

Rapport Referenz

Dieses Dokument beschreibt, wie man für das Präsentationsprogramm Rapports erstellt und bearbeitet, und listet den Kommandovorrat auf. Am Ende folgen einige Beispiel-Rapports mit Kommentaren.



Inhalt

Grundlagen	3
Was sind Rapports	3
Wo finde ich sie	3
Wo sind sie gespeichert	3
Wie erstelle und bearbeite ich sie	3
Rapports ausführen	4
Zusätzliche Ausführungs-Optionen	4
Automatische Parameter-Auswahl	5
Rapports Erstellen	6
Neuen Rapport erzeugen	6
Hinweise zum Aufzeichnen von Befehlen	6
Der Rapport-Editor	8
Textersetzung	9
Parameter	9
Der Aufbau eines Rapport-Befehls	12
Der Befehls-Assistent	13
Die Syntax im Rapport.sea	13
Rapport-Versionierung	15
Befehls-Referenz	16
Daten darstellen („Show“-Befehl)	16
Daten wegwischen	17
Daten nur laden (vorbereiten)	17
Darstellungsoptionen setzen	17
Befehle für eine Layout-Seite	18

Text anzeigen.....	18
Bezeichnungen von Parametern und Werten verwenden.....	19
Befehle für Archive	19
Rechteck- und Plot-Stile ändern.....	20
Plot-Stil ändern.....	20
Rechteck-Stil ändern	21
Balkengrafik-Stil ändern	21
Unter-Rapports aufrufen.....	22
Sprungbefehle	22
Bedingte Ausführung.....	23
Textvergleiche	23
Vergleich mit mehreren Werten	23
Größer-Kleiner-Vergleiche.....	23
Parameter vorhanden	24
Daten vorhanden.....	24
Daten-Werte prüfen.....	24
Bewertungsergebnis prüfen.....	25
Bedingung nicht erfüllt – ‚Else‘-Befehl	25
Parameter-Definition	27
Einfache Variablen.....	27
Numerische Parameter	27
Daten aus Messungen als Parameter.....	27
Bewertungsergebnis als Parameter	29
Globale und Automatische Parameter.....	29
Globale Parameter	29
Vordefinierte und Automatische Parameter.....	29
Weitere Befehle	30
Fehlersuche	30
Kommentare.....	30
Beispiele	31
Basis-Anzeige-Rapport	31
Verwendung von Unter-Rapports	31
Verzweigung und Schleife	31
Spektren und Ordnungsmarkierungen.....	32
Index.....	33

Grundlagen

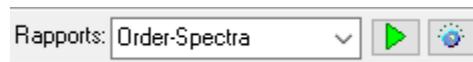
Was sind Rapports

Ein Rapport ist vergleichbar mit einem Makro für Word: es automatisiert eine Abfolge von Bedienschritten, die dann mit einem einzigen Befehl aufgerufen werden kann.

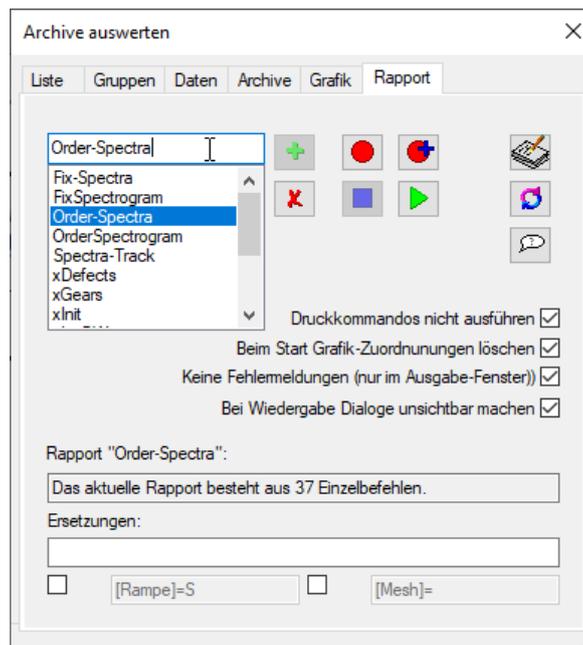
Ursprünglich waren Rapports genau das: Aufzeichnungen von Bedienschritten. Doch mit der Zeit haben sich die Rapports zu einer Art Programmiersprache entwickelt, mit Variablen, Bedingungen, Schleifen und zahlreichen Befehlen, die sich nicht durch Aufzeichnen von Bedienschritten erzeugen lassen.

Wo finde ich sie

Für die Rapports gibt es eine Auswahlliste in der Werkzeugleiste des Präsentationsprogramms:



Man wählt ein Rapport aus der Liste aus und drückt den grünen „Start“-Knopf. Der „Drehknopf“-Button öffnet die Rapport-Abteilung im generellen Bedienfenster:



Auch hier gibt es die Liste der Rapports und einen „Start“-Knopf. Die weiteren Bedienelemente und Funktionen werden später besprochen.

Wo sind sie gespeichert

Rapports sind Teil des Präsentations-Projektes und werden daher im Projekt-Verzeichnis gespeichert. Normalerweise werden sie in der Datei `Rapports.sea` abgelegt. (Es ist möglich, aber unüblich, einen anderen Dateinamen oder sogar mehrere Rapport-Dateien zu verwenden). Die Datei `Rapports.sea` ist trotz ihrer Dateiendung eine normale Text-Datei, die mit einem Text-Editor wie Notepad oder Notepad++ geöffnet und bearbeitet werden kann.

Wie erstelle und bearbeite ich sie

Die einfachste Methode, ein Rapport zu erstellen, ist das Aufzeichnen von Bedienschritten. Meistens möchte man das Ergebnis der Aufzeichnung aber noch bearbeiten, beispielsweise indem man durch die Verwendung von Parametern oder Verzweigungen mehr Flexibilität einführt.

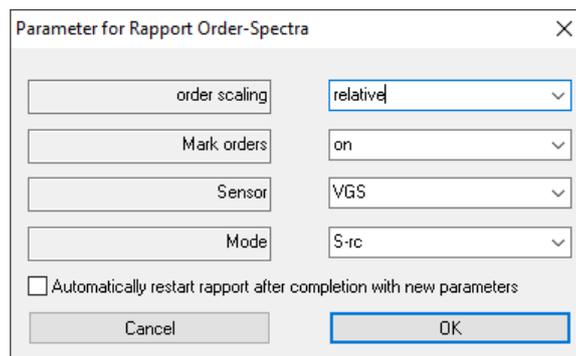
Die Präsentations-App enthält einen Editor für Rapports, mit dem man die vorhandenen Rapports ändern und erweitern kann. Des Weiteren kann man die `Rapports.sea`-Datei mit einem Texteditor öffnen und die Rapports dort bearbeiten. Manche Aufgaben lassen sich schneller und bequemer mit dem Editor in der Präsentations-App erledigen, für andere Aufgaben ist der Texteditor die bessere Wahl. Beide Methoden werden im Rahmen dieser Dokumentation erläutert.

Gleich an dieser Stelle folgender wichtiger *Hinweis*: beim Beenden speichert die Präsentations-App automatisch alle Änderungen, sowohl an Layout-Seiten als auch an Rapports. Wenn Sie das Präsentationsprojekt in der App geöffnet haben, können Sie keine Änderungen per Texteditor im `Rapports.sea` durchführen (Sie können zwar, aber die Änderungen gehen verloren, sobald sie die Präsentations-App schließen). Bevor Sie den Texteditor verwenden, müssen Sie die App beenden und die Datei ggf. neu laden.

Eine ähnliche Gefahr entsteht, wenn Sie die Präsentations-App mehrmals geöffnet haben (mit demselben Projekt): beim Schließen speichert jede dieser App-Instanzen alle Dateien entsprechend des eigenen Zustandes. Am Ende gewinnen also die Änderungen in der Instanz, die zuletzt geschlossen wird. Um die Präsentations-App zu schließen, ohne dass gespeichert wird, wählen Sie aus dem Menü **Datei (File)** den Befehl **Notausgang (Emergency Exit)**.

Rapports ausführen

Laden Sie Messungen, wählen Sie ein Rapport aus der Liste und drücken Sie den Start-Knopf.  Die meisten Rapports werden nach dem Start fragen, welche Werte für die Rapport-Parameter verwendet werden sollen:



Parameter	Value
order scaling	relative
Mark orders	on
Sensor	VGS
Mode	S-rc

Automatically restart rapport after completion with new parameters

Cancel OK

Die Namen, Anzahl und Art der Parameter und ihrer möglichen Werte werden beim Bearbeiten des Rapports angelegt.

Wählen Sie die Werte, die Sie für die aktuelle Ausführung des Rapports haben wollen. Wenn Sie das Häkchen bei **Rapport nach Beendigung mit neuen Parametern wiederholen (Automatically restart rapport after completion with new parameters)** setzen, dann wird das gewählte Rapport nach Ende der Ausführung automatisch neu gestartet.

Zusätzliche Ausführungs-Optionen

Wenn Sie die entsprechende Abteilung im Bedienfenster öffnen (siehe die Abbildung weiter oben; für die englische Version blättern Sie nach unten), dann finden Sie dort einige weitere Optionen für die Ausführung von Rapports.

Druckkommandos nicht ausführen: Es gibt Rapport-Befehle, mit denen die aktuelle Layout-Seite zum Standard-Drucker geschickt werden kann (oder auch alle geöffneten Seiten). Falls der eingestellte Standard-Drucker ein pdf-Drucker ist, kann man auf diesem Wege automatisch einen

Bericht als pdf-Datei speichern. Mit der Druckkommandos-Option kann man diese automatische Druckfunktion vorübergehend abschalten.

Bei Start Grafik-Zuordnungen löschen: Normalerweise werden bei Start eines Reports alle Verknüpfungen zwischen aktuell dargestellten Daten und den zugehörigen Grafik-Modulen gelöscht, damit es nicht zu einer Vermischung von alten Daten und Report-Daten kommt. Manchmal ist aber diese Vermischung ausdrücklich gewünscht. Dann entfernt man dieses Häkchen.

Keine Fehlermeldungen: Sollten bei der Ausführung eines Reports Fehler auftreten (z.B. weil Daten oder Grafik-Module nicht gefunden wurden), dann erscheinen normalerweise entsprechende Hinweisfenster. Mit dieser Option können die Hinweise auf Einträge im Ausgabefenster beschränkt werden.

Das Ausgabefenster öffnen Sie mit dem Werkzeugleisten-Knopf gleich links vom Fragezeichen:



Bei Wiedergabe Dialoge unsichtbar machen: Diese Option möchten Sie vermutlich immer aktiviert lassen. Dadurch werden das Bedienfenster und das Grafikmodul-Zuordnen-Fenster während der Report-Ausführung ausgeblendet. Wenn Sie wissen möchten, warum, dann machen Sie einfach mal das Häkchen aus.

Zusätzliche Ersetzungen: im unteren Bereich der Abteilung Report im Bedienfenster können sie Parameter-Ersetzungsregeln eingeben (und zwei davon durch Häkchen ein- und ausschalten), die parallel zu den normalen Parametern im Report wirksam werden. Die Syntax für diese Ersetzungen entspricht der im Report-Editor und wird dort beschrieben. In den meisten Fällen werden die zusätzlichen Ersetzungen aus dem Bedienfenster nur zu Testzwecken verwendet – oder gar nicht.

Ablauf protokollieren: Wenn Sie den Knopf  drücken (er verändert dann etwas sein Aussehen zu ) , werden im Ausgabefenster alle Report-Befehle protokolliert. Diese Funktion ist hilfreich, um Fehler in Reports zu finden oder die Frage zu beantworten, warum etwas nicht funktioniert.)

Automatische Parameter-Auswahl

Wenn ein Report mit Parametern gestartet wird, dann versucht die Präsentations-App, die Auswahllisten der Parameter-Werte auf diejenigen Werte einzuschränken, die aufgrund der aktuell geladenen Daten möglich sind.

Wenn beispielsweise einer der Parameter für den Prüfzustand (Mode) verwendet wird, dann reduziert die Präsentations-App beim Start eines Reports die Auswahl an möglichen Werten auf diejenigen Prüfzustände, die in den aktuell geladenen Daten vorkommen. Wenn es in den aktuell geladenen Daten nur einen einzigen Prüfzustand gibt, dann wird für den Parameter automatisch dieser Wert genommen und gar keine Auswahl mehr angezeigt.

Falls es für einen Parameter gar keinen möglichen Wert gibt, der mit den aktuell geladenen Daten kompatibel ist, dann wird eine entsprechende Meldung angezeigt und das Report gar nicht ausgeführt.

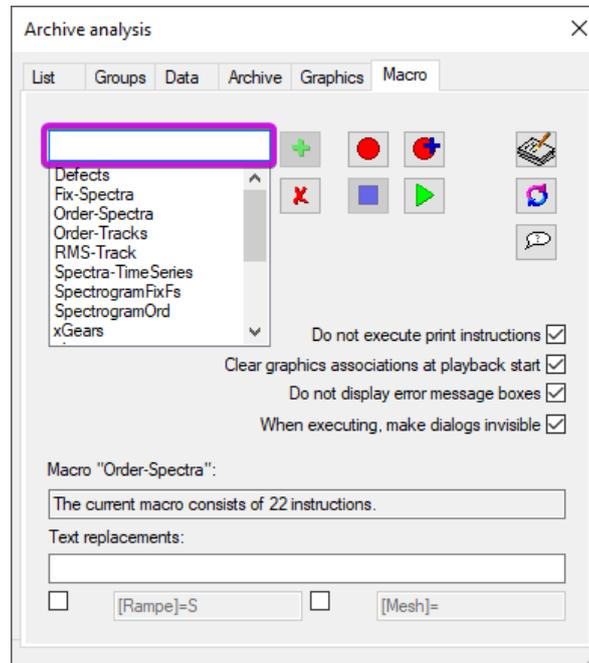
Die automatische Reduktion der Werteliste funktioniert nur für Parameter, deren mögliche Werte explizit im Report aufgelistet sind und bei denen man über die geladenen Daten Rückschlüsse auf die verfügbaren Werte ziehen kann.

Die automatische Parameter-Auswahl kann nicht deaktiviert werden.

Rapports Erstellen

Neuen Rapport erzeugen

Öffnen Sie die Abteilung **Rapport** des Bedienfensters. Geben Sie in das Eingabefeld oberhalb der Liste der vorhandenen Rapports einen neuen Namen ein:



Drücken Sie dann den Knopf , um ein Rapport dieses Namens anzulegen. Dieses Rapport ist noch leer, erscheint aber bereits in der Liste und ist dort automatisch ausgewählt. Drücken Sie nun den „Aufnahme“-Knopf .

Ab jetzt werden alle Befehle zum Darstellen von Daten, die Sie nun ausführen, im Rapport aufgezeichnet, bis Sie den „Stop“-Knopf  drücken. Mehr zur Aufzeichnung von Befehlen folgt im nächsten Abschnitt.

Sie können ein vorhandenes Rapport um weitere Befehle verlängern, indem Sie das Rapport in der Liste anwählen und dann den „Aufnahme fortsetzen“-Knopf  drücken.

Ein Rapport wird gelöscht, indem Sie es in der Liste anwählen und dann den „Löschen“-Knopf  drücken. (Wenn das aus Versehen passiert, können Sie sich z.B. dadurch retten, dass Sie den Notausgang nehmen, oder dass Sie eine Kopie der `Rapport.sea`-Datei erstellen, bevor Sie die Präsentations-App schließen bzw. speichern.)

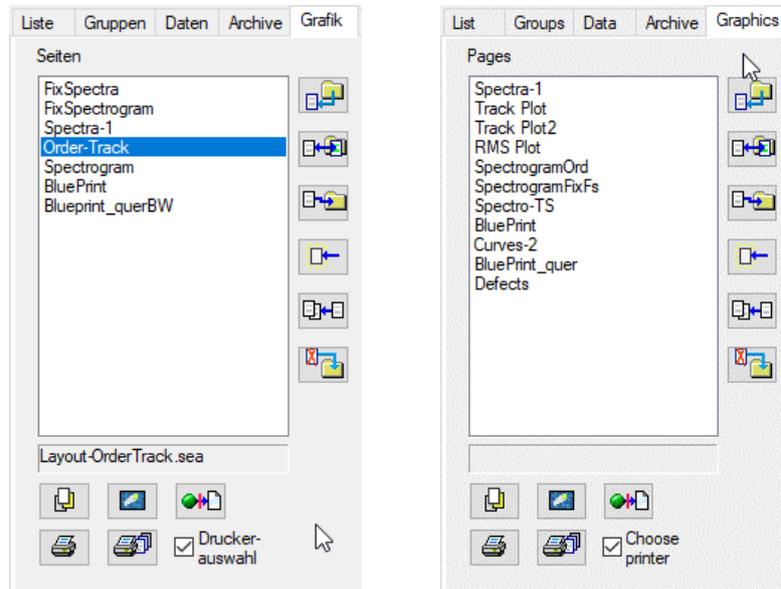
Hinweise zum Aufzeichnen von Befehlen

Nicht jede Aktion, die Sie während der Aufzeichnung eines Rapports ausführen, wird im Rapport gespeichert.

Nicht gespeichert werden beispielsweise alle Aktivitäten zum Laden (oder Ent-Laden) von Messungen, Markieren oder Gruppieren von Messungen, sowie alle Aktivitäten innerhalb von Layout-Seite (Skalierung ändern, Farben ändern usw.).

Da das Aufzeichnen der Bedienschritte meist nur dazu dient, das Grundgerüst eines Rapports zu erschaffen, das dann mit dem Editor ausgebaut und verfeinert wird, sollte man sich auf diejenigen Schritte konzentrieren, die am einfachsten durch Aufzeichnen zu erledigen sind.

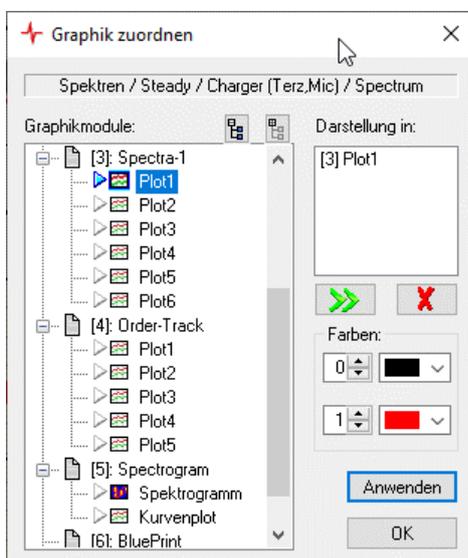
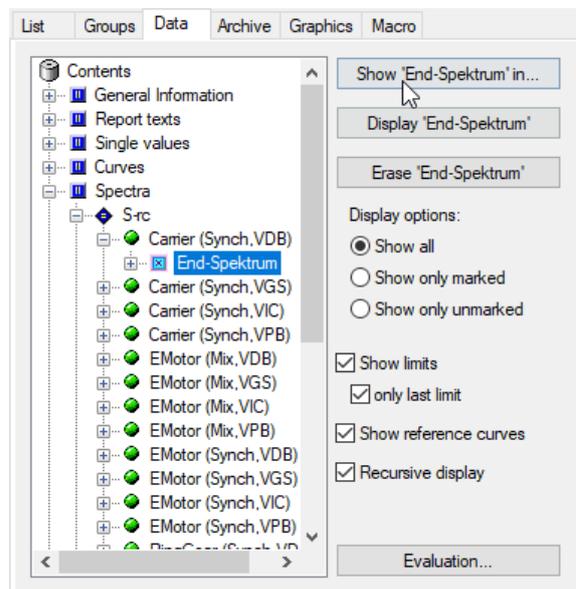
Beginnen Sie in der Abteilung **Grafik**, wählen Sie in der Liste die Layout-Seite, auf der Sie etwas darstellen möchten, und drücken Sie die Knöpfe für „Tafel wischen“  und „Seite in den Vordergrund“  :



Außerdem können Sie hier in der Abteilung **Grafik** die beiden Befehle „Aktuelle Seite drucken“  und „Alle geöffneten Seiten drucken“  in Reports aufzeichnen.

Wechseln Sie nun in die Abteilung **Daten** und führen Sie die folgenden Schritte in genau dieser Reihenfolge aus:

1. Wählen und aktivieren Sie im Inhaltsbaum einen Datenknoten, dessen Daten (z.B. ein bestimmtes Spektrum) Sie anzeigen möchten.
2. Drücken Sie den Knopf oben rechts („Darstellen in...“ bzw. „Show in...“).
3. Verknüpfen Sie ein Grafik-Modul mit diesem Daten-Knoten. Wählen Sie dazu das Grafik-Modul aus (siehe Abbildung),



drücken Sie den **[>>]**-Knopf

oder doppelklicken Sie auf das Grafik-Modul, und drücken Sie dann *entweder* **[Anwenden]** oder **[OK]**.

4. Drücken Sie in der Abteilung **Daten** des Bedienfensters den zweiten Knopf „...anzeigen“ bzw. „Display...“.

Führen Sie diese Schritte für alle Daten aus, deren Anzeige Sie aufzeichnen möchten – häufig brauchen Sie nur einen einzigen Datenknoten darzustellen, da Sie die Darstellung weiterer Daten sehr viel bequemer im Editor durch

Duplizieren des aufgezeichneten Befehls realisieren können.

Es ist wichtig, dass Sie in Schritt 3 die Grafik-Zuordnung nur durch entweder [Anwenden] oder [OK] bestätigen, und dass Sie während der vier Schritte keine anderen Bedienschritte ausführen, die in Rapports aufgezeichnet werden. (Erlaubt ist daher z.B. das Ändern von Plot-Skalierungen, weil es nicht aufgezeichnet wird.)

Die Präsentations-App fasst nämlich diese vier Bedien-Schritte nach der Aufzeichnung zu einem einzigen Befehl zusammen, dem „Show“-Befehl. Im Rapport-Editor sehen Sie später statt der einzelnen Schritte den „Show“-Befehl, der sehr viel einfacher zu bearbeiten und zu duplizieren ist.

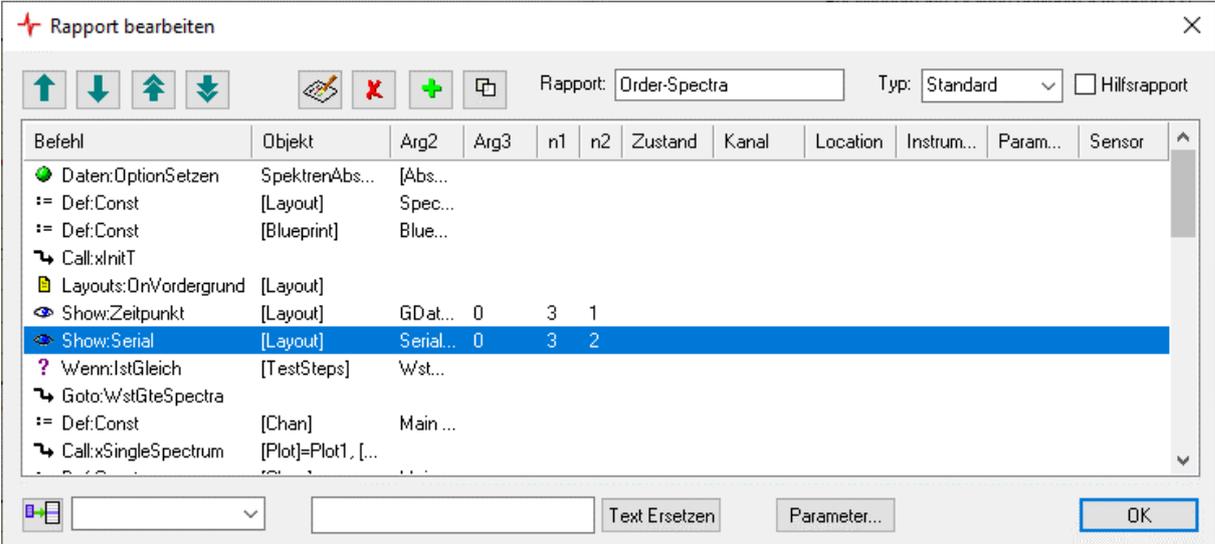
Falls es schiefgegangen ist und Sie später im Editor doch einzelne Befehle sehen: Löschen Sie einfach die fraglichen Befehle und benutzen Sie die „Aufnahme fortsetzen“-Funktion , um den Show-Befehl neu zu erstellen. Er erscheint dann zwar zunächst am Ende Ihres Rapports, aber Sie können ihn mit Hilfe des Editors leicht an die richtige Position verschieben.

Am Ende der Rapport-Aufzeichnung empfiehlt es sich, noch einmal den „Layout in den Vordergrund“-Befehl  aufzunehmen, da dieser Befehl zugleich eine Aktualisierung aller dargestellten Daten auslöst. Drücken Sie dann den „Stop“-Knopf , und Ihr Rapport-Grundgerüst ist fertig.

Natürlich können Sie dieses soeben aufgezeichnete Rapport auch direkt ausführen. Vielleicht ist das ja schon genau das, was Sie brauchen. In den meisten Anwendungsfällen werden Sie aber als nächstes den Editor öffnen und Ihr Rapport bearbeiten.

Der Rapport-Editor

Wechseln Sie im Bedienfenster in die Abteilung Rapport, wählen Sie in der Liste ein Rapport an und drücken Sie den Editor-Knopf , um dieses Rapport zu bearbeiten:



Befehl	Objekt	Arg2	Arg3	n1	n2	Zustand	Kanal	Location	Instrum...	Param...	Sensor
Daten:OptionSetzen	SpektrenAbs...	[Abs...									
:= Def:Const	[Layout]	Spec...									
:= Def:Const	[Blueprint]	Blue...									
Call:xInitT											
Layouts:OnVordergrund	[Layout]										
Show:Zeitpunkt	[Layout]	GDat...		0	3	1					
Show:Serial	[Layout]	Serial...		0	3	2					
? Wenn:IstGleich	[TestSteps]	Wst...									
Goto:WstGteSpectra											
:= Def:Const	[Chan]	Main...									
Call:xSingleSpectrum	[Plot]=Plot1, [...]										

Auf die Befehle, die im Hauptteil des Editor-Fensters aufgelistet werden, wird in der Rapport-Befehlsreferenz genauer eingegangen. An dieser Stelle sollen die Bedienelemente des Editors erläutert werden.

Mit den Knöpfen oben links     können Sie den angewählten Befehl um eine Zeile nach oben oder unten verschieben, oder ganz an den Anfang bzw. das Ende des Rapports setzen.

Die darauffolgenden vier Knöpfe     haben folgende Funktionen:

- Ausgewählten Befehl bearbeiten (kann auch durch Doppelklick auf einen Befehl aufgerufen werden).
- Ausgewählten Befehl löschen.
- Neuen Befehl unterhalb des ausgewählten Befehls einfügen.
- Ausgewählten Befehl duplizieren.

Mit den weiteren Bedienelementen oberhalb der Befehlsliste können Sie den Namen des Rapports ändern, seinen Typ festlegen (bleiben Sie erst einmal bei „Standard“) und es zum Hilfsrapport machen. Hilfsrapports erscheinen nicht in der Auswahlliste der Werkzeugleiste der Präsentations-App (also hier:   )

Mit dem Knopf  unten links können Sie den Text eines anderen Rapports unterhalb des ausgewählten Befehls einfügen (das andere Rapport suchen Sie in der Liste rechts vom Knopf aus).

Textersetzung

Das Ersetzungsfeld im Editor-Fenster dient vor allem dazu, in aufgezeichneten Befehlen die festen Werte durch Parameter zu ersetzen. Beispielsweise enthält ein durch Aufzeichnung entstandener Show-Befehl den Namen der Layout-Seite und des Prüfzustandes; für das fertige Rapport möchte man hier aber lieber die Parameter [Layout] und [Mode] stehen haben. (Mehr über Parameter folgt später; jetzt erst einmal nur Textersetzung).

Die Textersetzung im Editor-Fenster funktioniert ganz einfach: man gibt den zu ersetzenden Text ein, dann ein Gleichheitszeichen, dann den neuen Text, und drückt dann [Text Ersetzen]:

Die Ersetzung wird im gesamten Rapport durchgeführt. (Es gibt keine „Undo“-Funktion.)

Es gibt drei Zeichen, die besondere Beachtung verdienen: Leerzeichen, das Gleichheitszeichen, und das Komma.

Leerzeichen sind generell erlaubt, sowohl links als auch rechts des Gleichheitszeichens, aber sie werden auch exakt verarbeitet. Deshalb darf man um das Gleichheitszeichen herum keine Leerzeichen setzen. Würde das obige Beispiel so aussehen: „Page-Spectra = [Layout]“, dann würde der Editor nach dem Auftreten von „Page-Spectra “ (mit zusätzlichem Leerzeichen hinten) suchen, und diese durch „ [Layout]“ mit zusätzlichem Leerzeichen vorne) ersetzen.

Das erste gefundene *Gleichheitszeichen* trennt den Suchtext vom Ersetzungstext. Daher kann man auf diesem Wege keine Texte ersetzen, die Gleichheitszeichen enthalten. Glücklicherweise treten die auch praktisch nie auf.

Das *Komma* wird später in Rapport-Befehlen verwendet, um mehrere Parameter-Ersetzungen aufzulisten und zu trennen. Man kann Kommata also ohne Weiteres als Suchtext verwenden, muss sich beim Ersetzungstext aber sehr genau überlegen, was hinterher passieren soll.

Parameter

Parameter funktionieren ähnlich wie Variablen in regulären Programmiersprachen. Im Text des Rapport-Befehls sind sie Platzhalter für einen Wert, der erst ausgewählt wird, wenn das Rapport ausgeführt wird.

Wird beispielsweise in den Befehlen eines Rapports der Ausdruck [Mode] verwendet anstelle des Namens eines bestimmten Prüfzustandes, so kann man dieses Rapport für jeden beliebigen Prüfzustand verwenden (statt nur für den einen, für den man es aufgezeichnet hat).

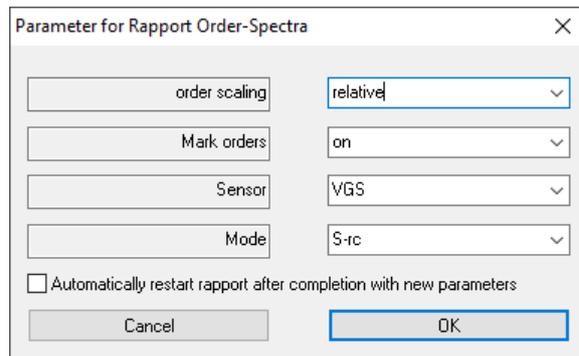
Verwendet man im Rapport den Parameter [Layout] anstelle des Namens einer bestimmten Layout-Seite, so kann man dieses Rapport zum Füllen mehrerer Seiten benutzen (die gleich aufgebaut sind, z.B. eine Seite pro Sensor).

Name und Beschreibung

Ein Parameter besitzt einen *Namen* (z.B. „Mode“ oder „Layout“), eine *Beschreibung* (z.B. „Prüfzustand“ oder „Seite“) und während der Ausführung einen *aktuellen Wert* (z.B. „3-Z“ oder „Page-Spectra“). Die Namen von Parametern werden gewöhnlich in eckigen Klammern geschrieben, also „[Mode]“ oder „[Layout]“.

Die Beschreibung eines Parameters wird verwendet, wenn dem Benutzer eine Auswahl möglicher Werte angeboten wird. Für diesen Fall werden auch den Werten Beschreibungs-Texte gegeben.

In dem Fenster zur Auswahl von Parameter-Werten (siehe Abbildung rechts) erscheinen immer die Beschreibungen; die wirklichen Namen der Parameter und die wirklichen Werte bleiben verborgen. (Beispielsweise heißt



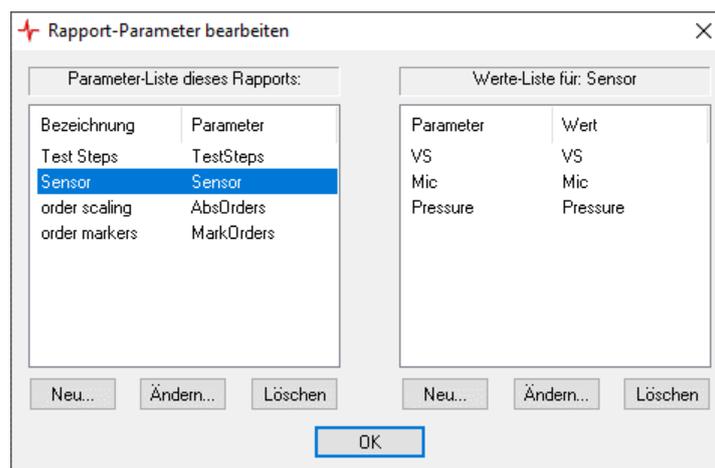
der erste Parameter in diesem Beispiel intern [AbsOrders], hat als Beschreibung den Text „order scaling“, und die Werte 1 und 0, die ihrerseits die Beschreibungen „absolute“ und „relative“ besitzen.)

Parameter erzeugen

Es gibt im Wesentlichen drei Methoden, um einen Parameter zu erzeugen: durch den Rapport-Befehl „Def“, durch Angabe des Wertes für ein Unter-Rapport („Call“-Befehl), und durch Anlegen einer Parameter- und Werte-Liste für ein Rapport.

Die „Def“- und „Call“-Befehle werden in der Befehlsreferenz erläutert; an dieser Stelle wird das Anlegen von Parameter- und Wertelisten beschrieben.

Der Rapport-Editor bietet ein grafisches Interface für die Verwaltung von Rapport-Parametern, indem man den Knopf [Parameter...] unterhalb der Befehlsliste drückt.



Auf der linken Seite werden die existierenden Parameter aufgelistet. Wählt man einen Parameter an, so erscheinen rechts davon die für diesen Parameter definierten Werte.

Beide Listen haben zwei Spalten: jeweils links erscheint die Beschreibung, rechts der tatsächliche Parameter-Name (hier ohne []) bzw. der tatsächliche Parameter-Wert. Wie man am Beispiel „Sensor“ und dessen Werten sieht, müssen sich Beschreibung und Name bzw. Wert nicht unterscheiden. Der Parameter „AbsOrders“ hat jedoch eine anderslautende Beschreibung.

Verwenden Sie die Knöpfe [Neu...], [Ändern...] und [Löschen...] unterhalb der Listen, um neue Parameter bzw. Wertelisten anzulegen und zu bearbeiten.

Parameter-Namen und -Werte dürfen *keine Leerzeichen* enthalten. Beschreibungen mit Leerzeichen sind erlaubt.

Parameter-Listen im Rapport.sea

Im **Rapport.sea** werden die Parameter eines Reports und deren Wertelisten in folgendem Format gespeichert:

```
Order-Spectra_Params: {
  TestSteps "Test Steps" BoostR
  Sensor     Sensor VS
  AbsOrders  "order scaling" 1
  MarkOrders "order markers" yes
}
Order-Spectra_Param_TestSteps: {
  BoostR "Boost Ramps"
  WstGte "Waste Gate Ramps"
}
Order-Spectra_Param_Sensor: {
  VS VS
  Mic Mic
  Pressure Pressure
} ;-- Order-Spectra_Param_Sensor
Order-Spectra_Param_AbsOrders: {
  1 absolute
  0 relative
}
Order-Spectra_Param_MarkOrders: {
  yes on
  no  off
}
```

In diesem Beispiel ist „Order-Spectra“ der Name des Reports. Dieser Name taucht im Namen jeder der Listen auf. Die Liste „(Rapport)_Params“ enthält alle Parameter. Zu jedem Parameter gibt es dann eine weitere Liste „(Rapport)_Param_(Parameter)“ mit den möglichen Werten.

In jeder dieser Listen wird in jeder Zeile zuerst der interne Name bzw. Wert aufgeführt, und als zweites die Beschreibung für das Auswahlfenster. Bei den Parametern gibt es noch eine dritte Spalte, die den zuletzt verwendeten Wert enthält.

Wie am obigen Beispiel schon in der ersten Parameter-Zeile „Test Steps“ erkennt, müssen Beschreibungen, die Leerzeichen enthalten, in Anführungszeichen eingeschlossen werden.

Vordefinierte Parameter

Der Parameter [#M] ist automatisch definiert und enthält die Anzahl der aktuell geladenen Messungen in der Liste.

Weiterhin definiert der A/B-Vergleichsassistent bei Ausführung seines Darstellungs-Rapports automatisch einen globalen Parameter [A/B-Vergleich] mit dem festen Wert „true“.

Der Auswerte-Assistent definiert bei Ausführung seines Reports zwei automatische Parameter, nämlich [Auswertung] mit dem Wert „true“ und [Auswerte-Verfahren] mit der Text-Beschreibung des Auswerteverfahrens (z.B. „Mittelwert + 3 x Standardabweichung“).

Weiterhin können im Rapport.sea globale Parameter definiert werden. Mehr dazu auf Seite 29.

Der Aufbau eines Rapport-Befehls

Ein Rapport-Befehl besteht aus 12 Elemente (die im Rapport-Editor als 12 Spalten erscheinen). Das erste Element des Befehls ist das Kommando; bei vielen Befehlen enthält das Kommando einen Doppelpunkt und eine genauere Spezifikation.

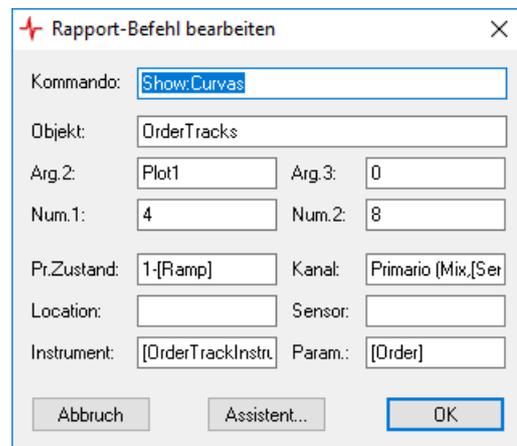
Die anderen 11 Elemente spezifizieren genauer, was das Kommando bewirkt. Sie können je nach Kommando unterschiedliche Funktionen haben oder auch gar nicht benutzt werden. Beim *Show*-Befehl beispielsweise beschreiben die 11 Elemente, welcher Datenknoten in welchem Grafik-Modul angezeigt werden soll, beim *Def*-Befehl beschreiben die ersten 3 Elemente den neuen Parameter und seinen Wert.

In der Befehlsliste des Rapport-Editors haben die Spalten für die Elemente Namen („Objekt“, „Arg2“ usw.). Diese Bezeichnungen dienen später in der Kommando-Dokumentation dazu, die einzelnen Elemente zu adressieren.

Wenn man im Editor auf einen Befehl doppelklickt (oder die Knöpfe *Bearbeiten*  oder *Neuer Befehl*  verwendet), so erscheint das Fenster zum Bearbeiten des Befehls, in dem die einzelnen Elemente aufgeführt sind.

Während die Elemente „Objekt“, „Arg2“, „Arg3“, „Num1“ und „Num2“ sehr variable für die verschiedenen Kommandos verwendet werden, ist haben die fünf Clavis-Elemente („Prüfzustand“ usw.) üblicherweise immer diese Bedeutung.

Obwohl die beiden „Num“-Elemente vom Namen her nahelegen, dass sie Zahlenwerte darstellen, muss dies nicht immer der Fall sein. Und auch wenn das Kommando für dieses Element einen Zahlenwert verlangt, kann im Befehl stattdessen beispielsweise ein Parameter erscheinen statt einer konkreten Zahlenangabe.



Darstellung im Rapport.sea

In der Textdatei `Rapport.sea` steht jeder Befehl in einer Zeile. Die Elemente des Befehls werden durch Leerzeichen getrennt. Wenn ein Element Leerzeichen enthalten soll, muss das ganze Element von Anführungszeichen eingeschlossen sein, wie in diesem Beispiel das Element „`Primario (Mix, [Sensor])`“:

```
Show:Curvas [Layout] Plot1 0 4 8 1-[Ramp] "Primario (Mix, [Sensor])" , [OrderTrackInstrument] [Order]
```

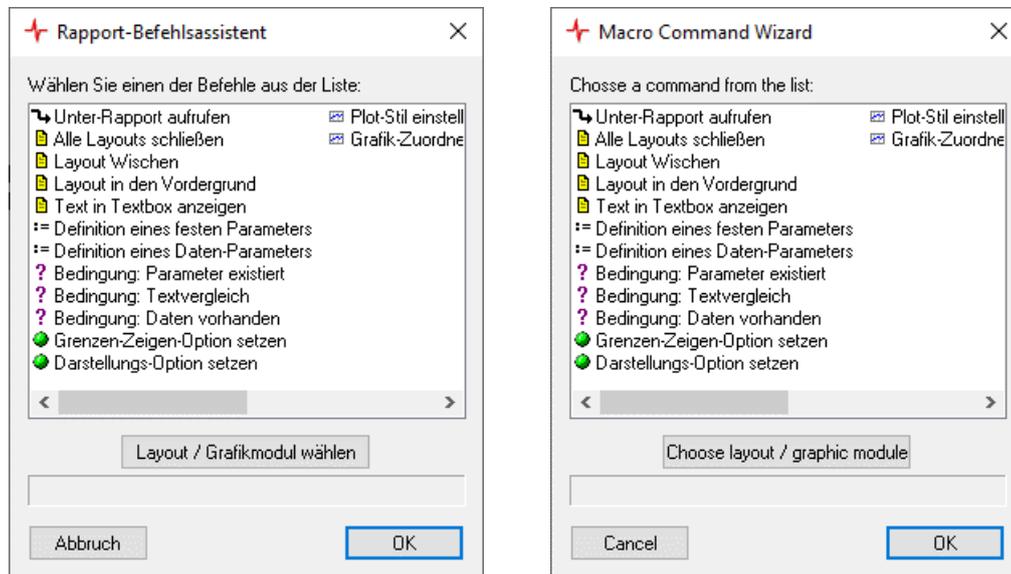
Elemente ohne Inhalt, wie in diesem Beispiel das `Location`-Element, werden durch ein Komma repräsentiert. Leere Elemente am Ende des Befehls können durch Kommata repräsentiert werden oder ganz weggelassen werden:

```
Layouts:DisplayText [Layout] Plot1.Title "Curva de {:Order} en 1-[Ramp]" , ,
```

Zwischen den Elementen und auch vor dem Befehl sind beliebig viele Leerzeichen erlaubt.

Der Befehls-Assistent

Wenn man im Rapport-Editor von Hand einen neuen Befehl erstellen möchte, und den korrekten Befehls-Namen oder das Grafikmodul nicht parat hat, hilft einem der Assistent. Im **Rapport-Befehl bearbeiten**-Fenster (siehe Abbildung oben) drückt man den Knopf **Assistent**, um ihn zu rufen:



Der Assistent bietet die am häufigsten verwendeten Befehle an (bis auf den Show-Befehl, der am besten durch Aufzeichnen erstellt wird). Die Befehle werden im Kapitel „Befehls-Referenz“ ab Seite 16 beschrieben. Wählen Sie einfach einen Befehl und drücken Sie **OK**.

Manche Befehle verwenden ein Grafikmodul (z.B. wenn es um das Darstellen von Text geht) oder Layout-Seite. Um für Ihren Befehl ein Grafikmodul oder Layout zu wählen, drücken Sie den entsprechenden Knopf, woraufhin sich das übliche **Grafikmodul Zuordnen**-Fenster öffnet. Verwenden Sie es wie beim direkten Darstellen von Daten und bestätigen Sie es mit **OK**. Wenn es nur um ein Layout geht, wählen Sie irgendein Grafikmodul auf der gewünschten Seite. Der Befehls-Assistent übernimmt dann das ausgewählte Layout und Grafikmodul in den Befehl.

Die Syntax im Rapport.sea

Die Datei **Rapport.sea** ist eine reine Textdatei (mit Codierung Utf-8, d.h. es können auch chinesische Schriftzeichen oder Emoji verwendet werden). Sie wird beim Starten der Präsentations-App eingelesen und im Speicher gehalten, und beim Beenden der App (oder beim Drücken des Speichern-Knopfes) zurückgeschrieben.

Jede Zeile in der Datei, die mit einem Semikolon ; beginnt, ist eine Kommentar-Zeile. Weiterhin dürfen beliebig Leerzeilen eingefügt oder weggelassen werden. Zusätzliche Leerzeichen zwischen den Elementen oder auch am Beginn und Ende von Zeilen werden ebenfalls ignoriert.

Der Inhalt von **Rapport.sea** besteht hauptsächlich aus Listen: Die Liste der bekannten Rapports, die Liste der Befehle für einen Rapport, Listen von Parametern und von Werten für Parameter. Die Syntax dieser Liste ist:

```
Listenname: {  
    Eintrag1  
    Eintrag2  
    ...  
}
```

Hinter dem Listennamen (das kann z.B. der Name eines Rapports sein) folgt ein Doppelpunkt, dann ein Leerzeichen, dann eine öffnende geschweifte Klammer. Auf den folgenden Zeilen steht

der Inhalt der Liste, eine Zeile pro Listen-Element. Die Liste wird abgeschlossen durch eine eigene Zeile, die mit einer schließenden geschweiften Klammer beginnt. Hinter der Klammer ist ein mit Semikolon eingeleiteter Kommentar zulässig (der z.B. angeben kann, welche Liste hier beendet wird). Für ein reales Beispiel siehe die Liste der Parameter und Werte im Abschnitt „Parameter-Listen im Rapport.sea“ (Seite 11).

Einige Einträge sind keine Listen. In diesem Fall stehen hinter dem Namen des Eintrags und dem Doppelpunkt direkt einer oder mehrere Werte.

```
Order-Spectra-Optionen: 0 1
```

Die genaue Bedeutung dieser Werte hängt vom Eintrag ab.

Liste der Rapports und der globalen Definitionen

Jedes `Rapport.sea` beginnt mit der Liste der bekannten Rapports. Diese trägt einfach den Namen „Rapports“:

```
Rapports: {
  Order-Spectra   Ordnungsspektren
  Fix-Spectra     FixFs-Spektren
  Spectra-Track   Spectra-Track
  OrderSpectrogram OrderSpectrogram
  FixSpectrogram FixSpectrogram
  xInit xInit
  xModes xModes
}
```

Auch bei dieser Liste, wie bei den Parametern, ist der erste Eintrag jeder Zeile der tatsächliche (interne) Name des Rapports, der zweite Eintrag der Anzeige-Name z.B. für die Werkzeugleiste.

Als nächstes folgt die Liste der globalen Parameter-Definitionen „GlobalDefs“:

```
GlobalDefs: {
  OrderTrackInstrument      "Spectral Track"
  OrderSpectrumInstrument  "End Spectrum"      "Peak Hold Spektrum"
  CurrentPosition          Göttingen          "Ort der Messung"
}
```

Das erste Element jeder Zeile ist der Name des Parameters, das zweite Element der Wert, den der Parameter haben soll. Weitere Einzelheiten finden Sie im Abschnitt „Globale und Automatische Parameter“ auf Seite 29.

Reihenfolge beliebig

Nach den globalen Definitionen folgen die einzelnen Rapports – jedes Rapport hat einen „Optionen“-Eintrag und eine Liste der Befehle, sowie die Liste der Parameter und die Listen der Werte zu diesen Parametern.

Allerdings ist die Anordnung der Listen in `Rapport.sea` beliebig. Die Liste „GlobalDefs“ kann also z.B. auch am Ende von `Rapport.sea` stehen. Die Listen mit den Parametern und Werten müssen nicht direkt nach der Befehls-Liste eines Rapports folgen. Die Präsentations-App selbst fügt neue Elemente immer am Ende an: wenn man zu einem Rapport einen neuen Parameter anlegt, so wird dessen Werte-Liste am Dateiende stehen, und nicht in der Nachbarschaft des Rapports, zu dem sie gehört.

Es ist aber kein Problem (und für die Übersichtlichkeit sogar ratsam), mit dem Texteditor die Listen in der Datei zu verschieben (ausschneiden, einfügen), um so alle Teile eines Rapports zusammenzufassen.

Die Reihenfolge der Elemente *innerhalb* einer Liste ist hingegen nicht beliebig. Man muss also immer ganze Listen verschieben.

Rapport-Versionierung

Das `Rapport.sea` kann optional einen Eintrag der folgenden Art enthalten (typischerweise am Anfang der Datei):

```
RequiresRapportVersion: 2019 02 28
```

Die drei Zahlen sind zu lesen als Jahr, Monat und Tag. Ist der Eintrag vorhanden, so überprüft die Präsentations-App beim Laden des `Rapport.sea`, ob die eigene Software-Version älter oder neuer ist als das angegebene Datum (im Beispiel der 28. Februar 2019). Ist die eigene Version älter, so besteht die Möglichkeit, dass im `Rapport.sea` Befehle verwendet werden, die das Präsentationsprogramm noch nicht kennt. Daher wird eine Warnung ausgegeben, dass einige Rapports möglicherweise nicht korrekt funktionieren werden.

Anmerkung: Diese Funktion existiert erst ab der Präsentations-Version 2019-03-01. Ältere Programmversionen kennen den Eintrag nicht und geben daher auch keine Warnungen aus.

Befehls-Referenz

Dieses Kapitel listet alle Befehle nach Funktionen gruppiert auf. Für eine alphabetische Liste werfen Sie einen Blick in den Index.

Viele Befehle gibt es mit einem deutschen und einem englischen Kommando. Beide Varianten sind austauschbar. Beim Aufzeichnen von Rapports hängt die Sprache der entstehenden Kommandos von der Sprache ab, die für die Präsentations-App eingestellt wurde. Bei der Wiedergabe spielt die Sprache keine Rolle.

Jeder Befehl hat (bis zu) 12 Elemente. Das erste Element ist das Kommando, die weiteren Elemente sind *Objekt*, *Ar2*, *Arg3*, *Num1*, *Num2* und die *Clavis*-Elemente. Siehe dazu auch „Der Aufbau eines Rapport-Befehls“ auf Seite 12.

Daten darstellen („Show“-Befehl)

Die häufigste Aufgabe im Rapport ist das Darstellen von Daten in einem Grafikmodul. Diese Funktion erledigt man mit dem Show-Befehl, der (mit Beispiel-Parametern) so aussieht

```
Show: Messkurven      Layout-Kurven  GSigCurvePlot5  0 4 8 1.M Mix N-Mon
ShowIfData: Messkurven  Layout-Kurven  GSigCurvePlot5  0 4 8 1.M Mix N-Mon
```

Der Show-Befehl wird beim Aufzeichnen eines Rapports erzeugt. Das Format des Show-Befehls ist wie folgt:

Hinter dem Doppelpunkt steht der Name des „Kapitels“ im Inhaltsbaum, aus dem die darzustellenden Daten stammen. Dieser Name wird allerdings nur bei manchen Datentypen zur Identifikation verwendet. Die weiteren Parameter haben folgende Werte:

Objekt = Name des Layouts, *Arg2* = Name des Grafikmoduls, in dem die Daten erscheinen sollen.

Arg3 = Farbe (Farb-Index). Optional kann hier auch die Farbe für die Grenze codiert werden. Multiplizieren Sie die Farbe für die Grenze mit 1000 und addieren sie diesen Wert zum normalen Farbindex. (Beispiel: 3005 bedeutet Farbe 5 für die normalen Kurven und 3 für die Grenzen.) Wenn keine Farbe für die Grenze festgelegt explizit wurde, wird die in der Abteilung **Grafik** für Grenzen ausgewählte Farbe verwendet.

Num1 und *Num2* codieren den Knoten im Inhaltsbaum; am besten erhalten Sie die Werte, indem Sie einen entsprechenden Show-Befehl durch Aufzeichnung erstellen.

Clavis = genaue Spezifikation des Datums (Knoten im Baum).

Im Gegensatz zum normalen Show-Befehl erzeugt die Variante *ShowIfData* keine Fehlermeldung, wenn der über *Num1*, *Num2* und *Clavis* spezifizierte Daten-Knoten nicht gefunden werden kann. (Siehe dazu auch den Bedingungs-Befehl „Daten vorhanden“ auf Seite 24.)

Der Show-Befehl entsteht durch die Zusammenfassung von vier Einzelbefehlen, nämlich *Daten:OnGraphAuswahl*, *GraphMod:OnHinzunehmen*, *GraphMod:OnApply* und *Daten:OnDatenAnzeigen*. Wenn Sie ein Rapport aufzeichnen und die auf Seite 6 beschriebenen Schritte nicht genau einhalten, werden Sie diese vier Einzelbefehle im Rapport-Editor sehen.

Wenn Sie bei der Aufzeichnung nicht die Kombination dieser vier Befehle einhalten (z.B. weil Sie zwei Mal *GraphMod:OnHinzunehmen* gedrückt haben, da sie die Daten in zwei Grafikmodulen parallel anzeigen wollen), kann die Präsentations-App die Zusammenfassung zum Show-Befehl nicht durchführen. Sie können dann entweder versuchen, die Elemente des Show-Befehl aus den

Einzelbefehlen selbst zusammensetzen, oder die Einzelbefehle löschen und (mit Aufzeichnung fortsetzen) einen neuen Versuch unternehmen.

Zur Erläuterung hier die vier Einzelbefehle:

Daten:OnGraphAuswahl oder **Data:OnGraphSelect**: Anwahl eines Daten-Knotens im Inhaltsbaum, für den mit den nächsten Befehlen Grafikmodule zugewiesen werden sollen. (Zur Aufzeichnung selektieren Sie den Knoten im Inhaltsbaum und drücken den **Xxx darstellen in-Knopf**.) Wird normalerweise im Show-Befehl vereinnahmt.

Daten:OnDatenAnzeigen oder **Data:OnDisplayData**: Anwahl eines Daten-Knotens, dessen Daten dargestellt werden sollen (entspricht dem Drücken des entsprechenden Knopfes in der Abteilung Daten). Auch dieser Befehl verschwindet normalerweise im Show-Befehl.

GraphMod:OnHinzunehmen codiert ein Grafik-Modul auf einer Layout-Seite. Diese beiden Namen erscheinen später als „Objekt“ und „Arg2“ im Show-Befehl. **GraphMod:OnApply** und **GraphMod:OnOK** stellen nur das Bestätigen des Grafikmodul-Auswählen-Fensters dar. Der **Num1**-Parameter ist der Farb-Index, der für die Darstellung der Daten verwendet werden soll.

Daten wegwischen

Daten:OnTafelWischen oder **Data:OnWipeTable**: Anwahl eines Daten-Knotens, dessen Daten aus allen Grafiken, in denen sie eingetragen sind, weggewischt werden sollen. Dieser Befehl wird in Rapports selten verwendet; meistens benutzt man den entsprechenden Befehl, um eine ganze Layout-Seite zu „wischen“ (siehe unten).

Daten:OnOptionen oder **Data:OnOptions**: Diese Befehle sind obsolet, können aber in älteren Rapports noch auftreten.

Daten nur laden (vorbereiten)

Mit dem Befehl **Daten:DatenVorbereiten** werden die Daten zu einem Datenknoten aus den Archiven geladen, ohne direkt dargestellt zu werden. Man gewinnt ihn z.B. durch Aufzeichnen und Abwandeln eines **Daten:OnGraphAuswahl**-Befehls.

Darstellungsoptionen setzen

Der Befehl **Daten:OptionSetzen** (oder **Data:SetOption**) kann verschiedenste Darstellungsoptionen setzen, abhängig von seinen Parametern. Diese Optionen gelten typischerweise global, also auch für bereits vor dem Rapport-Befehl zugewiesene Darstellungen. Beispielsweise wirkt das Setzen der Option „Grenzen sollen nicht angezeigt werden“ global auf alle Messkurven-Darstellungen, selbst wenn der entsprechende **Daten:OptionSetzen**-Befehl als allerletztes im Rapport steht.

Durch das **Arg1**-Argument des Befehls wird die zu setzende Option bestimmt, die anderen Parameter wählen die gewünschte Einstellung für die betreffende Option:

Funktion	Arg1	Einstellungen
Grenzen anzeigen	GrenzenZeigen	Arg2 = 0 : keine Grenzen, 1 : Grenzen zeigen, 2 : nur letzte Grenze
Spektren skalieren	SpektrenAbsOrdnung	Arg2 = 0 : Spektren nicht skalieren, 1 : Ordnungsspektren auf Referenz-Rotor skalieren

Beispiel: durch den Befehl

```
Daten:OptionSetzen GrenzenZeigen 2
```

wird bewirkt, dass für alle Messkurven jeweils nur eine Grenzkurve, nämlich die der letzten Messung der Liste, angezeigt wird.

Befehle für eine Layout-Seite

Die Funktion dieser Befehle erschließt sich aus dem Kommando-Namen:

```
Layouts:OnVordergrund "6 Plots"  
Layouts:OnSchliessen "6 Plots"  
Layouts:Schliessen "6 Plots"  
Layouts:AlleSchliessen  
Layouts:Minimieren "6 Plots"  
Layouts:Maximieren "6 Plots"  
  
Layouts:OnWischen "6 Plots"  
Layouts:AlleWischen  
Layouts:OnDrucken "6 Plots"  
Layouts:OnAlleDrucken Drucktitel
```

Das Argument („Objekt“) ist jeweils der Name einer Layout-Seite. Als Alternative gibt es die englischen Varianten:

```
Layouts:OnForeground "6 Plots"  
Layouts:OnClose "6 Plots"  
Layouts:Close "6 Plots"  
Layouts:CloseAll  
Layouts:Minimize "6 Plots"  
Layouts:Maximize "6 Plots"  
  
Layouts:OnWipe "6 Plots"  
Layouts:WipeAll  
Layouts:OnPrint "6 Plots"  
Layouts:OnPrintAll JobTitle
```

Der erste Befehl bringt die bezeichnete Layout-Seite in den Vordergrund, der zweite und dritte schließt sie, der vierte schließt alle Seiten. Der **OnWischen**-Befehl veranlasst das Wegwischen aller Daten aus Grafiken auf der bezeichneten Layout-Seite, der **AlleWischen**-Befehl das Wegwischen aller Daten. **OnDrucken** bewirkt das Ausdrucken der bezeichneten Seite. Der Objekt-Parameter ist jeweils der Name (Benutzer-Name) des betreffenden Layouts. **OnAlleDrucken** bewirkt das Ausdrucken aller geöffneten Layout-Seiten. Der Objekt-Parameter setzt den Namen des Druckjobs. Dieser ist für normale Drucker ohne Bedeutung; wird allerdings ein Pseudo-Drucker verwendet, der Dateien produziert (wie ein PDF-Writer), so wird aus dem Druckjob-Namen der Dateiname. Setzt man als Drucktitel den speziellen Wert `[AutoTitle]` ein (inklusive eckigen Klammern), so wird automatisch ein Name aus der Seriennummer und der Zeitmarke der ersten dargestellten Messung generiert.

Text anzeigen

Der Befehl `Layouts:TextAnzeigen` oder `Layouts:DisplayText` füllt eine Textbox mit einem im Rapport festgelegten Text. Ein Beispiel:

```
Layouts:TextAnzeigen Layout-6Plots Textbereich0 "Messung in [Chan]" 4
```

Der **Objekt**-Parameter gibt den Namen des Layouts an, **Arg2** den Namen der Textbox, **Arg3** den Text (der, wie im Beispiel gezeigt, mit Anführungszeichen umschlossen werden muss, wenn er Leerzeichen enthält). Das **num1**-Argument gibt den Farbindex an, falls der Text farbig sein soll; ist **num1** Null, so wird die im Layout eingestellte Farbe der Textbox verwendet.

Auch hier gibt es wieder eine Spezialität: wenn der anzuzeigende Text (`Arg3`) „\$Anzahl\$“ oder „\$Count\$“ lautet, wird die Anzahl momentan geladener Messungen in der Textbox ausgegeben. (Oder man verwendet den automatischen Parameter `[#M]` – siehe „Vordefinierte Parameter“ auf Seite 11.)

Ist das Ziel nicht nur eine einfache Textbox, sondern ein Listen-Textfeld, so wird diesem eine Zeile in der angegebenen Farbe hinzugefügt (d.h. der bisherige Inhalt bleibt erhalten).

Bezeichnungen von Parametern und Werten verwenden

Gelegentlich möchte man auf einer Layout-Seite anzeigen, *welche* Daten man mit dem Rapport dargestellt hat, und dazu die aktuell ausgewählten Werte der Rapport-Parameter in Textfeldern ausgeben. Nehmen wir als Beispiel den Befehl

```
Layouts:TextAnzeigen [Layout] SubTitle [Parameter] 4 0
```

Hierdurch wird der aktuelle Wert des `[Parameter]` in das Textfeld „SubTitle“ geschrieben. Aber es geht auch schöner:

Bezeichner	liefert	Beispiel
<code>[Parameter]</code>	Aktueller Wert des Parameters	3-D
<code>{Parameter}</code>	Benutzername (Beschreibung) des Parameters	„Prüfzustand“
<code>{:Parameter}</code>	Benutzername (Beschreibung) des aktuellen Wertes	„3. Gang Zug“

Mit dem Befehl

```
Layouts:TextAnzeigen [Layout] SubTitle "{Mode} = {:Mode}" 4 0
```

erhält man also im Textfeld bei diesem Beispiel die Anzeige „Prüfzustand = 3. Gang Zug“.

Die Benutzernamen oder Beschreibungen werden bei der Definition der Parameter und deren Werte mit angegeben; siehe dazu „Name und Beschreibung“ ab Seite 10.

Befehle für Archive

Es gibt einen Befehl zum Laden eines Archivs und einen, um es wieder zu „ent-laden“:

`Archive:ArchivLaden` oder `Archive:LoadArchive` hat als Objekt-Parameter den Dateinamen (vollständigen Pfad) des zu ladenden Archivs, und als optionales `Arg2` einen Namenszusatz, der gleich näher erläutert werden soll.

`Archive:ArchivEntfernen` oder `Archive:RemoveArchive` hat ebenfalls als Objekt-Parameter den Dateinamen.

Der Namenszusatz bei `Archive:ArchivLaden` ist eine Funktion, die nicht durch Aufzeichnen erreicht, sondern nur über den Rapport-Editor eingegeben werden kann. Wenn Sie ein Archiv mit Namenszusatz laden, bekommen dessen Daten im Inhaltsbaum eigene Knoten, die ebenfalls mit diesem Namenszusatz gekennzeichnet sind. Sie können diesen Daten dann Grafikmodule und Farben zuweisen, und zwar unabhängig von den Daten aus Archiven ohne Namenszusatz. Eine Anwendung dieser Funktion ist das Laden einer Referenzmessung. Sie können dadurch die Referenzkurven in einer anderen Farbe anzeigen lassen als die normalen Daten.

Rechteck- und Plot-Stile ändern

Plot-Stil ändern

Mit dem Befehl `Layouts:SetPlotStyle` kann man einige Eigenschaften eines Kurvenplots, Spektrogramms oder n-dimensionalen Plots per Rapport ändern:

```
Layouts:SetPlotStyle Layout-6Plots Plot0 "Titel" a b x1 x2 x3 x4 x5
```

Im dritten Argument (`Arg3`) kann man einen Titel für diesen Plot übergeben. Bleibt dieser Text leer (ein Komma), wird der Titel des Plots nicht geändert. Das `Num1`-Argument `a` codiert, was an dem Plot eingestellt werden soll, die Bedeutung des zweiten Arguments `b` sowie der fünf weiteren Argumente (die sonst für Prüfzustand, Kanal usw. stehen), hängt von `a` ab. Die folgende Tabelle gibt eine Übersicht:

a	1 = x-Achsen-skalierung einstellen	2 = y-Achsen-skalierung einstellen	3 = verbundene Skalierung parametrieren	4 = Hintergrundfarbe ändern
b	0 = feste Skalierung wie in x1, x2, x3 angegeben 1 = automatische Skalierung nach Datenbereich		1 = x-Achse, 2 = y-Achse, 3 = beide	0 = transparent 1 = Farbe laut x1 – x3
x1	Beginn der Achse (Minimum)		x-Faktor	RGB Rot-Anteil
x2	Ende der Achse (Maximum)		x-Offset	RGB Grün-Anteil
x3	Optional: Skalen-Schrittweite. Wenn nicht angegeben, wird die Skala automatisch geteilt		y-Faktor	RGB Blau-Anteil
x4			y-Offset	
x5			opt.: Plot-Name	
a	5 = z-Achsen-skalierung für Spektrogramm, 3D-Plot oder ND-Plot einstellen			
b	0 = feste Skalierung wie in x1, x2 angegeben 1 = automatische Skalierung nach Datenbereich			
x1	Beginn der Achse (Minimum)			
x2	Ende der Achse (Maximum)			
x3	Optional: Skalen-Schrittweite.			
x4				
x5				

Wie man sieht, werden die Parameter, die sonst Prüfzustand, Kanal usw. enthalten, bei `SetPlotStyle` als Zahlen interpretiert. Unbenutzte Argumente können wie immer entfallen. Ein Beispiel: mit dem Befehl

```
Layouts:SetPlotStyle Layout-6Plots Plot0 "Nandemo" 1 0 700 1400 30
```

wird der angezeigte Titel des Plot0 auf „Nandemo“ geändert und die x-Achse auf eine feste Skalierung von 700 bis 1400 mit einer Schrittweite von 30 eingestellt. Ohne die 30 am Ende würde eine automatische Schrittweite für das Intervall [700, 1400] berechnet.

Für `SetPlotStyle` mit `a=3` (verbundene Skalierung parametrieren) kann optional ein Plot-Name angegeben werden. Wird ein Name angegeben, so wird die Skalierung an den benannten Plot

gebunden. Wird kein Name angegeben, so wird die Bindung nicht geändert (bleibt wie zuvor oder ggf. über Dialog gesetzt). Wird ein Name angegeben, der aber nicht der Name eines Plots ist (z.B. „NONE“), so wird die Skalierungs-Verbindung getrennt.

Rechteck-Stil ändern

Der Befehl `Layouts:SetRectStyle` erlaubt es, die Farbe, Transparenz und Rahmen-Eigenschaften eines Rechtecks oder davon abgeleiteten Grafikmoduls zu ändern – das trifft auf jedes Modul zu, in dessen Einstellungen man die Reiter für „Position“ und „Rahmen“ findet. (Der „Rahmen“-Teil ist optional.)

```
Layouts:SetRectStyle [Layout] RectName cr cg cb fc tr fw fp
```

Die einzelnen Parameter haben folgende Funktion:

RectName	Wie üblich der Name des Grafikmoduls
cr, cg, cb	RGB-Werte für die Farbe des Rechtecks
fc	Verwendung der Farbe: 0 = die Farbe gilt für das Innere des Rechtecks. 1 = die angegebene Farbe soll für den Rahmen verwendet werden. -1 = keine Änderung der Rechteck- oder Rahmenfarbe.
tr	Transparenz: 0 = ausgefüllt, 1 = transparent, -1 = keine Änderung
fw	Rahmen-Breite in der aktuellen Einheit, -1 = keine Änderung
fp	Rahmen-Abstand („padding“) in der aktuellen Einheit, -1 = keine Änderung

Die Farben werden als RGB-Werte angegeben, jeweils im Bereich 0 – 255. Die Rahmenbreite und der Abstand werden so angegeben, wie man sie im Einstellfenster eingeben würde. (Der „Abstand“ trennt den Inhalt eines Rechtecks, z.B. einer Textbox, vom Rahmen. Bei einem Abstand von 0 berührt der Textinhalt die Rahmenlinie, was meist unerwünscht ist.)

Balkengrafik-Stil ändern

Um die Skalierung und einige andere Eigenschaften einer Balkengrafik zu ändern, gibt es den Befehl `Layouts:SetBarGraphStyle`.

```
Layouts:SetBarGraphStyle [Layout] BarGraphName scaling scMin scMax numBars
```

Die einzelnen Parameter haben folgende Funktion:

BarGraphName	Wie üblich der Name des Grafikmoduls
scaling	Skalierung: 0 = automatisch, 1 = mit festem Faktor, 2 = festes Intervall, -1 = keine Änderung
scMin	Minimum der Skalierung (für Intervall)
scMax	Maximum (für Intervall) oder Faktor der Skalierung
numBars	Platz für mindestens so viele Balken bereithalten; -1 = keine Änderung

Unter-Rapports aufrufen

Call:Report-Name: Der Call-Befehl ruft ein anderes Rapport auf, dessen Name hinter dem Doppelpunkt steht. Alle existierenden Parameter-Werte werden an das aufgerufene Rapport weitergegeben, haben also dort dieselben Werte. Zusätzlich können Sie weitere Aufruf-Parameter als **Objekt-Element** angeben:

```
Call:MTest "[Mode]=5-Z, [Chan]=Mix"
```

Mehrere Parameter können als Komma-Liste aufgeführt werden. Bei der Spezifikation dieser Aufruf-Parameter können andere Parameter verwendet werden:

```
Call:MTest "[Mode]=5- [Rampe]"
```

Hier wird **[Rampe]** ersetzt, z.B. durch „Z“, so dass für das aufgerufene Rapport „**MTest**“ der Parameter **[Mode]=5-Z** definiert ist.

Sprungbefehle

Mit **Goto:** und **Label:** kann innerhalb eines Rapports gesprungen werden (auch rückwärts). Durch den **Label-**Befehl wird eine beliebige Textmarke definiert. Durch den **Goto-**Befehl wird die Ausführung mit dem Befehl hinter dem entsprechenden **Label-**Befehl fortgesetzt.

```
Goto:(Textmarke)  
Label:(Textmarke)
```

Es kann nur innerhalb eines Rapports gesprungen werden, nicht in ein anderes Rapport oder Unter-Rapport.

Mit dem Befehl

```
Return:
```

Wird die Ausführung des aktuellen (Unter-)Rapports beendet, als ob der letzte Befehl erreicht worden wäre.

Durch Kombination von Unter-Rapports und Sprungbefehlen mit der Bedingten Ausführung kann man sehr komplexe Abläufe realisieren.

Bedingte Ausführung

Die Rapports sind keine echte Programmiersprache und stellen daher keineswegs alle Konstrukte einer echten Programmiersprache bereit. Die Befehle zur bedingten Ausführung können nur folgendes: wenn die im Befehl definierte Bedingung nicht erfüllt ist, wird der nächstfolgende Rapport-Befehl übersprungen. Optional kann man mit dem `Else`-Befehl zwischen zwei Alternativen wählen.

Alle Befehle zur Bedingten Ausführung beginnen mit `Wenn:` oder mit `If:`. Danach folgt die Definition der Bedingung. Unmittelbar nach dem Doppelpunkt kann ein Ausrufezeichen `!` stehen. Dadurch wird die Bedingung verneint. (Beispiele s.u.)

Textvergleiche

Diese Bedingung vergleicht zwei Texte (von denen meistens einer ein Parameter ist):

```
If:IsEqual Text1 Text2
Wenn:IstGleich Text1 Text2
```

Falls `Text1` buchstäblich gleich `Text2` ist, ist die Bedingung erfüllt und der nächste Befehl wird ausgeführt. Dieser Befehl macht erst Sinn, wenn man ihn zusammen mit Parametern verwendet:

```
Wenn:IstGleich [Channel] Mix
Call:SpecReport3
Wenn:!IstGleich [Channel] Mix
Call:SpecReport1
```

Die beiden ersten Befehle bewirken, dass das Unterrapport „SpecReport3“ genau dann aufgerufen wird, wenn der Parameter `[Channel]` den Wert „Mix“ hat. Die beiden letzten Befehle behandeln den anderen Fall: wenn `[Channel]` nicht den Wert „Mix“ hat (man beachte das `!` im Wenn-Befehl), so wird das Rapport „SpecReport1“ ausgeführt. (Siehe dazu auch die Beschreibung des „Else“-Befehls unten.)

Vergleich mit mehreren Werten

Man kann als Vergleichstext eine Liste von möglichen Werten angeben, die in geschweiften Klammern steht und durch Kommata getrennt wird:

```
If:IsEqual Text1 {Wert1,Wert2,Wert3}
```

Dieser Vergleich ist „wahr“, wenn `Text1` mit einem der aufgeführten Werte identisch ist. Wenn Vergleichstexte Leerzeichen enthalten, muss das ganze Element (inklusive der `{ }` Klammern) in Anführungszeichen eingeschlossen werden.

Größer-Kleiner-Vergleiche

```
If:Is< [Counter] 7
If:Is<= [Counter] 7
If:Is== [Counter] 7
If:Is!= [Counter] 7
If:Is>= [Counter] 7
If:Is> [Counter] 7
```

Diese Vergleiche sind speziell für die Verwendung mit numerischen Parametern (siehe Seite 27) gedacht, aber nicht darauf beschränkt. Der Wert des Parameters `[Counter]` wird mit dem zweiten Wert (hier Sieben) verglichen. Der zweite Wert kann natürlich auch ein Parameter sein. Ein Beispiel für die Verwendung findet sich auf Seite 31. Wenn einer der beiden Vergleichswerte

nicht als Zahl interpretiert werden kann, wird ein alphabetischer Vergleich der Werte ausgeführt. Alle diese Befehle gibt es auch in der Variante ohne das „Is“:

```
If:< [Counter] 7
If:<= [Counter] 7
If:== [Counter] 7
If:!= [Counter] 7
If:>= [Counter] 7
If:> [Counter] 7
```

Parameter vorhanden

```
If:Def Parameter-Name
Wenn:Definiert Parameter-Name
```

Diese Bedingung ist erfüllt, wenn es einen Parameter des angegebenen Namens gibt, unabhängig von dessen Wert. Beachten Sie, dass Sie den Namen des Parameters *ohne* eckige Klammern angeben müssen, also z.B. „Wenn:Definiert Chan“ und nicht „Wenn:Definiert [Chan]“.

```
Wenn:IstTasMessung oder If:IsTasMeasurement
```

sind erfüllt, wenn die hervorgehobene (oder die erste) Messung eine TasAlyser-Messung ist. Diese Bedingungen sind nicht erfüllt, wenn die Messung eine Rotas-Dpm42-Messung ist.

Daten vorhanden

Mit der If:Data-Bedingung kann überprüft werden, ob die aktuell geladenen Messungen bestimmte Daten-Knoten enthalten:

```
Wenn:Daten ... oder If:Data ...
```

Die weiteren Elemente dieses Befehls entsprechen denen eines Show-Befehls. Die Bedingung ist erfüllt, wenn im Inhaltsbaum ein entsprechender Daten-Eintrag vorhanden ist und dieser auch tatsächlich Daten enthält. Beispiel:

```
Wenn:Daten Spektren , , 0 4 3-Z [Chan]
Call:SpecReport3
```

Diese Befehle sorgen dafür, dass „SpecReport3“ nur dann ausgeführt wird, wenn es im betreffenden Datenknoten Daten gibt. Die bequemste Weise, die Elemente des Wenn:Daten-Befehls zu erzeugen, ist, im Rapport-Editor einen passenden Show-Befehl zu duplizieren und ihn von „Show:XYZ“ in „Wenn:Daten“ umzubenennen. Das „XYZ“ des Show-Befehls überträgt man in das erste Element („Objekt“) des Wenn-Befehls; die folgenden zwei Parameter kann man ignorieren oder durch ‚,‘ ersetzen. Wenn man keinen passenden Show-Befehl zur Hand hat, gibt es folgende Alternative: man zeichnet (per „Aufnahme fortsetzen“) nur die folgende Aktion auf: im Inhaltsbaum den betreffenden Datenknoten selektieren und „Darstellen in...“ drücken. Dann (ohne tatsächlich ein Grafikmodul auszuwählen) die Aufzeichnung beenden. Im Rapport steht jetzt ein Daten:OnGraphAuswahl-Befehl. Man benennt ihn um: aus „Daten:OnGraphAuswahl“ wird „Wenn:Daten“, und ersetzt eventuell einige Elemente durch [Parameter]. Dann verschiebt man den fertigen Befehl an die gewünschte Position.

Die Befehlssequenz

```
If:Data ...
Show:...
```

kann direkt als ShowIfData-Befehl formuliert werden; siehe „Daten darstellen („Show“-Befehl)“ auf Seite 16.

Daten-Werte prüfen

```
Wenn:DatenTextGleich Bewertung "n.i.O." , 3 4
```

Mit der `DatenTextGleich`-Bedingung wird überprüft, ob der „Daten-Text“ eines bestimmten Daten-Knotens (im Beispiel der Daten-Knoten „Bewertung“ aus dem „allgemeinen Informationen“ mit dem als zweites Element angegebenen Wert (im Beispiel „n.i.O.“) übereinstimmt. Im obigen Beispiel könnte als nächster Befehl (der nur ausgeführt wird, wenn die Bewertung „n.i.O.“ ist) ein Befehl folgen, der in fetter roter Schrift „n.i.O.“ ausgibt

Den `Wenn:DatenTextGleich`-Befehl generiert man am besten genauso wie den `Wenn:Daten-` Befehl oben: man modifiziert einen `Show-` oder einen `Daten:OnGraphAuswahl-` Befehl.

Nicht jeder Daten-Knoten besitzt einen sinnvollen Daten-Text, der sich für die `Wenn:DatenTextGleich`-Bedingung eignet. Für Daten-Knoten aus den „allgemeinen Informationen“ ist der Daten-Text gleich dem entsprechenden Daten-Wert der ersten Messung in der Liste.

```
Wenn:DatenWertGleich Ratio "2.54" , 3 6
Wenn:DatenWertGleich SingleValues "2.54" , 5 25 Puls - - L_Ges_Ks -
```

Dieser Befehl vergleicht den Wert einer Zusatzinformation oder den (Mittel-) Wert eines Einzahlkennwert-Eintrags mit dem angegebenen Zahlenwert.

Bewertungsergebnis prüfen

```
Wenn:BewertungIO (Mode-Name)
```

Der `BewertungIO`-Bedingungsbehehl erlaubt das Verzweigen abhängig vom Bewertungsergebnis der hervorgehobenen oder der ersten geladenen Messung. Ist keine Messung geladen, liefert er immer ‚falsch‘.

Die Angabe eines Mode-Namens ist optional. Wird kein Mode angegeben, dann bezieht sich die Abfrage auf das Gesamtergebnis. Wird ein Mode angegeben, der nicht vorhanden ist, dann wird das Ergebnis auf die Gesamtmessung verwendet (als wäre gar kein Mode angegeben).

```
Wenn:DatenGrenzeNio oder If:DataLimitNok
```

Dieser Befehl prüft für einen bestimmten Datenknoten, ob für die hervorgehobene Messung (die erste Messung, falls keine hervorgehoben ist) die Daten die zugehörige Grenze verletzt haben. Diese Bedingung liefert auch ‚unwahr‘, wenn es den fraglichen Datenknoten nicht gibt, er keine Daten enthält oder zu diesen Daten keine Grenze gespeichert ist. Ein Beispiel:

```
Wenn:DatenGrenzeNio Spektren , , 4 7 [Gang]-[Rampe] [Chan] OrderSpectra
Layouts:SetPlotStyle Kurven-2 Plot3 , 4 1 255 200 200
Sonst: , , , 0 0
Layouts:SetPlotStyle Kurven-2 Plot3 , 4 0
```

Hier wird geprüft, ob für das bezeichnete Ordnungsspektrum bei der hervorgehobenen Messung die Grenze verletzt wurde. Falls ja, wird beim entsprechenden Plot (hier „Plot3“ auf Layout-Seite „Kurven-2“) die Hintergrundfarbe auf Hellrot eingestellt; falls nicht, wird die Hintergrundfarbe auf transparent gesetzt (siehe den `SetPlotStyle`-Befehl auf Seite 156 und den `Sonst`-Befehl im folgenden Abschnitt).

Bedingung nicht erfüllt – ‚Else‘-Befehl

Auf den Befehl nach einem Bedingungs-Befehl kann optional der Befehl `Sonst:` oder `Else:` folgen (den Doppelpunkt nicht weglassen!). Ein Beispiel:

```
Wenn:IstGleich [Channel] Mix
Call:SpecReport3
Sonst:
Call:SpecReport1
```

Der Else-Befehl hat keinerlei Parameter.

Auch hier wieder der Hinweis: Rapports sind keine Programmiersprache. Wenn sich zwischen einem Wenn- und einem Sonst-Befehl mehr als ein Befehl befindet, wird (bei negativer Bedingung) nur der erste Befehl nach dem Wenn übersprungen. Wenn ein Sonst-Befehl ohne zugehöriges Wenn auftritt, wird der darauf folgende Befehl immer übersprungen. In keinem dieser Fälle gibt es eine Fehlermeldung!

Parameter-Definition

Mit den `Def`-Befehlen kann man innerhalb eines Reports Parameter erzeugen. Diese verhalten sich so, also ob sie beim Aufruf des Reports mit übergeben worden wären.

Wenn der Parameter bereits definiert war, wird durch einen `Def`-Befehl sein Wert geändert. Wird allerdings ein Parameter in einem Unter-Report (das durch einen Call-Befehl gerufen wird) definiert, so wird er mit dem Ende des Unter-Reports „vergessen“.

Einfache Variablen

Der einfache Befehl `Def:Const` definiert eine direkte Parameter-Ersetzung:

```
Def:Const [Parameter] Wert Bezeichnung
```

Durch diesen Befehl wird der Liste der bekannten Parameter der Eintrag `[Parameter]=Wert` hinzugefügt, wobei die Bezeichnung optional ist. Vor der Ausführung werden auf „Wert“ alle bisher geltenden Parameter angewandt. Hat man zum Beispiel den Parameter „[Rampe]=Z“, dann wird durch

```
Def:Const [Mode] 1-[Rampe] Prü fzustand
```

ein neuer Parameter `[Mode]` definiert, der aktuell den Wert „1-Z“ hat und die Beschreibung „Prüfzustand“ besitzt.

Trotz des Befehlsnamens „Const“ kann man dem Parameter jederzeit einen neuen Wert zuweisen (z.B. durch einen erneuten `Def:Const`-Befehl).

Numerische Parameter

Ein numerischer Parameter unterscheidet sich von einem mit `Def:Const` erzeugten Parameter dadurch, dass der Wert als Zahl verstanden wird und man ihn durch Rechenbefehle ändern kann.

```
Def:Num [Counter] 1
```

Rechenbefehle

Mit dem `Calc`-Befehl kann man den Wert eines numerischen Parameters ändern:

```
Calc:Set [Counter] 1
Calc:Add [Counter] 1
Calc:Sub [Counter] 1
Calc:Mul [Counter] 2
Calc:Div [Counter] 2
Calc:Sum [Counter] [Wert1] [Wert2]
Calc:Diff [Counter] [Wert1] [Wert2]
```

Der `Set`-Befehl setzt den Wert des Parameters. (Ein erneutes `Def:Num` hätte denselben Effekt.) `Add`, `Sub`, `Mul` und `Div` führen die betreffende Rechenoperation mit dem zweiten Wert aus. „`Calc:Mul [Counter] 2`“ multipliziert also den `[Counter]`-Wert mit 2. Statt der im Beispiel gezeigten festen Zahl 2 können natürlich auch Parameter stehen.

Die Befehls-Varianten `Sum` und `Diff` weisen dem `[Counter]` die Summe bzw. Differenz der beiden Werte zu.

Daten aus Messungen als Parameter

Es ist möglich, einen Parameter basierend auf Daten aus einer Messung zu definieren:

```
Def:Var [Parameter] Bezeichner , a num2
```

Durch diesen Befehl kann eine „Allgemeine Information“ oder ein Einzahlkennwert zu einer Parameter-Definition gemacht werden. Das Komma ist erforderlich als Platzhalter für das unbenutzte dritte Element (s.u.); das vierte Element `a` ist 3 für „Allgemeine Informationen“ oder 5

für Einzahlkennwerte. Das fünfte Argument `Num2`, der ‚Bezeichner‘ und die weiteren Argumente legen fest, welche Information gemeint ist. Ein Beispiel: der Befehl

```
Def:Var [Typname] Type , 3 3
```

deklariert einen neuen Parameter [`Typname`], dessen Wert die Bezeichnung des Typs der geladenen Messung ist (nicht „Type“, obwohl dies das zweite Element ist).

Wählen Sie für eine „Allgemeine Information“ `Num1 = 3` und folgende Werte für `Num2` und „Bezeichner“, um diese Informationen zu beschreiben:

Information	„Bezeichner“	Num2
Zeitpunkt	(irrelevant)	1
Seriennummer	(irrelevant)	2
Typname	(irrelevant)	3
Basistyp	(irrelevant)	44
Bewertungsergebnis	(irrelevant)	4
Prüfstandsname	(irrelevant)	5
Dateiname	(irrelevant)	15
Art der Messung	(irrelevant)	30
Zusatzinformation	Name der Information	6

Um eine Zusatzinformation („Additional Info“) in einen Parameter zu verwandeln, muss also der Name dieser Information angegeben werden und `Num2` den Wert 6 haben. Beispielsweise erzeugt

```
Def:Var [Übersetzung] Ratio , 3 6
```

einen Parameter der Art „[Übersetzung]=13.4“ (vorausgesetzt, dass es ein „additional Info“ namens „Ratio“ gibt, dessen Wert eine Zahl ist), der dann an anderer Stelle (etwa bei Einstellungen für eine Plot-Skalierung) weiterverwendet werden kann.

Einzahlkennwerte

Bei einem Einzahlkennwert ist der „Clavis“-Teil des Befehls (Zustand, Kanal, ...) zur Identifikation erforderlich, wohingegen der „Bezeichner“ irrelevant ist. Am besten erzeugt man auch einen solchen `Def:Var`-Befehl, indem man einen `show`-Befehl abändert. Ein Beispiel:

```
Def:Var [kmhSkalierung] , format 5 25 1-S - „Reference Speed“ DzMon
```

Welchen Wert der so definierte Parameter hat, hängt davon ab, ob man Wert, Grenze, Position oder Einheit des Einzahlkennwerts für den `Def:Var`-Befehl verwendet. (`Num2`: Wert=25, Grenze=26, Position=27, Einheit=28.) Optional kann als `format` eine `printf()`-artige Formatangabe zur Fließkommazahlen-Darstellung angegeben werden (z.B. „%5.2f“); wenn Sie darauf verzichten möchten, ersetzen Sie `format` durch ein Komma.

Vorgabewerte

Das dritte Argument im `Def:Var`-Befehl kann (statt für eine Format-Anweisung bei den Einzahlkennwerten) dafür verwendet werden, einen Vorgabe- oder Standardwert festzulegen. Falls im Archiv die Information, auf die sich der `Def:Var`-Befehl bezieht, nicht vorhanden ist, oder diese Information leer ist, wird dann der Vorgabewert verwendet. Beispiel (vergl. oben):

```
Def:Var [Übersetzung] Ratio 1.0 3 6
```

Format-Anweisungen für Einzahlkennwerte werden daran erkannt, dass sie mit einem ‚%‘ beginnen; alles andere (außer dem Komma) wird als Vorgabewert interpretiert.

Bewertungsergebnis als Parameter

Das Bewertungsergebnis der aktuellen bzw. hervorgehobenen Messung kann in einen Parameter-Wert übertragen werden (der dann z.B. für spätere Verzweigungen oder Set-Style-Befehle verwendet werden kann).

```
Def:EvalResult [Parameter] (Mode-Name)
```

Die Angabe eines Prüfzustands-Namens ist optional; wird kein Prüfzustand angegeben, dann bezieht sich der Befehl auf das Gesamtergebnis der aktuellen bzw. hervorgehobenen Messung.

Der Wert des Parameters nach diesem Befehl entspricht den Standard-Ergebniscodes: 0 = n.i.O., 1 = i.O., 2 = ohne Bewertung, 3 = Systemfehler. Der Benutzername des Parameterwertes (mit {Parameter} zu erhalten) ist die Text-Darstellung der Bewertung (also z.B. „i.O.“).

Wird ein Mode angegeben, der nicht vorhanden ist oder bei der hervorgehobenen Messung nicht gemessen wurde, dann bekommt der Parameter den Wert 2 = ohne Bewertung.

Globale und Automatische Parameter

Globale Parameter

Neben den `Def:`-Befehlen gibt es noch die Möglichkeit, im Rapport-See globale Parameter zu definieren. Diese werden in der Liste „GlobalDefs“ aufgeführt:

```
GlobalDefs: {  
  OrderTrackInstrument      "Spectral Track"  
  OrderSpectrumInstrument  "End Spectrum"      "Peak Hold Spektrum"  
  CurrentPosition          Göttingen          "Ort der Messung"  
}
```

Das erste Element jeder Zeile ist der Name des Parameters (ohne []). Das zweite Element ist der Wert, den der Parameter haben soll. (Er kann später durch `Def:`-Befehle geändert werden.) Der dritte Eintrag ist optional und liefert eine Beschreibung des Parameters. (Diese Beschreibung kann für Anzeige-Befehle verwendet werden, siehe Seite 19.)

Vordefinierte und Automatische Parameter

Es gibt einige Parameter, die automatisch vordefiniert werden und die Sie z.B. in `TextAnzeigen-` oder `If:Is>` Befehlen verwenden können:

```
[#M]                Anzahl der geladenen Messungen  
[$RapportName]     Name des aktuellen Rapports
```

Darüber hinaus können Sie Parameter definieren, deren Werte automatisch mit der Liste aller Prüfzustände, Kanäle, Typ-Bezeichnungen oder anderen Informationen gefüllt werden. Dies geschieht mit dem `Def:Auto`-Befehl, z.B.:

```
Def:Auto $Mode Prüfzustand , 0 0
```

oder

```
Def:Auto [Typname] Aggregate-Typ Type 3 3
```

Durch ersteren Befehl wird ein Parameter `[$Mode]` erzeugt, dessen Werte-Liste automatisch alle in den Daten vorkommenden Prüfzustände umfasst. Das im `Def:Auto`-Befehl stehende „Prüfzustand“ wird im Parameter-Auswahl-Dialog, der beim Start des Rapports erscheint, als Bezeichnung der Auswahlliste verwendet. Das Schlüsselwort „`$Mode`“ ist vordefiniert für die Prüfzustände; analog gibt es „`$Channel`“ für die Liste aller Kanäle.

Der zweite Beispiel-Befehl erzeugt einen Parameter `[Typname]`, dessen Werte-Liste alle vorkommenden Typen umfasst. Mit dieser Art von `Def:Auto`-Befehlen können Sie ganz analog zu den `Def:Var`-Befehlen (siehe oben) einen Parameter basierend auf einer Zusatzinformation

erzeugen. Beachten Sie einen kleinen Unterschied in der Syntax: beim `Def:Var`-Befehl ist das zweite Element der Name der Zusatzinformation und das dritte Element bleibt leer (ein Komma); beim `Def:Auto`-Befehl ist das zweite Element die Beschreibung des Parameters (Bezeichnung für die Auswahlliste) und das dritte Element der Name der Information.

Auch für die Automatischen Parameter gilt, dass sie im Parameter-Auswahl-Dialog nur dann auftauchen, wenn es mehr als einen Wert auszuwählen gibt; ansonsten nehmen sie automatisch den einen möglichen Wert an.

Der `Def:Auto`-Befehl kann im Haupt-Rapport oder einem von diesem gerufenen Unter-Rapport stehen. Anders als beim `Def:Var`-Befehl sind die `Def:Auto`-Parameter unabhängig davon, wo sie stehen, von Beginn des Reports an definiert. (Tatsächlich werden sie durch die Präsentations-App sogar schon vor Beginn der Ausführung des Reports angelegt.)

Weitere Befehle

Durch die Show-Befehle wird im Hintergrund das Fenster für die Zuordnung von Grafikmodulen geöffnet. Normalerweise bleibt dieses während der Ausführung des Reports unsichtbar (siehe die Option „Bei Wiedergabe Dialoge unsichtbar machen“ in der Abteilung Rapport des Bedienfensters; vergl. Seite 4). Am Ende des Reports aber wird es wieder sichtbar. Daher will man normalerweise, dass dieses Fenster am Ende des Reports geschlossen wird. Dies erreicht man mit

```
GraphMod:Schliessen  
GraphMod:OnOK
```

(Sie können auch „Schließen“ statt „Schliessen“ schreiben.) Möglicherweise möchten Sie das Bedienfenster aber am Ende des Reports explizit wieder anzeigen lassen. Dies erreichen Sie mit

```
ArchivPresent:DialogZeigen Abteilung
```

Mit der „Abteilung“ wird festgelegt, welche Seite des Fensters angezeigt werden soll. Die möglichen Werte sind „Liste“, „Gruppen“, „Archive“, „Rapport“. Wenn Sie keinen Parameter angeben, wird das Fenster mit der zuletzt verwendeten Abteilung geöffnet.

Fehlersuche

Wenn ein Rapport nicht so arbeitet, wie Sie es erwarten, können Sie im Rapport Befehle der Art

```
Stdout:...
```

verwenden. Sie werden wie Kommentare behandelt, also nicht ausgeführt. Stattdessen wird die ganze Zeile im Ausgabefenster ausgegeben. Mit Hilfe des `stdout`-Befehls können Sie beispielsweise die Ersetzung von Parametern beobachten: duplizieren Sie einfach den Befehl, den Sie untersuchen möchten, und ersetzen Sie den Anfang durch `stdout`:

```
Show:Messkurven Layout-[Layout] Plot0 0 4 8 [Mode] LsSk Ls_[Source]_H1  
Stdout:Messkurven Layout-[Layout] Plot0 0 4 8 [Mode] LsSk Ls_[Source]_H1
```

Dann erfahren Sie, wie die Präsentation ihre Parameter interpretiert.

Kommentare

Kommentare: Sie können im Rapport Kommentar-Zeilen einfügen. Jede Zeile, deren Befehl mit zwei Schrägstrichen `//` beginnt, wird als Kommentar betrachtet:

```
// "Dies ist ein Kommentar."
```

Beachten Sie, dass die Zeile trotzdem in ihre Teile zerlegt wird. Damit nicht irgendwelche Wörter ihres Kommentars mit den Befehls-Elementen kollidieren, sollten Sie ihren Kommentar immer in Anführungszeichen einschließen, wie im Beispiel gezeigt.

Beispiele

Dieses Kapitel stellt einige kommentierte Beispiel-Rapports vor, um die Verwendung der Befehle zu illustrieren. Gezeigt wird jeweils die Darstellung im `Rapport.sea`, wobei die Bearbeitung von Rapports und Parametern je nach Situation mit dem Texteditor oder dem Rapport-Editor der Präsentations-App bequemer sein kann.

Basis-Anzeige-Rapport

Dieses Rapport zeigt einfach die Spektren zu vier Prüfzuständen in entsprechenden Grafikmodulen an:

```
Fix-Spectra: {
  Def:Const [Layout] FixSpectra
  Layouts:OnWipe [Layout]
  Layouts:OnForeground [Layout]
  Show:Zeitpunkt [Layout] GDateTextBox0 0 3 1
  Show:Spectra [Layout] Plot1 , 4 7 Drive "Base (FixFs,[Sensor])" , Spectrum
  Show:Spectra [Layout] Plot2 , 4 7 Steady "Base (FixFs,[Sensor])" , Spectrum
  Show:Spectra [Layout] Plot3 , 4 7 Coast "Base (FixFs,[Sensor])" , Spectrum
  Show:Spectra [Layout] Plot4 , 4 7 Runout "Base (FixFs,[Sensor])" , Spectrum
  GraphMod:OnOK
}
Fix-Spectra_Params: {
  Sensor Sensor Mic
}
Fix-Spectra_Param_Sensor: {
  VS Vibration
  Mic Microphone
} ;-- Fix-Spectra_Param_Sensor
```

Zunächst wird der Name des Layouts in den Parameter `[Layout]` verwandelt. In diesem einfachen Rapport könnte auch jeweils direkt der Layout-Name verwendet werden, doch falls später z.B. Unter-Rapports hinzugefügt werden (siehe die Beispiele unten), ist es nützlich, wenn der Layout-Name schon als Parameter vorbereitet ist.

Durch die `Layouts:OnWipe` und `Layouts:OnForeground`-Befehle wird die Layout-Seite vorbereitet. Dann wird die „allgemeine Information“ des Messzeitpunkts in einem Textfeld angezeigt. Vier `Show`-Befehle stellen die Spektren zu vier verschiedenen Prüfzuständen („Drive“, „Steady“ usw.) in vier Plots dar (vergl. die Beschreibung des Show-Befehls auf Seite 16). Schließlich wird mit `GraphMod:OnOK` das Fenster zur Grafikmodul-Zuordnung geschlossen, das während der Rapport-Ausführung im Hintergrund geöffnet wird.

Das Rapport besitzt einen Parameter `[Sensor]` mit zwei möglichen Werten „VS“ und „Mic“ – siehe die Beschreibung zu Parametern im `Rapport.sea` auf Seite 11.

Verwendung von Unter-Rapports

Verzweigung und Schleife

Spektren und Ordnungsmarkierungen

In Ordnungsspektren kann man Markierungen für die Ordnungsgeber einzeichnen lassen. Diese Markierungen erscheinen als gestrichelte Linien im Kurvenplot.

Index

AlleSchliessen (Layouts)	18	Maximieren (Layout)	18
AlleWischen-Befehl	18	Maximize (Layout)	18
ArchivEntfernen-Befehl	19	Minimieren (Layout)	18
ArchivLaden-Befehl	19	Minimize (Layout)	18
Ausgabefenster	5, 30	Numerische Parameter	27
Automatische Parameter	29	OnAlleDrucken-Befehl	18
Bedingte Ausführung	23	OnClose (Layout)	18
Befehls-Assistent	13	OnDatenAnzeigen-Befehl	17
BewertungIO-Bedingung	25	OnDrucken (Layout)	18
Bewertungsergebnis		OnForeground (Layout)	18
als Parameter	29	OnGraphAuswahl-Befehl	17
Verzweigung (if)	25	OnHinzunehmen-Befehl	17
Bewertungsergebnis (Bedingung)	25	OnPrint (Layout)	18
Bezeichnungen von Parametern anzeigen	19	OnPrintAll-Befehl	18
Call-Befehl	22	OnSchliessen	18
Close (Layout)	18	OnTafelWischen-Befehl	17
CloseAll (Layout)	18	OnVordergrund	18
Darstellungsoptionen setzen	17	OnWipe (Layout)	18
DatenGrenzeNio-Bedingung	25	OnWischen (Layout)	18
DatenTextGleich (Bedingung)	25	OptionSetzen-Befehl	17
DatenVorbereiten-Befehl	17	Ordnungsgeber Markierung	32
Daten-Werte prüfen (Bedingung)	24	Plot-Stil ändern	20
Def:Auto	29	Rechteck-Stil ändern	21
Def:Const	27	RemoveArchive-Befehl	19
Def:Num	27	Return-Befehl	22
Def:Var	27, 29	Schliessen-Befehl (Layout)	18
Def-Befehle	27	SetPlotStyle-Befehl	20
DisplayText-Befehl	18	SetRectStyle-Befehl	21
Else (Bedingung)	25	Show-Befehl	16
GlobalDefs	29	ShowIfData-Befehl	16, 24
Globale Parameter	14, 29	Sonst (Bedingung)	25
Goto-Befehl	22	Sprungbefehle	22
GraphMod:Schliessen	30	TextAnzeigen-Befehl	18
Größer-Kleiner-Vergleiche	23	Textbox-Farbe	18
If:Data-Bedingung	24	Textvergleich (Bedingung)	23
If:DataLimitNok-Bedingung	25	Vergleich mit mehreren Werten	23
If:Def-Bedingung	24	Verneinung (Bedingung)	23
If:Is< Bedingung	23	Wenn:Daten-Bedingung	24
If:IsEqual-Bedingung	23	Wenn:DatenTextGleich-Bedingung	24
If-Befehle	23	Wenn:Definiert-Bedingung	24
Kommentare	30	Wenn:IstGleich-Bedingung	23
Label-Befehl	22	Wenn-Befehle	23
LoadArchive-Befehl	19	WipeAll-Befehl	18