phasen. Die '**select**'-Funktion selektiert die Amplituden aus dieser Liste, und die '**maxindex**'-Funktion sucht den Index des größten Wertes. Zusammen mit einer Erfassungsrate von 1000 Hz wird auf 1000 Hz skaliert und ergibt die Grundfrequenz ('base frequency') des Signals.

## Hinzufügen eines Displays

- Ziehen Sie ein 'Display Object'  vom Stock in das Panel.  
Nennen Sie es "Frequency".  
Ziehen Sie das 'Freq Math'-Objekt in die Action List.  
Wählen Sie die "Calculate"-Aktion.  
Ziehen Sie das 'Filtered Math'-Objekt in das "wave" -  
Parameterfeld.  
Ziehen Sie das "Frequency"-Display in die Action List.  
Ziehen Sie das 'Freq Math'-Objekt in das Parameterfeld des  
Displays.



Starten Sie das Programm erneut, und wählen Sie wieder den "example.dat"-File aus.



Gehen Sie zurück in den 'Edit'-Modus.

Wiederholung

# ***TestPoint-Konzepte***

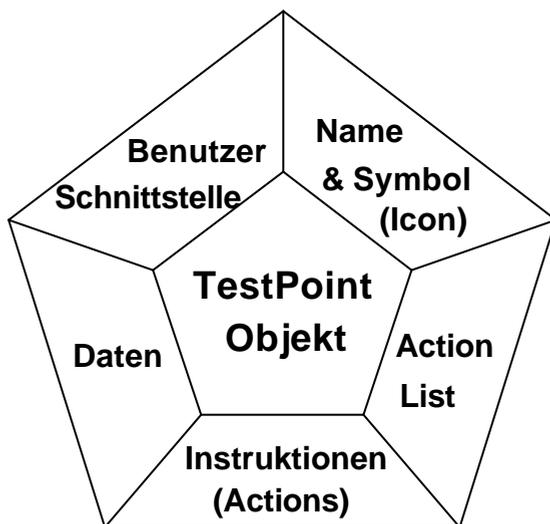
---

# Objekte

---

Ein Objekt in TestPoint repräsentiert eine Komponente Ihrer Anwendung wie zum Beispiel eine Schnittstelle, ein Meßgerät, eine Datei oder eine mathematische Berechnung. Ein Objekt verkörpert Aktionen und Information (Daten) dieser Komponente.

Objekte haben mehrere Charakteristika:



## Actions

Jedes TestPoint-Objekt hat eine oder mehrere Aktionen, die es ausführen kann. Zum Beispiel kann das 'Display Object' eine "set to"-Aktion ausführen, die einen Wert anzeigt. Das 'File Object' kann Files (Dateien) beschreiben, lesen, löschen usw.

## Daten

Jedes TestPoint-Objekt besitzt auch einen Datenwert. Dieser Datenwert kann sich ändern als Resultat der Aktionen dieses Objektes. Zum Beispiel setzt die "read sample"-Aktion der 'analog/digital input'-Objekte den Wert des Objektes entsprechend der eingelesenen Daten. Objekt-Datenwerte können z. B. bei mathematischen Operationen und in Parameter-Feldern von Action-Lines verwendet werden.

## Name & Icon (Symbol)

Jedes Objekt hat einen Namen, den der Benutzer frei wählen kann. In einigen Fällen wird dieser Name auch als 'Caption' (Bezeichnung) für das Objekt im Panel verwendet. Objekte haben auch ein kleines Icon (Symbol), welches die Art des Objektes symbolisiert. Zum Beispiel verwenden alle 'Pushbutton'-Objekte das -Symbol.

## User Interface (Benutzerschnittstelle)

Einige Objektkategorien besitzen auch eine Benutzerschnittstelle. Das 'Pushbutton'-, 'Switch'-, 'Data Entry'-, 'Display'-, 'Indicator'- und 'Graph'-Objekt sind Beispiele für Objekte, die eine solche Benutzerschnittstelle besitzen und im Panel erscheinen. Einige Objekte wie 'Pushbutton' und 'Switch' reagieren auf Benutzereingaben mittels Maus oder Tastatur. Objekte, die Eingaben durch den Benutzer akzeptieren, ändern ihren Datenwert, wenn der Benutzer Eingaben vornimmt. Zum Beispiel nimmt das 'Data Entry'-Objekt einen neuen Wert an, wenn der Benutzer einen solchen eingibt und anschließend die Eingabe- (Enter) oder TAB-Taste betätigt.

## **Action List**

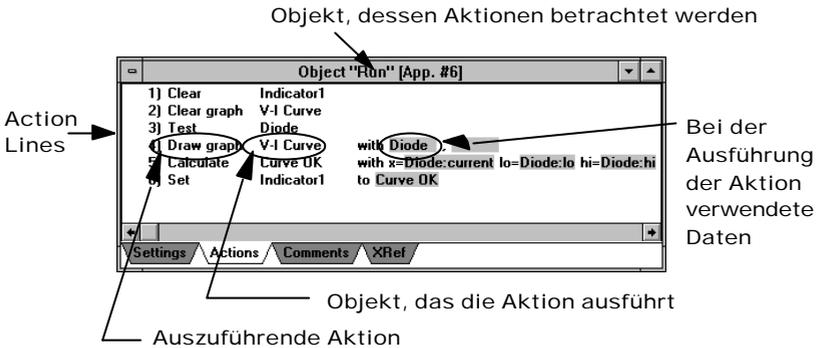
Vielen Objektkategorien ist eine 'Action List' zugeordnet. Die 'Action List' spezifiziert eine Folge von Aktionen, die ausgeführt wird, wenn das entsprechende Objekt einen Event (ein Ereignis) erhält. Zum Beispiel erhält das 'Pushbutton'-Objekt ein Ereignis, wenn dieses mit der Maus angeklickt oder durch die Tastatur aktiviert wird. Das 'File'-Objekt erhält ein Ereignis, wenn der Benutzer einen neuen Filenamen auswählt. 'Analog Input'-Objekte können Ereignisse erhalten, wenn die aufgenommenen Daten vorhanden sind.

## **'Child Objects'**

Das 'Panel'-Objekt unterscheidet sich von den anderen Objekten dadurch, daß es andere Objekte als "Child"-Objekte enthalten kann. Jede TestPoint-Anwendung besitzt immer ein Hauptpanel, welches die Hauptbenutzerschnittstelle darstellt. Objekte, die der Anwendung hinzugefügt werden, sind "Child"-Objekte des Hauptpanels. Darüberhinaus kann eine Anwendung mehrere Panels besitzen, um mehrere Fenster in einer Anwendung zu ermöglichen. Diese 'Panel'-Objekte werden auch zu 'Child'-Objekten des Hauptpanels.

# Action Lists

Mittels der 'Action Lists' wird in TestPoint bestimmt, was eine Anwendung ausführen soll (Programmlisting).



Action Lists werden erstellt, indem man die benötigten Objekte auswählt und sie in das 'Action List Window' zieht. TestPoint erzeugt dann eine Textzeile, die die von Ihnen gewählte Aktion beschreibt.

Eine Action List wird als Folge von Aktionen dargestellt. Neue Action Lines können durch Hereinziehen weiterer Objekte erzeugt werden. Action Lines können gelöscht und in der Reihenfolge geändert werden. Action Lists können auch Schleifen und Abfragen (IF/THEN/ELSE, Loop, 'Conditional'-Objekte) enthalten.

## **Daten**

---

Jedem TestPoint-Objekt ist ein Datenwert zugeordnet, welcher sich durch ausgeführte Aktionen ändern kann. Diese Datenwerte können auch als Parameter für andere Aktionen verwendet werden.

Zum Beispiel in der folgenden Action List:

- 1) Enter from        Voltmeter        up to 256 bytes, stop on EOS=LF
- 2) Set                Result            to Voltmeter

Zeile 2 übernimmt den momentanen Wert des Objektes "Voltmeter" in das Objekt "Result". Der Wert des Objektes "Voltmeter" wurde in Zeile 1 durch die "Enter from" Aktion verändert.

Aktionen, die Objektdatenwerte verändern, ersetzen generell den alten mit dem neuen Wert. Zum Beispiel in dieser Action List:

- 1) Enter from        Voltmeter        up to 256 bytes, stop on EOS=LF
- 2) Enter from        Voltmeter        up to 256 bytes, stop on EOS=LF
- 3) Set                Result            to Voltmeter

Das Objekt "Result" übernimmt den Wert des Objektes "Voltmeter" in Zeile 2. Der Wert des Objektes "Voltmeter" in Zeile 1 ist verloren, da er in Zeile 2 durch einen neuen ersetzt wird. Normalerweise folgt einer Aktion, die Daten erzeugt (z. B. von einem Instrument liest), eine Aktion, die die Daten weiterverarbeitet (z. B. auf dem Bildschirm grafisch darstellt).

# Ausführung

---

TestPoint führt Action Lists aus, wenn Ereignisse auftreten. Ereignisse können sein:

- Benutzereingabe (Maus und Tastatur)
- Timer Objekte
- Interrupts von Hardware wie z. B. IEC Bus-Schnittstelle oder A/D-Karten
- Dynamic Data Exchange (DDE) mit anderen Windows Anwendungen

oder, was sehr wichtig ist:

- Durch eine Aktion, die in irgendeiner anderen Action List ausgeführt wurde.

Zum Beispiel wenn ein 'Data Entry'-Objekt mit dem Namen "Amplitude" folgende Action List hat:

- 1) Calculate      Formula      with x=Amplitude
- 2) Set            Result      to Formula

und ein 'Pushbutton'-Objekt mit dem Namen "Preset" diese Action List hat:

- 1) Set            Amplitude      to 2.5

In diesem Fall setzt das Anklicken des "Preset"-Pushbuttons den Wert des 'Data Entry'- Objektes "Amplitude" auf 2.5. Dadurch wurde ein Wert in "Amplitude" eingegeben, und deshalb wird die Action List dieses Objektes nun ausgeführt, so, als ob ein neuer Wert direkt eingegeben worden wäre.

Ereignisse können sogar durch externen Code generiert werden, der unter Verwendung des 'Code'-Objekts in TestPoint eingebunden werden kann.

Jeder Objekt-Typ definiert die Ereignisse, auf die er reagiert, und er kann eine Action List haben, die entsprechend erstellt werden kann.

Einige Objekt-Typen, wie z. B. das 'Numeric Display'-Objekt generieren keine Ereignisse und besitzen auch keine Action List.

# Weitere Beispiele

---

Dieser Abschnitt enthält einige ausgewählte Beispielanwendungen, die mit Ihrem TestPoint System geliefert werden, inklusive Erklärungen, wie diese arbeiten.

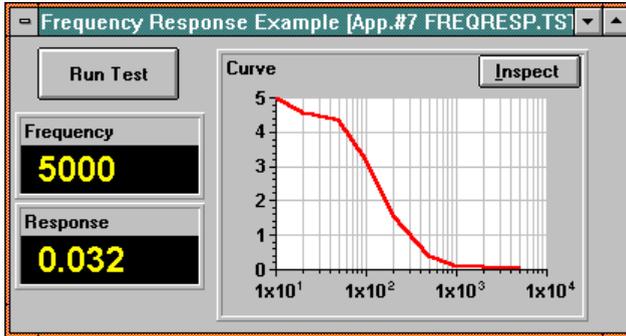
Sie können diese Beispiele in den TestPoint-Editor laden und den nachfolgenden Erklärungen folgen oder sie starten.

Wenn Sie aus dem **Debug**-Menü die Option '**Single Step mode**' auswählen, können Sie die Anwendung schrittweise ablaufen lassen und somit die Ergebnisse der einzelnen Zeilen verfolgen.

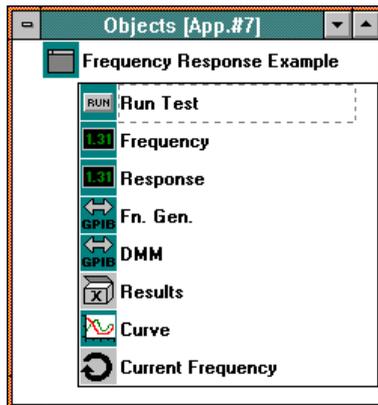
Das View/Data-Menükommando erlaubt es Ihnen, Datentypen und Werte von Objekten zu sehen, während die Anwendung aktiv ist.

## Beispiel Frequenzantwort

Diese Anwendung finden Sie im \TESTPT\EXAMPLES - Unterverzeichnis unter dem Namen **FREQRESP.TST**.



Das Beispiel benutzt einen Funktionsgenerator, um Kurvenformen in verschiedenen Frequenzen an ein Testobjekt zu senden, sowie ein Voltmeter, um die Amplitude an dessen Ausgang einzulesen. Die Kurve mit der Frequenzantwort wird in einer Grafik auf dem Bildschirm in halb-logarithmischer Darstellung angezeigt.



- 1) Clear Results
- 2) Clear Graph Curve
- 3) Decade Series Current Frequency from  $10^1$  to  $10^3$ , in 1,2,5 sequence
- 4) Set Frequency to Current Frequency
- 5) Output to Fn. Gen. with ":FREQ ", Current Frequency, term.=LF, send EOI?=1
- 6) Enter from DMM up to 256 bytes, stop on EOS=LF or EOI
- 7) Set Response to DMM
- 8) Append to Results from Current Frequency, DMM
- 9) End Current Frequency
- 10) Draw Graph Curve with Results

Wenn auf den 'RUN'-Pushbutton geklickt wird, wird die oben dargestellte Action List ausgeführt. Beachten Sie, daß in diesem Beispiel die Anwendung sofort beim Start (bei Mode=Run) ausgeführt wird. Dieses ist der Fall, weil die 'Exec. Actions at initialize' des Pushbutton aktiviert wurde.

Als erstes wird die Anwendung in Zeile 1 und 2 initialisiert. Das Objekt 'Result' ist ein Container, in dem die Frequenzen und Antwortspannungen akkumuliert werden.

Zeile 3 stellt den Anfang der Hauptschleife dar. Es wird das Loop-Objekt 'Current Frequency' mit der Aktion 'Decade Series' verwendet. Diese geht mit den Werten 10, 20, 50, 100, 200, 500, 1000, 2000 und 5000 durch die Anwendung. Die Schleife wird somit neunmal ausgeführt, wobei der Wert des Objektes 'Current Frequency' jeweils einen der oben genannten Werte einnimmt.

Zeile 4 stellt den aktuellen Frequenzwert im 'Display'-Objekt 'Frequency' dar. Damit kann der Anwender den aktuellen Zustand der Anwendung beobachten.

Zeile 5 sendet ein Kommando an den Funktionsgenerator, ein externes GPIB-Gerät. Die 'Output to'-Aktion wird benutzt, um die String-Konstante 'FREQ', gefolgt von dem aktuellen Frequenzwert, zum Gerät zu übertragen.

Zeile 6 liest die Spannungs-Antwort vom DMM (Digital-Multimeter) ein, ebenfalls ein GPIB-Gerät. Die 'Enter from'-Aktion liest Zeichen vom Gerät ein und wandelt diese dann automatisch in Zahlen um. Nachdem Zeile 6 ausgeführt wurde, ist der Datenwert des 'DMM'-Objektes der eingelesene Spannungswert.

Zeile 7 stellt den gemessenen Spannungswert im 'Display'-Objekt 'Response' dar.

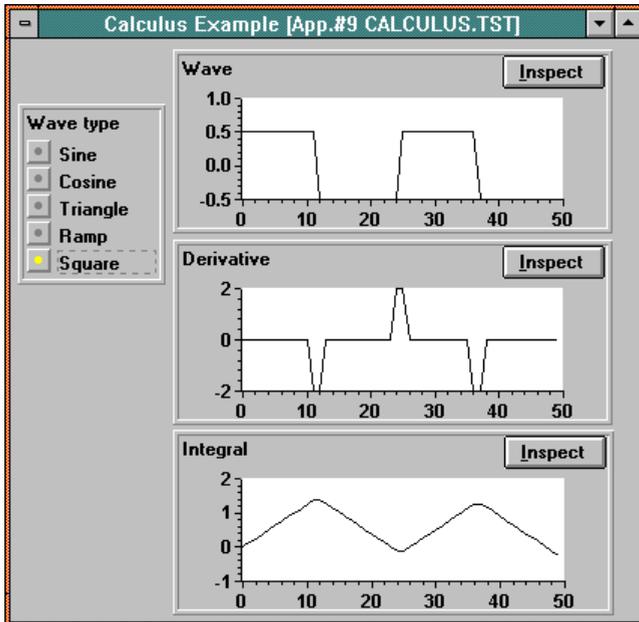
Zeile 8 akkumuliert das Ergebnis der Messung durch Anfügen des aktuellen Frequenz- und Spannungswertspeichers an den 'Container' 'Result'.

Nachdem die Schleife beendet ist, wird in Zeile 10 das Diagramm gezeichnet. Dazu werden die Werte aus dem 'Result'-Objekt benutzt. Das 'Graph'-Objekt ist auf halb-logarithmische Darstellung mit Gitter und veränderter Farbgebung der Linie vorkonfiguriert worden.

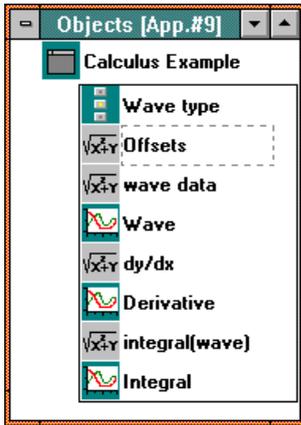
Beachten Sie, daß in diesem Beispiel der 'Demo mode' für GPIB-Objekte benutzt wurde. Dadurch benötigen Sie keine Geräte, sondern können stattdessen die Daten aus der Datei DEMO.DAT verwenden.

## Beispiel Analyse

Diese Anwendung finden Sie im \TESTPT\EXAMPLES - Unterverzeichnis unter dem Namen CALCULUS.TST.



Dieses Beispiel erzeugt eine wählbare Kurvenform und berechnet dann die Ableitung und das Integral. Alle drei Funktionen werden angezeigt.



### Formeln für das 'Math' Objekt:

Offsets: vector (0, 0, -0.499, -0.499, -0.499)

wave data: generate (50,25,fn) + offset[fn]

dy/dx: derivative2 (wave, pi()/12.5)

integral(wave): integrate (wave, pi()/12.5)

### Action List für das "Wave type" "Selector" Objekt:

- 1) Calculate Offsets
- 2) Calculate wave data with fn=Wave type  
offset=Offsets
- 3) Draw graph Wave with wave data
- 4) Calculate dy/dx with wave=wave data
- 5) Draw graph Derivative with dy/dx
- 6) Calculate integral(wave) with wave=wave data
- 7) Draw graph Integral with integral(wave)

Immer dann, wenn eine neue Auswahl aus dem ‘Selector’-Objekt ‘Wave type’ getroffen wird, wird dessen Action List ausgeführt.

Zeile 2 erzeugt die Kurvenform unter Benutzung der mathematischen TestPoint-Funktion ‘**generate()**’, die als Argument die Anzahl der zu erzeugenden Punkte, die Periodenzahl und den Kurventyp benötigt. Das Kurventype-Argument **fn** wird mit dem Wert des ‘Selector’- Objektes gleichgesetzt. Es wird, abhängig von der gewählten Funktion, ein Offset addiert, so daß die Funktion immer um den Nullpunkt zentriert ist. Beachten Sie, daß der Offset aus dem Index eines konstanten Vektors in Zeile 1 erzeugt wird.

Zeile 3 zeichnet das Kurvenform Diagramm.

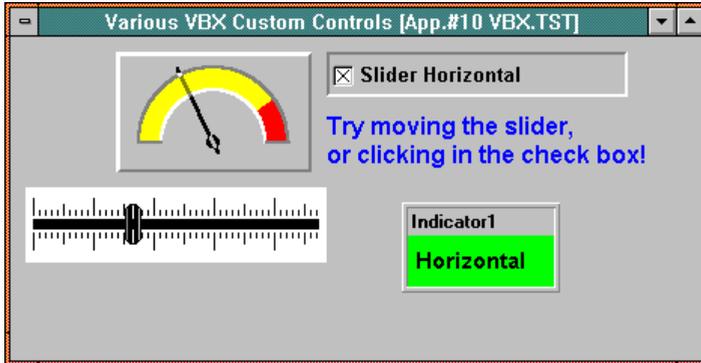
Zeile 4 und 5 berechnen die Ableitung und zeichnen das entsprechende Diagramm. Benutzt wird die mathematische TestPoint-Funktion ‘**derivative2()**’ mit den Argumenten Delta x und der zu verwendenden Kurvenform. TestPoint beinhaltet zwei Ableitungs-Funktionen: ‘**derivative()**’ und ‘**derivative2()**’. Die ‘**derivative2()**’- Funktion ist wesentlich schneller in der Ausführung, weil sie eine Anpassung zweiter Ordnung von jeweils drei Datenwerten vornimmt, während die ‘**derivative()**’-Funktion eine Anpassung vierter Ordnung mit jeweils fünf Datenpunkten vornimmt.

Zeile 6 und 7 berechnen das Integral und zeichnen das Diagramm der Kurvenform.

## ***Beispiel Custom Control (VBX)***

---

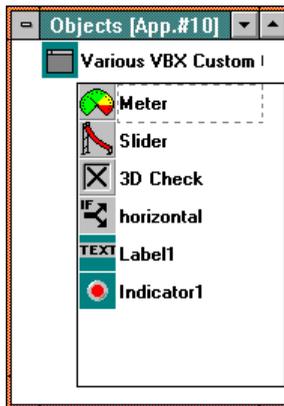
Diese Anwendung finden Sie im \TESTPT\EXAMPLES - Unterverzeichnis unter dem Namen **VBX.TST**.



TestPoint kann VBX (Visual Basic Extensions) Custom Controls benutzen, um Ihrem TestPoint-System neue Funktionen und neue Benutzerschnittstellen-Objekte hinzuzufügen. Hunderte dieser VBX Custom Controls werden von vielen Herstellern angeboten. VBX Custom Controls wurden ursprünglich benutzt, um in Visual Basic neue Funktionen zur Verfügung zu stellen, die unabhängig von der Sprache selbst sind. Es ist dadurch ein sehr aktiver Markt um diese Produkte entstanden. Dadurch sind sehr viele verschiedene Produkte zu sehr attraktiven Preisen erhältlich. Einige Pakete beinhalten 50 oder mehr Custom Controls und kosten dennoch unter DM 400,- (zur Zeit).

TestPoint arbeitet mit diesen Controls zusammen. Damit gibt es kein Limit an Funktionen mehr, denn diese können sehr einfach und preiswert dem System angefügt werden.

**Beachten Sie bitte, daß Sie dieses Beispiel außerhalb des TestPoint Editors (Doppelklick auf das Symbol im Programmmanager) starten müssen. Wenn Sie es in den Editor laden, werden Sie die Warnung erhalten, daß Sie keine Lizenz haben, die verwendeten Custom Controls zu benutzen. Wenn Sie sie in der Editorumgebung benutzen wollen, müssen Sie eine Lizenz des Herstellers der Custom Controls erwerben.**



Die ersten drei Objekte in dieser Liste sind alle VBX Custom Controls. Jedes hat ein eigenes Symbol, das vom Hersteller eingebunden wurde. In diesem speziellen Beispiel haben wir Controls von verschiedenen Herstellern verwendet. Die Analoganzeige stammt aus einem Paket mit Namen 'Gadget for Windows' von der Firma Universal Dynamics in San Diego, Californien. Der Schieberegler (Slider) und die 3D-Checkbox sind aus dem Paket 'VBTools' von der Firma MicroHelp in Marietta, Georgia.

Diese VBX Custom Controls arbeiten wie jedes andere TestPoint Objekt. Sie können Action Listen haben, wie diese hier des 'Slider' Objektes:

- 1) Set     Meter(CurrentValue)     to Slider

Das setzt den 'Current Value'-Wert des 'Meter'-Objektes (Analoganzeige) auf den Wert des 'Slider'-Objektes. Beides, 'Meter' und 'Slider', sind Custom Controls, und die Action List des einen nutzt den Datenwert zur Einstellung eines anderen. Der Effekt ist, daß die Nadel des 'Meter' Objektes sich bewegt, wenn der 'Slider' seinen Wert verändert.

Action Lists wie diese ändern Objekteinstellungen, die durch das Ziehen der entsprechenden Einstellung aus dem 'Settings'-Fenster eines Objektes in die Action List erzeugt werden.

Dies ist die Action List des '3D Checkbox'-Objektes:

- 1) Set Indicator1 to 3D Check
- 2) Set Slider(Style) to 3D Check
- 3) If/Then/Else horizontal with checkbox=3D Check
- 4) Move Slider x=10, y=100, w=200, h=50
- 5) Else if not horizontal
- 6) Move Slider x=10, y=10, w=50, h=200
- 7) End If horizontal

Zeile 1 setzt das 'Indicator'-Objekt auf den Wert der 'Checkbox' und zwar immer dann, wenn dieser sich geändert hat. Der Text des 'Indicators' zeigt je nach der gewählten Einstellung 'Horizontal' oder 'Vertical'.

Zeile 2 setzt den 'Style' des 'Slider'-Objektes. Dieser ändert die Richtung des Schiebereglers.

Die Zeilen 3 bis 7 ändern Lage und Größe des 'Slider'-Objektes auf dem Bildschirm in Abhängigkeit von der gewählten Richtung. Somit paßt sich das 'Slider'-Objekt immer optimal der gewählten Lage an. Die 'Move'-Aktion ist eine Aktion, die zum Panel-Objekt gehört.

# ***Mehr über die TestPoint-Objekte***

---

Sie haben die Möglichkeit, sich eine detaillierte Beschreibung jedes TestPoint-Objektes anzusehen, wenn Sie die Menükommandos Help, Index im TestPoint-Editor auswählen. Wenn Sie die Vollversion (keine Demo-Version) besitzen, können Sie im TestPoint-Handbuch im Abschnitt 'Techniken und Referenz' nachschlagen.

## **TestPoint Objekte**

Action Object	Indicator Object
Analog/Digital (A/D) Object	Indirect objects
Bar Indicator Object	Loop Object
Bitmap Switch Object	Math Object
Case Object	OLE Object
Code Object	Panel Object
Conditional Object	Picture Object
Container Object	Port I/O Object
Data Entry Object	Pushbutton Object
DDE Object	Report Object
Digital/Analog (D/A) Object	RS232 Object
Digital I/O (DIO) Object	Selector Object
Display Object	Slider Object
Error Handler Object	Switch Object
File Object	Task Object
GPIB Object	Text Object
Graph Object	Time Object
Grid Object	User-defined objects
Groups of objects	VBX Object

## **Weitere Informationen/Bestellung**

---

Bitte bestellen Sie TestPoint

in Deutschland bei

Keithley Instruments GmbH  
Landsbergerstr. 65  
82110 Germering  
Telefon: 089-84 93 07-40  
FAX: 089-84 93 07-59

in der Schweiz bei

Keithley Instruments SA  
Kriesbachstrasse 4  
8600 Dübendorf  
Telefon: 01-821-9444  
FAX: 01-820-3081

Informationen über besonders günstige Fünffach-Lizenzen, Ausbildungs- und Hochschulrabatte, Bundles (Paketpreise TestPoint mit unseren Einsteckkarten), oder über unsere attraktiven Sonderkonditionen für OEMs und System-Integratoren erhalten Sie auf Anfrage.

Sehr gerne informieren wir Sie telefonisch über TestPoint. Dabei erarbeiten wir die für Ihre individuelle Applikation optimale Lösung, ob es sich um Software, Hardware oder ein komplettes System handelt – kostenlos und unverbindlich. Rufen Sie uns dazu an unter der Tel.-Nr.

**0 89 / 84 93 07 - 40.**

Unsere Ingenieure beraten Sie ausführlich und kompetent.

Oder fordern Sie ein formelles und individuelles Angebot an, das wir Ihnen auf Wunsch noch am selben Tag zufaxen.

Selbstverständlich demonstrieren wir Ihnen die Leistungsfähigkeit von TestPoint auch in Ihrem Hause - einer unserer Außendienstingenieure ist immer auch in Ihrer Nähe.

Bitte beachten Sie auch unser breites Spektrum an PC-Einsteckmeßkarten, IEEE-Interfaceboards, GPIB-Meßgeräten und portablen Lösungen!

## *Index*

---

Action list	26; 43; 61; <b>76</b>	GPIB	24
Child object	75	Icon	74
Code Object	78	Indicator	31
Data	74; <b>77</b>	Interrupts	78
DDE		Math	32
triggers	78	Name	74
Demo mode	24; 41	Numeric	25
Display	25	Object	73
Edit mode	30; 46; 65	Popup	27; 44; 62
End-of-string	27	Run mode	30; 46; 65
Enter from	27	User interface	74
EOI	27	Voltmeter	24; 41
EOS	27		
Execution triggers	78		



## ***Notizen***

---



## ***Notizen***

---