

# CALENDAR V17

Kalender erstellen und bearbeiten

## Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an [manuals@ts.fujitsu.com](mailto:manuals@ts.fujitsu.com) senden.

## Zertifizierte Dokumentation nach DIN EN ISO 9001:2000

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2000 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH  
[www.cognitas.de](http://www.cognitas.de)

## Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions GmbH 2010.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.



Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions. Das vorliegende Dokument ist eine fachlich ergänzte Neuausgabe eines früheren Handbuchs zu einer bereits vor längerer Zeit freigegebene Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter <http://de.ts.fujitsu.com/>

---

# Inhalt

<b>1</b>	<b>Einleitung</b> . . . . .	<b>7</b>
<b>1.1</b>	<b>Zielgruppen des Handbuchs</b> . . . . .	<b>8</b>
<b>1.2</b>	<b>Konzept des Handbuchs</b> . . . . .	<b>8</b>
<b>2</b>	<b>Aufbau und Installation</b> . . . . .	<b>11</b>
<b>2.1</b>	<b>Der Aufbau eines Kalenders</b> . . . . .	<b>11</b>
2.1.1	Basis-Informationen . . . . .	11
2.1.2	Kalendertage . . . . .	13
2.1.3	Symbolische Datumsangaben: SYMDAT . . . . .	14
2.1.4	Feiertage . . . . .	17
<b>2.2</b>	<b>Kalenderdateien</b> . . . . .	<b>19</b>
2.2.1	Namenskonvention für Kalenderdateien . . . . .	19
2.2.2	Schutzmechanismen für Kalenderdateien . . . . .	19
2.2.3	Dateiattribute . . . . .	20
<b>2.3</b>	<b>Installation</b> . . . . .	<b>20</b>
2.3.1	Bestandteile . . . . .	20
2.3.2	Installation des Subsystems . . . . .	21
<b>3</b>	<b>Dialogschnittstelle: Der Kalender-Editor</b> . . . . .	<b>23</b>
<b>3.1</b>	<b>Funktionsübersicht</b> . . . . .	<b>23</b>
<b>3.2</b>	<b>Bedienung des Kalender-Editors</b> . . . . .	<b>24</b>
3.2.1	Starten und Beenden . . . . .	25
3.2.2	Maskenübersicht . . . . .	27
3.2.3	Allgemeiner Maskenaufbau . . . . .	28
3.2.4	Die Editor-Funktionen . . . . .	29

3.2.5	Beschreibung der Masken . . . . .	33
3.2.5.1	Maske C000: Main Hauptmenü des Kalender-Editors . . . . .	34
3.2.5.2	Maske C010: Edit Auswahl der Funktionen . . . . .	38
3.2.5.3	Maske B020: Basic Information Basis-Informationen ausgeben und bearbeiten . . . . .	43
3.2.5.4	Maske D010: List of Days Kalendertage auflisten und zur Auswahl markieren . . . . .	47
3.2.5.5	Maske D020: Day Information Kalendertage, d.h. Tagesinformationen bearbeiten . . . . .	50
3.2.5.6	Maske S010: List of Symbolic Dates SYMDAT-Liste ausgeben / zur Auswahl markieren . . . . .	54
3.2.5.7	Maske S020: Symbolic Date Information SYMDAT-Informationen ausgeben und bearbeiten. . . . .	57
3.2.5.8	Maske H010: List of Holidays Feiertage auflisten und zur Auswahl markieren . . . . .	62
3.2.5.9	Maske H020: Holiday Information Feiertags-Informationen ausgeben und bearbeiten. . . . .	66
<b>4</b>	<b>Programmschnittstelle . . . . .</b>	<b>71</b>
<b>4.1</b>	<b>Formatübersicht . . . . .</b>	<b>73</b>
<b>4.2</b>	<b>Operandenbeschreibung . . . . .</b>	<b>76</b>
<b>4.3</b>	<b>Datenbereiche für Funktionen . . . . .</b>	<b>83</b>
<b>4.4</b>	<b>Returncodes . . . . .</b>	<b>86</b>
<b>4.5</b>	<b>Beispiel . . . . .</b>	<b>90</b>
<b>4.6</b>	<b>Einzelformate . . . . .</b>	<b>97</b>
4.6.1	*AKTHOL Feiertag aktivieren . . . . .	98
4.6.2	*CLSECAL Kalender schließen . . . . .	100
4.6.3	*CREACAL Kalender erzeugen . . . . .	102
4.6.4	*CRECHOL Zyklischen Feiertag erzeugen . . . . .	106
4.6.5	*CRECSYM Zyklisches SYMDAT erzeugen . . . . .	108
4.6.6	*CRENHOL Nicht-zyklischen Feiertag erzeugen . . . . .	114
4.6.7	*CRENSYM Nicht-zyklisches SYMDAT erzeugen . . . . .	117
4.6.8	*DEAKHOL Feiertag deaktivieren . . . . .	121
4.6.9	*DELHOL Feiertag löschen . . . . .	123
4.6.10	*DELSYM SYMDAT löschen . . . . .	125
4.6.11	*MODBAS Basis-Information ändern . . . . .	127
4.6.12	*MODCHOL Zyklischen Feiertag ändern . . . . .	131

4.6.13	*MODDAY	Kalendertag ändern . . . . .	133
4.6.14	*MODCSYM	Zyklisches SYMDAT ändern . . . . .	137
4.6.15	*MODNHOL	Nicht-zyklischen Feiertag ändern . . . . .	142
4.6.16	*MODNSYM	Nicht-zyklisches SYMDAT ändern . . . . .	146
4.6.17	*OPENCAL	Kalender öffnen . . . . .	151
4.6.18	*SAVECAL	Kalender sichern . . . . .	154
4.6.19	*SHBAS	Basis-Information ausgeben . . . . .	156
4.6.20	*SHDAYHL	Kalendertag mit zugeordneten Feiertagen ausgeben . . . . .	159
4.6.21	*SHDAYIN	Kalendertag mit zugeordneten SYMDATs ausgeben . . . . .	163
4.6.22	*SHHOLIN	Feiertag mit zugeordneten Kalendertagen ausgeben . . . . .	168
4.6.23	*SHLODAY	Liste von Kalendertagen ausgeben . . . . .	173
4.6.24	*SHLOHOL	Liste von Feiertagen ausgeben . . . . .	178
4.6.25	*SHLOSYM	Liste von SYMDAT ausgeben . . . . .	182
4.6.26	*SHSYMIN	SYMDAT mit zugeordneten Kalendertagen ausgeben . . . . .	187
4.6.27	*SHNESTM	Nächsten Zeitpunkt zu SYMDATs ausgeben . . . . .	192
<b>4.7</b>	<b>Parameterliste</b>	. . . . .	<b>197</b>
<b>4.8</b>	<b>Makro-Syntax</b>	. . . . .	<b>202</b>
4.8.1	<b>Makroaufrufformat</b>	. . . . .	202
4.8.2	Elemente der Metasyntax	. . . . .	203
4.8.3	Datentypen der Operandenwerte	. . . . .	204
4.8.4	Zusätze zu Datentypen	. . . . .	204
4.8.5	Datentypen der Variablen	. . . . .	205
4.8.6	MF-Operand	. . . . .	205
<b>4.9</b>	<b>C-Headerfile (calendr.h)</b>	. . . . .	<b>208</b>
<b>5</b>	<b>Kommandoschnittstelle</b>	. . . . .	<b>221</b>
<hr/>			
	SHOW-CALENDAR		
	Informationen aus einer Kalenderdatei anfordern . . . . .		221
	Funktionsbeschreibung . . . . .		221
	Format . . . . .		222
	Operandenbeschreibung . . . . .		223
	Ausgabeformate . . . . .		229
	<b>Literatur</b>	. . . . .	<b>235</b>
	<b>Stichwörter</b>	. . . . .	<b>237</b>
<hr/>			



---

# 1 Einleitung

Mit dem Subsystem CALENDAR wird für das BS2000 eine zentrale Kalenderfunktion angeboten.

Das Kalendersystem erlaubt dem Anwender, einen neuen Kalender nach Maßgabe seiner zu verarbeitenden Daten zu erzeugen oder vorhandene Kalender zu lesen bzw. zu bearbeiten.

Ein Kalender ist eine Datei (Kalenderdatei), die für einen festgelegten Zeitraum (Kalendergrenzen) Informationen über Arbeitstage, Arbeitszeiten, freie Tage, Feiertage, zusätzliche symbolische Datumsangaben (SYMDAT) etc. enthält.

CALENDAR stellt grundlegende Strukturen wie z.B. eine Standard-Arbeitswoche und die fortlaufende Erzeugung der Tage im festgelegten Zeitraum zur Verfügung. Ebenso können die Feiertage für die Kalenderdatei einer sogenannten Feiertagsdatei entnommen werden.

Bei der Definition von Kalenderdaten durch den Anwender, der eine Kalenderdatei erzeugt oder verändert, handelt es sich im wesentlichen um

- Zuordnung von Informationen zu Kalendertagen
- Definition von Feiertagen
- Definition von symbolischen Datumsangaben (SYMDAT).

Über drei verschiedene Schnittstellen kann der Anwender auf die Daten einer Kalenderdatei zugreifen:

- Über die **Dialogschnittstelle** können mit dem Kalender-Editor die Daten für eine Kalenderdatei erzeugt, geändert oder ausgegeben werden. Die Bearbeitung erfolgt über FHS-Masken.
- Über die **Programmschnittstelle** können die Daten einer Kalenderdatei auf Programmebene (Assembler, C) erzeugt, bearbeitet und ausgegeben werden.
- Über die **Kommandoschnittstelle** können Kalenderdaten zur Ausgabe angefordert werden.  
Die Ausgabe erfolgt nach SYSOUT / SYSLST, falls der Anwender es wünscht auch in S-Variablen. So können die Daten aus Kalenderdateien auch in SDF-P-Prozeduren verwendet werden.

## 1.1 Zielgruppen des Handbuchs

Das vorliegende Handbuch wendet sich an die BS2000-Systembetreuung und alle Anwender, die eine zentrale Kalenderfunktion nutzen wollen.

## 1.2 Konzept des Handbuchs

Das Handbuch besteht aus fünf Kapiteln mit folgendem Inhalt:

### Kapitel 1, *Einleitung*

enthält eine Kurzbeschreibung des Subsystems CALENDAR, nennt die angesprochenen Zielgruppen und erläutert den Aufbau des Handbuchs.

### Kapitel 2, *Aufbau und Installation*

informiert über Struktur und Elemente eines Kalenders und gibt Auskunft über Namenskonvention, Schutzmechanismen und Dateiattribute einer Kalenderdatei. Anschließend werden Hinweise zur Installation gegeben und die zum Einsatz von CALENDAR benötigten Dateien genannt.

### Kapitel 3, *Dialogschnittstelle: Der Kalender-Editor*

beschreibt das Dienstprogramm „Kalender-Editor“, mit dem - maskengesteuert - Kalenderdaten von existierenden Kalendern gelesen und verändert sowie neue Kalender angelegt werden können.

### Kapitel 4, *Programmschnittstelle*

enthält eine ausführliche Beschreibung des Makros CALENDR sowie die Parameterliste und ein kommentiertes C-Headerfile. Im Anschluss an die Formatübersicht des Makros mit Operandenbeschreibung und einer Auflistung der Returncodes werden alle Einzelformate, nach Funktionsaufrufen alphabetisch geordnet, beschrieben. Weiter enthält das Kapitel eine Beschreibung der Makrosyntax und ein Beispiel zum Makro CALENDR.

### Kapitel 5, *Kommandoschnittstelle:*

beschreibt das Kommando SHOW-CALENDAR, mit dem Informationen aus dem angegebenen Kalender zur Ausgabe angefordert werden. Die wichtigsten Ausgabeformate sind abgebildet.

Am Ende des Handbuchs finden Sie ein Literatur- und Stichwortverzeichnis.



## Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Namen SYSRME.CALENDAR.<version>.D. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei können Sie mit dem Kommando /SHOW-FILE oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-DOCUMENT SYSRME.CALENDAR.<version>.D, LINE-SPACING=*BY-EBCDIC-CONTROL
```



---

## 2 Aufbau und Installation

### 2.1 Der Aufbau eines Kalenders

Die Informationsstruktur in einem Kalender wird gebildet durch vier grundlegende Elemente:

- Basis-Informationen (Kalendergrenzen und Standard-Arbeitswoche)
- Kalendertage
- Symbolische Datumsangaben (SYMDAT)
- Feiertage

#### 2.1.1 Basis-Informationen

Die Definition der Basis-Informationen „Kalendergrenzen“ und „Standard-Arbeitswoche“ ist notwendige Voraussetzung für alle weiteren Schritte bei der Erzeugung eines Kalenders.

##### Die Kalendergrenzen

Die zeitliche Begrenzung eines Kalenders wird gebildet durch den ersten Tag des Kalenders als Untergrenze und den letzten Tag des Kalenders als Obergrenze.

Zwischen dem ersten Tag und dem letzten Tag eines Kalenders dürfen maximal fünf Jahre liegen, d.h. Kalender, die sich über einen längeren Zeitraum als fünf Jahre erstrecken, sind nicht zulässig. Zusätzlich müssen die Kalendergrenzen zwischen dem 1. Januar 1970 (1970-01-01) und dem 31. Dezember 2030 (2030-12-31) liegen.

Unter Berücksichtigung dieser Einschränkungen können Kalendergrenzen jederzeit geändert werden. Dabei ist folgendes zu beachten:

Das Verschieben der Kalendergrenzen muss überlappend erfolgen, d.h. das neue Datum für Kalenderbeginn muss  $\leq$  dem alten Datum für Kalenderende sein und das neue Datum für Kalenderende muss  $\geq$  dem alten Datum für Kalenderbeginn sein. Andernfalls ist nicht sichergestellt, dass für die Berechnung der zyklischen symbolischen Datumsangaben (SYMDAT, Beschreibung siehe [Seite 14ff.](#)) eine definierte Berechnungsgrundlage vorhanden ist. Beim Verschieben der Kalendergrenzen durch den Anwender gehen diejenigen Informationen verloren, die nach der Verschiebung außerhalb der neuen Kalendergrenzen

liegen. Auch wenn die Änderungen wieder rückgängig gemacht werden, d.h. die Kalendergrenzen auf den ursprünglichen Wert zurückgesetzt werden, sind diese Informationen verloren.

### Die Standard-Arbeitswoche

Innerhalb der vom Anwender festgelegten Kalendergrenzen werden die einzelnen Tage von CALENDAR erzeugt und mit Initialwerten versorgt. Diese Initialwerte werden global in der Standard-Arbeitswoche definiert und gelten für alle Kalendertage, die nicht explizit abweichend definiert wurden. Auch eine Änderung der Werte in der Standard-Arbeitswoche ist für alle Kalendertage gültig, ausgenommen diejenigen, für die explizit eine abweichende Definition erfolgt ist.

Die Standard-Arbeitswoche enthält die Festlegungen

- der Wochentage
- des Attributes eines Wochentages
- der Arbeitszeit eines Wochentages.

**Wochentage** sind die Tage von Montag bis Sonntag.

Die Benennung der Wochentage erfolgt in englischer Sprache und kann ebensowenig geändert werden wie die Reihenfolge der Wochentage:

Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

Das **Attribut** eines Wochentages gibt an, ob der jeweilige Wochentag standardmäßig ein Arbeitstag oder ein freier Tag ist.

*Voreinstellung:*

Monday, Tuesday, Wednesday, Thursday, Friday: Arbeitstage.  
Saturday, Sunday: freie Tage.

Die **Arbeitszeit** eines Wochentages, definiert durch die beiden Werte Beginn und Ende, wird in Stunden und Minuten angegeben.

*Beispiel*

08:30 - 17:30

Zulässige Kombinationen für die Angabe von Beginnzeit und Endezeit sind:

- Beginnzeit = Endezeit
- Beginnzeit < Endezeit
- Beginnzeit > Endezeit (z.B. für Nachtschicht).

- CALENDAR prüft nur, ob die Angaben sinnvolle Uhrzeit-Angaben sind, d.h. ob die Stundenangaben im zulässigen Bereich 00 bis 23 und die Minutenangaben im zulässigen Bereich 00 bis 59 liegen.

*Voreinstellung:*

00:00 - 23:59

## 2.1.2 Kalendertage

Innerhalb der definierten Kalendergrenzen legt CALENDAR die Kalendertage in lückenloser Reihenfolge an. Die Anzahl der für eine Kalenderdatei angelegten Kalendertage hängt also ausschließlich von den angegebenen Kalendergrenzen ab. Vom Anwender können Kalendertage weder erzeugt noch gelöscht werden.

Ein Kalendertag ist ein Wochentag, der innerhalb der Kalendergrenzen eines Kalenders liegt.

Die Initialwerte für Attribut (Arbeitstag / freier Tag) und Arbeitszeit eines Kalendertags werden in der Standard-Arbeitswoche definiert.

Beide Werte können vom Anwender durch explizite Angabe modifiziert werden.

Für die letztgültige Definition eines Kalendertages als Arbeitstag oder als freier Tag ist zusätzlich die Festlegung der Feiertage zu berücksichtigen.

Stimmen die Angaben für den Attribut-Wert eines Kalendertags aus Standard-Arbeitswoche, Festlegung der Feiertage und explizite Angabe eines Attributwertes nicht überein, wird der letztgültige Wert nach folgender **Attribut-Regel** ermittelt:

1. Wurde das Attribut (Arbeitstag / freier Tag) für einen Kalendertag explizit angegeben, hat diese Definition immer Vorrang.

*Beispiel:*

Ein Kalendertag, der als Arbeitstag definiert wurde, bleibt auch dann Arbeitstag, wenn er gleichzeitig als Feiertag definiert ist („Feiertage“ siehe unten [Seite 17ff.](#)).

2. Wurde für einen Kalendertag
  - kein Attribut-Wert explizit angegeben oder dieser wieder zurückgesetzt (STD),
  - dieser Kalendertag als Feiertag definiert,wird die Feiertagsdefinition wirksam. Der Kalendertag ist damit ein freier Tag.
3. Wurde für einen Kalendertag
  - kein Attribut-Wert explizit angegeben oder dieser wieder zurückgesetzt
  - keine Feiertagsdefinition eingetragen,wird der Attribut-Wert für den Kalendertag der Standard-Arbeitswoche entnommen.

*Hinweis*

Wird der Attribut-Wert (Arbeitstag / freier Tag) für einen Kalendertag geändert, hat dies Auswirkungen auf die Berechnung der zyklischen symbolischen Datumsangaben (SYMDAT) in einem Kalender (Beschreibung siehe folgendes Kapitel).

### 2.1.3 Symbolische Datumsangaben: SYMDAT

Einer symbolischen Datumsangabe, kurz SYMDAT, werden Kalendertage eines Kalenders zugeordnet.

Ein SYMDAT bietet die Möglichkeit, logisch zusammengehörige Tage unter einem Namen zusammenzufassen und über diesen Namen zu verwenden.

*Beispiel*

ULTIMO.

Mit dem SYMDAT „ULTIMO“ kann die Gruppe des jeweils letzten Kalendertages eines jeden Monats innerhalb der definierten Kalendergrenzen zu einer logischen Einheit zusammengefasst und mit diesem Namen angesprochen werden.

Jedem SYMDAT kann eine **Zeitangabe** zugeordnet werden.

Diese Zeitangabe gilt dann für alle Kalendertage, die dem SYMDAT zugeordnet sind.

*Voreinstellung :*

00:00:00.

#### Namenskonvention für ein SYMDAT

Maximale Länge eines SYMDAT-Namens:

20 Zeichen.

Erlaubte Zeichen in einem SYMDAT-Namen:

„0“ . . „9“, „A“ . . „Z“, „#“, „.“, „\$“.

Einschränkungen:

„.“ (Punkt) darf weder an erster noch an letzter Stelle stehen.

Zwei aufeinanderfolgende Punkte sind nicht erlaubt.

An erster Stelle eines SYMDAT-Namens sind Großbuchstaben („A“ . . „Z“) und die Zeichen „#“ oder „\$“ erlaubt.

#### Zyklische und nicht-zyklische SYMDAT

Symbolische Datumsangaben sind zu unterscheiden nach zyklischen und nicht-zyklischen SYMDAT.

**Nicht-zyklisch** heißen diejenigen SYMDAT, deren zugehörige Datumsangaben, d.h. Kalendertage, nicht berechnet werden können, da die Intervalle zwischen den Einträgen nicht bekannt sind.

Für ein nicht-zyklisches SYMDAT müssen die gewünschten Kalendertage, jeder einzeln, explizit zugeordnet werden.

**Zyklisch** heißen diejenigen SYMDAT, für die die Abstände zwischen den zugeordneten Kalendertagen berechnet werden können, da sich die Intervalle zwischen den Kalendertagen zyklisch wiederholen.

Für zyklische SYMDAT werden nicht die zugeordneten Kalendertage definiert, sondern

- das Start-Datum
- die Zyklus-Art: Monats-Zyklus, Wochen-Zyklus, Arbeitstag-Zyklus, Tag-Zyklus
- der Zyklus-Wert: Anzahl der zu addierenden Monate / Wochen / Arbeitstage / Tage
- die Free-Day-Rule (d.h. die Alternative für die Tage, die gemäß Zyklus einem SYMDAT zuzuordnen wären, aber als freie Tage definiert sind): AFTER / BEFORE / SKIP / ON.

Aus diesen vier Informationen lässt sich für einen Kalender ein SYMDAT vollständig berechnen, d.h. alle betroffenen Kalendertage können dem zyklischen SYMDAT zugeordnet werden.

**Free-Day-Rule** bezeichnet die vom Anwender festzulegende Alternative, die bestimmt, wie im festgelegten Zyklus zu verfahren ist, wenn ein berechnetes SYMDAT auf einen freien Tag fällt. Der Anwender, der ein zyklisches SYMDAT festlegt, kann hier unter vier Möglichkeiten wählen:

- BEFORE (vor dem berechneten Tag eintragen).  
Dabei wird in dem Bereich zwischen dem berechneten Datum als Ausgangspunkt und dem Datum des vorherigen Eintrags + 1 (also rückwärts) der nächste Arbeitstag gesucht. Wird in diesem Zeitraum kein Arbeitstag gefunden, entfällt der Eintrag.
- AFTER (nach dem berechneten Tag eintragen).  
Dabei wird in dem Bereich zwischen dem berechneten Datum als Ausgangspunkt und dem Datum des nächsten Eintrags - 1 (also vorwärts) der nächste Arbeitstag gesucht. Wird in diesem Zeitraum kein Arbeitstag gefunden, entfällt der Eintrag.
- SKIP (überspringen).  
Der Eintrag für diesen Tag entfällt.
- ON (eintragen).  
Der Eintrag erfolgt unbedingt an dem berechneten Tag.

Vier verschiedene **Zyklen** werden angeboten:

### **Der Monats-Zyklus**

Das SYMDAT wird ab dem Start-Datum für jeden Monat am gleichen Tag berechnet. Zusätzlich kommt die Free-Day-Rule in Anwendung.

Fällt ein berechneter Kalendertag für ein SYMDAT auf ein Datum, das nicht vorhanden ist, (z.B. 29. und 30. Februar, 31. April ), wird als Ersatzwert der letzte Tag des Monats angenommen.

### **Der Wochen-Zyklus**

Das SYMDAT wird ab dem Start-Datum durch Addition von sieben Kalendertagen berechnet. Zusätzlich kommt die Free-Day-Rule in Anwendung.

Würde der Eintrag aufgrund der Free-Day-Rule mit dem vorherigen oder dem nächsten Eintrag kollidieren, fällt dieser Eintrag weg.

#### *Beispiel*

Start-Datum: 5.3.1993

Zyklus: Woche

Free-Day-Rule: BEFORE

Freie Tage: 6.3.1993 bis 15.3.1993

Die Berechnung des SYMDAT führt zu folgenden Einträgen:

Der erste Eintrag erfolgt für den 5.3.1993 (Start-Datum). Der nächste Eintrag wäre am 12.3.1993 (Wochen-Zyklus). Da dies ein freier Tag ist, wird der nächste davor liegende Arbeitstag gesucht (Free-Day-Rule). Da innerhalb des Zeitraums einer Woche kein Arbeitstag gefunden wird (Freie Tage), entfällt dieser Eintrag. Die nächsten Einträge erfolgen für den 19.3.1993 und 26.3.1993.

### **Der Arbeitstag-Zyklus**

Zur Berechnung des SYMDAT wird zunächst ab dem angegebenen Start-Datum der erste Arbeitstag gesucht. Der vom Anwender anzugebende Zyklus-Wert legt fest, wie viele Arbeitstage jeweils auf einen Eintrag addiert werden, um den nächsten Eintrag vorzunehmen bis zur Kalenderobergrenze.

Da für die Berechnung nur Arbeitstage berücksichtigt werden, kommt die Free-Day-Rule für den Arbeitstag-Zyklus nicht zur Anwendung.

### **Der Tag-Zyklus**

Der erste Eintrag ist das Start-Datum. Zu diesem Datum wird der Zyklus-Wert addiert. Der so berechnete Wert ist der nächste Eintrag, falls die Free-Day-Rule für einen eventuell berechneten freien Tag nicht etwas anderes festlegt.



## 2.1.4 Feiertage

Ein Feiertag ist ein Name, dem Kalendertage zugeordnet werden. Wird der Feiertag aktiviert, werden alle zugeordneten Kalendertage zu freien Tagen, außer er wurde explizit als Arbeitstag definiert (vergl. Beschreibung der Attribut-Regel oben auf [Seite 13f.](#))

### Namenskonvention für einen Feiertag

Maximale Länge eines Feiertags-Namens:  
30 Zeichen.

Erlaubte Zeichen in einem Feiertags-Namen:  
„0“ . . „9“, „A“ . . „Z“, „#“, „-“, „\$“.

Einschränkungen:

„.“ (Punkt) darf weder an erster noch an letzter Stelle stehen.

Zwei aufeinanderfolgende Punkte sind nicht erlaubt.

An erster Stelle eines Feiertags-Namens sind Großbuchstaben („A“ . . „Z“) und die Zeichen „#“ oder „\$“ erlaubt.

### Feiertags-Typen

Feiertage sind zu unterscheiden in zyklische Feiertage und nicht-zyklische Feiertage.

Zyklische Feiertage kehren jährlich zum gleichen Datum wieder, wie z.B. Neujahr.

Die Zuordnung Feiertag - Kalendertag nimmt bei zyklischen Feiertagen CALENDAR selbst vor.

Nicht-zyklischen Feiertagen mit wechselnden Tages- und Monatsdaten, z.B. Ostern, müssen die entsprechenden Kalendertage eines jeden Jahres im Bereich der Kalendergrenzen explizit zugeordnet werden.

### Die Aktivierung von Feiertagen

Ein definierter Feiertag wird erst dann als solcher wirksam, d.h. ein ihm zugeordneter Kalendertag wird erst dann zum freien Tag, wenn er als Feiertag aktiviert ist.

Der Anwender kann dies für jeden Feiertag nach seinen Wünschen festlegen (siehe Maske H010 und Beschreibung auf [Seite 64](#)).

Beim Erzeugen eines Feiertags wird dieser automatisch aktiviert. Die vordefinierten Feiertage sind aktiviert.

## Die Feiertag-Datei

Um dem Anwender die Definition der Feiertage für den Geltungsbereich jeder seiner Kalenderdateien zu ersparen, wird mit CALENDAR eine Feiertag-Datei ausgeliefert. Diese enthält vordefinierte Feiertage, die beim Erzeugen eines Kalenders automatisch in die Kalenderdatei des Anwenders übernommen werden.

Die Bearbeitung der Datei durch den Anwender erfolgt mit dem Standard-Editor EDT.

## Dateiattribute

- Dateiname: SYSDAT.CALENDAR.ver.HOLIDAY
- FCBTYPE = SAM
- RECFORM = V

## Dateiaufbau

Die Sätze der Feiertag-Datei haben folgenden Aufbau:

```

1   5   10  15  20  25  30  35  40  45  50  55  60  65  70  75  80
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
OSTERMONTAG                ,N,1991-04-01,1992-04-20,1993-04-12,1994-04-04
NEUJAHR                     ,C,****-01-01

```

## Erklärung

- Jeder Eintrag wird vom vorhergehenden Eintrag durch ein Komma getrennt.
  - Der erste Eintrag in jeder Zeile ist der Name des Feiertags und hat 30 Zeichen (mit Füll-Blanks)
  - Der zweite Eintrag in jeder Zeile definiert, ob der Feiertag ein zyklischer oder ein nicht-zyklischer ist.
  - Die übrigen Einträge geben die dem jeweiligen Feiertag zugeordneten Kalendertage an.
- Da der EDT nur eine maximale Satzlänge von 256 Zeichen unterstützt, können einem (nicht-zyklischen) Feiertag nicht mehr als 20 Kalendertage zugeordnet werden. Bei zyklischen Feiertagen (siehe oben das Beispiel „NEUJAHR“) wird prinzipiell nur ein Datum angegeben. Die Jahresangabe wird in diesem Fall durch den String „\*\*\*\*“ ersetzt.

## 2.2 Kalenderdateien

Wie schon eingangs erwähnt, wird für jeden Kalender eine eigene Kalenderdatei angelegt. Es gibt keine „zentrale“ Kalenderdatei, die alle vorhandenen Kalender enthält.

Jede Kalenderdatei enthält genau einen Kalender.

Kalenderdateien können auf einem Shared-Pubset liegen und so für mehrere Rechner zur Verfügung gestellt werden. Da der Name der Kalenderdatei und der Name des Kalenders identisch sind, wird über den Kalendernamen automatisch der Ablageort der Kalenderdatei (CATID und USERID) definiert.

### 2.2.1 Namenskonvention für Kalenderdateien

Maximale Länge eines Kalenderdatei-Namens:  
54 Zeichen.

Darüberhinaus gibt es keine Einschränkungen für den Namen einer Kalenderdatei.

### 2.2.2 Schutzmechanismen für Kalenderdateien

Alle Schutzmechanismen für BS2000-Dateien sind auch für Kalenderdateien zulässig. Der Anwender bzw. Erzeuger einer Kalenderdatei darf allerdings nicht verhindern, dass ein Leser bzw. schreibender Zugriff auf die Kalenderdatei möglich ist.

Um eine Kalenderdatei über die Kommandoschnittstelle zu lesen, muss die Kalenderdatei mit einem der beiden OPEN-Modi OPENMOD=\*READ bzw. \*READALL geöffnet werden. Dies setzt die Möglichkeit eines lesenden Zugriffs voraus.

Um in eine Kalenderdatei über die Dialogschnittstelle Kalender-Editor zu schreiben, muss die Kalenderdatei mit dem Open-Modus OPENMOD=\*UPDATE geöffnet werden. Dies setzt die Möglichkeit eines schreibenden Zugriffs voraus.

Der Anwender kann eine Kalenderdatei nur auf seiner eigenen Benutzerkennung anlegen. In einer Kalenderdatei dürfen nicht mehrere Anwender gleichzeitig Änderungen vornehmen. Dies wird dadurch sichergestellt, dass die Kalenderdatei beim Open-Modus \*UPDATE mit SHARUPD=WEAK und OPEN=INOUT geöffnet wird.

#### *Hinweise*

- Der schreibende Zugriff auf eine Kalenderdatei darf nur über das Subsystem CALENDAR erfolgen.
- Der lesende Zugriff auf eine Kalenderdatei ist auch außerhalb, z.B. mit dem Kommando SHOW-FILE, möglich.

### 2.2.3 Dateiattribute

Eine Kalenderdatei wird mit folgenden Attributen angelegt (siehe Ausgabe des Kommandos SHOW-FILE-ATTRIBUTES im Handbuch „Kommandos“ [1]):

- FILE-STRUC=PAM
- BUF-LEN=STD(2)
- BLK-CONTR=NO.

## 2.3 Installation

### 2.3.1 Bestandteile

Das Produkt CALENDAR gehört zum BS2000-Grundausbau und ist mit BS2000/OSD-BC ab V2.0 ablauffähig.

Das Produkt besteht aus folgenden Komponenten:

- Modulbibliothek: SYSLIB.CALENDAR.<version> .  
Die Bibliothek enthält die GCs für das CALENDAR-Subsystem.
- Modulbibliothek: SYSLNK.CALENDAR.<version> .  
Die Bibliothek enthält die Module für das CALENDAR-Subsystem.
- Modulbibliothek: SYSLNK.CALENDAR-TU.<version> .  
Die Bibliothek enthält die Module für den Kalender-Editor.
- Syntaxdatei: SYSSDF.CALENDAR.<version> .
- Subsystem-Deklaration: SYSSSC.CALENDAR.<version> .
- Meldungsdatei: SYSMES.CALENDAR.<version> .
- Repdatei für Objektkorrekturen: SYSREP.CALENDAR.<version> .
- Struktur- und Installationsinformation des CALENDAR-Subsystems für IMON: SYSSII.CALENDAR.<version> .
- Struktur- und Installationsinformation des Kalender-Editors für IMON: SYSSII.CALENDAR-TU.<version> .
- Beispiel einer Feiertagsdatei: SYSDAT.CALENDAR.<version>.HOLIDAY.  
Die Datei enthält vordefinierte Feiertage, die beim Erzeugen eines Kalenders in den Kalender übernommen werden. Sie kann bei Bedarf erweitert werden.  
Diese Datei kann als Vorlage für die eigentliche Feiertagsdatei \$TSOS.SYSDAT.CALENDAR.HOLIDAY verwendet werden.  
Mit Auslieferung der Beispielsdatei hat der Systemverwalter die Möglichkeit, seine alte Feiertagsdatei \$TSOS.SYSDAT.CALENDAR.HOLIDAY beizubehalten und evtl. mit dem Inhalt der Beispielsdatei zu ergänzen, oder gänzlich neu zu generieren (durch kopieren).

### 2.3.2 Installation des Subsystems

CALENDAR ist ein Subsystem des BS2000 und wird von DSSM verwaltet. Dazu muss die Subsystem-Deklaration SYSSSC.CALENDAR.<version> mit dem Programm SSCM in den Subsystemkatalog eingebracht werden.

Ausserdem muss die Feiertagsdatei SYSDAT.CALENDAR.HOLIDAY bereitgestellt werden (z.B. durch Umbenennen der ausgelieferten SYSDAT.CALENDAR.170.HOLIDAY nach SYSDAT.CALENDAR.HOLIDAY).

Standardmaessig wird beim Hochfahren des BS2000 das Subsystem CALENDAR und die zugehoerige Repdatei aktiviert.



Die Syntax- und Meldungsdatei müssen vom Systemverwalter global im System aktiviert werden.

Das Subsystem CALENDAR kann im laufenden Betrieb mit /STOP-SUBSYSTEM deaktiviert und mit /START-SUBSYSTEM wieder aktiviert werden.

Der Kalender-Editor wird mit dem Kommando START-CALENDAR-EDITOR gestartet. Um mit dem Kalender-Editor arbeiten zu können, muss das CALENDAR-Subsystem aktiviert sein.



---

## 3 Dialogschnittstelle: Der Kalender-Editor

Der Kalender-Editor ist ein Dienstprogramm, mit dem neue Kalenderdateien erzeugt und bereits vorhandene Kalenderdateien gelesen und verändert werden können. Die Oberfläche dieser Dialogschnittstelle ist über FHS-Masken realisiert.

Während der Bearbeitung wird die Kalenderdatei nicht automatisch zwischengesichert (explizite Angabe notwendig). Durch explizites Schließen oder Sichern der Kalenderdatei werden die im Hauptspeicher befindlichen Daten auf Platte geschrieben.

### 3.1 Funktionsübersicht

Die mit dem Kalender-Editor zu bearbeitenden Daten sind vier Bereichen zuzuordnen, die den oben beschriebenen vier grundlegenden Strukturelementen eines Kalenders entsprechen:

- Basis-Informationen
- Kalendertage
- SYMDAT
- Feiertage.

Folgende Übersicht führt die Funktionen des Kalender-Editors auf und ordnet den Namen derjenigen Maske zu, in der der Anwender die entsprechende Funktion im Dialog realisieren kann.

<b>Funktion</b>	<b>Maskenname</b>
Anlegen einer neuen Kalenderdatei	Maske C000
Basis-Informationen in einer Kalenderdatei lesen / bearbeiten	Maske B020
Feiertage ausgeben / kopieren / löschen / erzeugen / aktivieren	Maske H010
Feiertags-Informationen bearbeiten	Maske H020
Funktion der zu lesenden / zu bearbeitenden Informationen eines Kalenders auswählen	Maske C010
Kalenderdatei neu anlegen	Maske C000
Kalenderdatei zum Lesen / Bearbeiten eines bestehenden Kalenders öffnen	Maske C000
Kalendertage mit ihren Eigenschaften ausgeben und zur weiteren Bearbeitung auswählen	Maske D010
Kalendertage, d.h. Tages-Informationen bearbeiten	Maske D020
Öffnen einer bestehenden Kalenderdatei zum Lesen / Bearbeiten eines Kalenders	Maske C000
Sichern und Schließen einer geöffneten Kalenderdatei	Maske C000
SYMDAT (symbolische Datumsangaben) ausgeben / kopieren / löschen / erzeugen	Maske S010
SYMDAT (symbolische Datumsangaben) bearbeiten, d.h. ändern	Maske S020
Tages-Informationen (Kalendertage) ausgeben und zur weiteren Bearbeitung auswählen	Maske D010
Tages-Informationen, d.h. Kalendertage bearbeiten	Maske D020

## 3.2 Bedienung des Kalender-Editors

Dieses Kapitel beschreibt, wie der Kalender-Editor aufgerufen und beendet wird, erklärt allgemein den Maskenaufbau, gibt eine Übersicht über die generelle Maskenfolge im Dialogbetrieb und stellt ausführlich jede einzelne Maske dar.



### 3.2.1 Starten und Beenden

Der Kalender-Editor wird gestartet mit dem Kommando START-CALENDAR-EDITOR.

Nach dem Start befinden Sie sich im Hauptmenü, der Maske C000.

Über die Auswahl des Dateiöffnungs-Modus in dieser Maske wird festgelegt, ob

- Informationen aus einem vorhandenen Kalender ausgegeben werden sollen
- Informationen aus einem vorhandenen Kalender ausgegeben und bearbeitet werden sollen
- eine neue Kalenderdatei angelegt werden soll.

Das Programm wird regulär beendet durch Auswahl der Funktion „5“ in der Funktionsauswahl-Maske C000.

Eine Programmunterbrechung mit der Funktionstaste **[K2]** ist zulässig; das Programm verzweigt in den Systemmodus. Dabei sollte der Anwender im Auge behalten, dass geöffnete Dateien nicht geschlossen werden und der Hauptspeicher Informationen enthalten kann, die noch nicht gesichert wurden. Eine Rückkehr in den Kalender-Editor nach Programmunterbrechung ist also dringend angeraten.

#### Format

##### START-CALENDAR-EDITOR

**VERSION** = **\*STD** / <product-version 6..11> / <product-version 4..9 without-cor> /  
<product-version 3..8 without-man>

,**MONJV** = **\*NONE** / <full-filename 1..54 without-gen-vers>

,**CPU-LIMIT** = **\*JOB-REST** / <integer 1..32767>

#### Beschreibung der Operanden:

##### **VERSION** =

Gibt die Elementversion an. Nur zulässig für eine Programmbibliothek (Elementtyp R oder L).

##### **VERSION** = **\*STD**

Der Defaultwert für die höchste Elementversion bei Programmbibliotheken wird übernommen (siehe Handbuch "LMS" [12]).

##### **VERSION** = <product-version 6..11>

Vollständige Angabe der Produktversion.

**VERSION = <product-version 4..9 without-correction-state>**

Angabe der Produktversion ohne Korrekturstand.

**VERSION = <product-version 3..8 without-manual-release>**

Angabe der Produktversion ohne Freigabe- und Korrekturstand.

**MONJV = \*NONE / <full-filename 1..54 without-gen-vers>**

Name der JV, die das Programm überwachen soll.

Während des Programmablaufs setzt dann das System die JV auf entsprechende Werte:

\$R Programm läuft

\$T Programm normal beendet

\$A Programm abnormal beendet

Dieser Operand steht nur dem Anwender mit dem Software-Produkt JV zur Verfügung (siehe auch Handbuch „Jobvariablen“ [2]).

**CPU-LIMIT = \*JOB-REST / <integer 1..32767 seconds>**

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf.

Wurde der Auftrag ohne Zeitbeschränkung (CPU-LIMIT=\*NO) gestartet, läuft das Programm *ebenfalls ohne* Zeitbeschränkung (Angabe wird ignoriert)!

Überschreitet das Programm beim Ablauf die angegebene Zeit, wird im Dialog das Programm unterbrochen und die Meldung EXC0075 ausgegeben. Der Benutzer kann einen Dump anfordern, das Programm abbrechen oder fortsetzen.

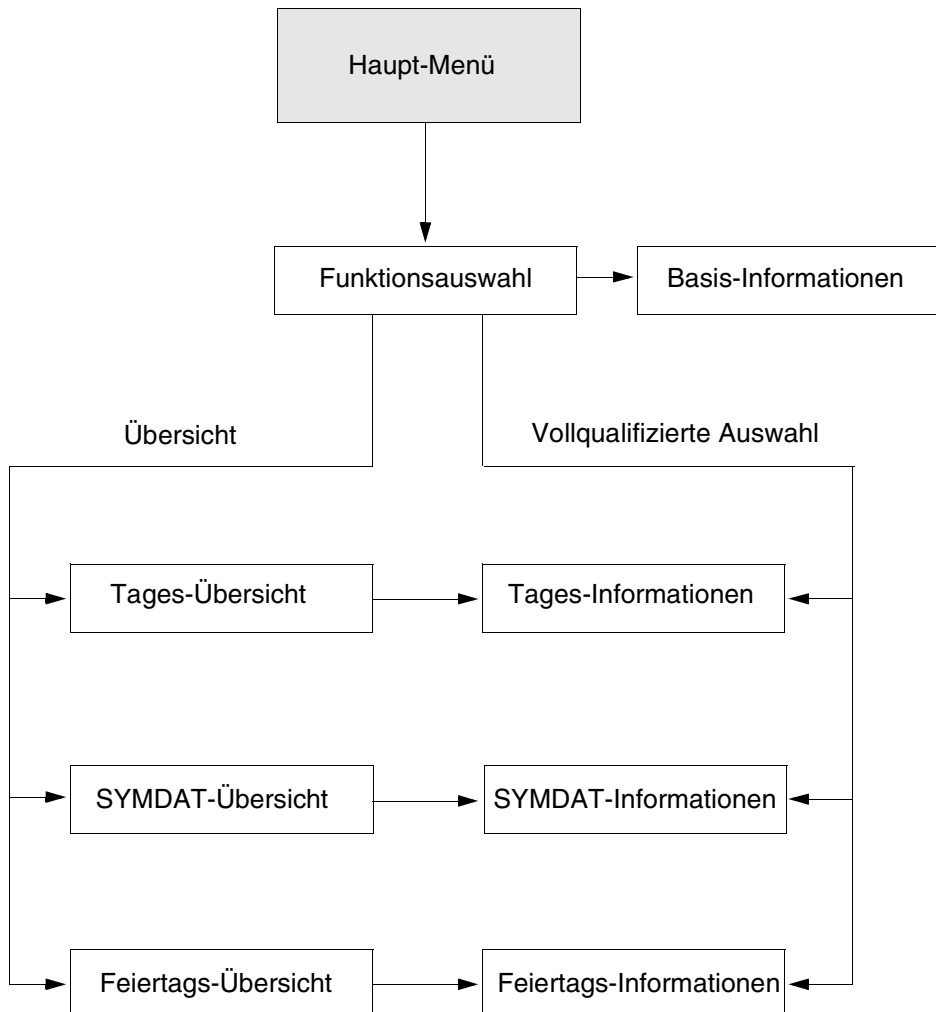
Im Stapelbetrieb wird das Programm beendet.

**CPU-LIMIT = \*JOB-REST**

Voreinstellung:

Wurde der Auftrag mit Zeitbeschränkung gestartet, so wird für das Programm der bei der Systemgenerierung festgelegte Wert als Zeitbeschränkung verwendet. Anderenfalls läuft das Programm ohne Zeitbeschränkung.

### 3.2.2 Maskenübersicht





*Anzeige- und Modifikationsbereich, Zeilen 4 - 19*

Aufbau und Inhalt dieses Bereichs sind maskenspezifisch.

*Operations-Auswahlbereich, Zeilen 20 - 21*

Die Informationen in den Zeilen 21 und 22 geben dem Anwender einen Überblick, welche Operationen im Anzeige- und Modifikationsbereich dieser Maske zulässig sind.

*Anweisungs-Eingabebereich, Zeile 22*

Dem Eingabefeld „Stmt“ (Statement) schließt sich eine Aufzählung aller Eingabemöglichkeiten an.

Bei der Maske B020 fehlt diese Aufzählung. Hier können nur die Zeichenfolgen „F12“ und „F13“ eingegeben werden, um dieselbe Wirkung wie die gleichnamigen Funktionstasten zu erzielen.

*Meldebereich, Zeilen 23 - 24*

Die Meldezeile informiert über eventuelle Fehlbedienungen und falsche Eingaben bzw. interne Fehler, z.B. FHS-Fehler, die aufgetreten sind.

### 3.2.4 Die Editor-Funktionen

Alle im Kalender-Editor ausführbaren grundlegenden Funktionen sind in diesem Kapitel zusammengefasst und werden nicht mehr gesondert bei der Beschreibung der einzelnen Masken beschrieben.

#### **Maske verlassen**

Die Masken C000 (Hauptmenü) und C010 (Auswahl der Funktionen) werden durch die Angabe einer Funktionsnummer verlassen.

Alle anderen Masken können mit der Funktionstaste **F12** bzw. **F13** verlassen werden. Die Funktionstaste **F12** bewirkt, dass die am Bildschirm erfolgte Eingabe nicht ausgewertet wird.

Die Funktionstaste **F13** bewirkt eine Auswertung der erfolgten Eingabe.

In beiden Fällen wird nachher zur Vorgängermaske zurückgekehrt.

Stehen einem Anwender diese beiden Funktionstasten nicht zur Verfügung, so kann er in das Eingabefeld „Stmt“ die Zeichenfolge „F12“ oder „F13“ eintragen.

In keinem Fall kann durch diese Beendigungsfunktion eine oder mehrere Ebenen übersprungen werden.

### Die Funktion „Blättern“

Die Blätter-Funktion, d.h Blättern vorwärts und Blättern rückwärts, wird im CALENDAR-Editor angeboten für

- alle Masken der Objektauswahl-Ebene: D010, S010, H010
- alle Masken der Objektbearbeitungs-Ebene: D020, S020, H020.

Der Aufruf dieser Blätterfunktion kann auf zwei verschiedene Arten erfolgen:

- entweder über eine Funktionstaste
- oder über die direkte Eingabe in das Eingabefeld „Stmt“ auf Zeile 22 der betreffenden Maske.

*Blättern rückwärts* wird erreicht

- entweder durch Drücken der Funktionstaste **F7**
- oder durch direkte Eingabe des Zeichens „-“ bzw. der Zeichenfolge „F7“ im Eingabefeld „Stmt“ (für diejenigen Anwender, denen die Funktionstaste nicht zur Verfügung steht).

*Blättern vorwärts* wird erreicht

- entweder durch Drücken der Funktionstaste **F8**
- oder durch direkte Eingabe des Zeichens „+“ bzw. der Zeichenfolge „F8“ im Eingabefeld „Stmt“ (für diejenigen Anwender, denen die Funktionstaste nicht zur Verfügung steht).

Soweit möglich, wird immer ganzseitig (Bildschirmseite) geblättert.

Stößt man beim Blättern an eine der Bereichsgrenzen (Anfang oder Ende der aufgelisteten Objekte), wird dies im Meldungsbereich der Maske angezeigt.

Der Kalender-Editor versucht immer, möglichst alle Zeilen des Ausgabebereichs aufzufüllen. Umfasst z.B. in einer Maske der Ausgabebereich zum Auflisten von Objekten 14 Zeilen, und der Anwender stößt beim Blättern an das Ende der Liste, werden die letzten 14 Datensätze ausgegeben (wenn 14 oder mehr Einträge vorhanden sind).

### Die Funktion „Positionieren“

In allen Masken der Objektauswahl-Ebene und der Objektbearbeitungs-Ebene gibt es ein *Eingabefeld* „Goto“. In diesem Feld kann der Name des gewünschten Objekts eingegeben werden. Zur Auswahl einer Gruppe von Objekten kann ein teilqualifizierter Name angegeben werden. Nach einer GOTO-Angabe wird die Liste mit Objekten immer neu aufgebaut.

Ist die GOTO-Angabe (alphabetisch) kleiner als das erste Element der Liste, beginnt die Liste mit dem ersten vorhandenen Objekt. Eine Meldung in der Meldungszeile 2 informiert den Anwender, dass das angegebene Objekt nicht existiert bzw. kleiner als das erste ausgegebene ist.

Ist das bei GOTO angegebene Objekt in der Liste enthalten, beginnt die Liste mit dem angegebenen Objekt.

Liegt die GOTO-Angabe zwischen zwei Elementen der Liste, beginnt die Liste mit dem al-

phabetisch nächsten Objekt. Eine entsprechende Meldung wird in der Meldungszeile 2 ausgegeben.

Ist die GOTO-Angabe (alphabetisch) größer als das letzte Element der Liste, beginnt die Liste mit dem letzten vorhandenen Objekt, gefolgt vom vorletzten usw., bis der Bildschirm voll ist bzw. soweit Objekte vorhanden sind. Eine entsprechende Meldung wird in der Meldungszeile 2 ausgegeben.

Erfolgt ein syntaktisch falscher Eintrag, wird eine entsprechende Meldung ausgegeben und ein korrekter Eintrag erwartet.

### Die Auswahl eines Objekts

Die in einer Kalenderdatei vorhandenen Objekte werden in den Masken der Objektauswahl-Ebene D010, H010 und S010 aufgelistet.

Der Anwender wählt durch Eingabe des Zeichens „X“ in das dafür vorgesehene Auswahl-feld ein Objekt aus. Verwendet der Anwender zum Markieren eines Objekts nicht das Zeichen „X“ sondern ein beliebiges anderes Zeichen ≠ Blank (Leerzeichen), wird dies wie ein „X“ interpretiert.

Dabei ist jedoch folgendes zu beachten:

Bei einigen Masken kann durch Angabe eines anderen Zeichens als „X“ eine weitere Funktion für das ausgewählte Objekt aufgerufen werden. Die entsprechenden Buchstaben und zugehörigen Funktionen sind für den Anwender in jeder entsprechenden Maske deutlich als Alternativauswahl mit abweichender Funktion erkennbar. Die Beschreibung erfolgt bei der Einzelbeschreibung der Masken.

### Die Auswertung der Eingaben

Mehrere auf einer Maske gleichzeitig gewünschten Aktionen (z.B. Markierungen in Eingabefeldern und Aufruf der Blätter-Funktion) werden in der folgenden Reihenfolge bearbeitet:

1. Prüfung der Funktionstaste **F12**.  
**F12** bewirkt die Rückkehr zur Vorgängermaske. Wurde die Funktion aufgerufen, werden keine weiteren Eingaben (wie z.B. Markierungen oder Blättern) ausgewertet.
2. Prüfung der Eingaben.  
Findet das Programm eine fehlerhafte Eingabe (z.B. ein ungültiges Datum), wird die Maske mit einer entsprechenden Fehlermeldung erneut ausgegeben. Die Eingabewerte des Anwenders bleiben dabei unverändert.
3. Auswertung vorhandener Markierungen.  
Nach Auswertung der Markierungen werden die entsprechenden Funktionen ausgeführt. Die Markierungen „X“ (Objektauswahl) und „C“ (Objekt kopieren) bewirken, dass das Programm in eine Folgemaske verzweigt.

4. Prüfung des Eingabefeldes „Stmt“ auf den Wert „New“.  
Bei den Masken, für die der Wert „New“ bei „Stmt“ als Möglichkeit vorgesehen ist (S010, H010), wird geprüft, ob dieser Wert angegeben wurde. Ist dies der Fall (d.h. der Anwender möchte ein neues Objekt erzeugen), verzweigt das Programm in die entsprechende Folgemaske.
5. Prüfung der Funktionstaste **F13**.  
**F13** bewirkt die Rückkehr zur Vorgängermaske. Wurde die Funktion aufgerufen, werden weder die Funktion „Blättern“ noch ein Wert im Feld „Goto“ (zum Positionieren) ausgewertet.
6. Auswertung des Eingabefeldes „Goto“.  
Erfolgt das Positionieren über „Goto“, wird die Funktion „Blättern“ ignoriert.
7. Auswertung der Angaben zur Funktion „Blättern“ und Ausführung.

Führt die Auswertung der Angaben des Anwenders zu neuen Werten in dieser Maske, dann wird die Maske mit den geänderten Daten neu angezeigt.  
Alle zugleich erfolgten Angaben, die nach obiger Reihenfolge nachrangig zu behandeln waren (z.B. Blätter-Funktion oder Angabe bei „Goto“ zum Positionieren), werden nicht mehr ausgewertet, d.h. müssen evtl. neu in die geänderte Maske eingegeben werden.

### Die Bearbeitung eines Objekts

Wurde die Kalenderdatei in der Hauptmenü-Maske C000 mit „Open Mode = Update“ geöffnet, können die Daten in einer der Masken D020, S020 oder H020 (Objektbearbeitungsebene) bearbeitet werden.

### Datenübertragung

Die Übertragung der Daten vom Bildschirm zum Programm CALENDAR-Editor erfolgt entweder mit der Funktionstaste **DUE** bzw. **ENTER** oder mit der Funktionstaste **F13**.  
Wurden am Bildschirm keine Eingaben vorgenommen, so bewirkt die Funktionstaste **DUE** bzw. **ENTER** eine Wiederholung der vorangegangenen Maskenausgabe.



## Funktionstasten (Übersicht)

Funktionstasten können in folgender Weise genutzt werden:

Taste	Bedeutung
<b>[DUE] / [ENTER]</b>	Auswertung von Eingaben am Bildschirm
<b>[F7]</b>	Blättern rückwärts
<b>[F8]</b>	Blättern vorwärts
<b>[F12]</b>	Rückkehr zur Vorgängermaske ohne Auswertung von Eingaben am Bildschirm
<b>[F13]</b>	Auswertung von Eingaben am Bildschirm und Rückkehr zur Vorgängermaske
<b>[K2]</b>	Programmunterbrechung

### 3.2.5 Beschreibung der Masken

Im diesem Abschnitt wird jede Maske des Kalender-Editors ausführlich dargestellt in folgenden Schritten

- Reihenfolge der Masken (Grafik)
- Funktionsbeschreibung
- Abbildung der Maske
- Beschreibung der Felder
- Folgemaske

#### *Hinweis*

Bei der Feldbeschreibung sind alle Eingabefelder durch einen Eingabepfeil " ▶ " gekennzeichnet.

### 3.2.5.1 Maske C000: Main Hauptmenü des Kalender-Editors

Nach Aufruf des Programms erscheint als erste Maske das Hauptmenü.

#### Funktionsbeschreibung

Das Hauptmenü ist die Funktionsauswahl-Ebene des Kalender-Editors.

Um eine Kalenderdatei zu lesen oder zu bearbeiten, muss sie zunächst geöffnet werden. Das Programm erwartet dazu:

- die Funktionsnummer „1“ für Open
- den Open-Modus: Read / Update / Create
- den Namen der existierenden oder zu erzeugenden Kalenderdatei

Vor dem Öffnen einer Kalenderdatei (und ebenso nach dem Schließen) kann alternativ zu „Open“ nur die Funktion „Exit“ angegeben werden, die das Programm wieder beendet.

Nach dem Öffnen einer bestehenden Kalenderdatei zum Bearbeiten (Update) verzweigt das Programm zur Maske C010.

Nach dem Öffnen bzw. Anlegen einer neuen Kalenderdatei (Create) verzweigt das Programm zur Maske B020.

Die Eingaben sind mit **ENTER** zu bestätigen.

#### Maske

```

C000      Main                               Calendar Editor Vxxx
Calendar Name: :10SN:$QM211.WERK.KALENDER

Function: 1

  1. Open      Open Mode:  U  (R=Read/U=Update/C=Create)
                Name: WERK.KALENDER

  2. Edit

  3. Save

  4. Close     Save (Y/N): Y

  5. Exit

Select Function with Number + Enter

Msg:
CLD2007 CALENDAR SUCCESSFULLY OPENED.

LTG                                             TAST

```

## Feld-Beschreibung

► Function:

Eingabefeld für die der gewünschten Funktion zugeordneten Ziffer (1..5).  
Die Auswahl der Funktion erfolgt durch Eintrag dieser Ziffer in dieses Feld.

Ist keine Kalenderdatei geöffnet, können nur die Funktionen 1 oder 5 angegeben werden.

### 1. OPEN

Die Eingabe der Funktionsziffer „1“ in das Feld „Function“ ist eine der notwendigen Angaben zum Öffnen einer Kalenderdatei. (Zusätzlich muss der Open-Modus für die Kalenderdatei und ihr Name bekanntgegeben werden.)

► Open Mode:

Open-Modus für die Kalenderdatei.

Mögliche Werte:

R (=Read), U (=Update), C (=Create)

Erfolgt keine Angabe, wird das Feld mit dem Wert „U“ vorbelegt.

Um eine neue Kalenderdatei anzulegen, muss die Voreinstellung mit dem Wert C (CREATE) überschrieben werden.

Um eine vorhandene Kalenderdatei zu lesen, sollte die Voreinstellung mit dem Wert R (READ) überschrieben werden, um den gleichzeitigen Zugriff mehrerer Anwender zu ermöglichen.

R=Read

Lesender Zugriff auf einen vorhandenen Kalender.

Die Datei wird in den Hauptspeicher übertragen, ohne dass Änderungen möglich sind und kann von mehreren Anwendern gleichzeitig gelesen werden. Die Kalenderdatei muss in diesem Zugriffsmodus beim Verlassen nicht geschlossen werden.

U=Update

Schreibender Zugriff auf einen vorhandenen Kalender.

Die Datei wird in den Hauptspeicher übertragen und kann dort immer nur von einem Anwender gleichzeitig bearbeitet werden. Während ein Anwender Änderungen vornimmt, ist sie für alle übrigen gesperrt.

Die erfolgten Änderungen werden erst in die Kalenderdatei zurückgeschrieben, wenn eine der Funktionen „Save“ (3) oder „Close“ (4) mit Save = Y erfolgt ist.

### C=Create

Anlegen einer neuen Kalenderdatei, d.h Erzeugen eines neuen Kalenders. Nachdem die Angabe der Funktionsziffer (1), des Open-Modus (C) und des Namens für den neuen Kalender mit **[DUE]** bzw. **[ENTER]** bestätigt sind, verzweigt das Programm zur Maske B020, in der die Kalendergrenzen und die Standard-Arbeitswoche definiert werden können.

► Name :

Name der zu lesenden / zu bearbeitenden / zu erzeugenden Kalenderdatei. Der Name muss angegeben werden, andernfalls wird eine entsprechende Fehlermeldung im Meldungsbereich ausgegeben. Der Name muss ein vollqualifizierter BS2000-Dateiname sein; über CATID und USERID wird der Ablageort der Datei definiert.

### 2. Edit (nur bei geöffneter Kalenderdatei)

Die Eingabe der Funktionsziffer „2“ in das Feld „Function“ bewirkt das Verzweigen zur Maske C010, um dort eine Funktion auszuwählen, aus dem Objekte bearbeitet werden sollen.

### 3. Save (nur bei geöffneter Kalenderdatei)

Die Eingabe der Funktionsziffer „3“ in das Feld „Function“ ermöglicht dem Anwender, seine geänderten Daten vor dem Schließen der Kalenderdatei zwischendurch zu sichern.

### 4. Close (nur bei geöffneter Kalenderdatei)

Die Eingabe der Funktionsziffer „4“ in das Feld „Function“ bewirkt, dass die Kalenderdatei geschlossen wird.

► Save (Y/N):

Angabe, ob beim Schließen der Datei die im Hauptspeicher erfolgten Änderungen in die Kalenderdatei zurückgeschrieben werden sollen.

Voreinstellung:

„Y“ (=YES), d.h. die erfolgten Änderungen werden in die Kalenderdatei übernommen. Überschreibt der Anwender die Voreinstellung durch explizite Angabe des Wertes „N“ (=NO), werden alle Änderungen seit dem letzten „Save“ nicht in die Kalenderdatei übernommen.

## 5. Exit

Die Eingabe der Funktionsziffer „5“ in das Feld „Function“ bewirkt die Beendigung des Kalender-Editors.

### *Hinweis*

Die Funktion „5“ kann nicht ausgeführt werden, wenn eine mit „Update“ oder „Create“ geöffnete Kalenderdatei nicht vorher geschlossen wurde.

### 3.2.5.2 Maske C010: Edit Auswahl der Funktionen



#### Funktionsbeschreibung

In dieser Maske kann eine Funktion ausgewählt werden, um dann in der entsprechenden Folgemaske die gewünschten Objekte zu lesen oder zu bearbeiten.

Funktionen, die über eine zugeordnete Funktionsziffer ausgewählt werden, betreffen die Objekte

- Basis-Informationen
- Kalendertage
- SYMDAT-Angaben
- Feiertage

*Basis-Informationen* können nur grundsätzlich ausgewählt werden, ohne sie einzeln mit Namen ansprechen zu können. Die entsprechende Funktionsziffer (1) verzweigt immer zur Objektbearbeitungs-Ebene in die Maske B020, um sie dort zu lesen oder zu bearbeiten.

Für die übrigen Funktionen, die die Objekte „Kalendertage“, „SYMDAT“ und „Feiertage“ betreffen, besteht auch die Möglichkeit, Objekte mit vollqualifiziertem Namen anzugeben. In diesem Fall wird in die entsprechende Maske der Objektbearbeitungs-Ebene verzweigt.

Bei keiner oder teilqualifizierter Angabe für den Objektnamen, wird in die entsprechende Maske der Objektauswahl-Ebene verzweigt, in der eine Bearbeitung von Objekten nicht möglich ist.

Die Funktionsauswahl (mit oder ohne zusätzliche Angaben zur Objektauswahl) ist mit **ENTER** zu bestätigen.

## Maske

C010	Edit	Calendar Editor Vxxx
Calendar Name: :10SN:\$QM211.WERK.KALENDER		
Function: 1		
1. Edit Basic Information		
2. Edit Days	Start with	: 1995-03-15
3. Edit Symbolic Dates		:
4. Edit Holidays		:
5. Return to Main Menu		
Select Function with Number + Enter		
Msg:		
LTG		TAST

## Feld-Beschreibung

### ► Function:

Eingabefeld für die der gewünschten Funktion zugeordneten Ziffer (1..5). Die Auswahl der Funktion erfolgt durch Eintrag dieser Ziffer in dieses Feld.

#### 1. Edit Basic Information

Auswahl der Funktion „Basis-Informationen“.

Basis-Informationen (d.h. Kalendergrenzen und Standard-Arbeitswoche) können nur als Gesamtheit angesprochen werden. Da ein Kalender nur *eine* Untergrenze und nur *eine* Obergrenze und auch nur *eine* Standard-Arbeitswoche enthält, ist eine Namensvergabe für die einzelnen Elemente nicht vorgesehen und überflüssig.

Das Programm verzweigt immer in die Maske B020, d.h. Objektbearbeitungs-Ebene. Ob die Basis-Informationen einer Kalenderdatei dort nur gelesen oder auch bearbeitet werden können, ist vom Open-Modus der Kalenderdatei abhängig (genauer siehe Beschreibung der Maske B020 auf [Seite 43ff.](#)).

## 2. Edit Days

Auswahl der Funktion „Kalendertage“.

Macht der Anwender keine zusätzlichen Angaben im Eingabefeld „Start with“, verzweigt das Programm in die Maske D010, in der die Kalendertage der geöffneten Kalenderdatei, beginnend mit dem ersten gefundenen Eintrag, ausgegeben werden.

Eine Bearbeitung der Kalendertage ist hier nicht möglich.

► Start with:

Auswahl des Startwertes, mit dem die Auflistung der Kalendertage beginnen soll.

Eine Eingabe in dieses Feld ist nicht zwingend. Als Eingaben sind voll- und teilqualifizierte Datumsangaben erlaubt. Voreinstellung gibt es keine.

Macht der Anwender hier keine Angabe, beginnt die Ausgabeliste mit dem ersten Tag des Kalenders.

Die Eingabe eines *teilqualifizierten* Datums in der Form yyyy-... (yyyy=Jahr, d.h. nur Jahresangabe) bewirkt, dass die Ausgabe ab dem ersten Tag des Jahres bzw. ab dem ersten Kalendertag (wenn der Kalender ab einem späteren Datum beginnt) erfolgt.

Die Eingabe eines *teilqualifizierten* Datums in der Form yyyy-mm-.. (mm=Monat, d.h. Jahres- und Monatsangabe) bewirkt, dass die Ausgabe ab dem ersten Tag des Monats im angegebenen Jahr bzw. ab dem ersten Kalendertag (wenn der Kalender ab einem späteren Datum beginnt) erfolgt.

Keine oder eine teilqualifizierte Angabe bewirken, dass das Programm in die Maske D010 verzweigt. Dort werden die Kalendertage der geöffneten Kalenderdatei, beginnend mit dem ersten gefundenen Eintrag, ausgegeben.

Eine Bearbeitung der Kalendertage ist hier nicht möglich.

Die Eingabe eines vollqualifizierten Datums in der Form yyyy-mm-dd (dd=Tagesangabe) bewirkt, dass das Programm in die Maske D020 (Objektbearbeitungs-Ebene) verzweigt. Dort wird der angegebene Kalendertag mit der Liste der diesem Tag zugeordneten SYMDATs zur Bearbeitung ausgegeben.

Voraussetzung ist natürlich, dass der angegebene Wert auch tatsächlich in der Kalenderdatei als Kalendertag vorhanden ist. Andernfalls bleibt der Anwender in der Maske C010 und erhält eine entsprechende Fehlermeldung ausgegeben.

## 3. Edit Symbolic Dates

Auswahl der Funktion „SYMDAT“ (Symbolische Datumsangaben).



Macht der Anwender keine zusätzlichen Angaben im zugehörigen Eingabefeld „Start with“, verzweigt das Programm in die Maske S010, in der die SYMDATs der geöffneten Kalenderdatei, beginnend mit dem ersten Element, ausgegeben werden. Eine Bearbeitung der SYMDATs ist hier nicht möglich.

► Start with : \_ \_  
:

Auswahl des Startwertes, mit dem die Auflistung der SYMDATs beginnen soll.

Eine Eingabe in dieses Feld ist nicht zwingend.

Eine Voreinstellung gibt es nicht.

Als Eingaben sind teil- und vollqualifizierte SYMDAT-Namen erlaubt.

Die Eingabe eines *teilqualifizierten* SYMDAT-Namens „name\*“ (Teil eines SYMDAT-Namens, abgeschlossen mit dem Zeichen „\*“) bewirkt folgendes:

Das Programm verzweigt in die Maske D010, in der die SYMDATs der geöffneten Kalenderdatei, beginnend mit dem ersten gefundenen Eintrag in alphabetischer Reihenfolge (nach Maßgabe der Eingabe „name\*“), ausgegeben werden. Eine Bearbeitung der SYMDATs ist hier nicht möglich.

Die Eingabe eines *vollqualifizierten* SYMDAT-Namens bewirkt, dass das Programm in die Maske S020 (Objektbearbeitungs-Ebene) verzweigt. Dort wird das angegebene SYMDAT mit der diesem SYMDAT zugeordneten Liste von Kalendertagen ausgegeben. Handelt es sich um ein zyklisches SYMDAT, sind die Zyklusdaten und die zugeordneten Kalendertage für die Bearbeitung gesperrt. Handelt es sich um ein nicht-zyklisches SYMDAT, sind die Zyklusangaben ebenfalls gesperrt, die zugeordneten Kalendertage können jedoch bearbeitet werden.

Voraussetzung ist natürlich, dass der angegebene Wert auch tatsächlich in der Kalenderdatei als SYMDAT vorhanden ist. Andernfalls bleibt der Anwender in der Maske C010 und erhält eine entsprechende Fehlermeldung ausgegeben.

#### 4. Edit Holidays

Auswahl der Funktion „Feiertage“.

Macht der Anwender keine zusätzlichen Angaben im zugehörigen Eingabefeld „Start with“, verzweigt das Programm in die Maske H010, in der die Feiertage der geöffneten Kalenderdatei, beginnend mit dem ersten Element, ausgegeben werden. Eine Bearbeitung der Feiertage ist hier nicht möglich.

```

Start with      :   _ _
                 :
▶               :

```

Auswahl des Startwertes, mit dem die Auflistung der Feiertage beginnen soll.

Eine Eingabe in dieses Feld ist nicht zwingend.

Eine Voreinstellung gibt es nicht.

Als Eingaben sind teil- und vollqualifizierte Feiertags-Namen erlaubt.

Die Eingabe eines *teilqualifizierten* Feiertags-Namens (Teil eines Feiertags-Namens, abgeschlossen mit dem Zeichen „\*“) bewirkt folgendes:  
Das Programm verzweigt in die Maske H010, in der die Feiertage der geöffneten Kalenderdatei, beginnend mit dem ersten gefundenen Eintrag in alphabetischer Reihenfolge (nach Maßgabe der Eingabe „name\*“), ausgegeben werden. Eine Bearbeitung der Feiertage ist hier nicht möglich.

Die Eingabe eines *vollqualifizierten* Feiertags-Namens bewirkt, dass das Programm in die Maske H020 (Objektbearbeitungs-Ebene) verzweigt. Dort wird der angegebene Feiertag mit der diesem Feiertag zugeordneten Liste von Kalendertagen ausgegeben. Handelt es sich um einen zyklischen Feiertag, sind die Zyklusdaten zugänglich, die zugeordneten Kalendertage jedoch für die Bearbeitung gesperrt. Handelt es sich um einen nicht-zyklischen Feiertag, sind die Zyklusdaten gesperrt, die zugeordneten Kalendertage können jedoch bearbeitet werden.

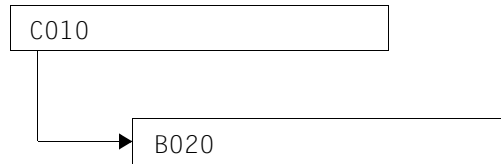
Voraussetzung ist natürlich, dass der angegebene Feiertag auch tatsächlich in der Kalenderdatei vorhanden ist. Andernfalls bleibt der Anwender in der Maske C010 und erhält eine entsprechende Fehlermeldung ausgegeben.

5. Return to main menu

Rückkehr in das Hauptmenü C000.

### 3.2.5.3 Maske B020: Basic Information

#### Basis-Informationen ausgeben und bearbeiten



#### Funktionsbeschreibung

In dieser Maske können

- die Basis-Informationen aus einem vorhandenen Kalender auf Bildschirm ausgegeben werden.  
Voraussetzung:  
Die Kalenderdatei muss mit „Open Mode = Read“ geöffnet worden sein.
- die Basis-Informationen aus einem vorhandenen Kalender auf Bildschirm ausgegeben und bearbeitet werden.  
Voraussetzung:  
Die Kalenderdatei muss mit „Open Mode = Update“ geöffnet worden sein.
- die Basis-Informationen für einen neuen Kalender festgelegt werden.  
Voraussetzung:  
Die Kalenderdatei muss mit „Open Mode = Create“ geöffnet worden sein.

Als Grundgerüst eines jeden Kalenders werden folgende Basis-Informationen definiert:

- die Kalendergrenzen mit
  - dem ersten Kalendertag
  - dem letzten Kalendertag
- die Standard-Arbeitswoche mit
  - der Festlegung jedes Standard-Wochentags als Arbeitstag oder freier Tag
  - der Festlegung der Standard-Arbeitszeit für jeden Standard-Wochentag.

Die Eingaben sind mit **ENTER** zu bestätigen.

**Maske**

B020 Basic Information		Calendar Editor Vxxx	
Calendar Name: :10SN:\$QM211.WERK.KALENDER			
Calendar Limits	Begin :	1994-01-30	
	End :	1996-01-30	
Week Definitions	Attribute	Working Hours	
	(W=Work, F=Free)	Begin	End
		(HH:MM)	(HH:MM)
Mon	W	08:00	16:30
Tue	W	08:00	16:30
Wed	W	08:00	16:30
Thu	W	08:00	16:30
Fri	W	08:00	16:30
Sat	F	00:00	00:00
Sun	F	00:00	00:00
Store with Enter		F12=Cancel, F13=Return	
Stmt:			
Msg :			
LTG		TAST	

**Feld-Beschreibung**

## Calendar Limits (Kalendergrenzen)

## ► Begin : ####-##-##

Angabe des ersten Kalendertages, mit dem der Kalender beginnen soll in der Form yyyy-mm-dd, z.B. 1995-06-01.

Für einen bereits vorhandenen Kalender kann hier die Kalenderuntergrenze neu festgelegt werden. Damit werden neue Kalendertage erzeugt und / oder alte gelöscht.

## ► End : ####-##-##

Angabe des letzten Kalendertages, mit dem der Kalender enden soll in der Form yyyy-mm-dd, z.B. 1999-12-31.

Für einen bereits vorhandenen Kalender kann hier die Kalenderobergrenze neu festgelegt werden. Damit werden neue Kalendertage erzeugt und / oder alte gelöscht.

*Hinweis*

Eine Änderung der Kalendergrenzen hat Auswirkungen auf zyklische SYMDAT (Beschreibung siehe [Seite 14ff.](#)).

### Week Definitions (Standard-Arbeitswoche)

- ▶ Mon                    –
- ▶ Tue                   –
- ▶ Wed                   –
- ▶ Thu                   –
- ▶ Fri                    –
- ▶ Sat                   –
- ▶ Sun                   –

Wochentage der Standard-Arbeitswoche.

Für jeden Tag der Standard-Arbeitswoche kann hier sein Attribut, d. h. Arbeitstag oder freier Tag, verändert werden.

Soll der Montag standardmäßig als Arbeitstag festgelegt werden, ist in das Eingabefeld rechts neben „Mon“ (Monday / Montag) das Zeichen „W“ (Work) einzutragen.

Soll der Sonntag standardmäßig als freier Tag festgelegt werden, ist in das Eingabefeld rechts neben „Sun“ (Sunday / Sonntag) das Zeichen „F“ (Free) einzutragen.

Analog ist mit den übrigen Wochentagen „Tue“ (Tuesday / Dienstag), „Wed“ (Wednesday / Mittwoch), „Thu“ (Thursday / Donnerstag), „Fri“ (Friday / Freitag) und „Sat“ (Saturday / Samstag) zu verfahren.

Die festgelegten Werte gelten (als Standard !) für den gesamten Zeitraum eines Kalenders, für jeden Montag, Dienstag etc., soweit nicht für konkrete Kalendertage Ausnahmen festgelegt werden.

### Working Hours (Standard-Arbeitszeit)

- ▶ In der Spalte unter „Begin“ wird für jeden Tag der Standard-Arbeitswoche der Beginn der Standard-Arbeitszeit festgelegt in der Form hh:mm (hh=Stunden, mm=Minuten).
- ▶ In der Spalte unter „End“ wird für jeden Tag der Standard-Arbeitswoche das Ende der Standard-Arbeitszeit festgelegt in der Form hh:mm (hh=Stunden, mm=Minuten).

### Hinweis

Die Veränderung eines Attributes eines Tags der Standard-Arbeitswoche hat Auswirkungen auf zyklische SYMDAT (Beschreibung siehe [Seite 15ff.](#)).

- ▶ Stmt :

Mögliche Angaben:

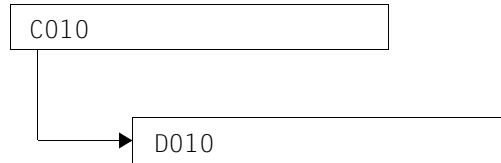
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Dadurch wird dieselbe Wirkung erzielt wie mit den gleichnamigen Funktionstasten (Beschreibung dieser Funktionstasten siehe [Seite 29](#)).

### **Folgemaske**

Nach Verlassen der Maske mit **F12** oder **F13** kehrt das Programm zurück in die Ausgangsmaske C010.

### 3.2.5.4 Maske D010: List of Days Kalendertage auflisten und zur Auswahl markieren



#### Funktionsbeschreibung

Diese Maske wird angeboten, wenn in der Maske C010 die Funktion „2“ aufgerufen wurde. Wurde keine Vorauswahl für die auszugebenden Kalendertage veranlasst, beginnt die Auflistung mit dem aktuellen Tagesdatum.

Für jeden Kalendertag steht eine Zeile zur Verfügung, in der

- das Datum
- der Wochentag
- das Attribut (Arbeitstag / freier Tag)
- das Feiertagskennzeichen (aktiviert / nicht aktiviert)
- die Arbeitszeit
- die Summe der diesem Datum zugeordneten SYMDAT

am Bildschirm ausgegeben werden.

Aus den aufgelisteten Kalendertagen können im Auswahlfeld am Anfang einer jeden Zeile die gewünschten Tage durch Markieren mit „X“ ausgewählt werden.

Die Eingaben sind mit **ENTER** zu bestätigen.

Nachdem die Auswahl mit **ENTER** bestätigt ist, verzweigt das Programm in die Folgemaske D020, in der die ausgewählten Kalendertage bearbeitet werden können.

Wurden mehrere Kalendertage ausgewählt, können diese in der Folgemaske nacheinander bearbeitet werden, ohne dass nach jeder Einzelbearbeitung zur Maske D010 zurückgekehrt werden muss.

**Maske**

Date	Day	Attr	Holiday	Working Hours	Sym.Date Nr.
1995-03-19	SUN	F	N	00:00 - 00:00	0000
1995-03-20	MON	W	N	08:00 - 16:03	0000
1995-03-21	TUE	W	N	08:00 - 16:30	0000
1995-03-22	WED	W	N	08:00 - 16:30	0000
1995-03-23	THU	W	N	08:30 - 16:30	0001
1995-03-24	FRI	W	N	08:00 - 16:30	0000
1995-03-25	SAT	F	N	00:00 - 00:00	0000
1995-03-26	SUN	F	N	00:00 - 00:00	0000
1995-03-27	MON	W	N	08:00 - 16:03	0000
1995-03-28	TUE	W	N	08:00 - 16:30	0000
1995-03-29	WED	W	N	08:00 - 16:30	0000
1995-03-30	THU	W	N	08:30 - 16:30	0001
1995-03-31	FRI	W	N	08:00 - 16:30	0001
1995-04-01	SAT	F	N	00:00 - 00:00	0000

Select with X + Enter  
 Stmt: ( + , - )  
 Msg :

F12=Cancel, F13=Return  
 Goto: - -

LTG TAST

**Feld-Beschreibung**

► -

Am Anfang jeder Zeile der aufgelisteten Kalendertage kann in diesem Eingabefeld der Kalendertag mit „X“ markiert werden.

Er wird dadurch zur weiteren Bearbeitung ausgewählt, die in der Folgemaske D020 durchgeführt werden kann.

Jede Zeile der Maske im Bereich der aufgelisteten Kalendertage enthält die Angaben für *einen* Kalendertag.

**Date**

Datumsangabe für den Kalendertag.

**Day**

Wochentagsangabe für den Kalendertag.

**Attr**

Attributangabe (Arbeitstag / freier Tag) für den Kalendertag.



### Holiday

Feiertagskennzeichen (Feiertag aktiviert / nicht aktiviert, vgl. [Seite 64](#)).

### Working Hours

Beginn und Ende der für diesen Kalendertag festgelegten Arbeitszeit, jeweils in der Form hh:mm (hh=Stunden, mm=Minuten).

### Sym.Date Nr.

Anzahl der SYMDAT, die diesem Kalendertag zugeordnet sind.

- Stmt: ( + , - )

#### Mögliche Angaben:

- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“ oder gleichbedeutend die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#)).

- Goto: - -

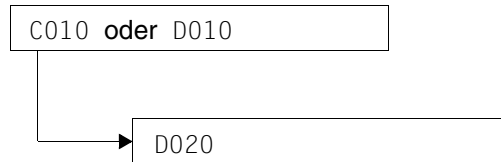
Die Angabe einer voll- oder teilqualifizierten Datumsangabe bewirkt, dass der Bildschirm neu aufgebaut wird. Die ausgegebene Liste beginnt mit dem ersten gefundenen Element. Ist der angegebene Kalendertag nicht vorhanden, wird eine Liste am Bildschirm ausgegeben, die mit dem nächstliegenden Element (Beschreibung auf [Seite 30](#)) beginnt, begleitet von einer entsprechenden Meldung in Meldungszeile 2.

## Folgemaske

Nach Verlassen der Maske mit **[F12]** oder **[F13]** kehrt das Programm zurück in die Ausgangsmaske C010.

Wurde mindestens ein Kalendertag markiert und diese Auswahl mit **[ENTER]** bestätigt, verzweigt das Programm in die Folgemaske D020

### 3.2.5.5 Maske D020: Day Information Kalendertage, d.h. Tagesinformationen bearbeiten



#### Funktionsbeschreibung

Diese Maske gibt hintereinander die Kalendertage aus, die in der Maske D010 zur Bearbeitung ausgewählt wurden.

Für jeden ausgewählten Kalendertag werden

- das Datum
- der Wochentag
- das Attribut (Arbeitstag / freier Tag)
- die Arbeitszeit
- die Anzahl der dem Datum zugeordneten SYMDAT
- falls vorhanden, der Feiertagsname
- die dem Kalendertag zugeordneten SYMDAT-Namen
- die Zeiten dieser SYMDAT

in der Maske ausgegeben.

Nur die Werte in den Feldern „Attr“ (Attribut) und „Working Hours“ (Arbeitszeit), die mit Standardwerten aus den Basis-Informationen vorbelegt sind, können jetzt in dieser Maske für den angezeigten Kalendertag vom Anwender verändert werden.

Die Reihenfolge, in der die zugeordneten SYMDAT aufgelistet werden, kann über das Feld „Change order“ gesteuert werden. Es wird entweder in alphabetischer Reihenfolge (SYMDAT-Namen) oder in numerischer Reihenfolge (Zeitangaben, die den SYMDAT zugeordnet sind) aufgelistet.

Das Umschalten der Sortierfolge erfolgt im Eingabefeld „Change order“:

- durch Angabe des Buchstabens „S“ für Sortierung nach SYMDAT-Namen
- durch Angabe des Buchstabens „T“ für Sortierung nach Zeit

Die Eingaben sind mit **[ENTER]** zu bestätigen.

**Maske**

```

D020      Day Information                                  Calendar Editor Vxxx
Calendar Name: :10SN:$QM211.WERK.KALENDER

Date      Day  Attr  Working Hours  SdNr  Holiday Name
          (W/F/S)  Begin    End
1995-03-31  FRI   S        08:00 - 16:30  0001

Symbolic Date      Time      Symbolic Date      Time
ULTIMO             17:00:00      : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :
                  : :               : :

Store with Enter
Stmt:      ( +, - )    S Change Order      Goto:      F12=Cancel, F13=Return
Msg :

-----
LTG                                             TAST
    
```

**Feld-Beschreibung**

Date

Datumsangabe für den Kalendertag.

Day

Wochentagsangabe für den Kalendertag.

► Attr (W/F/S)

Attributangabe für den Kalendertag:

- W: Arbeitstag
- F: arbeitsfreier Tag
- S: Wert aus der Standard-Arbeitswoche (siehe Beschreibung auf [Seite 12](#))

Working Hours

Begin End

► 00:00 - 23:59

Angaben für Beginn und Ende der für diesen Kalendertag festzulegenden Arbeitszeit, jeweils in der Form hh:mm (hh=Stunden, mm=Minuten).

SdNr

Anzahl der SYMDAT, die diesem Kalendertag zugeordnet sind.

Holiday Name

Feiertagsname für diesen Kalendertag (falls vorhanden).

Symbolic Date

Auflistung der SYMDAT-Namen.

Time

Tageszeit, die einem SYMDAT-Namen zugeordnet ist in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss=Sekunden).

► Stmt: ( +, - )

Mögliche Angaben:

- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“ oder gleichbedeutend die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#))

► \_ Change order

Eingabefeld zum Umschalten der Sortierfolge, in der die dem Kalendertag zugeordneten SYMDAT aufgelistet werden.

Die Reihenfolge ist entweder alphabetisch (SYMDAT-Namen) oder numerisch (Zeitangaben, die den SYMDAT-Namen zugeordnet sind).

Durch die Eingabe des Zeichens "S" oder "T" in dieses Feld wird die Sortierfolge in die jeweils andere umgeschaltet.

► Goto: : :

In dieser Maske ist es von der Sortierfolge (siehe „Change order“) abhängig, ob die Angabe eines SYMDAT-Namens im linken Bereich des Positionierfeldes „Goto“ oder die Angabe einer Zeitangabe (die einem SYMDAT zugeordnet ist) im rechten Bereich dieses Feldes sinnvoll ist. Je nach Sortierlogik ist immer ein Feld gesperrt. Eine Angabe in diesem Feld bei gleichzeitigem Umschalten der Sortierlogik wird nicht mehr ausgewertet (vgl. auch Beschreibung auf [Seite 30](#)).

**Folgemaske**

Nach Verlassen der Maske mit **F12** oder **F13** kehrt das Programm zurück in die Ausgangsmaske C010 oder D010.

### 3.2.5.6 Maske S010: List of Symbolic Dates SYMDAT-Liste ausgeben / zur Auswahl markieren



#### Funktionsbeschreibung

Diese Maske wird angeboten, wenn in der Maske C010 die Funktion „3“ aufgerufen wurde. Wurde keine Vorauswahl für die auszugebenden SYMDAT veranlasst, beginnt die Auflistung mit dem in alphabetischer Reihenfolge ersten SYMDAT.

Die aufgelisteten SYMDAT können in dieser Maske nicht bearbeitet werden.

Es können jedoch SYMDAT

- aus der Liste gelöscht
- zum Kopieren ausgewählt
- zum Bearbeiten ausgewählt
- neu erzeugt werden

Vor jedem SYMDAT-Namen (Symbolic Date) in dieser Maske befindet sich ein Eingabefeld. Verschiedene Eingaben in dieses Auswahlfeld sind möglich und bewirken unterschiedliche Funktionen:

- Eingabe des Zeichens „X“.  
Ein SYMDAT wird zur Bearbeitung ausgewählt.  
Das Programm verzweigt in die Maske S020, in der das SYMDAT zur Bearbeitung angeboten wird.
- Eingabe des Zeichens „C“.  
Ein SYMDAT wird zum Kopieren ausgewählt.  
Das Programm verzweigt in die Maske S020.  
Der Anwender muss dort zwingend einen neuen SYMDAT-Namen in das Feld „Symbolic Date“ eintragen (das nur in diesem Fall zum Eingabefeld wird). Damit ist der Kopiervorgang abgeschlossen und das kopierte Objekt kann in der Maske S020 weiter bearbeitet werden. Nicht verändert werden kann der Typ des SYMDAT, d.h. aus einem nicht-zyklischen SYMDAT kann nach dem Kopieren kein neues zyklisches SYMDAT entstehen und umgekehrt.
- Eingabe des Zeichens „D“.  
Ein mit „D“ markiertes SYMDAT wird aus dem Kalender gelöscht.

*Hinweis*

Alle eingegebenen Zeichen  $\neq$  „C“ oder „D“ werden wie „X“ interpretiert.

Um ein neues SYMDAT zu erzeugen, ist in das Feld „Stmt“ der Wert „New“ einzugeben. Das Programm verzweigt anschließend in die Maske S020. Dort muss, wie beim Kopiervorgang, der Name des neuen SYMDAT in das Feld „Symbolic Date“ eingetragen werden. Anschließend können alle übrigen Werte für das neue SYMDAT angegeben werden.

Die Eingaben sind mit **ENTER** zu bestätigen.

**Maske**

S010 List of Symbolic Dates		Calendar Editor Vxxx	
Calendar Name: :10SN:\$QM211.WERK.KALENDER			
Symbolic Date	Type	Symbolic Date	Type
FRIDAYFORUM	C	ULTIMO	C
Select with X or C(opy) or D(elete) + Enter		F12=Cancel, F13=Return	
Stmt: ( +, -, New )		Goto:	
Msg :			
LTG		TAST	

**Feld-Beschreibung**

► —

Vor jedem SYMDAT-Namen (Symbolic Date) befindet sich in dieser Maske ein Eingabe- bzw. Auswahlfeld, über das der Anwender Objekte auswählt und zugleich festlegt, welche Funktion auf ein ausgewähltes Objekt angewandt wird.

Eingabewerte: „C“, „D“, „X“ oder beliebiges Zeichen  $\neq$  „C“ oder „D“.

Funktionsbeschreibung siehe oben auf [Seite 54](#).

Symbolic Date

Name eines SYMDAT.

Type

Typ eines SYMDAT (zyklisch / nicht-zyklisch).

► Stmt: ( +, -, New )

Mögliche Angaben:

- „New“.  
Dem Programm wird mitgeteilt, dass ein neues SYMDAT angelegt werden soll.  
Es verzweigt zur Maske S020 und erwartet die Eingabe eines neuen SYMDAT-Namens in das Feld „Symbolic Date“.
- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“ oder gleichbedeutend die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#))

► Goto:

Die Angabe eines voll- oder teilqualifizierten SYMDAT-Namens bewirkt, dass der Bildschirm neu aufgebaut wird. Die ausgegebene Liste beginnt mit dem ersten gefundenen Element. Ist das angegebene SYMDAT nicht vorhanden, wird eine Liste am Bildschirm ausgegeben, die mit dem nächstliegenden Element (Beschreibung auf [Seite 30](#)) beginnt, begleitet von einer entsprechenden Meldung in Meldungszeile 2.

## Folgemaske

Nach Verlassen der Maske mit **[F12]** oder **[F13]** kehrt das Programm zurück in die Ausgangsmaske C010.

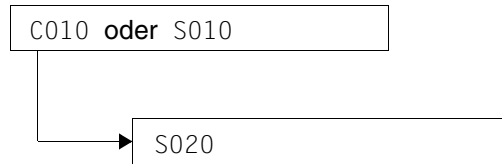
Wurde die Liste der SYMDATs durch Kopieren, Löschen oder die Angabe „New“ verändert, wird sie in der Maske S010 neu aufgebaut.

Wurde ein SYMDAT mit „X“ markiert (oder einem beliebigen Zeichen ≠ „C“ oder „D“) und diese Auswahl mit **[ENTER]** bestätigt, verzweigt das Programm in die Folgemaske D020

Wurde ein SYMDAT mit „C“ markiert oder im Feld „Stmt“ das Erzeugen eines neuen SYMDAT angestoßen und die jeweilige Angabe mit **[ENTER]** bestätigt, verzweigt das Programm in die Folgemaske S020.



### 3.2.5.7 Maske S020: Symbolic Date Information SYMDAT-Informationen ausgeben und bearbeiten.



#### Funktionsbeschreibung

Diese Maske gibt alle Definitionen des ausgewählten SYMDAT aus und listet alle Tage auf, die diesem SYMDAT zugeordnet sind.

Ein *zyklisches SYMDAT* wird durch folgende Definitionen bestimmt:

- SYMDAT-Name (Symbolic Date)
- Typ (Type): C (= zyklisch)
- Zeitangabe (Time)
- Zyklus-Wert (Cycle Value)
- Art des Zyklus (Cycle Type): Tag / Arbeitstag / Woche / Monat
- Alternative (Beschreibung der Free-Day-Rule siehe Seite [Seite 15](#))
- Datumsgrenzen für zugeordnete Tage (Cycle Limits)

Ein *nicht-zyklisches SYMDAT* wird durch folgende Definitionen bestimmt:

- SYMDAT-Name (Symbolic Date)
- Typ (Type): N (=nicht-zyklisch)
- Zeitangabe (Time)
- Zugeordnete Tage (Assigned Dates)

Außer dem SYMDAT-Typ können in dieser Maske alle Werte in den aufgeführten Feldern für das ausgewählte SYMDAT vom Anwender verändert werden. Das Typ-Feld ist immer gesperrt. Nur bei der Erzeugung eines SYMDAT (New) kann das Feld „Type“ mit einem Wert versorgt werden.

Die Eingaben sind mit **ENTER** zu bestätigen.

**Maske**

S020		Symbolic Date Information		Calendar Editor Vxxx	
Calendar Name: :10SN:\$QM211.WERK.KALENDER					
Symbolic Date : ULTIMO		Type : C (N/C)		Time : 17:00:00	
Cycle Value: 0001		Cycle Type: 4		Alternative: 1	
Cycle Limits: Begin: 1995-03-31		1. Day		1. Before	
End: 1995-12-31		2. Workday		2. After	
		3. Week		3. Skip	
		4. Month		4. On	
Assigned Dates	1995-03-31	1995-04-28	1995-05-31	1995-06-30	
	1995-07-31	1995-08-31	1995-09-29	1995-10-31	
	1995-11-30	1995-12-29	- -	- -	
	- -	- -	- -	- -	
	- -	- -	- -	- -	
	- -	- -	- -	- -	
	- -	- -	- -	- -	
	- -	- -	- -	- -	
Store with Enter					F12=Cancel, F13=Return
Stmt: ( +, - )					Goto: - -
Msg :					
LTG			TAST		

**Feld-Beschreibung**

## ► Symbolic Date :

Name des neu zu erzeugenden SYMDAT (wenn in der Maske S010 im Feld „Stmt“ der Wert „New“ eingegeben wurde).

Name des kopierten SYMDAT (wenn in der Maske S010 ein vorhandenes SYMDAT mit „C“ markiert wurde).

## ► Type : \_ (N/C)

Typ eines SYMDAT.

Werte:

N: non-cyclic = nicht-zyklisch

C: cyclic = zyklisch.

(Beschreibung siehe [Seite 14ff.](#)).

Der Typ eines SYMDAT kann prinzipiell nicht verändert werden. Wird dies trotzdem einmal gewünscht, muss das betreffende SYMDAT gelöscht und im Anschluss neu erzeugt werden.

- ▶ Time : 00:00:00

Zeitangabe, die dem SYMDAT zugeordnet ist  
(hh:mm:ss; hh = Stunden, mm = Minuten, ss = Sekunden).

- ▶ Cycle Value: 0000

Nur für zyklische SYMDAT.

Wert zur Berechnung des Zyklusses.

Der Wert wird zusammen mit der Angabe im Feld „Cycle Type“ ausgewertet.

Für nicht-zyklische SYMDAT wird dieses Feld ignoriert.

- ▶ Cycle Type:

Nur für zyklische SYMDAT.

Art des Zyklus, der für das ausgewählte SYMDAT definiert wird.

Die Auswahl erfolgt durch Angabe der zugeordneten Nummer:

1 für Day	Zyklus = Tag
2 für Workday	Zyklus = Arbeitstag
3 für Week	Zyklus = Woche
4 für Month	Zyklus = Monat

Eine genauere Beschreibung finden Sie im einführenden Teil auf [Seite 14ff.](#)

Für nicht-zyklische SYMDAT ist dieses Feld gesperrt.

- ▶ Alternative:

Nur für zyklische SYMDAT.

Festlegung der Alternative für den Fall, dass ein berechneter Eintrag auf einen freien Tag fällt. Die Auswahl erfolgt durch Angabe der zugeordneten Nummer:

1 für Before	d.h. davor eintragen
2 für After	d.h. danach eintragen
3 für Skip	d.h. nicht eintragen
4 für On	d.h. trotzdem eintragen

Eine genauere Beschreibung finden Sie im einführenden Teil auf [Seite 15](#).  
Für nicht-zyklische SYMDAT ist dieses Feld gesperrt.

- ▶ Cycle Limits:      Begin:      - -
- ▶                      End:           - -

Nur für zyklische SYMDAT.

Datumsgrenzen, d.h. Zeitabschnitt, innerhalb dessen die Tage liegen müssen, die dem SYMDAT zugeordnet werden können.

„Begin“ bezeichnet den ersten Tag dieses Zeitabschnitts, d.h. die Untergrenze.

„End“ bezeichnet den letzten Tag des Zeitabschnitts, d.h. die Obergrenze.

Die Werte für „Begin“ und „End“ werden eingegeben in der Form yyyy-mm-dd (yyyy=Jahresangabe 4-stellig; mm=Monatsangabe 2-stellig; dd=Tagesangabe 2-stellig).

Das Feld „Begin“ muss mit einem Wert versorgt werden, eine Angabe im Feld „End“ liegt im Ermessen des Anwenders.

Für nicht-zyklische SYMDAT ist dieses Feld gesperrt.

- ▶ Assigned      yyyy-mm-dd      yyyy-mm-dd      yyyy-mm-dd      yyyy-mm-dd  
    Dates

Datumsangaben für jeden der Kalendertage, die einem *nicht-zyklischen* SYMDAT zugeordnet werden.

Der Wert für jeden Kalendertag wird eingegeben in der Form yyyy-mm-dd (yyyy= Jahresangabe 4-stellig; mm=Monatsangabe 2-stellig; dd=Tagesangabe 2-stellig).

Das Löschen einer Datumsangabe aus der Liste erfolgt durch Überschreiben des Eintrags mit Nullen (0).

Ist der Bereich in dieser Maske, in dem die Datumsangaben aufgelistet werden, nicht groß genug, um alle gewünschten Einträge aufzunehmen, kann der Anwender eine zusätzliche Tabelle anfordern durch Eingabe des Zeichens „+“ in das Feld „Stmt“ (genauer siehe Beschreibung dieses Feldes unten).

Wurde zur Bearbeitung der Daten für diese Maske ein *zyklisches* SYMDAT ausgewählt, so werden hier im Bereich „Assigned Dates“ die berechneten zugeordneten Tage ausgegeben. Diese Datumsangaben können jedoch nicht verändert werden.

► Stmt: ( +, - )

Mögliche Angaben:

- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“.  
Zusätzlich zur Standardfunktion „Blättern vorwärts“ hat dieses Zeichen hier die Funktion, eine Fortsetzungs-Tabelle für den Bereich „Assigned Dates“ anzufordern, falls der Bereich zur Eingabe von Datumsangaben für ein nicht-zyklisches SYMDAT nicht ausreichend ist.
- die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#))

► Goto: : :

Die Angabe eines SYMDAT-Namens bewirkt, dass der Bildschirm neu aufgebaut wird. Die ausgegebene Liste beginnt mit dem ersten gefundenen Element. Ist das angegebene SYMDAT nicht vorhanden, wird eine Liste am Bildschirm ausgegeben, die mit dem nächstliegenden Element (Beschreibung auf [Seite 30](#)) beginnt, begleitet von einer entsprechenden Meldung in Meldungszeile 2.

### Folgemaske

Nach Verlassen der Maske mit **[F12]** oder **[F13]** kehrt das Programm zurück in die Ausgangsmaske C010 oder S010.

### 3.2.5.8 Maske H010: List of Holidays Feiertage auflisten und zur Auswahl markieren



#### Funktionsbeschreibung

Diese Maske wird angeboten, wenn in der Maske C010 die Funktion „4“ aufgerufen wurde. Wurde keine Vorauswahl für die auszugebenden Feiertage veranlasst, beginnt die Auflistung mit dem in alphabetischer Reihenfolge ersten Feiertag.

Die aufgelisteten Feiertage können in dieser Maske nicht bearbeitet werden. Es können jedoch Feiertage

- aus der Liste gelöscht werden
- zum Kopieren ausgewählt werden
- zum Bearbeiten ausgewählt werden
- das Erzeugen eines neuen Feiertags angestoßen werden.

Vor jedem Feiertags-Namen (Holiday) in dieser Maske befindet sich ein Eingabefeld. Folgende Werte können in dieses Feld eingegeben werden:

- Eingabe des Zeichens „X“. Der markierte Feiertag wird zur Bearbeitung ausgewählt. Das Programm verzweigt in die Maske H020, in der die Informationen des Feiertags (ausgenommen sein Name und sein Typ) verändert werden können.
- Eingabe des Zeichens „C“. Der markierte Feiertag wird zum Kopieren ausgewählt. Das Programm verzweigt in die Maske H020. Der Anwender muss dort zwingend einen neuen Feiertags-Namen in das Feld „Holiday“ eintragen (das nur in diesem Fall zum Eingabefeld wird). Alle übrigen Informationen werden von dem kopierten Feiertag übernommen und können, mit Ausnahme des Typs, vom Anwender verändert werden.
- Eingabe des Zeichens „D“. Ein mit „D“ markierter Feiertag wird aus dem Kalender gelöscht.

#### *Hinweis*

Alle eingegebenen Zeichen  $\neq$  „C“ oder „D“ werden wie „X“ interpretiert.

Um einen neuen Feiertag zu erzeugen, ist in das Feld „Stmt“ der Wert „New“ einzugeben. Das Programm verzweigt anschließend in die Maske H020. Dort muss, wie beim Kopiervorgang, der Name des neuen Feiertags in das Feld „Holiday“ eingetragen werden. Anschließend können alle übrigen Werte für diesen Feiertag angegeben werden.

Die Eingaben sind mit **ENTER** zu bestätigen.

## Maske

```

H010      List of Holidays                      Calendar Editor Vxxx
Calendar Name: :10SN:$QM211.WERK.KALENDER

          Holiday                               Type       Active
          CHRISTMAS                             C             Y
          EARLYMAYDAY                           C             Y
          EASTER                                 N             Y

Select with X or C(opy) or D(elete) + Enter
Activate with Y(es) or N(o) + Enter
Stmt:      ( +, -, New)                        Goto:         F12=Cancel, F13=Return
Msg :

-----
LTG                                             TAST
  
```

## Feld-Beschreibung

► —

Vor jedem Feiertags-Namen (Holiday) befindet sich in dieser Maske ein Eingabe- bzw. Auswahlfeld, über das der Anwender Objekte auswählt und zugleich festlegt, welche Funktion auf ein ausgewähltes Objekt angewandt wird.

Eingabewerte: „C“, „D“, „X“ oder beliebiges Zeichen ≠ „C“ oder „D“.

Funktionsbeschreibung siehe oben auf [Seite 62](#)

Holiday

Name des Feiertags.

Type

Typ des Feiertags: zyklisch / nicht-zyklisch

► Active

Für jeden Feiertag wird in diesem Feld am Ende der Maskenzeile festgelegt, ob er aktiviert werden soll oder nicht.

Aktivieren heißt, für den betreffenden Kalender festlegen, ob dieser Feiertag als Feiertag behandelt werden soll. Nur ein „aktiver“ Feiertag wird bei der Entscheidung, ob ein Tag ein Arbeitstag oder ein freier Tag ist, als Feiertag berücksichtigt.

Werte:

„Y“ (yes): der Feiertag wird aktiviert.

„N“ (no): der Feiertag wird nicht aktiviert.

► Stmt:( +, -, New)

Mögliche Angaben:

- „New“.  
Dem Programm wird mitgeteilt, dass ein neuer Feiertag angelegt werden soll. Es verzweigt zur Maske H020 und erwartet die Eingabe eines neuen Feiertags-Namens in das Feld „Holiday“.
- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“ oder gleichbedeutend die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#)).

► Goto:

Die Angabe eines voll- oder teilqualifizierten Feiertags-Namens bewirkt, dass der Bildschirm neu aufgebaut wird. Die ausgegebene Liste beginnt mit dem ersten gefundenen Element. Ist der angegebene Feiertag nicht vorhanden, wird eine Liste am Bildschirm ausgegeben, die mit dem nächstliegenden Element (Beschreibung auf [Seite 30](#)) beginnt, begleitet von einer entsprechenden Meldung in Meldungszeile 2.

## Folgemaske

Nach Verlassen der Maske mit **[F12]** oder **[F13]** kehrt das Programm zurück in die Ausgangsmaske C010.

Wurde

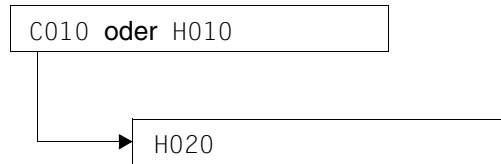
- ein Feiertag mit „X“ markiert (oder einem beliebigen Zeichen ≠ „C“ oder „D“)
- ein Feiertag mit „C“ markiert
- im Feld „Stmt“ das Erzeugen eines neuen Feiertags angestoßen

und die jeweilige Funktionswahl mit **[ENTER]** bestätigt, verzweigt das Programm in die Folgemaske H020



Wurde die Liste der Feiertage durch Kopieren, Löschen oder die Angabe „New“ verändert, wird sie in der Maske H010 neu aufgebaut.

### 3.2.5.9 Maske H020: Holiday Information Feiertags-Informationen ausgeben und bearbeiten.



#### Funktionsbeschreibung

Diese Maske gibt alle Definitionen des ausgewählten Feiertags aus. Ist der Feiertag ein nicht-zyklischer, werden alle Tage aufgelistet, die diesem Feiertag zugeordnet sind.

Feiertage werden eingeteilt in

- zyklische Feiertage
- nicht-zyklische Feiertage.

Zyklische Feiertage fallen jedes Jahr auf den gleichen Tag, z.B. Weihnachten. Nicht-zyklische Feiertage sind veränderlich, z.B. Ostern.

Ein zyklischer Feiertag wird durch folgende Definitionen bestimmt:

- Feiertags-Name (Holiday)
- Typ (Type): C (= zyklisch)
- Monats- und Tagesangabe (Date for Cyclic Holiday)

Ein nicht-zyklischer Feiertag wird durch folgende Definitionen bestimmt:

- Feiertags-Name (Holiday)
- Typ (Type): N (=nicht-zyklisch)
- Zugeordnete Tage (Assigned Dates)

Alle Werte in den aufgeführten Feldern können in dieser Maske für den ausgewählten Feiertag - nach Maßgabe seines Typs - vom Anwender verändert werden.

Die Eingaben sind mit **ENTER** zu bestätigen.

## Maske

```
H020      Holiday Information                      Calendar Editor Vxxx
Calendar Name: :10SN:$QM211.WERK.KALENDER

Holiday: CHRISTMAS                               Type: C (N/C)

Date for Cyclic Holiday: 12-25

Assigned   1994-12-25      1995-12-25      1996-12-25      - -
Dates
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -
          - -             - -             - -             - -

Store with Enter
Stmt:      ( +, - )
Msg :

                                 F12=Cancel, F13=Return
                                 Goto:      - -

-----
LTG                                           TAST
```

## Feld-Beschreibung

► Holiday:

Name des zur Bearbeitung ausgewählten Feiertags (wenn in der Maske H010 dieser Feiertag zur weiteren Bearbeitung mit dem Zeichen „X“ markiert wurde; ein beliebiges Zeichen ≠ „C“ oder „D“ wird wie „X“ interpretiert.).

Name des kopierten Feiertags (wenn in der Maske H010 ein vorhandener Feiertag mit „C“ markiert wurde).

Name des neu zu erzeugenden Feiertags (wenn in der Maske H010 im Feld „Stmt“ der Wert „New“ eingegeben wurde).

► Type : (N/C)

Typ eines Feiertags.

Feiertage, die jährlich zum selben Tag wiederkehren, sind vom Typ „zyklisch“ (Jahreszyklus). Veränderliche Feiertage sind vom Typ „nicht-zyklisch“.

Werte:

N: non-cyclic = nicht-zyklisch

C: cyclic = zyklisch.

- Date for Cyclic Holiday: -

Nur für zyklische Feiertage.

Monats- und Tagesangabe für einen zyklischen Feiertag in der Form mm-dd (mm=Monatsangabe 2-stellig; dd=Tagesangabe 2-stellig).

Die Definition gilt für jedes Jahr innerhalb der festgelegten Kalendergrenzen.

*Beispiel* (Definition der Neujahrstage für die Zeitspanne eines Kalenders)

01-01

- Assigned Dates    yyyy-mm-dd    yyyy-mm-dd    yyyy-mm-dd    yyyy-mm-dd

Nur für nicht-zyklische Feiertage.

Datumsangaben für jeden der Kalendertage, die einem *nicht-zyklischen* Feiertag zugeordnet werden.

Der Wert für jeden Kalendertag wird eingegeben in der Form yyyy-mm-dd (yyyy= Jahresangabe 4-stellig; mm=Monatsangabe 2-stellig; dd=Tagesangabe 2-stellig).

Das Löschen einer Datumsangabe aus der Liste erfolgt durch Überschreiben des Eintrags mit Nullen (0).

Ist der Bereich in dieser Maske, in dem die Datumsangaben aufgelistet werden, nicht groß genug, um alle gewünschten Einträge aufzunehmen, kann der Anwender eine zusätzliche Tabelle anfordern durch Eingabe des Zeichens „+“ in das Feld „Stmt“ (genauer siehe Beschreibung dieses Feldes unten).

- Stmt:( +, - )

Mögliche Angaben:

- das Zeichen „-“ oder gleichbedeutend die Zeichenfolge „F7“
- das Zeichen „+“.

Zusätzlich zur Standardfunktion „Blättern vorwärts“ hat dieses Zeichen hier die Funktion, eine Fortsetzungs-Tabelle für den Bereich „Assigned Dates“ anzufordern, falls der Bereich zur Eingabe von Datumsangaben für einen nicht-zyklischen Feiertag nicht ausreichend ist.

- die Zeichenfolge „F8“
- die Zeichenfolge „F12“
- die Zeichenfolge „F13“

Die Wirkung ist jeweils identisch mit dem Betätigen der gleichnamigen Funktionstaste (Übersicht siehe [Seite 33](#)).

## ▶ Goto:                                   :   :

Ist der zu bearbeitende Feiertag vom Typ „nicht-zyklisch“, bewirkt die Angabe einer voll- oder teilqualifizierten Datumsangabe in diesem Feld, dass der Bildschirm neu aufgebaut wird. Die ausgegebene Liste beginnt mit dem ersten gefundenen Element. Ist der angegebene Tag nicht vorhanden, wird eine Liste am Bildschirm ausgegeben, die mit dem nächstliegenden Element (Beschreibung auf Seite 30) beginnt, begleitet von einer entsprechenden Meldung in Meldungszeile 2.

**Folgemaske**

Nach Verlassen der Maske mit [F12] oder [F13] kehrt das Programm zurück in die Ausgangsmaske C010 oder H010.



---

## 4 Programmschnittstelle

Das Subsystem CALENDAR bietet eine Programmschnittstelle für Assembler und C.  
Schnittstellename für Assembler: CALENDR.  
Schnittstellename für C: CALEND.h.

Über die Schnittstelle können Kalenderdaten erzeugt, geändert und zur Ausgabe angefordert werden. Die Funktionsaufrufe werden über den Operanden FUNCT gesteuert, wobei folgende Reihenfolge eingehalten werden muss:

1. Kalender öffnen bzw. erzeugen.
2. Daten erzeugen, ändern, löschen, ausgeben.
3. Kalender schließen.

Eine Übersicht der möglichen Funktionsaufrufe finden Sie auf Seite 97. Im Anschluss daran werden die Funktionsaufrufe in alphabetischer Reihenfolge ausführlich beschrieben.

Die Objekte einer Kalenderdatei: Basis-Informationen, Kalendertage, SYMDAT, Feiertage, auf die über die Programmschnittstelle zugegriffen werden kann, sind im Abschnitt „Aufbau eines Kalenders“ (Seite 11ff) beschrieben.

Aus Performancegründen sollten folgende Regeln immer berücksichtigt werden:

- Der Kalender sollte nur für den tatsächlich benötigten Zeitraum angelegt werden.
- Bei der Erzeugung eines Kalenders sollten vor der Erzeugung von SYMDAT zuerst die Standard-Arbeitswoche, die Kalendertage und die Feiertage definiert werden.
- Bei der Änderung des Attributes für mehrere Kalendertage sollte zwischendurch keine Show-Funktion (SH...) aufgerufen werden.
- Werden mehrere Feiertage hintereinander erzeugt oder gelöscht oder aktiviert oder deaktiviert, sollte zwischendurch keine Show-Funktion (SH...) aufgerufen werden.

Performance-kritisch sind

- das Öffnen eines sehr großen Kalenders.  
Die Größe des Kalenders hängt von der Anzahl der Kalendertage innerhalb der Kalendergrenzen und von der Anzahl an SYMDAT und Feiertagen ab.

- die Berechnung von zyklischen SYMDAT bei einer großen Anzahl von zyklischen SYMDAT.  
Die SYMDAT-Berechnung erfolgt erst, wenn über Show-Funktionen Daten angefordert werden, bei denen zyklische SYMDAT betroffen sind.  
Die SYMDAT-Berechnung wird durch folgende Aktionen ausgelöst:
  - Änderung des Attributes eines Wochentags in der Standard-Arbeitswoche
  - Änderung des Attributes eines Kalendertags
  - Aktivieren oder Deaktivieren von Feiertagen.



## 4.1 Formatübersicht

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,XPAND = PARAM / ENTSTDW / ENTDATE / ENTDAY / ENTSYM /         ENTSYMS / ENTHOL / STDWEEK / CCB / DAYLIST / BASINF /         LODAY / DAYINF / DAYHOL / LOSYM / SYMINF / LOHOL /         HOLINF / NESTM  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,FUNCT = *CREACAL / *OPENCAL / *CLSECAL / *SAVECAL / *MODBAS /         *MODDAY / *CRENSYM / *CRECSYM / *DELSYM / *MODNSYM /         *MODCSYM / *CRENHOL / *CRECHOL / *DELHOL / *MODNHOL /         *MODCHOL / *AKTHOL / *DEAKHOL / *SHBAS / *SHLODAY /         *SHDAYIN / *SHDAYHL / *SHLOSVM / *SHSYMIN / *SHLOHOL /         *SHHOLIN / *SHNESTM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,CLDNAME = &lt;c-string 1..54: filename 1..54 with catid&gt; /           &lt;var: char:54 filename 1..54 with catid&gt; /           &lt;reg: A(char:54 filename 1..54 with catid)&gt;  ,OPENMOD = *READ / *READALL / *UPDATE / &lt;var: enum-of _open_s:1&gt;           / &lt;reg: enum-of _open_s:1&gt;  ,LIMIT1 = *UNCHANGED / *STD / &lt;c-string 10..10: date 10..10&gt;           &lt;var: char:10: date 10..10&gt; /           &lt;reg: A(char:10: date 10..10)&gt;  ,LIMIT2 = *UNCHANGED / *STD / &lt;c-string 10..10: date 10..10&gt;           &lt;var: char:10: date 10..10&gt;           &lt;reg: A(char:10: date 10..10)&gt;  ,WEEKADR = *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,SAVE = *<u>YES</u> / *NO / &lt;var: enum-of _save_s:1&gt;         &lt;reg: enum-of _save_s:1&gt; </pre>

Makro	Operanden
CALENDR (Forts.)	<pre> ,DATE = &lt;c-string 10..10:  date 10..10&gt; /       &lt;var: char:10:  date 10..10&gt; /       &lt;reg: A(char:10:  date 10..10)&gt;  ,ATTRIBT = *UNCHANGED / *FREE / *WORK / *STD /           &lt;var: enum-of _attribute_s:1&gt; /           &lt;reg: enum-of _attribute_s:1&gt;  ,WKTIME1 = *UNCHANGED / *STD / &lt;c-string 5..5:  time 5..5&gt; /           &lt;var: char:5:  time 5..5&gt; /           &lt;reg: A(char:5:  time 5..5)&gt;  ,WKTIME2 = *UNCHANGED / *STD / &lt;c-string 5..5:  time 5..5&gt; /           &lt;var: char:5:  time 5..5&gt;           &lt;reg: A(char:5:  time 5..5)&gt;  ,NAME = &lt;c-string 1..30:  name 1..30&gt; /        &lt;var: char:30:  name 1..30&gt; /        &lt;reg: A(char:30:  name 1..30)&gt;  ,ADDDAY = (adresse, anzahl)          adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)          anzahl  : &lt;integer 0..1860&gt; /                  &lt;var: int:4:  integer 0..1860&gt;                  &lt;reg: int:4:  integer 0..1860&gt;  ,DELDAY = (adresse, anzahl)          adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)          anzahl  : &lt;integer 0..1860&gt; /                  &lt;var: int:4:  integer 0..1860&gt;                  &lt;reg: int:4:  integer 0..1860&gt;  ,TIME = *UNCHANGED / *STD / &lt;c-string 8..8:  time 8..8&gt; /        &lt;var: char:8:  time 8..8&gt; / &lt;reg: A(char:8:  time 8..8)&gt;  ,CYCLTYP = *UNCHANGED / *DAY / *WORKDAY / *WEEK / *MONTH /           &lt;var: enum-of _cycle_s:1&gt;           &lt;reg: enum-of _cycle_s:1&gt;  ,CYCLVAL = *UNCHANGED / &lt;integer 1..9999&gt; /           &lt;var: int:4:  integer 1..9999&gt;           &lt;reg: int:4:  integer 1..9999&gt;  ,ALTERN = *UNCHANGED / *BEFORE / *AFTER / *SKIP / *ON /           &lt;var: enum-of _alternative_s:1&gt; /           &lt;reg: enum-of _alternative_s:1&gt; </pre>

Makro	Operanden
CALENDR (Forts.)	<pre> ,DATE1 = *UNCHANGED / &lt;c-string 10..10: date 10..10&gt; / *FIRST /         *NEXT / &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *UNCHANGED / *UNDEFINED / *LAST /         &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,NAME1 = *FIRST / &lt;c-string 1..30: name 1..30&gt;         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,NAME2 = *LAST / &lt;c-string 1..30: name 1..30&gt;         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,NUMB = *NO / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt;  ,OUTPUT = (adresse, länge)         adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)         länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,SORT = *TIME / *SYMDAT / &lt;var: enum-of _sort_s:1&gt; /         &lt;reg: enum-of _sort_s:1&gt; </pre>

## 4.2 Operandenbeschreibung

### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff. Der Operand Version = 1 muss immer angegeben werden.

### XPAND=

Erzeugung von Layouts

- für die Parameterliste
- für die Standard-Arbeitswoche
- für Bereiche, die eine Liste von Tagen enthalten
- für verschiedene Ausgabebereiche.

Operandenwerte:

<b>PARAM</b>	Layout für die Parameterliste.
<b>ENTSTDW</b>	Layout für die Standard-Arbeitswoche.
<b>ENTDATE</b>	Layout für einen Datums-Eintrag.
<b>ENTDAY</b>	Layout für einen Tages-Eintrag.
<b>ENTSYM</b>	Layout für einen SYMDAT-Eintrag.
<b>ENTSYMS</b>	Layout für einen SYMDAT-Kurzeintrag.
<b>ENTHOL</b>	Layout für einen Feiertags-Eintrag.
<b>STDWEEK</b>	Layout für die Standard-Arbeitswoche (Kurzeintrag).
<b>CCB</b>	Layout für den Control-Block.
<b>DAYLIST</b>	Layout für eine Liste von Kalendertagen, die zugeordnet oder gelöscht werden sollen.
<b>BASINF</b>	Layout für die Basis-Informationen.
<b>LODAY</b>	Layout für eine Liste von Kalendertagen.
<b>DAYINF</b>	Layout für einen Kalendertag mit zugeordneten SYMDAT.
<b>DAYHOL</b>	Layout für einen Kalendertag mit zugeordneten Feiertagen.
<b>LOSYM</b>	Layout für eine Liste von SYMDAT.
<b>SYMINF</b>	Layout für ein SYMDAT mit zugeordneten Kalendertagen.
<b>LOHOL</b>	Layout für eine Liste von Feiertagen.
<b>HOLINF</b>	Layout für einen Feiertag mit zugeordneten Kalendertagen.
<b>NESTM</b>	Layout fuer Zeit (DATE, TIME) und Symdatnam

### VERSION=1

Version der Schnittstelle. Das Layout der Ausgabebereiche wird versionsspezifisch festgelegt. Außerdem wird überprüft, ob die Version vom Subsystem unterstützt wird. Aktuell wird nur Version 1 unterstützt.

**CALLER=**

Wird nur bei MF=E ausgewertet.

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR). Entsprechend dieser Angabe wird der Funktionsaufruf erzeugt.

**USER / SYSTEM**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*CREACAL / \*OPENCAL / \*CLSECAL / \*SAVECAL / \*MODBAS / \*MODDAY  
 \*CRENSYM / \*CRECSYM / \*DELSYM / \*MODNSYM / \*MODCSYM / \*CRENHOL  
 \*CRECHOL / \*DELHOL / \*MODNHOL / \*MODCHOL / \*AKTHOL / \*DEAKHOL  
 \*SHBAS / \*SHLODAY / \*SHDAYIN / \*SHDAYHL / \*SHLOSYM / \*SHSYMIN  
 \*SHLOHOL / \*SHHOLIN / \*SHNESTM / <var: enum-of \_funct\_s:1> /  
 <reg: enum-of \_funct\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**CCB=**

Control-Block. Die Adresse des Control-Blocks muss bei jedem Funktionsaufruf für eine Kalenderdatei angegeben werden. Der Control-Block darf vom Anwender nicht verändert werden. Beim Öffnen einer Kalenderdatei wird dieser Control-Block automatisch versorgt, beim Schließen der Kalenderdatei werden die Werte wieder gelöscht.

Nach den Funktionsaufrufen zum Erzeugen eines neuen Kalenders bzw. zum Öffnen eines vorhandenen Kalenders wird die betreffende Kalenderdatei über die Control-Block-Adresse angesprochen und nicht mehr über den Dateinamen.

Werden mehrere Kalender gleichzeitig geöffnet, benötigt der Anwender für jede Kalenderdatei einen eigenen Control-Block.

Das Layout für den Bereich wird mit MF=D und XPAND=CCB erzeugt

**<var: pointer> / (<reg: pointer>)**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**CLDNAME=**

Name der Kalenderdatei, d.h. des Kalenders.

Der Operand wird nur beim Erzeugen eines neuen bzw. Öffnen eines vorhandenen Kalenders ausgewertet.

**<c-string 1..54: filename 1..54 with catid> /  
 <var: char:54 filename 1..54 with catid> /  
 <reg: A(char:54 filename 1..54 with catid)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**OPENMOD=**

Open-Modus. Mit dem Eröffnungsmodus wird entschieden, wie der Anwender auf die Daten der Kalenderdatei zugreifen kann.

**\*READ / \*READALL / \*UPDATE / <var: enum-of \_open\_s:1> /**

**<reg: enum-of \_open\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**LIMIT1=**

Untergrenze des Kalenders, d.h. Datum des ersten Kalendertags der Kalenderdatei. Die Untergrenze muss kleiner sein als die Obergrenze.

**\*UNCHANGED / \*STD / <c-string 10..10: date 10..10> / <var: char:10: date 10..10> /**

**<reg: A(char:10: date 10..10)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**LIMIT2=**

Obergrenze des Kalenders, d.h. Datum des letzten Kalendertags der Kalenderdatei. Die Obergrenze muss größer sein als die Untergrenze.

Der Zeitraum zwischen Unter- und Obergrenze eines Kalenders darf nicht größer sein als fünf Jahre.

**\*UNCHANGED / \*STD / <c-string 10..10: date 10..10> /**

**<var: char:10: date 10..10> / <reg: A(char:10: date 10..10)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**WEEKADR=**

Adresse des Bereichs, der die Standard-Arbeitswoche enthält.

Die Angabe ist möglich beim Erzeugen eines Kalenders (FUNCT=\*CREACAL) und beim Ändern der Basis-Informationen (FUNCT=\*MODBAS).

Der Datenbereich enthält für jeden Wochentag einen Eintrag für den Namen des Wochentags, für das Attribut des Wochentags (Arbeitstag / freier Tag) und für die Standard-Arbeitszeit dieses Wochentags.

Wird der Operand mit der Funktion \*MODBAS, also beim Ändern der Basis-Informationen verwendet, kann als Eingabebereich derjenige genommen werden, den der Anwender mit der Funktion \*SHBAS als Ausgabe erhält. Dieser Bereich kann vorher modifiziert werden.

**\*NO / <var: pointer> / (<reg: pointer>)**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**SAVE=**

Festlegung, ob die in der Kalenderdatei durchgeführten Änderungen gespeichert werden. Der Operand wird nicht ausgewertet, wenn der Kalender mit OPENMOD=\*READ oder \*READALL geöffnet wurde, da in diesen Fällen Änderungen prinzipiell nicht möglich sind..

**\*YES / \*NO / <var: enum-of \_save\_s:1> / <reg: enum-of \_save\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**DATE=**

Datum eines Kalendertags, der ausgegeben oder verändert werden soll.

Die Angabe muss als vollständiges Datum in der Form yyyy-mm-dd (yyyy=Jahresangabe, mm=Monatsangabe, dd=Tagesangabe) erfolgen.

Für einen zyklischen Feiertag muss als Jahresangabe die Zeichenkette „\*\*\*\*“ angegeben werden. In diesem Fall wird über diesen Operanden der Tag definiert, der dem Feiertag zugeordnet ist.

Zusammen mit dem Operanden TIME kann der Operand einen Zeitpunkt angeben, ab dem die nächste SYMDAT-Zeit ausgegeben werden soll.

**<c-string 10..10: date 10..10> / <var: char:10: date 10..10> /  
<reg: A(char:10: date 10..10)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**ATTRIBT=**

Attribut eines Kalendertags.

Der Kalendertag kann entweder als Arbeitstag oder als freier Tag definiert werden.

**\*UNCHANGED / \*FREE / \*WORK / \*STD / <var: enum-of \_attribute\_s:1> /  
<reg: enum-of \_attribute\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**WKTIME1=**

Beginn der Arbeitszeit für den entsprechenden Kalendertag. Die Angabe erfolgt in der Form hh:mm (hh=Stunden, mm=Minuten).

**\*UNCHANGED / \*STD / <c-string 5..5: time 5..5> / <var: char:5: time 5..5> /  
<reg: A(char:5: time 5..5)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**WKTIME2=**

Ende der Arbeitszeit für den entsprechenden Kalendertag. Die Angabe erfolgt in der Form hh:mm (hh=Stunden, mm=Minuten).

**\*UNCHANGED / \*STD / <c-string 5..5: time 5..5> / <var: char:5: time 5..5> /  
<reg: A(char:5: time 5..5)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**NAME=**

Name eines SYMDAT (Beschreibung auf Seite 14ff) bzw. eines Feiertags (Beschreibung auf Seite 17f).

Es muss ein vollständiger Name angegeben werden. Der Name eines SYMDAT kann maximal 20 Zeichen, der Name eines Feiertags maximal 30 Zeichen lang sein.

**<c-string 1..30: name 1..30> / <var: char:30: name 1..30> /  
<reg: A(char:30: name 1..30)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**ADDDAY = (adresse, anzahl)**

Datenbereich, der die Kalendertage enthält, die beim Erzeugen und Ändern von nicht-zyklischen SYMDAT oder nicht-zyklischen Feiertagen diesen zugeordnet werden sollen. Der Bereich enthält für jeden zugeordneten Kalendertag das Datum. Das Layout für den Bereich wird mit MF=D und XPAND=DAYLIST erzeugt. Das Layout für einen einzelnen Datumseintrag erhält man mit XPAND=ENTDATE. Damit ist der Zugriff auf die Angabe für Jahr, Monat und Tag möglich.

**adresse:** \*NO / <var: pointer> / (<reg: pointer>)

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**anzahl :** <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**DELDAY = (adresse, anzahl)**

Datenbereich mit Kalendertagen, deren Zuordnung zu nicht-zyklischen SYMDAT oder nicht-zyklischen Feiertagen bei deren Bearbeitung gelöscht werden soll. Der Bereich enthält für jeden Kalendertag, zu dem die Zuordnung gelöscht werden soll, das Datum. Das Layout für den Bereich wird mit MF=D und XPAND=DAYLIST erzeugt. Das Layout für einen einzelnen Datumseintrag erhält man mit XPAND=ENTDATE. Damit ist der Zugriff auf die Angabe für Jahr, Monat und Tag möglich.

**adresse:** \*NO / <var: pointer> / (<reg: pointer>)

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**anzahl :** <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**TIME=**

Zeitangabe, die einem SYMDAT zugeordnet werden kann. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss= Sekunden). Zusammen mit dem Operanden DATE kann der Operand einen Zeitpunkt angeben, ab dem die naechste SYMDAT-Zeit ausgegeben werden soll. In diesem Fall sind die Operandenwerte \*UNCHANGED und \*STD nicht zulaessig.

\*UNCHANGED / \*STD / <c-string 8..8: time 8..8> / <var: char:8: time 8..8> /  
<reg: A(char:8: time 8..8)>

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**CYCLTYP=**

Art des Zyklus (Beschreibung auf Seite 15f). Der Operand definiert zusammen mit den Operanden CYCLVAL und ALTERN für zyklische SYMDAT die Zyklus-Art.



**\*UNCHANGED / \*DAY / \*WORKDAY / \*WEEK / \*MONTH /**  
**<var: enum-of \_cycle\_s:1> / <reg: enum-of \_cycle\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### **CYCLVAL=**

Zyklus-Wert.

Der Operand bestimmt für zyklische SYMDAT- abhängig von der Art des Zyklus (CYCLTYP) - die Anzahl der Tage bzw. Arbeitstage bzw. Wochen bzw. Monate, die zwischen zwei Einträgen liegen. Die Angabe bezieht sich also immer auf den Operanden CYCLTYP.

**\*UNCHANGED / <integer 1..9999> / <var: int:4: integer 1..9999> /**  
**<reg: int:4: integer 1..9999>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### **ALTERN=**

Alternative (siehe Beschreibung der Free-Day-Rule auf Seite 15).

Der Operand legt bei zyklischen SYMDAT die Alternative für den Fall fest, dass ein berechneter SYMDAT-Eintrag auf einen freien Tag fällt.

**\*UNCHANGED / \*BEFORE / \*AFTER / \*SKIP / \*ON /**  
**<var: enum-of \_alternative\_s:1> / <reg: enum-of \_alternative\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### **DATE1=**

Erstes Datum eines Bereichs.

Der Operand definiert bei zyklischen SYMDAT das Start-Datum für die Berechnung des zyklischen SYMDAT. Erforderlich ist in diesem Fall die Angabe eines vollständigen Datums in der Form yyyy-mm-dd (yyyy=Jahresangabe, mm=Monatsangabe, dd=Tagesangabe). Wird dieses Start-Datum geändert, ändern sich auch die berechneten SYMDAT-Einträge. Für die Ausgabe von Kalenderdaten dient der Operand zur Einschränkung der Ausgabe. Er definiert den ersten Eintrag eines Bereichs von Kalendertagen.

Die Tages- und Monatsangabe kann weggelassen werden: Gültig sind die Angaben yyyy-mm-dd, yyyy-mm oder yyyy.

Der letzte Eintrag des Ausgabebereichs kann entweder über den Operanden DATE2 oder über den Operanden NUMB definiert werden.

**\*UNCHANGED / <c-string 10..10: date 10..10> / \*FIRST / \*NEXT /**  
**<var: char:10: date 10..10> / <reg: A(char:10: date 10..10)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### **DATE2=**

Letztes Datum eines Bereichs.

Der Operand definiert bei zyklischen SYMDAT das Ende-Datum für die Berechnung des zyklischen SYMDAT. Erforderlich ist in diesem Fall die Angabe eines vollständigen Datums in der Form yyyy-mm-dd (yyyy=Jahresangabe, mm=Monatsangabe, dd=Tagesangabe).

Liegt das Ende-Datum außerhalb der Kalendergrenzen, werden die SYMDAT-Einträge

trotzdem nur bis zur Kalendergrenze berechnet.

Für die Ausgabe von Kalenderdaten dient der Operand zur Einschränkung der Ausgabe. Er definiert den letzten Eintrag eines Bereichs von Kalendertagen.

Die Tages- und Monatsangabe kann weggelassen werden: Gültig sind die Angaben yyyy-mm-dd, yyyy-mm oder yyyy.

Der erste Eintrag des Ausgabebereichs wird über den Operanden DATE1 definiert.

**\*UNCHANGED / \*UNDEFINED / \*LAST / <c-string 10..10: date 10..10> / <var: char:10: date 10..10> / <reg: A(char:10: date 10..10)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### NAME1=

Erster Eintrag eines Bereichs von SYMDAT oder Feiertagen.

Der Name eines SYMDAT kann maximal 20 Zeichen, der Name eines Feiertags maximal 30 Zeichen lang sein. Eine teilqualifizierte Angabe des Namens, mit dem Zeichen „\*“ am Ende des Namens, ist erlaubt. Auf diese Weise kann der Anwender die Ausgabe von SYMDAT- oder Feiertags-Einträgen einschränken.

Der letzte Eintrag des Bereichs kann entweder über den Operanden NAME2 oder über den Operanden NUMB definiert werden.

**\*FIRST / <c-string 1..30: name 1..30> / <var: char:30: name 1..30> / <reg: A(char:30: name 1..30)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### NAME2=

Letzter Eintrag eines Bereichs von SYMDAT oder Feiertagen.

Der Name eines SYMDAT kann maximal 20 Zeichen, der Name eines Feiertags maximal 30 Zeichen lang sein. Eine teilqualifizierte Angabe des Namens, mit dem Zeichen „\*“ am Ende des Namens, ist erlaubt. Auf diese Weise kann der Anwender die Ausgabe von SYMDAT- oder Feiertags-Einträgen einschränken.

Der erste Eintrag des Bereichs wird über den Operanden NAME1 definiert.

**\*LAST / <c-string 1..30: name 1..30> / <var: char:30: name 1..30> / <reg: A(char:30: name 1..30)>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

### NUMB=

Anzahl der gewünschten Einträge. Die Angabe schränkt die Ausgabe der Daten ein. Die Angabe bezieht sich immer auf einen der Operanden NAME1 oder DATE1.

Die Angabe eines negativen Werts bewirkt die Ausgabe der Einträge, die vor dem durch NAME1 oder DATE1 definierten Eintrag liegen.

Ist der Operandwert für NUMB ≠ \*NO, werden die Operanden NAME2 und DATE2 nicht ausgewertet.

**\*NO / <integer -4096..4096> / <var: int:4: integer -4096..4096> / <reg: int:4: integer -4096..4096>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**OUTPUT=**

Ausgabebereich.

Bei allen Ausgabefunktionen muss die Adresse und Länge eines Ausgabebereichs angegeben werden, in dem die gewünschten Daten abgelegt werden sollen.

Ist der Ausgabebereich zu klein, erhält der Anwender einen entsprechenden Returncode ausgegeben, der ihn darüber informiert, dass der Ausgabebereich keine oder nur einen Teil der Daten enthält.

**adresse: <var: pointer> / (<reg: pointer>)**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**länge : <integer 0..1000000> / <var: int:4: integer 0..1000000>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

**SORT=**

Nur für die Funktion \*SHDAYIN (Tagesinformationen ausgeben).

Sortierkriterium: Die zugeordneten SYMDAT sollen entweder nach den SYMDAT-Namen oder nach der Zeit sortiert werden.

**\*TIME / \*SYMDAT / <var: enum-of \_sort\_s:1> / <reg: enum-of \_sort\_s:1>**

Beschreibung der Operandenwerte siehe Einzelformate Seite 97ff.

## 4.3 Datenbereiche für Funktionen

Für mehrere Funktionen müssen Bereiche mit spezifischem Layout zur Verfügung gestellt werden, die über den Operanden XPAND aufgebaut werden.

Für die Funktionen \*CREACAL und \*MODBAS wird ein Bereich benötigt, der die Standard-Arbeitswoche enthält.

Für die Funktionen \*CRENSYM, \*MODNSYM, \*CRENHOL und \*MODNHOL wird ein Bereich benötigt, der eine Liste von Tagen enthält.

Für die Ausgabefunktionen \*SHBAS, \*SHLODAY, \*SHDAYIN, \*SHDAYHL, \*SHLOSYM, \*SHSYMIN, \*SHLOHOL, \*SHHOLIN werden verschiedene Ausgabebereiche benötigt.

Der Ausgabebereich fuer die Funktion \*SHNESTM enthält ein Datum (DATE), eine Uhrzeit (TIME) und den Namen eines SYMDAT.

In der folgenden Tabelle sind den einzelnen Funktionsaufrufen die von ihnen benötigten Layouts (zu erzeugen über den Operanden XPAND) zugeordnet.

## Übersicht

XPAND=	03	06	07	11	15	16	19	20	21	22	23	24	25	26	27
STDWEEK	x			x											
ENTSTDW	x			x			x								
DAYLIST		x	x		x	x									
ENTDATE		x	x		x	x				x				x	
BASINF							x								
LODAY											x				
ENTDAY											x				
DAYINF									x						
ENTSYMS									x						
DAYHOL								x							
ENTHOL								x				x			
LOSYM													x		
ENTSYM													x		
SYMINF														x	
LOHOL												x			
HOLINF										x					
SHNESTM															x

Zuordnung der Nummern zu den Funktionen (Die Numerierung stimmt überein mit der

Reihenfolge der Beschreibung der Funktionen in [Abschnitt „Einzelformate“ auf Seite 97](#)):

03	*CREACAL	Kalender erzeugen
06	*CRENHOL	Nicht-zyklischen Feiertag erzeugen
07	*CRENSYM	Nicht-zyklisches SYMDAT erzeugen
11	*MODBAS	Basis-Informationen ändern
15	*MODNHOL	Nicht-zyklischen Feiertag ändern
16	*MODNSYM	Nicht-zyklisches SYMDAT ändern
19	*SHBAS	Basis-Informationen ausgeben
20	*SHDAYHL	Kalendertag mit zugeordneten Feiertagen ausgeben
21	*SHDAYIN	Kalendertag mit zugeordneten SYMDAT ausgeben
22	*SHHOLIN	Feiertag mit zugeordneten Kalendertagen ausgeben
23	*SHLODAY	Liste von Kalendertagen ausgeben
24	*SHLOHOL	Liste von Feiertagen ausgeben
25	*SHLOSYM	Liste von SYMDAT ausgeben
26	*SHSYMIN	SYMDAT mit zugeordneten Kalendertagen ausgeben
27	*SHNESTM	Nächsten Zeitpunkt zu einer Menge von SYMDATs ausgeben

## 4.4 Returncodes

Nach jedem Makroaufruf wird eine Rückinformation (Returncode) im Standard-Header hinterlegt. Der Returncode enthält Informationen über den Erfolg oder Misserfolg des aufgerufenen Makros.

Der Standard-Header hat für alle Makros den gleichen Aufbau, er unterscheidet sich nur durch die symbolischen Feldnamen, die im ersten Zeichen (PREFIX=<p>) und in den drei folgenden Zeichen (MACID=<mac>) variieren können.

Die Namen der Returncode-Felder lauten MRET, SR1 und SR2 und sind in der folgenden Darstellung fett gedruckt.

Standard-Header

<p><mac>HDR	DS	0A		
<p><mac>FHE	DS	0XL8	0	GENERAL PARAMETER AREA HEADER
*				
<p><mac>IFID	DS	0A	0	INTERFACE IDENTIFIER
<p><mac>FCTU	DS	AL2	0	FUNCTION UNIT NUMBER
*				BIT 15 HEADER FLAG BIT,
*				MUST BE RESET UNTIL FURTHER NOTICE
*				BIT 14-12 UNUSED, MUST BE RESET
*				BIT 11-0 REAL FUNCTION UNIT NUMBER
<p><mac>FCT	DS	AL1	2	FUNCTION NUMBER
<p><mac>FCTV	DS	AL1	3	FUNCTION INTERFACE VERSION NUMBER
*				
<p><mac>RET	DS	0A	4	GENERAL RETURN CODE
<p><mac>SRET	DS	0AL2	4	SUB RETURN CODE
<p><mac> <b>SR2</b>	DS	AL1	4	SUB RETURN CODE 2
<p><mac> <b>SR1</b>	DS	AL1	5	SUB RETURN CODE 1
<p><mac> <b>MRET</b>	DS	0AL2	6	MAIN RETURN CODE
<p><mac>MR2	DS	AL1	6	MAIN RETURN CODE 2
<p><mac>MR1	DS	AL1	7	MAIN RETURN CODE 1
<p><mac>FHL	EQU	8	8	GENERAL OPERAND LIST HEADER LENGTH

## Bedeutung von Subcode 1 (SR1)

Subcode 1 teilt die Returncodes in Klassen ein.

### 00 Kein Fehler.

Der Auftrag wurde vollständig und fehlerfrei bearbeitet.

### 01 Parameterfehler

Die Auftragsdaten sind fehlerhaft. Meist liegt ein Programmierfehler vor. Nach Korrektur der Auftragsdaten kann der Auftrag wiederholt werden.

### 20 Interner Fehler

Bei der Auftragsbearbeitung trat ein Fehler auf, der in einer von CALENDAR benötigten Systemkomponente liegt. Der Anwender kann einen Fehler dieser Klasse nicht mit eigenen Mitteln beheben. Der Abbruch des Anwenderprogramms ist meist die einzig sinnvolle Reaktion.

### 40 Sonstige Fehler

In dieser Klasse sind Fehler zusammengefasst, auf die der Aufrufer jeweils spezifisch reagieren muss.

Bei einem Returncode  $\neq 0$  wird im allgemeinen der Funktionsaufruf abgebrochen und die Aktion nicht ausgeführt.

Ausnahmen dazu bilden folgende Returncodes:

- Returncode X'0008':  
Es existiert keine Feiertagsdatei. Der Kalender wird trotzdem erzeugt, enthält jedoch keine vordefinierten Feiertage.
- Returncode X'001D':  
Der Ausgabebereich ist zu klein, um die angeforderten Daten komplett aufzunehmen. Er enthält aber zumindest die Anzahl der vorhandenen und der ausgewählten Daten. Mit dieser Information kann die Größe des Ausgabebereichs neu berechnet werden.
- Returncode X'0025':  
Das Startdatum eines zyklischen SYMDAT mit CYCLTYP=WORKDAY wurde beim Verschieben der Kalendergrenzen auf Null gesetzt, da das SYMDAT keine zugeordneten Tage innerhalb der neuen Kalendergrenzen hat.

(SC2)	SC1	Maincode	Bedeutung
00	00	0000	Kein Fehler
01	01	0001	Operandenfehler: Falscher Wert
02	01	0001	Operandenfehler: Adresse nicht angegeben
00	40	0002	Kalender bereits vorhanden
00	40	0003	Kalender existiert nicht
00	40	0004	DVS-Fehler beim Zugriff auf die Kalenderdatei
00	40	0005	Datei ist keine Kalenderdatei
00	40	0006	Kalender ist nicht konsistent

(SC2)	SC1	Maincode	Bedeutung
00	40	0007	Änderungen in der Kalenderdatei im Moment nicht möglich
00	40	0008	Feiertags-Datei nicht vorhanden
00	40	0009	DVS-Fehler beim Zugriff auf die Feiertags-Datei
00	40	000A	Zu viele Kalendertage innerhalb der Kalendergrenzen
00	40	000B	Ende-Datum kleiner als Start-Datum
00	40	000C	Kein weiterer Speicherplatz verfügbar
00	40	000D	Control-Block fehlerhaft
00	40	000E	Datum außerhalb der Kalendergrenzen
00	40	000F	SYMDAT bereits vorhanden
00	40	0010	SYMDAT existiert nicht
00	40	0011	SYMDAT ist ein zyklisches SYMDAT
00	40	0012	SYMDAT ist ein nicht-zyklisches SYMDAT
00	40	0013	Zuordnung zum Datum bereits vorhanden
00	40	0014	Zuordnung zum Datum existiert nicht
00	40	0015	Falsche Datumsangabe
00	40	0016	Feiertag bereits vorhanden
00	40	0017	Feiertag existiert nicht
00	40	0018	Feiertag ist ein zyklischer Feiertag
00	40	0019	Feiertag ist ein nicht-zyklischer Feiertag
00	40	001A	Feiertag bereits aktiviert
00	40	001B	Feiertag nicht aktiviert
00	40	001C	Ausgabebereich zu klein: Keine Information vorhanden
00	40	001D	Ausgabebereich zu klein: Information teilweise vorhanden
00	40	001E	Definition des SYMDAT-Bereichs fehlerhaft
00	40	001F	Definition des Feiertags-Bereichs fehlerhaft
00	40	0020	Bereich für die Standard-Arbeitswoche fehlerhaft
00	40	0021	Ein Eintrag in der Feiertags-Datei ist fehlerhaft
00	40	0022	LIMIT1 < 1970 oder LIMIT2 > 2030 oder neues LIMIT1 > altes LIMIT2
00	40	0023	Warnung: Start-Datum des zyklischen SYMDAT außerhalb der Kalendergrenzen
00	40	0024	Funktion nicht erlaubt mit OPENMOD = *READ oder *READALL
00	40	0025	Start-Datum für zyklisches SYMDAT mit CYCLTYP= WORKDAY wurde gelöscht, da es sich außerhalb der Kalendergrenzen befindet
00	40	0026	Interner Programmfehler
00	40	0027	Kalenderversion wird von der Schnittstellenversion nicht unterstützt
00	40	0028	Es wurden mehr als 4096 SYMDAT erzeugt



<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
00	40	0029	Es wurden mehr als 1024 Feiertage erzeugt
00	01	FFFF	Funktion wird nicht unterstützt
00	03	FFFF	Version wird nicht unterstützt
00	04	FFFF	Ausrichtungsfehler

## 4.5 Beispiel

Dieses Beispiel enthält folgende Aufrufe der Schnittstelle CALENDR:

1. Kalenderdatei öffnen. Dabei wird der Open-Modus \*UPDATE verwendet, da unter Punkt 4 Daten geändert werden
2. Informationen über ein SYMDAT ausgeben. Danach wird der Ausgabebereich ausgewertet. Für jeden zugeordneten Tag werden das Jahr, der Monat und der Tag übernommen.
3. Liste von Kalendertagen ausgeben. Über den Operanden NUMB wird die Ausgabe auf einen einzigen Tag eingeschränkt. Danach werden die Attribut-Flags des Tages ausgewertet. Das erste Flag gibt Auskunft darüber, ob der Tag ein freier Tag oder ein Arbeitstag ist. Das zweite Flag zeigt an, wie der Tag definiert wurde  
 ‚W‘ : Der Tag wurde explizit als Arbeitstag definiert  
 ‚F‘: Der Tag wurde explizit als freier Tag definiert  
 ‚S‘: Das Attribut des Tages wurde nicht explizit definiert, es wird der Standard-Arbeitswoche entnommen
4. Attribut des Kalendertags ändern.
5. Kalenderdatei wieder schliessen. Dabei werden die Änderungen gespeichert.

### Programm

```

TESTCAL  START
TESTCAL  AMODE ANY
TESTCAL  RMODE ANY
          BASR  10,0
          USING *,10
*
          REQM  1           Speicher fuer Parameterliste anfordern
          LR   4,1
          USING GCLDPMDL,4   DSECT fuer die Parameterliste
          REQM  6           Speicher fuer Ausgabebereich anfordern
          LR   5,1
*
          MVC  GCLDPMDL(256),PARAM      Parameterliste uebertragen
          MVC  GCLDPMDL+256(GCLD#-256),PARAM+256
*
          LA   3,KAL1           Adresse des Control-Block
*
*
* Parameterliste modifizieren (Kalender oeffnen)
*
OPEN      CALENDR MF=M,VERSION=1,FUNCT=*OPENCAL,CCB=(3),           -
          OPENMOD=*UPDATE,CLDNAME='KALENDER'
```

```

*
* Funktion aufrufen
*
        CALENDR MF=E,VERSION=1,PARAM=(4)
*
* Returncode auswerten
*
STOP1   CLC   GCLDMRET,=AL2(GCLDRSUC)
        BNE   FEHLER
*
*
* Parameterliste modifizieren (SYMDAT-Information zeigen)
*
SHOWSYM CALENDR MF=M,VERSION=1,FUNCT=*SHSYMIN,OUTPUT=((5),18656),   -
        NAME='ULTIMO',DATE1=*FIRST,DATE2=*LAST,CCB=(3)
*
* Funktion aufrufen
*
        CALENDR MF=E,VERSION=1,PARAM=(4)
*
* Returncode auswerten
*
        CLC   GCLDMRET,=AL2(GCLDRSUC)
        BNE   FEHLER
*
* Ausgabebereich auswerten
*
        USING GCLDOSI,5           DSECT fuer den Ausgabebereich
        LA    6,GCLDSIA           R6 = Adresse des ersten
*                               zugeordneten Tages
        USING GCLDDTEN,6         DSECT fuer Datums-Eintrag
        LH    7,GCLDSIGD          R7 = Anzahl der vorhandenen Tage
M1      MVC   JAHR,GCLDDTY        Jahr uebernehmen
        MVC   MONAT,GCLDDTMO      Monat uebernehmen
        MVC   TAG,GCLDDTDY        Tag uebernehmen
*
*
* Daten auswerten. Wenn der dem SYMDAT zugeordnete Tag gleich
* dem Tagesdatum ist, koennen bestimmte Aktionen ausgefuehrt
* werden.
*
        LA    6,GCLDDTE#(6)       auf naechsten Tag positionieren
STOP2   BCT   7,M1                naechsten Tag bearbeiten
*
*
* Parameterliste modifizieren (Liste von Tagen zeigen)
*
* Über den Operanden NUMB=1 wird nur ein einziger Tag angefordert
*
SHOWLOD CALENDR MF=M,VERSION=1,FUNCT=*SHLODAY,OUTPUT=((5),18656),   -

```

```

                DATE1='1995-04-25',NUMB=1,CCB=(3)
*
* Funktion aufrufen
*
                CALENDR MF=E,VERSION=1,PARAM=(4)
*
* Returncode auswerten
*
                CLC   GCLDMRET,=AL2(GCLDRSUC)
                BNE   FEHLER
*
* Ausgabebereich auswerten
*
                USING GCLDOLOD,5           DSECT fuer den Ausgabebereich
                LA    6,GCLDLDA           R6=Adresse des ersten Tages-Eintrags
                USING GCLDDYEN,6         DSECT fuer Tages-Eintrag
                MVC   ATTR,GCLDDEA       Attribut des Tages uebernehmen
                MVC   ATTRDEF,GCLDDEAD   Explizit definiertes Attribut
STOP3  CLI   GCLDDEA,C'F'               Ist der Tag ein freier Tag
                BE    FREEDAY           Aktion fuer freien Tag ausfuehren
                CLI   GCLDDEAD,C'W'     Tag wurde als Arbeitstag definiert
                BNE   CLOSE             Kalender schliessen
*
*
* Parameterliste modifizieren (Kalendertag aendern)
*
MODDAY  CALENDR MF=M,VERSION=1,FUNCT=*MODDAY,DATE='1995-04-25',   -
                ATTRIBT=*STD,WKTIME1=*UNCHANGED,WKTIME2=*UNCHANGED, -
                CCB=(3),PARAM=(4)
*
* Funktion aufrufen
*
                CALENDR MF=E,VERSION=1,PARAM=(4)
*
* Returncode auswerten
*
                MVC   RETCODE,GCLDMRET
STOP4  CLC   RETCODE,=AL2(GCLDRSUC)
                BNE   FEHLER
*
*
* Parameterliste modifizieren (Kalender schliessen)
*
CLOSE   CALENDR MF=M,VERSION=1,FUNCT=*CLSECAL,SAVE=*YES,CCB=(3)
*
* Funktion aufrufen
*
                CALENDR MF=E,VERSION=1,PARAM=(4)

```

```

*
* Returncode auswerten
*
STOP5    CLC    GCLDMRET,=AL2(GCLDRSUC)
          BNE   FEHLER
          B     ENDE
*
*
* Aktionen fuer freien Tag ausfuehren
*
FREEDAY  EQU    *
*
*
ENDE     TERM                                Programm beenden
*
*
FEHLER   EQU    *
*
* Fehlerbehandlung (z.B. Returncode ausgeben)
*
          TERM  MODE=ABNORMAL,UNIT=STEP
*
*
* Speicherbereiche
*
JAHR     DS     CL4                          Jahr
MONAT    DS     CL2                          Monat
TAG      DS     CL2                          Tag
*
RETCODE  DS     AL2
*
ATTR     DS     CL1
ATTRDEF  DS     CL1
*
* Speicherbereich fuer den Control-Block
*
          DS     0F
KAL1     DS     0CL12
          DS     CL4
          DS     A
          DS     F
*
* Parameterliste
*
PARAM    CALENDR MF=L,VERSION=1
*
* DSECT fuer Parameterliste

```

```

*
      CALENDR MF=D,VERSION=1,XPAND=PARAM
*
* DSECT fuer Datums-Eintrag
*
      CALENDR MF=D,VERSION=1,XPAND=ENTDATE
*
* DSECT fuer SYMDAT-Information
*
      CALENDR MF=D,VERSION=1,XPAND=SYMINF
*
* DSECT fuer Liste von Tagen
*
      CALENDR MF=D,VERSION=1,XPAND=LODAY
*
* DSECT fuer einen Tages-Eintrag
*
      CALENDR MF=D,VERSION=1,XPAND=ENTDAY
*
      END

```

## Übersetzung

```

(IN)    DEL-SYS-FILE *OMF
(IN)    START-PROG $ASSEMBH
(OUT)   % BLS0500 PROGRAM ,ASSEMBH', VERSION ,1.1A00' OF ,<date>' LOADED
(OUT)   % BLS0552 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS. ALL
        RIGHTS RESERVED
(OUT)   (MSG) % % BLS0519 PROGRAM ,:V4B:$TSOS.ASSEMBH' LOADED
(OUT)   % ASS6010 V 1.1A10 OF BS2000 ASSEMBH READY
(IN)    COMPIL SOURCE=S.TESTCAL,MACROLIB=INTLIB.CALENDAR.112,LISTING=PAR
        (OUTPUT=LST.TESTCAL),TEST-SUPPORT=YES
(OUT)   % ASS6011 ASSEMBLY TIME: 581 MSEC
(OUT)   % ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
(OUT)   % ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
(OUT)   % ASS6006 LISTING GENERATOR TIME: 167 MSEC
(IN)    END
(OUT)   % ASS6012 END OF ASSEMBH

```

## Ablaufprotokoll

```

(IN)    LOAD-PROG *MOD(*OMF),TEST=AID
(OUT)   % BLS0517 MODULE ,TESTCAL' LOADED
(IN)    %IN STOP1
(IN)    %IN STOP2
(IN)    %IN STOP3

```

```

(IN)      %IN STOP4
(IN)      %IN STOP5
(IN)      %R
(EMP ON)  STOPPED AT LABEL: STOP1 , SRC_REF: 83 , SOURCE: TESTCAL , PROC: TESTCAL
(IN)      %D KAL1 %XL12 -----(01)
(OUT)    *** TID: 00010074 *** TSN: 0C25 *****
(NL)     CURRENT PC: 0000004E   CSECT: TESTCAL *****
(NL)     V'00000188' = KAL1    + #'00000000'
(NL)     00000188 (00000000) C3C3C240 01000000 7EFFFFFF          CCB ....=
(IN)      %R
(EMP ON)  STOPPED AT LABEL: STOP2 , SRC_REF: 128 , SOURCE: TESTCAL , PROC: TESTCAL
(IN)      %D JAHR -----(02)
(OUT)    SRC_REF:   128 SOURCE: TESTCAL  PROC: TESTCAL *****
(NL)     JAHR              = |1995|
(IN)      %D MONAT
(OUT)    MONAT           = |01|
(IN)      %D TAG
(OUT)    TAG              = |31|
(IN)      %D %7
(OUT)    CURRENT PC: 000000AA   CSECT: TESTCAL *****
(NL)     %7                = 0000000C
(IN)      %D %5-> %XL256 -----(03)
(OUT)    V'00002000' = ABSOLUT + #'00002000'
(NL)     00002000 (00002000) 0001000C 000C000C E4D3E3C9 D4D64040          .....ULTIMO
(NL)     00002010 (00002010) 40404040 40404040 40404040 D5F1F67A          N16:
(NL)     00002020 (00002020) F1F57AF0 F0400000 40404040 40404040          15:00 ..
(NL)     00002030 (00002030) 40404040 40404040 40404040 4000F1F9          .19
(NL)     00002040 (00002040) F9F560F0 F160F3F1 F1F9F9F5 60F0F260          95-01-311995-02-
(NL)     00002050 (00002050) F2F8F1F9 F9F560F0 F360F3F1 F1F9F9F5          281995-03-311995
(NL)     00002060 (00002060) 60F0F460 F3F0F1F9 F9F560F0 F560F3F1          -04-301995-05-31
(NL)     00002070 (00002070) F1F9F9F5 60F0F660 F3F0F1F9 F9F560F0          1995-06-301995-0
(NL)     00002080 (00002080) F760F3F1 F1F9F9F5 60F0F860 F3F1F1F9          7-311995-08-3119
(NL)     00002090 (00002090) F9F560F0 F960F3F0 F1F9F9F5 60F1F060          95-09-301995-10-
(NL)     000020A0 (000020A0) F3F1F1F9 F9F560F1 F160F3F0 F1F9F9F5          311995-11-301995
(NL)     000020B0 (000020B0) 60F1F260 F3F10000 00000000 00000000          -12-31.....
(NL)     000020C0 (000020C0) 00000000 00000000 00000000 00000000          .....
(NL)     REPEATED LINES:      2
(NL)     000020F0 (000020F0) 00000000 00000000 00000000 00000000          .....
(IN)      %R
:
:
:
(EMP ON)  STOPPED AT LABEL: STOP2 , SRC_REF: 128 , TESTCAL , PROC: TESTCAL
(IN)      %R
(EMP ON)  STOPPED AT LABEL: STOP3 , SRC_REF: 163 , SOURCE: TESTCAL , PROC: TESTCAL
(IN)      %D ATTR -----(4)
(OUT)    SRC_REF:   163 SOURCE: TESTCAL  PROC: TESTCAL *****
(NL)     ATTR              = |W|

```

```

(IN)      %D ATTRDEF
(OUT)    ATTRDEF          = |W|
(IN)      %D %5-> %XL40 -----(05)
(OUT)    CURRENT PC: 000000EC    CSECT: TESTCAL *****
(NL)     V'00002000' = ABSOLUT  + #'00002000'
(NL)     00002000 (00002000) 0001016D 00010001 F1F9F9F5 60F0F460    ..._...1995-04-
(NL)     00002010 (00002010) F2F5E3E4 C5E6E6E6 D5F0F07A F0F0F2F3    25TUEWWWN00:0023
(NL)     00002020 (00002020) 7AF5F9F0 00000000                      :590....
(IN)      %R
(EMP ON) STOPPED AT LABEL: STOP4 , SRC_REF: 191 , SOURCE: TESTCAL , PROC: TESTCAL
(IN)      %D RETCODE -----(06)
(OUT)    SRC_REF:   191 SOURCE: TESTCAL  PROC: TESTCAL *****
(NL)     RETCODE          = 0000  ..
(IN)      %R
(EMP ON) STOPPED AT LABEL: STOP5 , SRC_REF: 213 , SOURCE: TESTCAL , PROC: TESTCAL
(IN)      %D KAL1 %XL12 -----(07)
(OUT)    CURRENT PC: 00000146    CSECT: TESTCAL *****
(NL)     V'00000188' = KAL1      + #'00000000'
(NL)     00000188 (00000000) 40404040 00000000 00000000          .....
(IN)      %R

```

### Erklärung

- (01) Der Kalender wurde geöffnet, der Control-Block ist mit Werten versorgt
- (02) Die SYMDAT-Informationen zum SYMDAT ,ULTIMO' wurden ausgegeben. Die Variablen JAHR, MONAT und TAG enthalten die Werte des ersten zugeordneten Tages.
- (03) Anzeige des kompletten Ausgabebereichs fuer die Funktion \*SHSYMIN
- (04) Die Liste von Tagen wurde ausgegeben, dabei wurde die Ausgabe auf einen Tag eingeschränkt. Die Variable ATTR enthaelt das Attribut, das sich für diesen Tag aufgrund der Definitionen ergibt. Die Variable ATTRDEF enthält das Attribut, das für diesen Tag explizit definiert wurde
- (05) Anzeige des kompletten Ausgabebereichs fuer die Funktion \*SHLODAY
- (06) Die Funktion \*MODDAY wurde ausgeführt. Das Attribut für den Kalendertag wurde geändert. Die Variable RETCODE enthält den Returncode, den die Funktion liefert.
- (07) Der Kalender wurde wieder geschlossen. Der Control-Block (KAL1) ist gelöscht und kann nun wieder für einen anderen Kalender verwendet werden.



## 4.6 Einzelformate

### Funktionsübersicht

<b>Funktion</b>	<b>Kurzbeschreibung</b>
FUNCT = *AKTHOL	Feiertag aktivieren
FUNCT = *CLSECAL	Kalender schließen
FUNCT = *CREACAL	Kalender erzeugen
FUNCT = *CRECHOL	Zyklischen Feiertag erzeugen
FUNCT = *CRECSYM	Zyklisches SYMDAT erzeugen
FUNCT = *CRENHOL	Nicht-zyklischen Feiertag erzeugen
FUNCT = *CRENSYM	Nicht-zyklisches SYMDAT erzeugen
FUNCT = *DEAKHOL	Feiertag deaktivieren
FUNCT = *DELHOL	Feiertag löschen
FUNCT = *DELSYM	SYMDAT löschen
FUNCT = *MODBAS	Zyklischen Feiertag ändern
FUNCT = *MODCHOL	Basis-Information ändern
FUNCT = *MODCSYM	Kalendertag ändern
FUNCT = *MODDAY	Zyklisches SYMDAT ändern
FUNCT = *MODNHOL	Nicht-zyklischen Feiertag ändern
FUNCT = *MODNSYM	Nicht-zyklisches SYMDAT ändern
FUNCT = *OPENCAL	Kalender öffnen
FUNCT = *SAVECAL	Kalender sichern
FUNCT = *SHBAS	Basis-Information ausgeben
FUNCT = *SHDAYHL	Kalendertag mit zugeordneten Feiertagen ausgeben
FUNCT = *SHDAYIN	Kalendertag mit zugeordneten SYMDAT ausgeben
FUNCT = *SHHOLIN	Feiertag mit zugeordneten Kalendertagen ausgeben
FUNCT = *SHLODAY	Liste von Kalendertagen ausgeben
FUNCT = *SHLOHOL	Liste von Feiertagen ausgeben
FUNCT = *SHLOSYM	Liste von SYMDAT ausgeben
FUNCT = *SHSYMIN	SYMDAT mit zugeordneten Kalendertagen ausgeben
FUNCT = *SHNESTM	Nächsten Zeitpunkt zu einer Menge von SYMDATs ausgeben

## 4.6.1 \*AKTHOL Feiertag aktivieren

Mit der Funktion \*AKTHOL wird ein Feiertag aktiviert.

Nur wenn ein Feiertag aktiviert ist, werden die zugeordneten Kalendertage zu Feiertagen (und damit eventuell zu freien Tagen).

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *AKTHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt;         &lt;reg: A(char:30: name 1..30)&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

##### \*AKTHOL

Feiertag aktivieren.

##### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe [Seite 199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

##### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

#### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

#### CALLER = USER / SYSTEM

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block, über dessen Adresse die geöffnete Kalenderdatei angesprochen wird, in der ein Feiertag aktiviert werden soll. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des Feiertags, der aktiviert werden soll. Ausführliche Beschreibung siehe Seite [79](#).

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Namen des Feiertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

## 4.6.2 \*CLSECAL Kalender schließen

Mit der Funktion \*CLSECAL wird eine Kalenderdatei geschlossen.

Der zu schließende Kalender wird über die Control-Block-Adresse angegeben. Damit ist gewährleistet, dass nur eine Kalenderdatei geschlossen wird, die vorher geöffnet wurde (mit \*CREACAL oder \*OPENCAL).

### Format

Makro	Operanden
CALENDR	MF = C / D / L / M / E  ,FUNCT = *CLSECAL / <var: enum-of _funct_s:1> / <reg: enum-of _funct_s:1>  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = <var: pointer> / (<reg: pointer>)  ,SAVE = *YES / *NO / <var: enum-of _save_s:1> / <reg: enum-of _save_s:1>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff.

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### \*CLSECAL

Kalenderdatei schließen.

#### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

#### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

#### CALLER = USER / SYSTEM

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der zu schließenden Kalenderdatei. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**SAVE=**

Festlegung, ob die in der Kalenderdatei durchgeführten Änderungen gesichert und in die Kalenderdatei übernommen werden. Nicht-gesicherte Änderungen sind nach dem Schließen der Datei verloren.

Der Operand wird nicht ausgewertet, wenn die Kalenderdatei mit OPEN=\*READ oder \*READALL geöffnet wurde, da in diesen Fällen keine Änderungen in der Datei möglich sind.

**\*YES**

Die Änderungen werden gesichert.

**\*NO**

Die Änderungen werden nicht gesichert.

**<var: enum-of \_save\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_save\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

### 4.6.3 \*CREACAL Kalender erzeugen

Mit der Funktion \*CREACAL wird ein neuer Kalender angelegt. Die Funktion kann nur ausgeführt werden, wenn keine gleichnamige Kalenderdatei vorhanden ist.

Die Kalenderdatei wird angelegt und mit SHARUPD=NO und OPEN=OUTIN geöffnet. Sie bleibt so für andere Anwender gesperrt, bis sie geschlossen wird.

CALENDAR legt beim Erzeugen einer neuen Kalenderdatei einen Control-Block an. Die Adresse dieses Bereichs muss über den Operanden CCB bei allen Funktionsaufrufen angegeben werden. Das Layout dieses Bereichs kann mit XPAND=CCB erzeugt werden. Ist eine Feiertagsdatei vorhanden, werden die enthaltenen Feiertage in die zu erzeugende Kalenderdatei übernommen. Dabei werden diese vordefinierten Feiertage auch aktiviert (siehe \*AKTHOL).

#### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *CREACAL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,CLDNAME = &lt;c-string 1..54: filename 1..54 with catid&gt; /           &lt;var: char:54 filename 1..54 with catid&gt; /           &lt;reg: A(char:54 filename 1..54 with catid)&gt;  ,LIMIT1 = *STD / &lt;c-string 10..10: date 10..10&gt; /           &lt;var: char:10: date 10..10&gt; /           &lt;reg: A(char:10: date 10..10)&gt;  ,LIMIT2 = *STD / &lt;c-string 10..10: date 10..10&gt; /           &lt;var: char:10: date 10..10&gt; /           &lt;reg: A(char:10: date 10..10)&gt;  ,WEEKADR = *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)</pre>

## Operandenbeschreibung

### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### \*CREACAL

Kalenderdatei erzeugen.

#### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

### CALLER = USER / SYSTEM

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

### CCB=

Control-Block für die zu erzeugende Kalenderdatei. Ausführliche Beschreibung siehe Seite [77](#).

#### <var: pointer>

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

#### (<reg: pointer>)

Register mit der Adresse des Control-Blocks.

### CLDNAME=

Name des zu erzeugenden Kalenders. Ausführliche Beschreibung siehe Seite [77](#).

#### <c-string 1..54: filename 1..54 with catid>

Name des Kalenders als Direktangabe.

#### <var: char:54 filename 1..54 with catid>

Name des Feldes mit dem Kalendernamen; nur in Verbindung mit MF=M zulässig.

#### <reg: A(char:54 filename 1..54 with catid)>

Register mit der Adresse des Feldes, das den Namen des Kalenders enthält; nur in Verbindung mit MF=M zulässig.

**LIMIT1=**

Datum des ersten Kalendertags der zu erzeugenden Kalenderdatei.  
Der Wert muss kleiner sein als der bei LIMIT2 angegebene.

**\*STD**

Als Datum des ersten Kalendertags wird das aktuelle Tagesdatum angenommen.

**<c-string 10..10: date 10..10>**

Datum des ersten Kalendertags in der Kalenderdatei als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des ersten Kalendertags in der Kalenderdatei; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**LIMIT2=**

Datum des letzten Kalendertags der zu erzeugenden Kalenderdatei.  
Der Zeitraum zwischen Unter- und Obergrenze eines Kalenders darf nicht größer sein als fünf Jahre. Ausführliche Beschreibung siehe Seite [78](#).

**\*STD**

Als Datum des letzten Kalendertags wird das Datum des ersten Kalendertags + 1 Jahr angenommen.

**<c-string 10..10: date 10..10>**

Datum des letzten Kalendertags in der Kalenderdatei als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des letzten Kalendertags in der Kalenderdatei; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des letzten Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**WEEKADR=**

Adresse des Bereichs, der die Standard-Arbeitswoche enthält. Ausführliche Beschreibung siehe Seite [78](#).

**\*NO**

Es wird die Voreinstellung für die Standardarbeitswoche verwendet:  
Arbeitstage: Montag, Dienstag, Mittwoch, Donnerstag, Freitag  
Freie Tage: Samstag, Sonntag  
Arbeitszeit: 00:00 bis 23:59.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:



- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse der Standard-Arbeitswoche; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse der Standard-Arbeitswoche.

**Benötigte Layout-Bereiche:****Standard-Arbeitswoche (XPAND = STDWEEK)**

126	1	*	Standard-week	
127	1	GCLDSWA	DS	7CL14
128	1	*		
129	1	*		
130	1	GCLDSWA#	EQU	7
131	1	GCLDPSW#	EQU	*-GCLDSWA

area for standard-week entry  
is described with  
XPAND=ENTSTDW

**Standard-Arbeitswoche (XPAND = ENTSTDW)**

11	1	*	Standard-week-entry	
12	1	GCLDWEDY	DS	CL3
13	1	GCLDWEAT	DS	CL1
14	1	*		
15	1	GCLDWET1	DS	CL5
16	1	GCLDWET2	DS	CL5
17	1	GCLDSWE#	EQU	*-GCLDWEDY

week-day (MON - SUN)  
attribute of day (W=workday,  
F=free day)  
begin of working time  
end of working time

#### 4.6.4 \*CRECHOL Zyklischen Feiertag erzeugen

Mit der Funktion \*CRECHOL wird ein zyklischer Feiertag (Beschreibung Seite 17f) erzeugt. Die Funktion kann nur ausgeführt werden, wenn kein gleichnamiger zyklischer oder nicht-zyklischer Feiertag im betreffenden Kalender existiert.

Ein Feiertag, der vom Anwender erzeugt wird, wird auch aktiviert.

##### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *CRECHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,DATE = &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;</pre>

##### Operandenbeschreibung

###### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff.

###### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

###### \*CRECHOL

Zyklischen Feiertag erzeugen.

###### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

###### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der ein zyklischer Feiertag erzeugt wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zu erzeugenden zyklischen Feiertags. Ausführliche Beschreibung siehe Seite 79.

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**DATE=**

Datum des zu erzeugenden zyklischen Feiertags. Da ein zyklischer Feiertag jährlich wiederkehrend auf dasselbe Tages- und Monatsdatum fällt, ist für die Jahresangabe der String „\*\*\*\*“ anzugeben, also „\*\*\*\*-mm-dd“. Ausführliche Beschreibung siehe Seite 79.

**<c-string 10..10: date 10..10>**

Datum des Feiertags als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Feiertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: name 1..10)>**

Register mit der Adresse des Feldes, das das Datum des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

#### **4.6.5 \*CRECSYM Zyklisches SYMDAT erzeugen**

Mit der Funktion \*CRECSYM wird ein zyklisches SYMDAT (Beschreibung Seite [15](#)) erzeugt.

Die Funktion kann nur ausgeführt werden, wenn kein gleichnamiges zyklisches SYMDAT im betreffenden Kalender vorhanden ist.

**Format**

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *CRECSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,TIME = *STD / &lt;c-string 8..8: time 8..8&gt; /         &lt;var: char:8: time 8..8&gt; /         &lt;reg: A(char:8: time 8..8)&gt;  ,CYCLTYP = *DAY / *WORKDAY / *WEEK / *MONTH /           &lt;var: enum-of _cycle_s:1&gt; /           &lt;reg: enum-of _cycle_s:1&gt;  ,CYCLVAL = &lt;integer 1..9999&gt; / &lt;var: int:4: integer 1..9999&gt; /           &lt;reg: int:4: integer 1..9999&gt;  ,ALTERN = *BEFORE / *AFTER / *SKIP / *ON /           &lt;var: enum-of _alternative_s:1&gt; /           &lt;reg: enum-of _alternative_s:1&gt;  ,DATE1 = &lt;c-string 10..10: date 10..10&gt; /          &lt;var: char:10: date 10..10&gt; /          &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *UNDEFINED / &lt;c-string 10..10: date 10..10&gt; /          &lt;var: char:10: date 10..10&gt; /          &lt;reg: A(char:10: date 10..10)&gt;</pre>

**Operandenbeschreibung****MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*CRECSYM**

Zyklisches SYMDAT erzeugen.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, für die das zyklische SYMDAT erzeugt wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zyklischen SYMDAT, das erzeugt werden soll. Ausführliche Beschreibung siehe Seite 79.

**<c-string 1..20: name 1..20>**

Name des zyklischen SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**TIME=**

Zeitangabe, die dem zu erzeugenden zyklischen SYMDAT zugeordnet werden soll. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss=Sekunden).

**\*STD**

Voreinstellung für die Zeitangabe: 00:00:00

**<c-string 8..8: time 8..8>**

Zeitangabe für das zyklische SYMDAT als Direktangabe.

**<var: char:8: time 8..8>**

Name des Feldes mit der Zeitangabe für das zyklische SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:8: time 8..8)>**

Register mit der Adresse des Feldes, das die Zeitangabe für das SYMDAT enthält; nur in Verbindung mit MF=M zulässig.

**CYCLTYP=**

Art des Zyklus (Beschreibung auf Seite 15f). Der Operand definiert zusammen mit den Operanden CYCLVAL und ALTERN den Zyklus.

**\*DAY**

Zyklus = Tag.

**\*WORKDAY**

Zyklus = Arbeitstag.

**\*WEEK**

Zyklus = Woche.

**\*MONTH**

Zyklus = Monat.

**<var: enum-of \_cycle\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_cycle\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**CYCLVAL=**

Zyklus-Wert.

Der Operand bestimmt für das zu erzeugende zyklische SYMDAT- abhängig von der Art des Zyklus (CYCLTYP) - die Anzahl der Tage bzw. Arbeitstage bzw. Wochen bzw. Monate, die zwischen zwei Einträgen liegen. Die Angabe bezieht sich also immer auf den Operanden CYCLTYP.

**<integer 1..9999>**

Anzahl der Tage / Arbeitstage / Wochen / Monate zwischen zwei Einträgen als Direktangabe.

**<var: int:4: integer 1..9999>**

Name des Feldes mit dem Zyklus-Wert für das zyklische SYMDAT.

**<reg: int:4: integer 1..9999>**

Register, das den Zykluswert enthält; nur in Verbindung mit MF=M zulässig.

**ALTERN=**

Alternative (siehe Beschreibung der Free-Day-Rule auf Seite 15).

Der Operand legt für das zu erzeugende zyklische SYMDAT die Alternative für den Fall fest, dass ein berechneter SYMDAT-Eintrag auf einen freien Tag fällt.

**\*BEFORE**

Der Eintrag erfolgt vor dem freien Tag.

**\*AFTER**

Der Eintrag erfolgt nach dem freien Tag.

**\*SKIP**

Der Eintrag entfällt.

**\*ON**

Der Eintrag erfolgt am freien Tag.

**<var: enum-of \_alternative\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_alternative\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**DATE1=**

Start-Datum für die Berechnung des zu erzeugenden zyklischen SYMDAT. Ausführliche Beschreibung siehe Seite 81.

**<c-string 10..10: date 10..10>**

Start-Datum als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Startdatum für das zyklische SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Startdatum enthält; nur in Verbindung mit MF=M zulässig.

**DATE2=**

Ende-Datum für die Berechnung des zyklischen SYMDAT. Ausführliche Beschreibung siehe Seite 81.

**\*UNDEFINED**

Da das Enddatum für die Berechnung zyklischer SYMDAT nicht definiert ist, wird als Ersatzwert das Kalenderende verwendet.



**<var: char:10: date 10..10>**

Name des Feldes mit dem Endedatum für das zyklische SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Endedatum enthält; nur in Verbindung mit MF=M zulässig.

## 4.6.6 \*CRENHOL Nicht-zyklischen Feiertag erzeugen

Mit der Funktion \*CRENHOL wird ein nicht-zyklischer Feiertag (Beschreibung Seite 17f) erzeugt. Die Funktion kann nur ausgeführt werden, wenn kein gleichnamiger (zyklischer oder nicht-zyklischer) Feiertag im betreffenden Kalender vorhanden ist. Ein Feiertag, der vom Anwender erzeugt wird, wird auch aktiviert.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *CRENHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt;         &lt;reg: A(char:30: name 1..30)&gt;  ,ADDDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl  : &lt;integer 0..1860&gt; /                    &lt;var: int:4: integer 0..1860&gt; /                    &lt;reg: int:4: integer 0..1860&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff.

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### \*CRENHOL

Nicht-zyklischen Feiertag erzeugen.

#### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der der nicht-zyklische Feiertag erzeugt wird. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zu erzeugenden nicht-zyklischen Feiertags. Ausführliche Beschreibung siehe Seite [79](#).

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**ADDDAY = (adresse, anzahl)**

Datenbereich, der die Kalendertage enthält, die dem Feiertag zugeordnet werden. Ausführliche Beschreibung siehe Seite [72](#).

adresse: \*NO / <var: pointer> / (<reg: pointer>)

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Bereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Bereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Bereichs.

anzahl : <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig.
<reg: int:4: integer 0..1860>	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

### Benötigte Layout-Bereiche:

Liste von Kalendertagen (XPAND = DAYLIST)

```

145 1 *   day-list
146 1 GCLDPDA DS   1860CL10
147 1 *
148 1 GCLDPDA# EQU 1860
149 1 GCLDPDY# EQU *-GCLDPDA

```

arae for days entry is  
described with XPAND=ENTDATE

Datums-Eintrag (XPAND = ENTDATE)

```

300 1 *   Date-entry
301 1 GCLDDTY DS   CL4
302 1 GCLDDTH1 DS CL1
303 1 GCLDDTMO DS CL2
304 1 GCLDDTH2 DS CL1
305 1 GCLDDTDY DS CL2
306 1 GCLDDTE# EQU *-GCLDDTY

```

year  
hyphen  
month  
hyphen  
day

## 4.6.7 \*CRENSYM Nicht-zyklisches SYMDAT erzeugen

Mit der Funktion \*CRENSYM wird ein nicht-zyklisches SYMDAT erzeugt (Beschreibung siehe Seite 14f). Die Funktion kann nur ausgeführt werden, wenn kein gleichnamiges (zyklisches oder nicht-zyklisches) SYMDAT im betreffenden Kalender vorhanden ist.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *CRENSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,ADDDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl  : &lt;integer 0..1860&gt; /                    &lt;var: int:4: integer 0..1860&gt; /                    &lt;reg: int:4: integer 0..1860&gt;  ,TIME = *STD / &lt;c-string 8..8: time 8..8&gt; /         &lt;var: char:8: time 8..8&gt; /         &lt;reg: A(char:8: time 8..8)&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff.

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*CRENSYM**

Nicht-zyklisches SYMDAT erzeugen.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der das nicht-zyklische SYMDAT erzeugt wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zu erzeugenden nicht-zyklischen SYMDAT. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..20: name 1..20>**

Name des SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**ADDDAY = (adresse, anzahl)**

Datenbereich, der die Kalendertage enthält, die dem SYMDAT zugeordnet werden.

Alle Kalendertage, die zugeordnet werden, müssen innerhalb der Kalendergrenzen liegen. Ausführliche Beschreibung siehe Seite 72.

**adresse: \*NO / <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Bereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Bereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Bereichs.

**anzahl : <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>**

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig.
<reg: int:4: integer 0..1860>	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

### **TIME=**

Zeitangabe, die dem zu erzeugenden SYMDAT zugeordnet werden soll. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss=Sekunden).

#### **\*STD**

Voreinstellung für die Zeitangabe: 00:00:00

#### **<c-string 8..8: time 8..8>**

Zeitangabe für das nicht-zyklische SYMDAT als Direktangabe.

#### **<var: char:8: time 8..8>**

Name des Feldes mit der Zeitangabe für das nicht-zyklische SYMDAT; nur in Verbindung mit MF=M zulässig.

#### **<reg: A(char:8: time 8..8)>**

Register mit der Adresse des Feldes, das die Zeitangabe enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

Liste von Kalendertagen (XPAND = DAYLIST)

145	1	*	day-list		
146	1	GCLDPDA	DS	1860CL10	arae for days entry is
147	1	*			described with XPAND=ENTDATE
148	1	GCLDPDA#	EQU	1860	
149	1	GCLDPDY#	EQU	*-GCLDPDA	

Datums-Eintrag (XPAND = ENTDATE)

300	1	*	Date-entry		
301	1	GCLDDTY	DS	CL4	year
302	1	GCLDDTH1	DS	CL1	hyphen
303	1	GCLDDTMO	DS	CL2	month
304	1	GCLDDTH2	DS	CL1	hyphen
305	1	GCLDDTDY	DS	CL2	day
306	1	GCLDDTE#	EQU	*-GCLDDTY	



## 4.6.8 **\*DEAKHOL** Feiertag deaktivieren

Mit der Funktion **\*DEAKHOL** wird ein Feiertag deaktiviert.

Die Deaktivierung eines Feiertags bewirkt, dass er nicht mehr als solcher berücksichtigt wird, ohne jedoch gleich aus der Kalenderdatei gelöscht zu werden. Er kann zu einem späteren Zeitpunkt wieder aktiviert werden.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *DEAKHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;</pre>

### Operandenbeschreibung

#### **MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### **FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*DEAKHOL**

Feiertag deaktivieren.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

#### **VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block, über dessen Adresse die geöffnete Kalenderdatei angesprochen wird, in der ein Feiertag deaktiviert werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des Feiertags, der deaktiviert werden soll. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

## 4.6.9 \*DELHOL Feiertag löschen

Mit der Funktion \*DELHOL wird ein Feiertag gelöscht. Zyklische und nicht-zyklische Feiertage werden auf dieselbe Weise, durch Angabe der Control-Block-Adresse der Kalenderdatei und ihres Namens, gelöscht.

Vordefinierte Feiertage aus der Feiertagsdatei sollten nicht gelöscht, sondern nur deaktiviert werden. Es gibt keine Möglichkeit, nach dem Erzeugen des Kalenders einen vordefinierten Feiertag aus der Feiertagsdatei zu übernehmen.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *DELHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*DELHOL**

Feiertag löschen.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

#### **VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block, über dessen Adresse die geöffnete Kalenderdatei angesprochen wird, in der ein Feiertag gelöscht werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des Feiertags, der gelöscht werden soll. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

### 4.6.10 \*DELSYM SYMDAT löschen

Mit der Funktion \*DELSYM wird ein SYMDAT-Eintrag gelöscht. Zyklische und nicht-zyklische SYMDAT werden auf dieselbe Weise, durch Angabe der Control-Block-Adresse der Kalenderdatei und ihres Namens, gelöscht.

#### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *DELSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;</pre>

#### Operandenbeschreibung

##### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

##### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

##### **\*DELSYM**

SYMDAT löschen.

##### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

##### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

##### **VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

##### **CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block, über dessen Adresse die geöffnete Kalenderdatei angesprochen wird, in der ein SYMDAT gelöscht werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des SYMDAT, das gelöscht werden soll. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..20: name 1..20>**

Name des SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

### 4.6.11 \*MODBAS Basis-Information ändern

Mit der Funktion \*MODBAS können die Kalendergrenzen (Start- und Ende-Datum) und die Standard-Arbeitswoche geändert werden.

Beim Ändern von Kalendergrenzen ist folgendes zu beachten:

- Die Daten zu Kalendertagen, die nach dem Ändern außerhalb der neuen Grenzen liegen, gehen verloren.
- Die Verknüpfung zu Kalendertagen, die nach dem Ändern außerhalb der neuen Grenzen liegen, wird bei nicht-zyklischen SYMDAT gelöscht.
- Bei zyklischen SYMDAT mit CYCLTYP=\*WORKDAY wird das Start-Datum korrigiert, wenn es nach dem Ändern außerhalb der neuen Grenzen liegen sollte.
- Werden durch das Ändern der Kalendergrenzen neue Tage erzeugt, so wird für diese Kalendertage das Attribut und die Arbeitszeit mit Standardwerten vorbelegt (siehe auch Seite 12f). Aktivierte Feiertage werden auch für die neuen Tage berücksichtigt.
- Beim nächstfolgenden Zugriff - nach dem Ändern der Kalendergrenzen - auf Informationen, bei denen zyklische SYMDAT betroffen sind, müssen diese zyklischen SYMDAT neu berechnet werden.

Zum Ändern der Standard-Arbeitswoche muss die Adresse des Bereichs angegeben werden, der die Definitionen für diese enthält (WEEKADR).

Das Layout wird mit XPAND = STDWEEK erzeugt. Der Inhalt dieses Bereichs sollte vom Ausgabebereich der Funktion \*SHBAS übernommen werden. Nur die benötigten Werte sollten geändert werden.

**Format**

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *MODBAS / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,LIMIT1 = *UNCHANGED / &lt;c-string 10..10: date 10..10&gt; /           &lt;var: char:10: date 10..10&gt; /           &lt;reg: A(char:10: date 10..10)&gt;  ,LIMIT2 = *UNCHANGED / &lt;c-string 10..10: date 10..10&gt; /           &lt;var: char:10: date 10..10&gt; /           &lt;reg: A(char:10: date 10..10)&gt;  ,WEEKADR = *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)</pre>

**Operandenbeschreibung****MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff](#).

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*MODBAS**

Basis-Information ändern.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).



**CCB=**

Control-Block der Kalenderdatei, in der die Basis-Informationen geändert werden sollen. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**LIMIT1=**

Untergrenze des Kalenders, d.h. Datum des ersten Kalendertags der Kalenderdatei. Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben, um zu verhindern, dass ein bei einer anderen Funktion eingetragener Wert übernommen wird. Ausführliche Beschreibung siehe Seite [78](#).

**\*UNCHANGED**

Der bestehende Wert bleibt unverändert.

**<c-string 10..10: date 10..10>**

Datum des ersten Kalendertags in der Kalenderdatei als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des ersten Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum enthält; nur in Verbindung mit MF=M zulässig.

**LIMIT2=**

Obergrenze des Kalenders, d.h. Datum des letzten Kalendertags der Kalenderdatei. Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben, um zu verhindern, dass ein bei einer anderen Funktion eingetragener Wert übernommen wird. Ausführliche Beschreibung siehe Seite [78](#).

**\*UNCHANGED**

Der bestehende Wert bleibt unverändert.

**<c-string 10..10: date 10..10>**

Datum des letzten Kalendertags in der Kalenderdatei als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des letzten Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum enthält; nur in Verbindung mit MF=M zulässig.

**WEEKADR=**

Adresse des Bereichs, der die Standard-Arbeitswoche enthält.

Soll die Standard-Arbeitswoche nicht verändert werden, ist der Operandenwert \*NO anzugeben, um zu verhindern, dass ein bei einer anderen Funktion eingetragener Wert übernommen wird. Ausführliche Beschreibung siehe Seite 78.

**\*NO**

Die Standard-Arbeitswoche bleibt unverändert.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse der Standard-Arbeitswoche dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse der Standard-Arbeitswoche; nur in Verbindung mit MF=M zulässig.

**<reg: pointer>**

Register mit der Adresse der Standard-Arbeitswoche.

**Benötigte Layout-Bereiche:****Standard-Arbeitswoche (XPAND = STDWEEK)**

126	1	*	Standard-week		
127	1	GCLDSWA	DS	7CL14	area for standard-week entry
128	1	*			is described with
129	1	*			XPAND=ENTSTDW
130	1	GCLDSWA#	EQU	7	
131	1	GCLDPSW#	EQU	*-GCLDSWA	

**Standard-Arbeitswoche (XPAND = ENTSTDW)**

11	1	*	Standard-week-entry		
12	1	GCLDWEDY	DS	CL3	week-day (MON – SUN)
13	1	GCLDWEAT	DS	CL1	attribute of day (W=workday, F=free day)
14	1	*			
15	1	GCLDWET1	DS	CL5	begin of working time
16	1	GCLDWET2	DS	CL5	end of working time
17	1	GCLDSWE#	EQU	*-GCLDWEDY	

## 4.6.12 \*MODCHOL Zyklischen Feiertag ändern

Mit der Funktion \*MODCHOL wird ein zyklischer Feiertag geändert. Er kann jedoch nicht in einen nicht-zyklischen umgewandelt werden.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *MODCHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,DATE = &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*MODCHOL**

Zyklischen Feiertag ändern.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

#### **VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der ein zyklischer Feiertag geändert werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zu ändernden zyklischen Feiertags. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**DATE=**

Datum des zu ändernden zyklischen Feiertags. Da ein zyklischer Feiertag jährlich wiederkehrend auf dasselbe Tages- und Monatsdatum fällt, ist für die Jahresangabe der String „\*\*\*\*“ anzugeben, also „\*\*\*\*-mm-dd“. Ausführliche Beschreibung siehe Seite 79.

**<c-string 10..10: date 10..10>**

Datum des Feiertags als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Feiertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

### 4.6.13 \*MODDAY Kalendertag ändern

Mit der Funktion \*MODDAY kann das Attribut und die Arbeitszeit eines Kalendertags geändert werden.

Änderungen des Attributwertes (ATTRBT) und der Arbeitszeit (WKTIME1 und WKTIME2) betreffen den angegebenen Kalendertag (DATE) und sind Abweichungen von den Definitionen der Standard-Arbeitswoche. Die neuen Werte können später wieder rückgesetzt werden, d.h. die Definitionen aus der Standard-Arbeitswoche werden wieder gültig. Das Ändern des Attributes hat Auswirkungen auf die Berechnung von zyklischen SYMDAT.

Deshalb werden die zyklischen SYMDAT der Kalenderdatei beim nächsten Zugriff auf Daten, bei denen zyklische SYMDAT betroffen sind, neu berechnet.

#### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *MODDAY / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,DATE = &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,ATTRIBT = *UNCHANGED / *FREE / *WORK / *STD /            &lt;var: enum-of _attribute_s:1&gt;            &lt;reg: enum-of _attribute_s:1&gt;  ,WKTIME1 = *UNCHANGED / *STD / &lt;c-string 5..5: time 5..5&gt; /            &lt;var: char:5: time 5..5&gt; /            &lt;reg: A(char:5: time 5..5)&gt;  ,WKTIME2 = *UNCHANGED / *STD / &lt;c-string 5..5: time 5..5&gt; /            &lt;var: char:5: time 5..5&gt; /            &lt;reg: A(char:5: time 5..5)&gt;</pre>

#### Operandenbeschreibung

##### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*MODDAY**

Kalendertag ändern.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der ein Kalendertag geändert werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**DATE=**

Datum des zu ändernden Kalendertags. Ausführliche Beschreibung siehe Seite 79.

**<c-string 10..10: date 10..10>**

Datum des Kalendertags als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**ATTRIBT=**

Attribut des Kalendertags. Der Kalendertag kann entweder als Arbeitstag oder als freier Tag definiert werden.

Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben.

**\*UNCHANGED**

Der vorhandene Wert bleibt unverändert.

**\*FREE**

Der Kalendertag ist ein freier Tag.

**\*WORK**

Der Kalendertag ist ein Arbeitstag.

**\*STD**

Die Angabe für das Attribut wird der Standard-Arbeitswoche entnommen.

**<var: enum-of \_attribute\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_attribute\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**WKTIME1=**

Beginn der Arbeitszeit für den angegebenen Kalendertag. Die Angabe erfolgt in der Form hh:mm (hh=Stunden, mm=Minuten).

Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben.

**\*UNCHANGED**

Der vorhandene Wert bleibt unverändert.

**\*STD**

Die Angabe für den Beginn der Arbeitszeit wird der Standard-Arbeitswoche entnommen.

**<c-string 5..5: time 5..5>**

Beginn der Arbeitszeit als Direktangabe.

**<var: char:5: time 5..5>**

Name des Feldes mit dem Arbeitszeitbeginn; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:5: time 5..5)>**

Register mit der Adresse des Feldes, das den Arbeitszeitbeginn enthält; nur in Verbindung mit MF=M zulässig.

**WKTIME2=**

Ende der Arbeitszeit für den entsprechenden Kalendertag. Die Angabe erfolgt in der Form hh:mm (hh=Stunden, mm=Minuten).

Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben.

**\*UNCHANGED**

Der vorhandene Wert bleibt unverändert.

**\*STD**

Die Angabe für das Ende der Arbeitszeit wird der Standard-Arbeitswoche entnommen.

**<c-string 5..5: time 5..5>**

Ende der Arbeitszeit als Direktangabe.

**<var: char:5: time 5..5>**

Name des Feldes mit dem Arbeitszeitende; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:5: time 5..5)>**

Register mit der Adresse des Feldes, das das Arbeitszeitende enthält; nur in Verbindung mit MF=M zulässig.



### 4.6.14 \*MODCSYM Zyklisches SYMDAT ändern

Mit der Funktion \*MODCSYM kann ein zyklisches SYMDAT (siehe Beschreibung auf Seite 15ff) verändert werden. Es kann jedoch nicht in ein nicht-zyklisches umgewandelt werden.

Das Start-Datum (DATE1) muss innerhalb der Kalendergrenzen liegen und es muss kleiner sein als das Ende-Datum (DATE2), falls dieses definiert wurde. Das Ende-Datum kann auch außerhalb der Kalendergrenzen liegen.

#### Format

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,FUNCT = *MODCSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,TIME = *UNCHANGED / *STD / &lt;c-string 8..8: time 8..8&gt; /         &lt;var: char:8: time 8..8&gt; /         &lt;reg: A(char:8: time 8..8)&gt;  ,CYCLTYP = *UNCHANGED / *DAY / *WORKDAY / *WEEK / *MONTH /           &lt;var: enum-of _cycle_s:1&gt; /           &lt;reg: enum-of _cycle_s:1&gt;  ,CYCLVAL = *UNCHANGED / &lt;integer 1..9999&gt; /           &lt;var: int:4: integer 1..9999&gt; /           &lt;reg: int:4: integer 1..9999&gt;  ,ALTERN = *UNCHANGED / *BEFORE / *AFTER / *SKIP / *ON /           &lt;var: enum-of _alternative_s:1&gt; /           &lt;reg: enum-of _alternative_s:1&gt; </pre>

Makro	Operanden
CALENDR (Forts.)	<pre> ,DATE1 = *UNCHANGED / &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *UNCHANGED / *UNDEFINED /         &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt; </pre>

**MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff](#).

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*MODCSYM**

Zyklisches SYMDAT ändern.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der das zyklische SYMDAT geändert wird. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des zyklischen SYMDAT, das verändert werden soll. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..20: name 1..20>**

Name des SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**TIME=**

Zeitangabe, die dem zu ändernden zyklischen SYMDAT zugeordnet werden soll. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss= Sekunden). Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**\*STD**

Voreinstellung für die Zeitangabe: 00:00:00

**<c-string 8..8: time 8..8>**

Zeitangabe für das zyklische SYMDAT als Direktangabe.

**<var: char:8: time 8..8>**

Name des Feldes mit der Zeitangabe; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:8: time 8..8)>**

Register mit der Adresse des Feldes, das die Zeitangabe enthält; nur in Verbindung mit MF=M zulässig.

**CYCLTYP=**

Art des Zyklus (Beschreibung auf Seite 15f). Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben. Ausführliche Beschreibung siehe Seite 80.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**\*DAY**

Zyklus = Tag.

**\*WORKDAY**

Zyklus = Arbeitstag.

**\*WEEK**

Zyklus = Woche.

**\*MONTH**

Zyklus = Monat.

**<var: enum-of \_cycle\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_cycle\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**CYCLVAL=**

Zyklus-Wert.

Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben. Ausführliche Beschreibung siehe Seite 81.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**<integer 1..9999>**

Anzahl der Tage / Arbeitstage / Wochen / Monate zwischen zwei Einträgen als Direktangabe.

**<var: int:4: integer 1..9999>**

Name des Feldes mit der Anzahl der Tage / Arbeitstage / Wochen / Monate als Integer-Zahl; Feld-Länge = 4 Byte

**<reg: int:4: integer 1..9999>**

Register, das den Zykluswert enthält; nur in Verbindung mit MF=M zulässig.

**ALTERN=**

Alternative (siehe Beschreibung der Free-Day-Rule auf Seite 15).

Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben. Ausführliche Beschreibung siehe Seite 81.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**\*BEFORE**

Der Eintrag erfolgt vor dem freien Tag.

**\*AFTER**

Der Eintrag erfolgt nach dem freien Tag.

**\*SKIP**

Der Eintrag entfällt.

**\*ON**

Der Eintrag erfolgt am freien Tag.

**<var: enum-of \_alternative\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_alternative\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**DATE1=**

Start-Datum für die Berechnung des zu ändernden zyklischen SYMDAT. Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben. Ausführliche Beschreibung siehe Seite 81.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**<c-string 10..10: date 10..10>**

Start-Datum als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Startdatum; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Startdatum enthält; nur in Verbindung mit MF=M zulässig.

**DATE2=**

Enddatum für die Berechnung des zyklischen SYMDAT. Soll der Wert nicht verändert werden, ist der Operandenwert \*UNCHANGED anzugeben. Ausführliche Beschreibung siehe Seite 81.

**\*UNCHANGED**

Der Wert bleibt unverändert.

**\*UNDEFINED**

Da das Enddatum für die Berechnung zyklischer SYMDAT nicht definiert ist, wird als Ersatzwert das Kalenderende verwendet.

**<c-string 10..10: date 10..10>**

Enddatum als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Enddatum; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Enddatum enthält; nur in Verbindung mit MF=M zulässig.

### 4.6.15 \*MODNHOL Nicht-zyklischen Feiertag ändern

Mit der Funktion \*MODNHOL wird ein nicht-zyklischer Feiertag geändert, d.h. es werden zusätzliche Kalendertage mit dem Feiertag verknüpft (ADDDAY) oder bestehende Verknüpfungen gelöscht (DELDAY). Sollte bei einem dieser beiden Operanden keine Änderung vorgenommen werden, muss dort der Operandenwert \*NO angegeben werden.

#### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *MODNHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,ADDDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl : &lt;integer 0..1860&gt; /                   &lt;var: int:4: integer 0..1860&gt; /                   &lt;reg: int:4: integer 0..1860&gt;  ,DELDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl : &lt;integer 0..1860&gt; /                   &lt;var: int:4: integer 0..1860&gt; /                   &lt;reg: int:4: integer 0..1860&gt;</pre>

#### Operandenbeschreibung

##### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

##### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*MODNHOL**

Nicht-zyklischen Feiertag ändern.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, in der ein nicht-zyklischer Feiertag geändert werden soll. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des nicht-zyklischen Feiertags, der geändert werden soll. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertagsnamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**ADDDAY = (adresse, anzahl)**

Datenbereich, der die Kalendertage enthält, die dem Feiertag zugeordnet werden sollen. Hat der Feiertag bereits eine Verknüpfung zu einem dieser Tage, wird keine zusätzliche Verknüpfung erzeugt, sondern der Funktionsaufruf abgebrochen. Ausführliche Beschreibung siehe Seite 72.

**adresse: \*NO / <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: <ul style="list-style-type: none"> <li>- Adresse des Bereichs als direkte Adressangabe in der Form A(feld).</li> <li>- Name des Feldes mit der Adresse des Bereichs; nur in Verbindung mit MF=M zulässig.</li> </ul>
(<reg: pointer>):	Register mit der Adresse des Bereichs.

**anzahl : <integer 0..1860> / <var: int:4: integer 0..1860> / <reg: int:4: integer 0..1860>**

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig..
<reg: int:4: integer 0..1860>:	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

**DELDAY = (adresse, anzahl)**

Datenbereich mit Kalendertagen, deren Zuordnung zu dem unter NAME angegebenen nicht-zyklischen Feiertag gelöscht werden soll. Befindet sich in dem Bereich ein Kalendertag, zu dem der Feiertag keine Verknüpfung hat, wird der Funktionsaufruf abgebrochen, und es wird keine Verknüpfung gelöscht. Ausführliche Beschreibung siehe Seite 80.

**adresse: \*NO / <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: <ul style="list-style-type: none"> <li>- Adresse des Bereichs als direkte Adressangabe in der Form A(feld).</li> <li>- Name des Feldes mit der Adresse des Bereichs; nur in Verbindung mit MF=M zulässig.</li> </ul>
(<reg: pointer>):	Register mit der Adresse des Bereichs.



**anzahl : <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>**

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig.
<reg: int:4: integer 0..1860>:	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

### Benötigte Layout-Bereiche:

#### Liste von Kalendertagen (XPAND = DAYLIST)

```

145 1 *   day-list
146 1 GCLDPDA DS   1860CL10           arae for days entry is
147 1 *                                       described with XPAND=ENTDATE
148 1 GCLDPDA# EQU  1860
149 1 GCLDPDY# EQU  *-GCLDPDA

```

#### Datums-Eintrag (XPAND = ENTDATE)

```

300 1 *   Date-entry
301 1 GCLDDTY DS   CL4                year
302 1 GCLDDTH1 DS  CL1                hyphen
303 1 GCLDDTMO DS  CL2                month
304 1 GCLDDTH2 DS  CL1                hyphen
305 1 GCLDDTDY DS  CL2                day
306 1 GCLDDTE# EQU *-GCLDDTY

```

### 4.6.16 \*MODNSYM Nicht-zyklisches SYMDAT ändern

Mit der Funktion \*MODNSYM wird ein nicht-zyklisches SYMDAT geändert, d.h. es werden zusätzliche Kalendertage mit dem SYMDAT-Eintrag verknüpft (ADDDAY) oder bestehende Verknüpfungen gelöscht (DELDAY). Sollte bei einem dieser beiden Operanden keine Änderung vorgenommen werden, muss dort der Operandenwert \*NO angegeben werden. Soll die zugeordnete Zeit unverändert bleiben, muss bei TIME der Wert \*UNCHANGED angegeben werden.

#### Format

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,FUNCT = *MODNSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,TIME = *UNCHANGED / *STD / &lt;c-string 8..8: time 8..8&gt; /         &lt;var: char:8: time 8..8&gt; /         &lt;reg: A(char:8: time 8..8)&gt;  ,ADDDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl: &lt;integer 0..1860&gt; /                   &lt;var: int:4: integer 0..1860&gt; /                   &lt;reg: int:4: integer 0..1860&gt;  ,DELDAY = (adresse, anzahl)           adresse : *NO / &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           anzahl: &lt;integer 0..1860&gt; /                   &lt;var: int:4: integer 0..1860&gt; /                   &lt;reg: int:4: integer 0..1860&gt; </pre>

## Operandenbeschreibung

### **MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

### **FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*MODNSYM**

Nicht-zyklisches SYMDAT ändern.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

### **VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

### **CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

### **CCB=**

Control-Block der Kalenderdatei, in der das nicht-zyklische SYMDAT geändert werden soll. Ausführliche Beschreibung siehe Seite [77](#).

#### **<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

#### **(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

### **NAME=**

Name des zu ändernden nicht-zyklischen SYMDAT. Ausführliche Beschreibung siehe Seite [71](#).

#### **<c-string 1..20: name 1..20>**

Name des SYMDAT als Direktangabe.

#### **<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**TIME=**

Zeitangabe, die dem zu ändernden SYMDAT zugeordnet ist. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss=Sekunden).

**\*UNCHANGED**

Der Wert bleibt unverändert.

**\*STD**

Voreinstellung für die Zeitangabe: 00:00:00

**<c-string 8..8: time 8..8>**

Zeitangabe für das zyklische SYMDAT als Direktangabe.

**<var: char:8: time 8..8>**

Name des Feldes mit der Zeitangabe; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:8: time 8..8)>**

Register mit der Adresse des Feldes, das die Zeitangabe enthält; nur in Verbindung mit MF=M zulässig.

**ADDDAY = (adresse, anzahl)**

Datenbereich, der die Kalendertage enthält, die dem SYMDAT zugeordnet werden sollen. Alle Tage müssen innerhalb der Kalendergrenzen liegen. Hat das SYMDAT bereits eine Verknüpfung zu einem dieser Tage, wird keine zusätzliche Verknüpfung erzeugt, sondern der Funktionsaufruf abgebrochen. Ausführliche Beschreibung siehe Seite 72.

**adresse: \*NO / <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Bereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Datenbereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Datenbereichs.

**anzahl** : <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig.
<reg: int:4: integer 0..1860>:	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

### DELDAY = (adresse, anzahl)

Datenbereich mit Kalendertagen, deren Zuordnung zu dem unter NAME angegebenen nicht-zyklischen SYMDAT gelöscht werden soll. Befindet sich in dem Bereich ein Kalendertag, zu dem der Feiertag keine Verknüpfung hat, wird der Funktionsaufruf abgebrochen, und es wird keine Verknüpfung gelöscht. Ausführliche Beschreibung siehe Seite 80.

**adresse**: \*NO / <var: pointer> / (<reg: pointer>)

Operandenwert	Bedeutung
*NO:	Keine Bereichsangabe.
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Bereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Datenbereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Datenbereichs.

**anzahl** : <integer 0..1860> / <var: int:4: integer 0..1860> /  
<reg: int:4: integer 0..1860>

Operandenwert	Bedeutung
<integer 0..1860>:	Anzahl der Kalendertage in diesem Bereich als Direktangabe
<var: int:4: integer 0..1860>:	Name des Feldes mit der Anzahl der Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig.
<reg: int:4: integer 0..1860>:	Register, das die Anzahl der Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

Liste von Kalendertagen (XPAND = DAYLIST)

145	1	*	day-list		
146	1	GCLDPDA	DS	1860CL10	arae for days entry is
147	1	*			described with XPAND=ENTDATE
148	1	GCLDPDA#	EQU	1860	
149	1	GCLDPDY#	EQU	*-GCLDPDA	

Datums-Eintrag (XPAND = ENTDATE)

300	1	*	Date-entry		
301	1	GCLDDTY	DS	CL4	year
302	1	GCLDDTH1	DS	CL1	hyphen
303	1	GCLDDTMO	DS	CL2	month
304	1	GCLDDTH2	DS	CL1	hyphen
305	1	GCLDDTDY	DS	CL2	day
306	1	GCLDDTE#	EQU	*-GCLDDTY	

## 4.6.17 \*OPENCAL Kalender öffnen

Mit der Funktion \*OPENCAL wird der angegebene Kalender geöffnet.

Ein Kalender kann nur geöffnet werden, wenn bereits eine Kalenderdatei mit diesem Namen vorhanden ist. Andernfalls wird der Funktionsaufruf abgebrochen.

Beim Öffnen eines Kalenders legt CALENDAR einen Control-Block an, über den der geöffnete Kalender weiterhin angesprochen wird. Die Adresse des Control-Blocks ist über den Operanden CCB anzugeben, das Layout für den Bereich wird mit XPAND=CCB erzeugt.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *OPENCAL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,CLDNAME = &lt;c-string 1..54: filename 1..54 with catid&gt; /           &lt;var: char:54 filename 1..54 with catid&gt; /           &lt;reg: A(char:54 filename 1..54 with catid)&gt;  ,OPENMOD = *READ / *READALL / *UPDATE / &lt;var: enum-of _open_s:1&gt; /           &lt;reg: enum-of _open_s:1&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*OPENCAL**

Kalenderdatei öffnen.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der zu öffnenden Kalenderdatei. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**CLDNAME=**

Name der zu öffnenden Kalenderdatei, d.h. des zu öffnenden Kalenders. Ausführliche Beschreibung siehe Seite [77](#).

**<c-string 1..54: filename 1..54 with catid>**

Name des Kalenders als Direktangabe.

**<var: char:54 filename 1..54 with catid>**

Name des Feldes mit dem Kalendernamen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:54 filename 1..54 with catid)>**

Register mit der Adresse des Feldes, das den Namen des Kalenders enthält; nur in Verbindung mit MF=M zulässig.

**OPENMOD=**

Open-Modus. Ausführliche Beschreibung siehe Seite [78](#).

**\*READ**

Die Kalenderdatei wird mit SHARUPD=WEAK und OPEN=INPUT zum Lesen geöffnet. Die Kalenderdaten können ausgegeben, aber nicht verändert werden. Mit diesem Open-Modus können beliebig viele Anwender gleichzeitig auf eine Kalenderdatei zugreifen. Jeder Folge-Zugriff auf den Kalender (z.B. \*SHBAS, \*SHLOSYM etc.) beinhaltet einen Dateizugriff.



**\*READALL**

Die Kalenderdatei wird mit SHARUPD=WEAK und OPEN=INPUT zum Lesen geöffnet. Die Daten können ausgegeben, aber nicht verändert werden. Mit diesem Open-Modus können beliebig viele Anwender gleichzeitig auf eine Kalenderdatei zugreifen. Im Unterschied zu \*READ wird der Kalender komplett in den Speicher eingelesen und dort bis zum Schließen des Kalenders gehalten. Dadurch sind bei allen Folgeaufrufen keine weiteren Dateizugriffe nötig. Gleichzeitig steigt jedoch der Speicherbedarf gegenüber dem Open-Modus \*READ.

**\*UPDATE**

Die Kalenderdatei wird mit SHARUPD=WEAK und OPEN=INOUT zum Bearbeiten geöffnet. Dadurch kann immer nur ein Anwender den Kalender mit OPENMOD=\*UPDATE öffnen. Die Daten können ausgegeben und verändert werden. Der Kalender wird komplett in den Speicher eingelesen. Die Änderungen erfolgen im Speicher. Erst nach dem Schließen der Kalenderdatei werden die veränderten Daten in die Kalenderdatei zurückgeschrieben.

**<var: enum-of \_open\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_open\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**Layout des Control-Blocks (XPAND = CCB)**

```

136 1 *   control-block
137 1 GCLDCBTX DS   CL4           text
138 1 GCLDCBAD DS   A             address
139 1 GCLDCBCP DS   F             check pattern
140 1 GCLDPCB# EQU  *-GCLDCBTX

```

## 4.6.18 \*SAVECAL Kalender sichern

Mit der Funktion \*SAVECAL wird der über die Control-Block-Adresse angegebene geöffnete Kalender gesichert. Dabei werden alle Kalenderdaten in die Kalenderdatei auf Platte zurückgeschrieben. Die Kalenderdatei im Speicher bleibt geöffnet, die Kalenderdaten bleiben im Speicher erhalten, so dass die Datei nach dem Sichern sofort weiter bearbeitet werden kann.

### Format

Makro	Operanden
CALENDR	MF = C / D / L / M / E  ,FUNCT = *SAVECAL / <var: enum-of _funct_s:1> / <reg: enum-of _funct_s:1>  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = <var: pointer> / (<reg: pointer>)

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite 205ff.

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### \*SAVECAL

Kalenderdatei sichern.

#### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

#### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

#### CALLER = USER / SYSTEM

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der zu sichernden Kalenderdatei. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

## 4.6.19 \*SHBAS Basis-Information ausgeben

Mit der Funktion \*SHBAS wird die Ausgabe der Basis-Informationen der angegebenen Kalenderdatei angefordert. Das Layout des Ausgabebereichs wird mit XPAND=BASINF erzeugt.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHBAS / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### \*SHBAS

Basis-Information ausgeben.

#### <var: enum-of \_funct\_s:1>

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### <reg: enum-of \_funct\_s:1>

Register, das rechtsbündig den Wert des Equates enthält.

#### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, deren Basis-Informationen ausgegeben werden sollen. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**OUTPUT= (adresse,länge)**

Ausgabebereich für die Basis-Informationen. Ausführliche Beschreibung siehe Seite [83](#).

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge : <integer 0..1000000> / <var: int:4: integer 0..1000000> /**

**<reg: int:4: integer 0..1000000>**

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:****Basis-Informationen (XPAND=BASINF)**

154	1	*	Basis-Information		
155	1	GCLDBIVS	DS	H	version of interface
156	1	GCLDBIND	DS	H	number of days in calendar
157	1	GCLDBINS	DS	H	number of SYMDAT in calendar
158	1	GCLDBINH	DS	H	number of holidays in calendar
159	1	GCLDBICN	DS	CL54	calendar name
160	1	GCLDBIL1	DS	CL10	first date in calendar
161	1	GCLDBIL2	DS	CL10	last date in calendar
162	1	GCLDBIFS	DS	CL20	first SYMDAT in calendar
163	1	GCLDBILS	DS	CL20	last SYMDAT in calendar
164	1	GCLDBIFH	DS	CL30	first holiday in calendar
165	1	GCLDBILH	DS	CL30	last holiday in calendar
166	1	GCLDBIS	DS	7CL14	arae for standard-week entry
167	1	*			is described with
168	1	*			XPAND=ENTSTDW
169	1	GCLDBIS#	EQU	7	
170	1	GCLDOBI#	EQU	*-GCLDBIVS	

**Standard-Arbeitswoche (XPAND = ENTSTDW)**

11	1	*	Standard-week-entry		
12	1	GCLDWEDY	DS	CL3	week-day (MON - SUN)
13	1	GCLDWEAT	DS	CL1	attribute of day (W=workday, F=free day)
14	1	*			
15	1	GCLDWET1	DS	CL5	begin of working time
16	1	GCLDWET2	DS	CL5	end of working time
17	1	GCLDSWE#	EQU	*-GCLDWEDY	

## 4.6.20 \*SHDAYHL Kalendertag mit zugeordneten Feiertagen ausgeben

Mit der Funktion \*SHDAYHL wird die Ausgabe eines Kalendertages mit zugeordneten Feiertagen aus der angegebenen Kalenderdatei angefordert. Es werden nur die Feiertage berücksichtigt, die aktiviert sind. Der Umfang der Ausgabe kann nicht eingeschränkt werden; normalerweise sollte einem Kalendertag auch nur ein Feiertag zugeordnet sein. Das Layout des Ausgabebereichs wird mit XPAND=DAYHOL erzeugt.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHDAYHL / &lt;var: enum-of _funct_s:1&gt; /           &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,DATE = &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*SHDAYHL**

Kalendertag mit zugeordneten Feiertagen ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe des Kalendertages angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**DATE=**

Datum des auszugebenden Kalendertags. Ausführliche Beschreibung siehe Seite 79.

**<c-string 10..10: date 10..10>**

Datum des Kalendertags als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**OUTPUT=**

Ausgabebereich für den Kalendertag mit zugeordneten Feiertagen. Ausführliche Beschreibung siehe Seite 83.



**adresse:** <var: pointer> / (<reg: pointer>)

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge :** <integer 0..1000000> / <var: int:4: integer 0..1000000> /  
 <reg: int:4: integer 0..1000000>

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

**Kalendertag mit zugeordneten Feiertagen (XPAND = DAYHOL)**

215	1	*	day_holiday		
216	1	GCLDDHVS	DS	H	version of interface
217	1	GCLDDHAH	DS	H	number of assigned holidays
218	1	GCLDDHGH	DS	H	number of given holidays
219	1	GCLDDHDT	DS	CL10	date
220	1	GCLDDHA	DS	1024CL34	area for holidays entry is
221	1	*			described with XPAND=ENTHOL
222	1	GCLDDHA#	EQU	1024	
223	1	GCLDODH#	EQU	*-GCLDDHVS	

**Feiertags-Eintrag (XPAND = ENTHOL)**

114	1	*	holiday-entry		
115	1	GCLDHEHN	DS	CL30	holiday-name
116	1	GCLDHEHT	DS	CL1	holiday-type (C=cyclic, N=non cyclic)
117	1	*			holiday is assigned (Y=yes, N=no)
118	1	GCLDHEAS	DS	CL1	number of assigned days
119	1	*			
120	1	GCLDHEAD	DS	H	
121	1	GCLDHLE#	EQU	*-GCLDHEHN	

**4.6.21 \*SHDAYIN Kalendertag mit zugeordneten SYMDATs ausgeben**

Mit der Funktion \*SHDAYIN wird die Ausgabe eines Kalendertages mit zugeordneten SYMDATs aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=DAYINF erzeugt.

**Format**

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,FUNCT = *SHDAYIN / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,DATE = &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,NAME1 = *FIRST / &lt;c-string 1..30: name 1..30&gt;          &lt;var: char:30: date 1..30&gt; /          &lt;reg: A(char:30: date 1..30)&gt;  ,NAME2 = *LAST / &lt;c-string 1..30: name 1..30&gt;          &lt;var: char:30: date 1..30&gt;          &lt;reg: A(char:30: date 1..30)&gt;  ,NUMB = *NO / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt; </pre>

**Operandenbeschreibung****MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHDAYIN**

Tages-Information (Kalendertag) ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe des Kalendertages angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**DATE=**

Datum des auszugebenden Kalendertags. Ausführliche Beschreibung siehe Seite 79.

**<c-string 10..10: date 10..10>**

Datum des Kalendertags als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**OUTPUT= (adresse,länge)**

Ausgabebereich für den Kalendertag mit zugeordneten SYMDAT. Ausführliche Beschreibung siehe Seite 83.

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge : <integer 0..1000000> / <var: int:4: integer 0..1000000> / <reg: int:4: integer 0..1000000>**

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

#### **NAME1=**

Name, ab dem die zugeordneten SYMDAT ausgegeben werden sollen.

##### **\*FIRST**

Der erste SYMDAT-Eintrag im Kalender soll das erste auszugebende SYMDAT sein.

**<c-string 1..30: name 1..30>**

Name des ersten auszugebenden SYMDAT als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den SYMDAT-Namen enthält; nur in Verbindung mit MF=M zulässig.

#### **NAME2=**

Name, bis zu dem die zugeordneten SYMDAT ausgegeben werden sollen

Wird das letzte auszugebende SYMDAT über den Operanden NUMB definiert, wird dieser Operand nicht ausgewertet.

##### **\*LAST**

Der letzte SYMDAT-Eintrag im Kalender soll das letzte auszugebende SYMDAT sein.

**<c-string 1..30: name 1..30>**

Name des letzten auszugebenden SYMDAT als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem SYMDAT-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den SYMDAT-Namen enthält; nur in Verbindung mit MF=M zulässig.

### **NUMB=**

Anzahl der auszugebenden SYMDAT, beginnend mit dem bei NAME1 angegebenen. Eine Angabe ≠ \*NO bewirkt, dass der Operand NAME2 nicht ausgewertet wird.

**\*NO**

Die Ausgabe wird durch die Angabe bei NAME2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden SYMDAT als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden SYMDAT als Integer-Zahl; Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden SYMDAT enthält; nur in Verbindung mit MF=M zulässig.

### **SORT=**

Sortierkriterium: Die zugeordneten SYMDAT sollen entweder nach den SYMDAT-Namen oder nach der Zeit sortiert werden.

**\*TIME**

Sortierkriterium: Zeit.

**\*SYMDAT**

Sortierkriterium: SYMDAT-Namen.

**<var: enum-of \_sort\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 200. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_sort\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

### **Benötigte Layout-Bereiche:**

Kalendertag mit zugeordneten SYMDAT (XPAND = DAYINF)

188	1	*	day_information		
189	1	GCLDDIVS	DS	H	version of interface
190	1	GCLDDIAS	DS	H	number of assigned SYMDAT
191	1	GCLDDISS	DS	H	number of selected SYMDAT
192	1	GCLDDIGS	DS	H	number of given SYMDAT
193	1	GCLDDIDT	DS	CL10	date
194	1	GCLDDID	DS	CL3	week-day (MON - SUN)
195	1	GCLDDIA	DS	CL1	attribute (W=workday, F=free
196	1	*			day)
197	1	GCLDDIA1	DS	CL1	attribute defined for day
198	1	*			(W=workday, F=free day,
199	1	*			S=standard)
200	1	GCLDDIA2	DS	CL1	attribute of day in
201	1	*			standard-week (W=workday,
202	1	*			F=free day)
203	1	GCLDDIHO	DS	CL1	holiday (Y=yes, N=no)
204	1	GCLDDIT1	DS	CL5	begin of working time
205	1	GCLDDIT2	DS	CL5	end of working time
206	1	GCLDDIF1	DS	CL1	filler
207	1	GCLDDIS	DS	4096CL28	area for SYMDAT entry is
208	1	*			described with XPAND=ENTSYMS
209	1	GCLDDIS#	EQU	4096	
210	1	GCLDDI#	EQU	*-GCLDDIVS	

**SYMDAT Kurzeintrag (XPAND = ENTSYMS)**

106	1	*	SYMDAT_short-entry		
107	1	GCLDSSN	DS	CL20	SYMDAT-name
108	1	GCLDSSST	DS	CL8	time
109	1	GCLDSE#	EQU	*-GCLDSSN	

## 4.6.22 \*SHHOLIN Feiertag mit zugeordneten Kalendertagen ausgeben

Mit der Funktion \*SHHOLIN wird die Ausgabe eines Feiertags mit zugeordneten Kalendertagen aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=HOLINF erzeugt.

Bei einem nicht-zyklischen Feiertag werden alle Kalendertage, die innerhalb der vom Anwender definierten Grenzen liegen, ausgegeben, auch wenn sie außerhalb der Kalendergrenzen liegen.

Bei einem zyklischen Feiertag werden nur die innerhalb der Kalendergrenzen liegenden Kalendertage ausgegeben.

Sind die Kalendergrenzen kleiner als ein Jahr, besteht die Möglichkeit, dass ein zyklischer Feiertag keine zugeordneten Kalendertage hat. Der Zähler steht in diesem Fall auf 0, das für den Feiertag definierte Tagesdatum („\*\*\*-mm-dd“) wird trotzdem ausgegeben.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHHOLIN / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..30: name 1..30&gt; /         &lt;var: char:30: name 1..30&gt; /         &lt;reg: A(char:30: name 1..30)&gt;  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,DATE1 = &lt;c-string 10..10: date 10..10&gt; / *FIRST / *NEXT /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *LAST / &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;</pre>



Makro	Operanden
CALENDR (Forts.)	,NUMB = <b>*NO</b> / <integer -4096..4096> / <var: int:4: integer -4096..4096> / <reg: int:4: integer -4096..4096>

## Operandenbeschreibung

### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

#### **\*SHHOLIN**

Feiertags-Informationen ausgeben.

#### **<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite [199](#). Die Angabe ist nur in Verbindung mit MF=M zulässig.

#### **<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

### VERSION = 1

Version 1. Aktuell wird nur diese Version unterstützt.

### CALLER = USER / SYSTEM

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

### CCB=

Control-Block der Kalenderdatei, aus der die Ausgabe des Feiertags angefordert wird. Ausführliche Beschreibung siehe Seite [77](#).

#### **<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

#### **(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

### NAME=

Name des auszugebenden Feiertags. Ausführliche Beschreibung siehe Seite [71](#).

**<c-string 1..30: name 1..30>**

Name des Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem Feiertags-Namen; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

### **OUTPUT= (adresse,länge)**

Ausgabebereich für den Feiertag mit zugeordneten Kalendertagen. Ausführliche Beschreibung siehe Seite [83](#).

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge : <integer 0..1000000> / <var: int:4: integer 0..1000000> / <reg: int:4: integer 0..1000000>**

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

### **DATE1=**

Datum, ab dem die zugeordneten Kalendertage ausgegeben werden sollen. Ausführliche Beschreibung siehe Seite [81](#).

**<c-string 10..10: date 10..10>**

Name des ersten zugeordneten Kalendertags als Direktangabe.

**\*FIRST**

Die Ausgabe der zugeordneten Kalendertage beginnt mit dem ersten Tag dieses Kalenders.

**\*NEXT**

Die Ausgabe der zugeordneten Kalendertage beginnt mit dem auf das aktuelle Tagesdatum folgenden Tag.

**<var: char:10: date 10..10>**

Name des Feldes mit dem ersten zugeordneten Kalendertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum enthält; nur in Verbindung mit MF=M zulässig.

**DATE2=**

Datum, bis zu dem die zugeordneten Kalendertage ausgegeben werden sollen.

Wird der letzte zugeordnete Kalendertag über den Operanden NUMB definiert, wird DATE2 nicht ausgewertet. Ausführliche Beschreibung siehe Seite [81](#)

**\*LAST**

Der letzte Kalendertag im Kalender soll der letzte auszugebende Kalendertag sein.

**<c-string 1..30: name 1..30>**

Name des letzten zugeordneten Kalendertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem letzten zugeordneten Kalendertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum enthält; nur in Verbindung mit MF=M zulässig.

**NUMB=**

Anzahl der auszugebenden Kalendertage, beginnend mit dem bei DATE1 angegebenen. Eine Angabe  $\neq$  \*NO bewirkt, dass der Operand DATE2 nicht ausgewertet wird. Ausführliche Beschreibung siehe Seite [82](#).

**\*NO**

Die Ausgabe wird durch die Angabe bei DATE2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden Kalendertage als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:****Feiertag mit zugeordneten Kalendertagen (XPAND = HOLINF)**

281	1	*	holiday-information		
282	1	GCLDHIVS	DS	H	version of interface
283	1	GCLDHIAD	DS	H	number of assigned days
284	1	GCLDHISD	DS	H	number of selected days
285	1	GCLDHIGD	DS	H	number of given days
286	1	GCLDHIHN	DS	CL30	holiday-name
287	1	GCLDHIHT	DS	CL1	holiday-type (C=cyclic N=non cyclic)
288	1	*			holiday is assigned (Y=yes, N=no)
289	1	GCLDHIAS	DS	CL1	
290	1	*			
291	1	GCLDHIA	DS	1860CL10	area for assigned days entry
292	1	*			is described with
293	1	*			XPAND=ENTDATE
294	1	GCLDHIA#	EQU	1860	
295	1	GCLDOHI#	EQU	*-GCLDHIVS	

**Datums-Eintrag (XPAND = ENTDATE)**

300	1	*	Date-entry		
301	1	GCLDDTY	DS	CL4	year
302	1	GCLDDTH1	DS	CL1	hyphen
303	1	GCLDDTMO	DS	CL2	month
304	1	GCLDDTH2	DS	CL1	hyphen
305	1	GCLDDTDY	DS	CL2	day
306	1	GCLDDTE#	EQU	*-GCLDDTY	

### 4.6.23 \*SHLODAY Liste von Kalendertagen ausgeben

Mit der Funktion \*SHLODAY wird die Ausgabe einer Liste von Kalendertagen aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=LODAY erzeugt.

#### Format

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,FUNCT = *SHLODAY / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,DATE1 = &lt;c-string 10..10: date 10..10&gt; / *FIRST / *NEXT /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *LAST / &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,NUMB = *NO / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt; </pre>

#### Operandenbeschreibung

##### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

##### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHLODAY**

Liste von Kalendertagen ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe der Liste mit Kalendertagen angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**OUTPUT= (adresse,länge)**

Ausgabebereich für die Liste mit Kalendertagen. Ausführliche Beschreibung siehe Seite 83.

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge** : <integer 0..1000000> / <var: int:4: integer 0..1000000> /  
<reg: int:4: integer 0..1000000>

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**DATE1=**

Erster Kalendertag, der ausgegeben werden soll. Ausführliche Beschreibung siehe Seite 81.

**<c-string 10..10: date 10..10>**

Name des ersten auszugebenden Kalendertags als Direktangabe.

**\*FIRST**

Die Ausgabe der Kalendertage beginnt mit dem ersten Tag dieses Kalenders.

**\*NEXT**

Die Ausgabe der Kalendertage beginnt mit dem auf das aktuelle Tagesdatum folgenden Tag.

**<var: char:10: date 10..10>**

Name des Feldes mit dem ersten auszugebenden Kalendertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das den Namen des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**DATE2=**

Letzter Kalendertag, der ausgegeben werden soll.

Wird der letzte Kalendertag über den Operanden NUMB definiert, wird DATE2 nicht ausgewertet. Ausführliche Beschreibung siehe Seite 81.

**\*LAST**

Der letzte auszugebende Kalendertag soll der letzte Kalendertag im Kalender sein.

**<c-string 1..10: name 1..10>**

Name des letzten auszugebenden Kalendertags als Direktangabe.

**<var: char:10: name 1..10>**

Name des Feldes mit dem letzten auszugebenden Kalendertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das den Namen des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**NUMB=**

Anzahl der auszugebenden Kalendertage, beginnend mit dem bei DATE1 angegebenen. Eine Angabe  $\neq$  \*NO bewirkt, dass der Operand DATE2 nicht ausgewertet wird. Ausführliche Beschreibung siehe Seite [82](#).

**\*NO**

Die Ausgabe wird durch die Angabe bei DATE2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden Kalendertage als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden Kalendertage enthält; nur in Verbindung mit MF=M zulässig.



**Benötigte Layout-Bereiche:**

## Liste von Kalendertagen (XPAND = LODAY)

175	1	*	List-of-days		
176	1	GCLDLDVS	DS	H	version of interface
177	1	GCLDLDAD	DS	H	number of days in calendar
178	1	GCLDLDSD	DS	H	number of selected days
179	1	GCLDLDGD	DS	H	number of given days
180	1	GCLDLDA	DS	1860CL32	area for days entry is
181	1	*			described with XPAND=ENTDAY
182	1	GCLDLDA#	EQU	1860	
183	1	GCLDOLD#	EQU	*-GCLDLDVS	

## Kalendertag (XPAND = ENTDAY)

64	1	*	Day-entry		
65	1	GCLDDEDT	DS	CL10	date
66	1	GCLDDED	DS	CL3	week-day (MON - SUN)
67	1	GCLDDEA	DS	CL1	attribute (W=workday, F=free day)
68	1	*			
69	1	GCLDDEAD	DS	CL1	attribute defined for day
70	1	*			(W=workday, F=free day, S=standard)
71	1	*			
72	1	GCLDDEAS	DS	CL1	attribute of day in
73	1	*			standard-week (W=workday, F=free day)
74	1	*			
75	1	GCLDDEHO	DS	CL1	holiday (Y=yes, N=no)
76	1	GCLDDET1	DS	CL5	begin of working time
77	1	GCLDDET2	DS	CL5	end of working time
78	1	GCLDDEFI	DS	CL1	filler
79	1	GCLDDEH	DS	H	number of assigned holidays
80	1	GCLDDES	DS	H	number of assigned SYMDAT
81	1	GCLDDYE#	EQU	*-GCLDDEDT	

## 4.6.24 \*SHLOHOL Liste von Feiertagen ausgeben

Mit der Funktion \*SHLOHOL wird die Ausgabe einer Liste von Feiertagen aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=LOHOL erzeugt.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHLOHOL / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,NAME1 = *FIRST / &lt;c-string 1..30: name 1..30&gt;          &lt;var: char:30: date 1..30&gt; /          &lt;reg: A(char:30: date 1..30)&gt;  ,NAME2 = *LAST / &lt;c-string 1..30: name 1..30&gt;          &lt;var: char:30: date 1..30&gt; /          &lt;reg: A(char:30: date 1..30)&gt;  ,NUMB = *NO / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHLOHOL**

Liste von Feiertagen ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe einer Liste von Feiertagen angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**OUTPUT=**

Ausgabebereich für die Feiertags-Liste. Ausführliche Beschreibung siehe Seite 83.

**adresse: <var: pointer> / (<reg: pointer>)**

<b>Operandenwert</b>	<b>Bedeutung</b>
<b>&lt;var: pointer&gt;:</b>	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
<b>(&lt;reg: pointer&gt;):</b>	Register mit der Adresse des Ausgabebereichs.

**länge** : <integer 0..1000000> / <var: int:4: integer 0..1000000> /  
 <reg: int:4: integer 0..1000000>

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**NAME1=**

Erster Feiertag, der ausgegeben werden soll.

**\*FIRST**

Der erste Feiertag im Kalender soll der erste ausgegebene Feiertag sein.

**<c-string 1..30: name 1..30>**

Name des ersten auszugebenden Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem ersten auszugebenden Feiertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**NAME2=**

Letzter Feiertag, der ausgegeben werden soll.

Wird der letzte auszugebende Feiertag über den Operanden NUMB definiert, wird dieser Operand nicht ausgewertet.

**\*LAST**

Der letzte Feiertag im Kalender soll der letzte auszugebende Feiertag sein.

**<c-string 1..30: name 1..30>**

Name des letzten auszugebenden Feiertags als Direktangabe.

**<var: char:30: name 1..30>**

Name des Feldes mit dem letzten auszugebenden Feiertag; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:30: name 1..30)>**

Register mit der Adresse des Feldes, das den Namen des Feiertags enthält; nur in Verbindung mit MF=M zulässig.

**NUMB=**

Anzahl der auszugebenden Feiertage, beginnend mit dem bei NAME1 angegebenen.  
Eine Angabe ≠ \*NO bewirkt, dass der Operand NAME2 nicht ausgewertet wird. Ausführliche Beschreibung siehe Seite 82.

**\*NO**

Die Ausgabe wird durch die Angabe bei NAME2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden Kalendertage als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden Kalendertage als Integer-Zahl;  
Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden Feiertage enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

## Liste von Feiertagen (XPAND = LOHOL)

268	1	*	List-of-holidays		
269	1	GCLDLHVS	DS	H	version of interface
270	1	GCLDLHAH	DS	H	number of holidays in calendar
271	1	GCLDLHSH	DS	H	number of selected holidays
272	1	GCLDLHGH	DS	H	number of given holidays
273	1	GCLDLHA	DS	1024CL34	area for holidays entry is
274	1	*			described with XPAND=ENTHOL
275	1	GCLDLHA#	EQU	1024	
276	1	GCLDOLH#	EQU	*-GCLDLHVS	

## Feiertag (XPAND = ENTHOL)

114	1	*	holiday-entry		
115	1	GCLDHEHN	DS	CL30	holiday-name
116	1	GCLDHEHT	DS	CL1	holiday-type (C=cyclic, N=non cyclic)
117	1	*			
118	1	GCLDHEAS	DS	CL1	holiday is assigned (Y=yes, N=no)
119	1	*			
120	1	GCLDHEAD	DS	H	number of assigned days
121	1	GCLDHLE#	EQU	*-GCLDHEHN	

## 4.6.25 \*SHLOSYM Liste von SYMDAT ausgeben

Mit der Funktion \*SHLOSYM wird die Ausgabe einer Liste von SYMDAT aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=LOSYM erzeugt.

### Format

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHLOSYM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,NAME1 = *FIRST / &lt;c-string 1..20: name 1..20&gt;          &lt;var: char:20: date 1..20&gt; /          &lt;reg: A(char:20: date 1..20)&gt;  ,NAME2 = *LAST / &lt;c-string 1..20: name 1..20&gt;          &lt;var: char:20: date 1..20&gt; /          &lt;reg: A(char:20: date 1..20)&gt;  ,NUMB = *<u>NO</u> / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt;</pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

#### FUNCT=

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHLOSYM**

Liste von SYMDAT (symbolische Datumsangaben) ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe der SYMDAT-Liste angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**OUTPUT=**

Ausgabebereich für die SYMDAT-Liste. Ausführliche Beschreibung siehe Seite 83.

**adresse: <var: pointer> / (<reg: pointer>)**

<b>Operandenwert</b>	<b>Bedeutung</b>
<b>&lt;var: pointer&gt;:</b>	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
<b>(&lt;reg: pointer&gt;):</b>	Register mit der Adresse des Ausgabebereichs.

**länge** : <integer 0..1000000> / <var: int:4: integer 0..1000000>  
 <reg: int:4: integer 0..1000000>

<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**NAME1=**

Erstes SYMDAT, das ausgegeben werden soll.

**\*FIRST**

Das erste SYMDAT im Kalender ist das als erstes auszugebende.

**<c-string 1..20: name 1..20>**

Name des ersten auszugebenden SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem ersten auszugebenden SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

Eine ausführliche Beschreibung des Operanden NAME1 finden Sie auf Seite [82](#).

**NAME2=**

Letztes SYMDAT, das ausgegeben werden soll.

Wird das letzte auszugebende SYMDAT über den Operanden NUMB definiert, wird NAME2 nicht ausgewertet.

**\*LAST**

Das letzte SYMDAT im Kalender soll das letzte auszugebende SYMDAT sein.

**<c-string 1..20: name 1..20>**

Name des letzten auszugebenden SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem letzten auszugebenden SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.



**NUMB=**

Anzahl der auszugebenden SYMDAT, beginnend mit dem bei NAME1 angegebenen.  
Eine Angabe  $\neq$  \*NO bewirkt, dass der Operand NAME2 nicht ausgewertet wird. Ausführliche Beschreibung siehe Seite [82](#).

**\*NO**

Die Ausgabe wird durch die Angabe bei NAME2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden Kalendertage als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden Kalendertage als Integer-Zahl;  
Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden SYMDAT enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

## Liste von SYMDAT (XPAND = LOSYM)

228	1	*	List-of-SYMDAT		
229	1	GCLDLSVS	DS	H	version of interface
230	1	GCLDLSAS	DS	H	number of SYMDAT in calendar
231	1	GCLDLSSS	DS	H	number of selected SYMDAT
232	1	GCLDLSGS	DS	H	number of given SYMDAT
233	1	GCLDLSA	DS	4096CL56	area for SYMDAT entry is
234	1	*			described with XPAND=ENTSYM
235	1	GCLDLSA#	EQU	4096	
236	1	GCLDOLS#	EQU	*-GCLDLSVS	

## SYMDAT-Eintrag (XPAND = ENTSYM)

86	1	*	SYMDAT-entry		
87	1	GCLDSESN	DS	CL20	SYMDAT-name
88	1	GCLDSEST	DS	CL1	SYMDAT-type (C=cyclic, N=non cyclic)
89	1	*			
90	1	GCLDSET	DS	CL8	time
91	1	GCLDSECT	DS	CL1	kind of cycle (D=day, W=week,0=workday, M=month)
92	1	*			cycle-value
93	1	GCLDSECV	DS	H	alternative for free days
94	1	GCLDSEAL	DS	CL1	(B=before, A=after, S=skip, O=on)
95	1	*			
96	1	*			filler
97	1	GCLDSEF1	DS	CL1	start-date for cyclic SYMDAT
98	1	GCLDSED1	DS	CL10	end-date for cyclic SYMDAT
99	1	GCLDSED2	DS	CL10	number of assigned days
100	1	GCLDSEAD	DS	H	
101	1	GCLDSYE#	EQU	*-GCLDSESN	

**4.6.26 \*SHSYMIN SYMDAT mit zugeordneten Kalendertagen ausgeben**

Mit der Funktion \*SHSYMIN wird die Ausgabe eines SYMDAT mit zugeordneten Kalendertagen aus der angegebenen Kalenderdatei angefordert.

Das Layout des Ausgabebereichs wird mit XPAND=SYMINF erzeugt.

**Format**

Makro	Operanden
CALENDR	<pre>MF = C / D / L / M / E  ,FUNCT = *SHSYMIN / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,NAME = &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,DATE1 = &lt;c-string 10..10: date 10..10&gt; / *FIRST / *NEXT /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,DATE2 = *LAST / &lt;c-string 10..10: date 10..10&gt; /         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,NUMB = *NO / &lt;integer -4096..4096&gt; /         &lt;var: int:4: integer -4096..4096&gt; /         &lt;reg: int:4: integer -4096..4096&gt;</pre>

**Operandenbeschreibung****MF=**

Der MF-Operand steuert die Form der Makroauflösung, siehe Seite [205ff.](#)

**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHSYMIN**

SYMDAT-Informationen ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert; siehe Seite 199. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe des SYMDAT mit zugeordneten Kalendertagen angefordert wird. Ausführliche Beschreibung siehe Seite 77.

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**NAME=**

Name des auszugebenden SYMDAT. Ausführliche Beschreibung siehe Seite 71.

**<c-string 1..20: name 1..20>**

Name des SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**OUTPUT=**

Ausgabebereich für das SYMDAT mit zugeordneten Kalendertagen. Ausführliche Beschreibung siehe Seite 83.

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge : <integer 0..1000000> / <var: int:4: integer 0..1000000>  
<reg: int:4: integer 0..1000000>**

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

### DATE1=

Erster zugeordneter Kalendertag, der ausgegeben werden soll. Ausführliche Beschreibung siehe Seite 81.

**<c-string 10..10: date 10..10>**

Datum des ersten auszugebenden Kalendertags als Direktangabe.

### \*FIRST

Die Ausgabe der zugeordneten Kalendertage beginnt mit dem ersten Tag dieses Kalenders.

### \*NEXT

Die Ausgabe der zugeordneten Kalendertage beginnt mit dem auf das aktuelle Tagesdatum folgenden Tag.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des ersten auszugebenden Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**DATE2=**

Letzter zugeordneter Kalendertag, der ausgegeben werden soll.

Wird der letzte zugeordnete Kalendertag über den Operanden NUMB definiert, wird DATE2 nicht ausgewertet. Ausführliche Beschreibung siehe Seite [81](#).

**\*LAST**

Der letzte Kalendertag im Kalender soll der letzte auszugebende Kalendertag sein.

**<c-string 1..10: name 1..10>**

Datum des letzten auszugebenden Kalendertags.

**<var: char:10: name 1..10>**

Name des Feldes mit dem Datum des letzten auszugebenden Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**NUMB=**

Anzahl der auszugebenden Kalendertage, beginnend mit dem bei DATE1 angegebenen. Eine Angabe  $\neq$  \*NO bewirkt, dass der Operand DATE2 nicht ausgewertet wird. Ausführliche Beschreibung siehe Seite [82](#).

**\*NO**

Die Ausgabe wird durch die Angabe bei DATE2 begrenzt.

**<integer -4096..4096>**

Anzahl der auszugebenden Kalendertage als Direktangabe.

**<var: int:4: integer -4096..4096>**

Name des Feldes mit der Anzahl der auszugebenden Kalendertage als Integer-Zahl; Feld-Länge = 4 Byte.

**<reg: int:4: integer -4096..4096>**

Register, das die Anzahl der auszugebenden Kalendertage enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:****SYMDAT mit zugeordneten Kalendertagen (XPAND = SYMINF)**

241	1	*	SYMDAT_information		
242	1	GCLDSIVS	DS	H	version of interface
243	1	GCLDSIAD	DS	H	number of assigned days
244	1	GCLDSISD	DS	H	number of selected days
245	1	GCLDSIGD	DS	H	number of given days
246	1	GCLDSISN	DS	CL20	SYMDAT-name
247	1	GCLDSIST	DS	CL1	SYMDAT-type (C=cyclic, N=non cyclic)
248	1	*			
249	1	GCLDSIT	DS	CL8	time
250	1	GCLDSICT	DS	CL1	kind of cycle (D=day, W=week, O=workday, M=month)
251	1	*			
252	1	GCLDSICV	DS	H	cycle-value
253	1	GCLDSIAL	DS	CL1	alternative for free days
254	1	*			(B=before, A=after, S=skip, O=on)
255	1	*			
256	1	GCLDSID1	DS	CL10	start-date for cyclic SYMDAT
257	1	GCLDSID2	DS	CL10	end-date for cyclic SYMDAT
258	1	GCLDSIF1	DS	CL1	filler1
259	1	GCLDSIA	DS	1860CL10	area for assigned days entry
260	1	*			is described with
261	1	*			XPAND=ENTDATE
262	1	GCLDSIA#	EQU	1860	
263	1	GCLDOSI#	EQU	*-GCLDSIVS	

**Datums-Eintrag (XPAND = ENTDAT)**

300	1	*	Date-entry		
301	1	GCLDDTY	DS	CL4	year
302	1	GCLDDTH1	DS	CL1	hyphen
303	1	GCLDDTMO	DS	CL2	month
304	1	GCLDDTH2	DS	CL1	hyphen
305	1	GCLDDTDY	DS	CL2	day
306	1	GCLDDTE#	EQU	*-GCLDDTY	

## 4.6.27 \*SHNESTM Nächsten Zeitpunkt zu SYMDATs ausgeben

Mit der Funktion \*SHNESTM wird ab einem Startzeitpunkt zu einer Menge von SYMDATs aus der angegebenen Kalenderdatei der nächste Zeitpunkt angefordert. Das Layout des Ausgabebereichs wird mit XPAND=NESTM erzeugt.

### Format

Makro	Operanden
CALENDR	<pre> MF = C / D / L / M / E  ,FUNCT = *SHNESTM / &lt;var: enum-of _funct_s:1&gt; /         &lt;reg: enum-of _funct_s:1&gt;  ,VERSION = 1  ,CALLER = <u>USER</u> / SYSTEM  ,CCB = &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)  ,OUTPUT = (adresse, länge)           adresse: &lt;var: pointer&gt; / (&lt;reg: pointer&gt;)           länge: &lt;integer 0..1000000&gt; /                 &lt;var: int:4: integer 0..1000000&gt; /                 &lt;reg: int:4: integer 0..1000000&gt;  ,NAME1 = *FIRST / &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,NAME2 = *LAST / &lt;c-string 1..20: name 1..20&gt; /         &lt;var: char:20: name 1..20&gt; /         &lt;reg: A(char:20: name 1..20)&gt;  ,DATE = &lt;c-string 10..10: date 10..10&gt;         &lt;var: char:10: date 10..10&gt; /         &lt;reg: A(char:10: date 10..10)&gt;  ,TIME = &lt;c-string 8..8: time 8..8&gt; /         &lt;var: char:8: time 8..8&gt; /         &lt;reg: A(char:8: time 8..8)&gt; </pre>

### Operandenbeschreibung

#### MF=

Der MF-Operand steuert die Form der Makroauflösung, siehe [Seite 205](#).



**FUNCT=**

Funktionscode, über den die gewünschte Kalenderfunktion angegeben wird.

**\*SHNESTM**

Nächsten Zeitpunkt aus einer Menge von SYMDATs ausgeben.

**<var: enum-of \_funct\_s:1>**

Name des Feldes mit dem Wert des Equates. Die Equates sind in der C- oder D-Form definiert. Die Angabe ist nur in Verbindung mit MF=M zulässig.

**<reg: enum-of \_funct\_s:1>**

Register, das rechtsbündig den Wert des Equates enthält.

**VERSION = 1**

Version 1. Aktuell wird nur diese Version unterstützt.

**CALLER = USER / SYSTEM**

Aufrufer der Schnittstelle: Anwender (TU) oder System (TPR).

**CCB=**

Control-Block der Kalenderdatei, aus der die Ausgabe des SYMDAT mit zugeordneten Kalendertagen angefordert wird. Ausführliche Beschreibung siehe Seite [77](#).

**<var: pointer>**

Alternativ sind folgende 2 Angaben möglich:

- Adresse des Control-Blocks dieser Kalenderdatei als direkte Adressangabe in der Form A(feld).
- Name des Feldes mit der Adresse des Control-Blocks; nur in Verbindung mit MF=M zulässig.

**(<reg: pointer>)**

Register mit der Adresse des Control-Blocks.

**OUTPUT=**

Ausgabebereich für das SYMDAT mit zugeordneten Kalendertagen. Ausführliche Beschreibung siehe Seite [83](#).

**adresse: <var: pointer> / (<reg: pointer>)**

Operandenwert	Bedeutung
<var: pointer>:	Alternativ sind folgende 2 Angaben möglich: - Adresse des Ausgabebereichs als direkte Adressangabe in der Form A(feld). - Name des Feldes mit der Adresse des Ausgabebereichs; nur in Verbindung mit MF=M zulässig.
(<reg: pointer>):	Register mit der Adresse des Ausgabebereichs.

**länge** : <integer 0..1000000> / <var: int:4: integer 0..1000000>  
 <reg: int:4: integer 0..1000000>

Operandenwert	Bedeutung
<integer 0..1000000>:	Länge des Ausgabebereichs in Byte als Direktangabe.
<var: int:4: integer 0..1000000>:	Name des Feldes mit der Anzahl der Bytes als Integer-Zahl; Feld-Länge = 4 Byte; nur in Verbindung mit MF=M zulässig
<reg: int:4: integer 0..1000000>:	Register, das die Längenangabe für den Ausgabebereich enthält; nur in Verbindung mit MF=M zulässig.

**NAME1=**

Name des ersten SYMDAT, zu dem Zeitpunkte in Betracht kommen. Ausführliche Beschreibung siehe [Seite 82](#).

**\*FIRST**

Das erste SMDAT im Kalender ist die Untergrenze.

**<c-string 1..20: name 1..20>**

Name des ersten SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem ersten SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des ersten SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**NAME2=**

Name des letzten SYMDAT, zu dem Zeitpunkte in Betracht kommen. Ausführliche Beschreibung siehe [Seite 82](#).

**\*LAST**

Das letzte SMDAT im Kalender soll das letzte auszugebende SYMDAT sein.

**<c-string 1..20: name 1..20>**

Name des letzten SYMDAT als Direktangabe.

**<var: char:20: name 1..20>**

Name des Feldes mit dem letzten SYMDAT; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:20: name 1..20)>**

Register mit der Adresse des Feldes, das den Namen des letzten SYMDATs enthält; nur in Verbindung mit MF=M zulässig.

**DATE=**

Datum, ab dem (zusammen mit der Uhrzeit TIME) der nächste Zeitpunkt ermittelt werden soll. Ausführliche Beschreibung siehe [Seite 79](#).

**<c-string 10..10: date 10..10>**

Datum als Direktangabe.

**<var: char:10: date 10..10>**

Name des Feldes mit dem Datum des Kalendertags; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:10: date 10..10)>**

Register mit der Adresse des Feldes, das das Datum des Kalendertags enthält; nur in Verbindung mit MF=M zulässig.

**TIME=**

Uhrzeit, ab der (zusammen mit dem Datum DATE) der nächste Zeitpunkt ermittelt werden soll. Die Angabe erfolgt in der Form hh:mm:ss (hh=Stunden, mm=Minuten, ss= Sekunden). Ausführliche Beschreibung siehe [Seite 80](#).

**<c-string 8..8: time 8..8>**

Uhrzeit als Direktangabe.

**<var: char:8: time 8..8>**

Name des Feldes mit der Uhrzeit; nur in Verbindung mit MF=M zulässig.

**<reg: A(char:8: time 8..8)>**

Register mit der Adresse des Feldes, das die Uhrzeit enthält; nur in Verbindung mit MF=M zulässig.

**Benötigte Layout-Bereiche:**

Datum, Uhrzeit und Name des zugehörigen SYMDAT (XPAND = NESTM)

264	1	*	List-of-SYMDAT		
265	1	GCLDNSVS	DS	H	version of interface
266	1	GCLDNSTD	DS	CL10	next date
267	1	GCLDNSTT	DS	CL8	next time
268	1	GCLDNSNM	DS	CL20	SYMDAT name
269	1	*			
269	1	GCLDONS#	EQU	*-GCLDNSVS	

## 4.7 Parameterliste

```

    PARLIST CALENDR MF=D,VERSION=1,XPAND=PARAM
1 PARLIST MFTST MF=D,PREFIX=G,MACID=CLD,ALIGN=F,
1
1          DMACID=CLD,SUPPORT=(E,D,C,M,L),DNAME=CLDPMDL
2 PARLIST DSECT ,
2
2          *,##### PREFIX=G, MACID=CLD #####
1 *      parameterarea description
1 GCLDHDR FHDR MF=(C,GCLD),EQUATES=NO      Standard-Header
2 GCLDHDR DS      0A
2 GCLDFHE DS      OXL8                      0  GENERAL PARAMETER AREA HEADER
2 *
2 GCLDIFID DS      0A                      0  INTERFACE IDENTIFIER
2 GCLDFCTU DS      AL2                     0  FUNCTION UNIT NUMBER
2 *
2 *                                          BIT 15  HEADER FLAG BIT,
2 *                                          MUST BE RESET UNTIL FURTHER NOTICE
2 *                                          BIT 14-12 UNUSED, MUST BE RESET
2 *                                          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 GCLDFCT DS      AL1                      2  FUNCTION NUMBER
2 GCLDFCTV DS      AL1                     3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 GCLDRET DS      0A                      4  GENERAL RETURN CODE
2 GCLDSRET DS      0AL2                   4  SUB RETURN CODE
2 GCLDSR2 DS      AL1                     4  SUB RETURN CODE 2
2 GCLDSR1 DS      AL1                     5  SUB RETURN CODE 1
2 GCLDMRET DS      0AL2                   6  MAIN RETURN CODE
2 GCLDMR2 DS      AL1                     6  MAIN RETURN CODE 2
2 GCLDMR1 DS      AL1                     7  MAIN RETURN CODE 1
2 GCLDFHL EQU      8                      8  GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *      main return codes
1 GCLDRSUC EQU      0                      no error
1 GCLDRINT EQU      1                      parameter error
1 GCLDRCEX EQU      2                      calendar already existing
1 GCLDRCNE EQU      3                      calendar does not exist
1 GCLDRDEC EQU      4                      DMS-error on calendar-file
1 GCLDRNCL EQU      5                      file is not a calendar
1 GCLDRCIN EQU      6                      calendar is inconsistent
1 GCLDRNUP EQU      7                      update of calendar not
1 *                                          possible at the moment
1 GCLDRHFN EQU      8                      holiday-file does not exist
1 GCLDRDHF EQU      9                      DMS-error on holiday-file
1 GCLDRTMD EQU     10                      too many days between calendar
1 *                                          limits
1 GCLDRELS EQU     11                      end-date less than start-date
1 GCLDRNME EQU     12                      no more memory available
1 GCLDRCBI EQU     13                      control-block invalid

```

1	GCLDRDTE	EQU	14	date outside of calendar
1	*			limits
1	GCLDRSYE	EQU	15	SYMDAT already existing
1	GCLDRSYN	EQU	16	SYMDAT does not exist
1	GCLDRCSY	EQU	17	SYMDAT is a cyclic SYMDAT
1	GCLDRNSY	EQU	18	SYMDAT is a non-cyclic SYMDAT
1	GCLDRDTE	EQU	19	assignment to date already
1	*			existing
1	GCLDRDTN	EQU	20	assignment to date does not
1	*			exist
1	GCLDRDTI	EQU	21	invalid date
1	GCLDRHLE	EQU	22	holiday already existing
1	GCLDRHLN	EQU	23	holiday does not exist
1	GCLDRCHL	EQU	24	holiday is a cyclic holiday
1	GCLDRNHL	EQU	25	holiday is a non-cyclic
1	*			holiday
1	GCLDRHAC	EQU	26	holiday already activated
1	GCLDRHNA	EQU	27	holiday is not activated
1	GCLDRNIF	EQU	28	output-area too small: no
1	*			information available
1	GCLDRSIF	EQU	29	output-area too small: some
1	*			information available
1	GCLDRWSY	EQU	30	definition of SYMDAT-area
1	*			invalid
1	GCLDRWHL	EQU	31	definition of holiday-area
1	*			invalid
1	GCLDRSWI	EQU	32	area for standardweek is
1	*			incorrect
1	GCLDRHFI	EQU	33	entry of holiday-file is
1	*			incorrect
1	GCLDRILI	EQU	34	LIMIT1 < 1970 or LIMIT2 > 2030
1	*			or new LIMIT1 > old LIMIT2
1	GCLDRSDO	EQU	35	start-date of cyclic SYMDAT
1	*			outside of calendar-limits
1	*			(warning)
1	GCLDRFNA	EQU	36	function not allowed with
1	*			OPENMOD = *READ or *READALL
1	GCLDRSDD	EQU	37	start date for cyclic SYMDAT
1	*			with CYCLTYP=WORKDAY has been
1	*			deleted as it is outside
1	*			calendar limits
1	GCLDRPRE	EQU	38	internal error in program
1	GCLDRCVI	EQU	39	version of calendar ist not
1	*			supported by version of
1	*			interface
1	GCLDRTMS	EQU	40	more than 4096 SYMDAT have
1	*			been created
1	GCLDRTMH	EQU	41	more than 1024 holidays have

1 *			been created
1 GCLDRFNS	EQU	65535	function not supported
1 GCLDRVNA	EQU	65535	version not supported
1 GCLDRAER	EQU	65535	alignment_error
1 *			
1 GCLDPFCT	DS	FL1	funct =
1 *	function	codes	
1 GCLDFCRC	EQU	1	create calendar
1 GCLDFOPC	EQU	2	open calendar
1 GCLDFCLC	EQU	3	close calendar
1 GCLDFSVC	EQU	4	save calendar
1 GCLDFMDB	EQU	5	modify basis information
1 GCLDFMDY	EQU	6	modify day
1 GCLDFCNS	EQU	7	create non cyclic symdat
1 GCLDFCCS	EQU	8	create cyclic symdat
1 GCLDFDES	EQU	9	delete symdat
1 GCLDFMNS	EQU	10	modify non cyclic symdat
1 GCLDFMCS	EQU	11	modify cyclic symdat
1 GCLDFCNH	EQU	12	create non cyclic holiday
1 GCLDFCCH	EQU	13	create cyclic holiday
1 GCLDFDEH	EQU	14	delete holiday
1 GCLDFMNH	EQU	15	modify non cyclic holiday
1 GCLDFMCH	EQU	16	modify cyclic holiday
1 GCLDFAKH	EQU	17	aktivate holiday
1 GCLDFDAH	EQU	18	deaktivate holiday
1 GCLDFSBA	EQU	19	show basis information
1 GCLDFS LD	EQU	20	show list of days
1 GCLDFS DI	EQU	21	show day-information
1 GCLDFS DH	EQU	22	show day with holidays
1 GCLDFS LS	EQU	23	show list of symdat
1 GCLDFS SI	EQU	24	show symdat-information
1 GCLDFS LH	EQU	25	show list of holidays
1 GCLDFS HI	EQU	26	show holiday-information
1 *			
1 GCLDPOPM	DS	FL1	openmod =
1 *	open	modus	
1 GCLDREAD	EQU	1	read
1 GCLDRDAL	EQU	2	read complete calendar
1 GCLDUPDT	EQU	3	update
1 *			
1 GCLDPATR	DS	FL1	attribute =
1 *	attribute	of day	
1 GCLDATUC	EQU	0	unchanged
1 GCLDFREE	EQU	1	free day
1 GCLDWORK	EQU	2	work day
1 GCLDSTWK	EQU	3	standard wekk
1 *			
1 GCLDPTYP	DS	FL1	cycltyp =

1 *	kind of cycle		
1	GCLDCYUC	EQU 0	unchanged
1	GCLDDAY	EQU 1	cycle = day
1	GCLDWKD	EQU 2	cycle = work-day
1	GCLDWEK	EQU 3	cycle = week
1	GCLDMON	EQU 4	cycle = month
1 *			
1	GCLDPALT	DS FL1	alternative =
1 *	alternative		
1	GCLDALUC	EQU 0	unchanged
1	GCLDBEFR	EQU 1	insert before free day
1	GCLDAFT	EQU 2	insert after free day
1	GCLDSKIP	EQU 3	dont insert
1	GCLDON	EQU 4	insert on free day
1 *			
1	GCLDPSAV	DS FL1	save =
1 *	save		
1	GCLDYES	EQU 1	save calendar
1	GCLDNO	EQU 2	dont save calendar
1 *			
1	GCLDPSOR	DS FL1	sort =
1 *	sort		
1	GCLDSSYM	EQU 1	sort by symdat
1	GCLDSTIM	EQU 2	sort by time
1 *			
1	GCLDPFL1	DS AL1	indicator
1	GCLDPRS1	EQU X'FF'	reserved
1	GCLDPFL2	DS AL1	indicator
1	GCLDPRS2	EQU X'FF'	reserved
1	GCLDPFI1	DS CL3	filler
1	GCLDPNUM	DS F	number of days or SYMDAT
1	GCLDPCVL	DS F	number of days between two
1 *			entries
1	GCLDPDMS	DS F	DMS returncode
1	GCLDPCCB	DS A	address of control-block
1	GCLDPWKA	DS A	address of standard week
1	GCLDPADD	DS A	address of days to add
1	GCLDPAD#	DS F	number of days in area
1	GCLDPDEL	DS A	address of days to delete
1	GCLDPDE#	DS F	number of days in area
1	GCLDPOUT	DS A	address of output-area
1	GCLDPOTL	DS F	length of output-area
1	GCLDPCLN	DS CL54	calendar-name
1	GCLDPNAM	DS CL30	name of SYMDAT or holiday
1	GCLDPNM1	DS CL30	first name (*FIRST = first
1 *			SYMDAT or holiday in
1 *			calendar)
1	GCLDPNM2	DS CL30	last name (*LAST = last SYMDAT



1 *			or holiday in calendar)	
1	GCLDPDAT	DS	CL10	date
1	GCLDPDT1	DS	CL10	start-date (*UNCH = start-date
1 *				of SYMDAT is unchanged,
1 *				*FIRST = first date in
1 *				calendar)
1	GCLDPDT2	DS	CL10	end-date (*UNCH = end-date of
1 *				SYMDAT is unchanged, *UNDEF =
1 *				end-date of SYMDAT is
1 *				undefined, *LAST = last date
1 *				in calendar)
1	GCLDPLM1	DS	CL10	first date in calendar (*UNCH
1 *				= calendar-limit is
1 *				unchanged)
1	GCLDPLM2	DS	CL10	last date in calendar (*UNCH =
1 *				calendar-limit is unchanged)
1	GCLDPTIM	DS	CL8	time (*UNCH = time is
1 *				unchanged, *STD = Standard
1 *				time)
1	GCLDPWT1	DS	CL5	begin of working time (*UNCH =
1 *				working time is unchanged,
1 *				*STD = Standard working time)
1	GCLDPWT2	DS	CL5	end of working time (*UNCH =
1 *				working time is unchanged,
1 *				*STD = Standard working time)
1	GCLD#	EQU	*-GCLDHDR	
		END		

## 4.8 Makro-Syntax

### 4.8.1 Makroaufrufformat

Das Makroaufrufformat ist aus zwei Spalten aufgebaut; die erste Spalte enthält den Makronamen, die zweite Spalte die möglichen Operanden.

Makroname	Operanden
<makroname>	<operand <sub>1</sub> > ,<operand <sub>2</sub> >

Beim Aufruf des Makros muss der Makroname mit mindestens einer Leerspalte vom ersten Operanden getrennt sein. Mehrere Operanden müssen durch Kommata getrennt angegeben werden.

Im Makroaufrufformat werden bestimmte Zeichen (Metazeichen) verwendet, die in den folgenden Tabellen erläutert werden.

## 4.8.2 Elemente der Metasyntax

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter oder Konstanten, die in dieser Form vom Benutzer angegeben werden müssen. Schlüsselwörter müssen mit * beginnen, falls alternativ sowohl Schlüsselwörter als auch Namen von Konstanten oder Variablen angegeben werden können.	DIB FORCED=*YES
Kleinbuchstaben	Kleinbuchstaben bezeichnen Datentypen der Werte oder Variablen, die vom Benutzer angegeben werden können.	DIB = <var: pointer>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch die Datentypen beschrieben wird.	<var: pointer>
<u>Unterstreich</u>	Der Unterstrich kennzeichnet den Default-Wert eines Operanden. Hat ein Operand keinen Default-Wert, so ist die Angabe eines Operanden Pflicht.	FORCED = * <u>NO</u> / *YES
=	Das Gleichheitszeichen verbindet den Operandennamen mit den dazugehörigen Operandenwerten.	DATA = <var: pointer>
/	Der Schrägstrich trennt alternative Operandenwerte.	FORCED = * <u>NO</u> / *YES
list-poss(n)	Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. n gibt die maximale Anzahl der Listenelemente an. Enthält die Liste mehr als ein Element, muss sie in runden Klammern eingeschlossen werden.	FLAG=list-poss(3): *SLI / *SKIP / *DC  Angabe: FLAG=*SKIP FLAG=( *SLI, *DC)

Einem Operanden wird durch ein Gleichheitszeichen ein Operandenwert zugewiesen, welcher aus einem definierten Wertevorrat stammt.

Dieser Wertevorrat wird durch einen Datentyp bestimmt. Tabelle 3 enthält die Datentypen der Operandenwerte.

### 4.8.3 Datentypen der Operandenwerte

Datentyp	Zeichenvorrat	Anmerkungen
c-string	EBCDIC-Zeichen	ist in Hochkommata einzuschließen
integer	[+-] 0..2147483647	ist eine dezimale Zahl
var:	leitet eine variable Angabe ein. Nach dem Doppelpunkt folgt der Typ der Variablen (siehe Tabelle „Datentypen der Variablen“ )	<var:var-type>
reg:	Register 0..15	Angabe: (<reg:var-type>)

### 4.8.4 Zusätze zu Datentypen

Zusatz	Bedeutung
n..m	für Datentyp integer bedeutet n..m eine Intervallangabe; n: Mindestwert m: Maximalwert
	für Datentyp c-string bedeutet n..m eine Längenangabe in Bytes; n: Mindestlänge m: Maximallänge mit $n < m$
n	bei Datentyp c-string bedeutet n eine Längenangabe in Bytes; n muss exakt eingehalten werden.

Die Operandenwerte können direkt als Zeichenkette oder Integer-Zahl (siehe Datentypen "c-string" und "integer" ) eingegeben werden oder indirekt über eine Variable (siehe Datentyp "var:" ) bezeichnet werden. Tabelle 5 enthält die Datentypen, die für Variablen möglich sind.

### 4.8.5 Datentypen der Variablen

Datentyp	Beschreibung	Definition im Programm
char:n	Die Variable ist eine Zeichenkette von n Zeichen. Fehlt die Längenangabe, wird n=1 angenommen.	CLn
int:n	Die Variable ist eine Integer-Zahl, die n Bytes belegt. Fehlt die Längenangabe, wird n=1 angenommen. Bedingung: $n \leq 4$	FLn
enum-of E:n	Die Variable ist die Aufzählung E, die n Bytes belegt. Fehlt die Längenangabe, wird n=1 angenommen. ( $n \leq 4$ )	XLn
pointer	Die Variable ist eine Adresse oder ein Adresswert.	A

### 4.8.6 MF-Operand

Mit dem Operanden MF wird die Art der Makrogenerierung bestimmt.

In Abhängigkeit von den Operandenwerten für MF unterscheidet man fünf Formen des Makroaufrufs:

Makroname	Operanden
<makroname>	MF=L [,operand <sub>1</sub> ,...,operand <sub>n</sub> ]
	MF=M ,operand <sub>1</sub> ,...,operand <sub>n</sub> [,PREFIX=p][,MACID=mac]
	MF=D [,PREFIX=p]
	MF=C [,PREFIX=p][,MACID=mac]
	MF=E [,PARAM=<adr> / (<reg>)]

MF=C [,PREFIX=p][,MACID=mac]

Es wird nur der Datenbereich generiert. Operandenwerte werden nicht ausgewertet. Jedes Feld hat einen Feldnamen und erläuternde Equates, falls erforderlich. Der Datenbereich wird durch ein Längenequate beendet. Der Standard-Header (siehe Seite 86) muss i.d.R. vom Anwender initialisiert werden.

Mit dem Operanden PREFIX kann der Anwender das erste Zeichen der Feldnamen und Equates bestimmen. p = 1 Buchstabe.

Mit dem Operanden MACID kann der Anwender das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates bestimmen. mac = 1..3 Zeichen.

Weitere Operanden werden in der C-Form nicht ausgewertet; ausgenommen VERSION = 1 und XPAND.

MF=D [,PREFIX=p]

Es wird eine DSECT generiert. Jedes Feld hat einen Feldnamen und erläuternde Equates, falls erforderlich. Die DSECT wird durch ein Längenequate beendet. Es wird nicht auf den anfänglichen Adresspegel umgeschaltet.

Die **DSECT** beschreibt die Struktur eines Speicherbereiches, ohne selbst Speicherplatz zu belegen. Der bei DSECT angegebene symbolische Name wird in einen ESD-Satz eingetragen. Der Adresspegel wird auf Null gesetzt.

Mit dem Präfix p kann der Anwender das erste Zeichen der Feldnamen und Equates bestimmen. p = 1 Buchstabe.

XPAND wird ausgewertet und legt fest, welche DSECT erzeugt wird.

MF=E [,PARAM=<adr> / (<reg>)]

Es werden nur die zum Aufruf des Funktionsbausteins notwendigen Befehle generiert. Der Befehlssteil endet bei CALLER=USER mit einem SVC. Im Makroaufruf muss die Adresse des Datenbereichs mit den Operandenwerten angegeben sein.

Der Operand PARAM bezeichnet die Adresse des Datenbereichs. adr = Adresse des Datenbereichs.

reg = Register, das die Adresse des Datenbereichs enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

Wenn PARAM nicht angegeben wird, muss vor einem Makroaufruf Register 1 mit der Adresse der Parameterliste versorgt werden.

Weitere Operanden werden in der E-Form nicht ausgewertet.

MF=L [,operand 1,...,operand n] [PREFIX=p]

Es wird nur der Datenbereich generiert, unter Beachtung der im Makroaufruf angegebenen Operandenwerte. Der Datenbereich enthält keine Feldnamen und keine erläuternden Equates. Der Standard-Header (siehe Seite 86) ist initialisiert. Der Makroaufruf wird im Definitionsteil des Programms abgesetzt.

Bei Shared-Code-Programmierung darf dieser Aufruf nicht im invarianten Programmteil liegen, wenn er variable Daten enthält. Der Datenbereich wird im invarianten Programmteil mit konstanten Werten initialisiert, vor dem E-Form-Aufruf in einen ablauflokalen Datenbereich kopiert und dort ggf. modifiziert. Die Modifizierung wird z.B. mit der M-Form realisiert, falls sie für die betreffende Schnittstelle angeboten wird.

MF=M ,operand 1,...,operand n [,PREFIX=p][,MACID=mac]

Es werden Befehle (z.B. MVCs) generiert, die während des Programmlaufs in einem mit MF=L bereits initialisierten Datenbereich bzw. bei Shared-Code-Programmierung in einer ablauflokalen Kopie des mit MF=L initialisierten Datenbereichs Felder mit den Operandenwerten überschreiben, die im Makroaufruf angegeben werden. Damit bietet die M-Form eine komfortable Möglichkeit, die Operandenwerte, mit denen ein Makro aufgerufen wird, **dynamisch** dem Programmlauf anzupassen.

Bei MF=M werden für Funktionsoperanden keine Standardwerte angenommen, d.h., alle Operanden müssen explizit angegeben werden.

Da die bei MF=M generierten Befehle die symbolischen Adressen und Equates der C-Form oder der D-Form benutzen, ist bei der Verwendung der M-Form sicherzustellen, dass diese Namen für die Adressierung des zu modifizierenden Datenbereichs zur Verfügung stehen. Insbesondere ist darauf zu achten, dass bei einem Makroaufruf mit MF=M ggf. die Operanden PREFIX und MACID mit den gleichen Werten angegeben werden wie im zugehörigen MF=C- bzw. MF=D-Aufruf.

Mit dem Operanden PREFIX kann der Anwender das erste Zeichen der Feldnamen und Equates bestimmen. p = 1 Buchstabe.

Mit dem Operanden MACID kann der Anwender das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates bestimmen. mac = 1..3 Zeichen.

## 4.9 C-Headerfile (calendr.h)

Beschreibungen in doppelten spitzen Klammern (<<...>>) sind eingefügte Kommentare bzw. Erklärungen

```

/* main return codes */
/* _mret_code */
#define CALENDRsuccessful 0 /* no error */
#define CALENDRinternal_error 1 /* parameter error */
#define CALENDRcalendar_existing 2 /* calendar already existing */
#define CALENDRcalendar_not_existing 3 /* calendar does not exist */
#define CALENDRDMS_error_calendar 4 /* DMS-error on calendar-file */
#define CALENDRno_calendar 5 /* file is not a calendar */
#define CALENDRcalendar_inconsistent 6 /* calendar is inconsistent */
#define CALENDRno_update_possible 7 /* update of calendar not
/* possible at the moment */
#define CALENDRholiday_file_not_existing 8
/* holiday-file does not
/* exist
#define CALENDRDMS_error_holiday_file 9
/* DMS-error on holiday-file */
#define CALENDRtoo_many_days 10 /* too many days between
/* calendar limits
#define CALENDREnddate_less_startdate 11
/* end-date less than
/* start-date
#define CALENDRno_memory 12 /* no more memory available */
#define CALENDRcontrol_block_invalid 13
/* control-block invalid */
#define CALENDRdate_outside 14 /* date outside of calendar
/* limits
#define CALENDRSYMDAT_existing 15 /* SYMDAT already existing */
#define CALENDRSYMDAT_not_existing 16 /* SYMDAT does not exist */
#define CALENDRcyclic_SYMDAT 17 /* SYMDAT is a cyclic SYMDAT */
#define CALENDRnon_cyclic_SYMDAT 18 /* SYMDAT is a non-cyclic
/* SYMDAT
#define CALENDRassignment_existing 19 /* assignment to date
/* already existing
#define CALENDRassignment_not_existing 20
/* assignment to date does
/* not exist
#define CALENDRdate_invalid 21 /* invalid date */
#define CALENDRholiday_existing 22 /* holiday already existing */
#define CALENDRholiday_not_existing 23 /* holiday does not exist */
#define CALENDRcyclic_holiday 24 /* holiday is a cyclic
/* holiday

```



```
#define CALENDRnon_cyclic_holiday 25 /* holiday is a non-cyclic */
/* holiday */
#define CALENDRholiday_activated 26 /* holiday already activated */
#define CALENDRholiday_not_activated 27
/* holiday is not activated */
#define CALENDRno_information 28 /* output-area to small: no */
/* information available */
#define CALENDRsome_information 29 /* output-area to small: some */
/* information available */
#define CALENDRwrong_SYMDAT_area 30 /* definition of SYMDAT-area */
/* invalid */
#define CALENDRwrong_holiday_area 31 /* definition of holiday-area */
/* invalid */
#define CALENDRstandardweek_incorrect 32
/* area for standardweek is */
/* incorrect */
#define CALENDRholiday_file_incorrect 33
/* entry of holiday-file is */
/* incorrect */
#define CALENDRinvalid_limits 34 /* LIMIT1 < 1970 or LIMIT2 > */
/* 2030 or new LIMIT1 > old */
/* LIMIT2 */
#define CALENDRstart_date_outside 35 /* start-date of cyclic */
/* SYMDAT outside of */
/* calendar-limits (warning) */
#define CALENDRfunction_not_allowed 36 /* function not allowed with */
/* OPENMOD = *READ or */
/* *READALL */
#define CALENDRSYMDAT_start_date_deleted 37
/* start date for cyclic */
/* SYMDAT with */
/* CYCLTYP=WORKDAY has been */
/* deleted as it is outside */
/* calendar limits */
#define CALENDRprogram_error 38 /* internal error in program */
#define CALENDRcalendar_version_not_supported 39
/* version of calendar ist */
/* not supported by version */
/* of interface */
#define CALENDRtoo_many_symdat 40 /* more than 4096 SYMDAT have */
/* been created */
#define CALENDRtoo_many_holiday 41 /* more than 1024 holidays */
/* have been created */
#define CALENDRtoo_many_assigned_days 42
/* more than 1860 days */
/* assigned to a holiday */
#define CALENDRlinkage_error_function 65535
/* function not supported */
```

```
#define CALENDRIlinkage_error_version 65535
/* version not supported */
#define CALENDRIlinkage_error_alignment 65535
/* alignment_error */

/* function codes */
/* ENUM _funct_s */
#define CALENDRCreacal 1 /* create calendar */
#define CALENDROpenca 2 /* open calendar */
#define CALENDRClseca 3 /* close calendar */
#define CALENDRSaveca 4 /* save calendar */
#define CALENDRmodbas 5 /* modify basis information */
#define CALENDRmodday 6 /* modify day */
#define CALENDRCrency 7 /* create non cyclic symdat */
#define CALENDRCrency 8 /* create cyclic symdat */
#define CALENDRdelncy 9 /* delete symdat */
#define CALENDRmodnncy 10 /* modify non cyclic symdat */
#define CALENDRmodncy 11 /* modify cyclic symdat */
#define CALENDRCrenhol 12 /* create non cyclic holiday */
#define CALENDRCrecho 13 /* create cyclic holiday */
#define CALENDRdelhol 14 /* delete holiday */
#define CALENDRmodnhol 15 /* modify non cyclic holiday */
#define CALENDRmodcho 16 /* modify cyclic holiday */
#define CALENDRAkthol 17 /* activate holiday */
#define CALENDRdeakhol 18 /* deactivate holiday */
#define CALENDRshbas 19 /* show basis information */
#define CALENDRshloday 20 /* show list of days */
#define CALENDRshdayin 21 /* show day-information */
#define CALENDRshdayhl 22 /* show day with holidays */
#define CALENDRshlosym 23 /* show list of symdat */
#define CALENDRshsymin 24 /* show symdat-information */
#define CALENDRshlohol 25 /* show list of holidays */
#define CALENDRshholin 26 /* show holiday-information */
#define CALENDRshnestm 27 /* show next symdat time */

/* open modus */
/* ENUM _open_s */
#define CALENDRread 1 /* read */
#define CALENDRreadall 2 /* read complete calendar */
#define CALENDRupdate 3 /* update */

/* attribute of day */
/* ENUM _attribute_s */
#define CALENDRunchanged 0 /* unchanged */
#define CALENDRfree_day 1 /* free day */
```

```

#define CALENDRwork_day 2          /* work day          */
#define CALENDRstd_week 3         /* standard wekk     */

/* alternative */
/* ENUM _alternative_s */
#define CALENDRunchanged 0        /* unchanged         */
#define CALENDRbefore 1          /* insert before free day */
#define CALENDRafter 2           /* insert after free day */
#define CALENDRskip 3            /* dont insert       */
#define CALENDRon 4              /* insert on free day */

/* kind of cycle */
/* ENUM _cycle_s */
#define CALENDRunchanged 0        /* unchanged         */
#define CALENDRday 1             /* cycle = day       */
#define CALENDRworkday 2         /* cycle = work-day  */
#define CALENDRweek 3            /* cycle = week      */
#define CALENDRmonth 4           /* cycle = month     */

/* save */
/* ENUM _save_s */
#define CALENDRyes 1              /* save calendar    */
#define CALENDRno 2              /* dont save calendar */

/* sort */
/* ENUM _sort_s */
#define CALENDRsymdat 1           /* sort by symdat   */
#define CALENDRtime 2            /* sort by time     */

/* parameterarea description */
struct CALENDR_md1 {
    /* Standardheader */
    struct ESMFHDR hdr;
    unsigned char function_code; /* funct =          */
    unsigned char open_modus;   /* openmod =       */
    unsigned char attribute;    /* attribute =     */
    unsigned char cycle_typ;    /* cycltyp =      */
    unsigned char alternative;   /* alternative =   */
    unsigned char save;         /* save =         */
    unsigned char sort;         /* sort =         */

    /* indicator */
    struct /* flag_1 */ {
        char reserved_1: 8; /* reserved */
    } flag_1;
};

```

```

/* indicator */
struct /* flag_2 */ {
    char reserved_2: 8; /* reserved */
} flag_2;

char filler1[3]; /* filler */
long number; /* number of days or SYMDAT */
unsigned long cycle_value; /* number of days between two
/* entries */
unsigned long DMS_returncode; /* DMS returncode */
void* control_block; /* address of control-block */
void* week_address; /* address of standard week */
void* days_to_add; /* address of days to add */
unsigned long days_to_add_num; /* number of days in area */
void* days_to_del; /* address of days to delete */
unsigned long days_to_del_num; /* number of days in area */
void* output_area; /* address of output-area */
unsigned long output_area_len; /* length of output-area */
char calendar_name[54]; /* calendar-name */
char name[30]; /* name of SYMDAT or holiday */
char name1[30]; /* first name (*FIRST = first
/* SYMDAT or holiday in
/* calendar)
char name2[30]; /* last name (*LAST = last
/* SYMDAT or holiday in
/* calendar)
char date[10]; /* date
char date1[10]; /* start-date (*UNCH =
/* start-date of SYMDAT is
/* unchanged, *FIRST = first
/* date in calendar)
char date2[10]; /* end-date (*UNCH = end-date
/* of SYMDAT is unchanged,
/* *UNDEF = end-date of
/* SYMDAT is undefined, *LAST
/* = last date in calendar)
char calendar_limit1[10]; /* first date in calendar
/* (*UNCH = calendar-limit is
/* unchanged)
char calendar_limit2[10]; /* last date in calendar
/* (*UNCH = calendar-limit is
/* unchanged)
char time[8]; /* time (*UNCH = time is
/* unchanged, *STD = Standard
/* time)
char work_time1[5]; /* begin of working time
/* (*UNCH = working time is

```

```

/* unchanged, *STD = Standard */
/* working time) */
char work_time2[5]; /* end of working time (*UNCH */
/* = working time is */
/* unchanged, *STD = Standard */
/* working time) */

};

/* Benutzerschnittstelle */
#ifdef __SNI_HOST_BS2000
void _SVC(int, void*);
#define GCLDSTU(p) _SVC(193, &p)
#endif

/* Systemschnittstelle */
#ifdef __SNI_HOST_BS2000
#pragma ISL GCLDSTP
void GCLDSTP(struct CALENDR_md1*);
#else
void GCLDSTP(struct CALENDR_md1*);
#endif

/* Standard-week-entry */
struct CALENDR_stdw_entry_md1 {
    << Eintrag der Standard-Arbeitswoche; enthält die Namen der Wochentage, das >>
    << Attribut für jeden Wochentag, den Beginn und das Ende der Standard-Arbeitszeit >>
    char swe_day[3]; /* week-day (MON - SUN) */
    char swe_attribute[1]; /* attribute of day */
    /* (W=workday, F=free day) */
    char swe_time1[5]; /* begin of working time */
    char swe_time2[5]; /* end of working time */
};

/* Date-entry */
struct CALENDR_date_entry_md1 {
    << Datums-Eintrag; enthält eine vierstellige Angabe für das Jahr, je eine zweistellige >>
    << für den Monat und den Tag >>
    char dte_year[4]; /* year */
    char dte_hyphen1[1]; /* hyphen */
    char dte_month[2]; /* month */
    char dte_hyphen2[1]; /* hyphen */
    char dte_day[2]; /* day */
};

```

```

/* Day-entry */
struct CALENDR_day_entry_md1 {

```

<< Kalendertags-Eintrag; enthält das Datum, den Wochentagsnamen, das Attribut, >>  
 << das Attribut für diesen Wochentag in der Standard-Arbeitswoche, die Feiertags- >>  
 << definition, Beginn und Ende der Arbeitszeit, die Anzahl der zugeordneten Feier- >>  
 << tage und SYMDAT >>

```

    char dye_date[10];           /* date */
    char dye_day[3];            /* week-day (MON - SUN) */
    char dye_attribute[1];      /* attribute (W=workday,
                                /* F=free day) */
    char dye_attr_def[1];       /* attribute defined for day
                                /* (W=workday, F=free day,
                                /* S=standard) */
    char dye_attr_std[1];       /* attribute of day in
                                /* standard-week (W=workday,
                                /* F=free day) */
    char dye_holiday[1];       /* holiday (Y=yes, N=no) */
    char dye_time1[5];          /* begin of working time */
    char dye_time2[5];          /* end of working time */
    char dye_filler[1];         /* filler */
    unsigned short dye_assigned_holidays;
                                /* number of assigned
                                /* holidays */
    unsigned short dye_assigned_SYMDAT;
                                /* number of assigned SYMDAT */
};

```

```

/* SYMDAT-entry */
struct CALENDR_SYMDAT_entry_md1 {

```

<< SYMDAT-Eintrag; enthält den Namen, den Typ, die Zeit, Zyklus-Art und Zyklus- >>  
 << wert, die Alternative für freie Tage, Start- und Ende-Datum für zyklische >>  
 << SYMDAT sowie die Anzahl der zugeordneten Kalendertage >>

```

    char sye_SYMDAT_name[20];   /* SYMDAT-name */
    char sye_SYMDAT_type[1];    /* SYMDAT-type (C=cyclic,
                                /* N=non cyclic) */
    char sye_SYMDAT_time[8];    /* time */
    char sye_cycle_type[1];     /* kind of cycle (D=day,
                                /* W=week, O=workday,
                                /* M=month) */
    unsigned short sye_cycle_value; /* cycle-value */
    char sye_alternative[1];    /* alternative for free days
                                /* (B=before, A=after,
                                /* S=skip, O=on) */
    char sye_filler1[1];        /* filler */

```

```

    char sye_date1[10];           /* start-date for cyclic      */
                                   /* SYMDAT                      */
    char sye_date2[10];           /* end-date for cyclic SYMDAT */
    unsigned short sye_assigned_days;
                                   /* number of assigned days    */
};

/* SYMDAT_short-entry */
struct CALENDR_SYMDAT_short_entry_md1 {

<< SYMDAT-Kurzeintrag; enthält den Namen und die zugeordnete Zeit >>

    char sse_SYMDAT_name[20];     /* SYMDAT-name                */
    char sse_SYMDAT_time[8];      /* time                        */
};

/* holiday-entry */
struct CALENDR_holiday_entry_md1 {

<< Feiertags-Eintrag; enthält den Namen, den Feiertagstyp, den Aktivierungszustand >>
<< und die Anzahl der zugeordneten Tage >>

    char hle_holiday_name[30];    /* holiday-name                */
    char hle_holiday_type[1];    /* holiday-type (C=cyclic,    */
                                   /* N=non cyclic)              */
    char hle_assigned[1];        /* holiday is assigned        */
                                   /* (Y=yes, N=no)              */
    unsigned short hle_assigned_days;
                                   /* number of assigned days    */
};

/* Standard-week */
struct CALENDR_stdweek_md1 {

<< Standard-Arbeitswoche (Kurzeintrag); enthält den Bereich der Standard- >>
<< Arbeitswoche >>

    char stdweek_area[7][14];     /* arae for standard-week    */
                                   /* entry is described with    */
                                   /* XPAND=ENTSTDW              */
};

/* control-block */
struct CALENDR_ccb_md1 {

<< Control-Block; enthält die Adresse eines internen Datenbereichs, der beim >>
<< Erzeugen eines Kalenders angelegt wurde >>

    char cb_text[4];             /* text                        */
    void* cb_address;           /* address                    */
    unsigned long cb_check_pattern; /* check pattern              */
};

```

```

};

/* day-list */
struct CALENDAR_daylist_md1 {
<< Liste von Kalendertagen (Kurzeintrag); enthält den Bereich der Kalendertags-Liste >>
    char pdy_day_area[1860][10]; /* area for days entry is */
    /* described with */
    /* XPAND=ENTDATE */
};

/* Basis-Information */
struct CALENDAR_out_basinf_md1 {
<< Eintrag für Basis-Informationen; enthält die Schnittstellenversion, die Anzahl der >>
<< Kalendertage, der SYMDAT und Feiertage im Kalender, den Kalendernamen, den >>
<< ersten und letzten Datumseintrag im Kalender, das erste und letzte SYMDAT >>
<< im Kalender, den ersten und letzten Feiertag im Kalender und den Bereich der >>
<< Standard-Arbeitswoche >>
    unsigned short bi_version; /* version of interface */
    unsigned short bi_number_day; /* number of days in calendar */
    unsigned short bi_number_SYMDAT; /* number of SYMDAT in */
    /* calendar */
    unsigned short bi_number_holiday; /* number of holidays in */
    /* calendar */
    char bi_calendar_name[54]; /* calendar name */
    char bi_limit1[10]; /* first date in calendar */
    char bi_limit2[10]; /* last date in calendar */
    char bi_first_SYMDAT[20]; /* first SYMDAT in calendar */
    char bi_last_SYMDAT[20]; /* last SYMDAT in calendar */
    char bi_first_holiday[30]; /* first holiday in calendar */
    char bi_last_holiday[30]; /* last holiday in calendar */
    char bi_stdweek_area[7][14]; /* area for standard-week */
    /* entry is described with */
    /* XPAND=ENTSTDW */
};

/* List-of-days */
struct CALENDAR_out_loday_md1 {
<< Eintrag für eine Liste von Kalendertagen; enthält die Schnittstellenversion, die >>
<< Anzahl der Kalendertage insgesamt, der ausgewählten und der ausgegebenen >>
<< Tage sowie den Bereich der Kalendertags-Liste >>

```



```

    unsigned short lod_version;      /* version of interface      */
    unsigned short lod_available_days;
                                    /* number of days in calendar */
    unsigned short lod_selected_days;
                                    /* number of selected days   */
    unsigned short lod_given_days;   /* number of given days      */
    char lod_day_area[1860][32];     /* area for days entry is    */
                                    /* described with             */
                                    /* XPAND=ENTDAY              */
};

/* day_information */
struct CALENDR_out_dayinf_md1 {
    << Eintrag für Kalendertag mit zugeordneten SYMDAT; enthält die Schnittstellen- >>
    << version, die Anzahl der zugeordneten, der ausgewählten und der ausgegebenen >>
    << SYMDAT, das Datum, den Wochentagsnamen, das Attribut für diesen Tag sowie >>
    << den entsprechenden Wochentag in der Standard-Arbeitswoche, Feiertagsdefinition >>
    << Beginn und Ende der Arbeitszeit sowie den Bereich für SYMDAT-Einträge >>
    unsigned short di_version;      /* version of interface      */
    unsigned short di_assigned_SYMDAT;
                                    /* number of assigned SYMDAT */
    unsigned short di_selected_SYMDAT;
                                    /* number of selected SYMDAT */
    unsigned short di_given_SYMDAT; /* number of given SYMDAT   */
    char di_date[10];               /* date                      */
    char di_day[3];                 /* week-day (MON - SUN)     */
    char di_attribute[1];           /* attribute (W=workday,    */
                                    /* F=free day)              */
    char di_attr_def[1];            /* attribute defined for day */
                                    /* (W=workday, F=free day,  */
                                    /* S=standard)              */
    char di_attr_std[1];            /* attribute of day in      */
                                    /* standard-week (W=workday, */
                                    /* F=free day)              */
    char di_holiday[1];            /* holiday (Y=yes, N=no)    */
    char di_time1[5];               /* begin of working time    */
    char di_time2[5];               /* end of working time      */
    char di_filler1[1];             /* filler                    */
    char di_SYMDAT_area[4096][28];  /* area for SYMDAT entry is */
                                    /* described with            */
                                    /* XPAND=ENTSYMS           */
};

/* day_holiday */
struct CALENDR_out_dayhol_md1 {

```

<< Eintrag für Kalendertag mit zugeordneten Feiertagen; enthält die Schnittstellen- >>  
 << version, die Anzahl der zugeordneten Feiertage, die Anzahl der ausgegebenen >>  
 << Feiertage, das Datum sowie den Bereich für den Eintrag eines Feiertags >>

```

    unsigned short dh_version;          /* version of interface      */
    unsigned short dh_assigned_holiday; /* number of assigned        */
                                        /* holidays                  */
    unsigned short dh_given_holiday;    /* number of given holidays  */

    char dh_date[10];                   /* date                      */
    char dh_holiday_area[1024][34];     /* area for holidays entry is */
                                        /* described with            */
                                        /* XPAND=ENTHOL             */
};

```

```

/* List-of-SYMDAT                                     */
struct CALENDAR_out_losym_md1 {

```

<< Eintrag für SYMDAT-Liste; enthält die Schnittstellenversion, die Anzahl aller, der >>  
 << ausgewählten und der ausgegebenen SYMDAT sowie den Bereich der >>  
 << SYMDAT-Liste >>

```

    unsigned short los_version;          /* version of interface      */
    unsigned short los_available_SYMDAT; /* number of SYMDAT in      */
                                        /* calendar                  */
    unsigned short los_selected_SYMDAT;  /* number of selected SYMDAT */
    unsigned short los_given_SYMDAT;     /* number of given SYMDAT   */

    char los_SYMDAT_AREA[4096][56];     /* area for SYMDAT entry is */
                                        /* described with            */
                                        /* XPAND=ENTSYM            */
};

```

```

/* SYMDAT_information                                 */
struct CALENDAR_out_syminf_md1 {

```

<< Eintrag für SYMDAT mit zugeordneten Kalendertagen; enthält die Schnittstellen- >>  
 << version, die Anzahl der zugeordneten, der ausgewählten und der ausgegebenen >>  
 << Kalendertage, den SYMDAT-Namen, den Typ, die zugeordnete Zeit, Zyklus-Art >>  
 << und Zyklus-Wert, die Alternative für freie Tage, Start- und Ende-Datum für ein >>  
 << zyklisches SYMDAT sowie den Bereich für den Eintrag der zugeordneten Kalen- >>  
 << dertage >>

```

unsigned short si_version;      /* version of interface */
unsigned short si_assigned_days; /* number of assigned days */
unsigned short si_selected_days; /* number of selected days */
unsigned short si_given_days;   /* number of given days */
char si_SYMDAT_name[20];       /* SYMDAT-name */
char si_SYMDAT_type[1];        /* SYMDAT-type (C=cyclic,
/* N=non cyclic) */
char si_SYMDAT_time[8];        /* time */
char si_cycle_type[1];         /* kind of cycle (D=day,
/* W=week, O=workday,
/* M=month) */
unsigned short si_cycle_value;  /* cycle-value */
char si_alternative[1];        /* alternative for free days
/* (B=before, A=after,
/* S=skip, O=on) */
char si_date1[10];             /* start-date for cyclic
/* SYMDAT */
char si_date2[10];             /* end-date for cyclic SYMDAT */
char si_filler1[1];           /* filler1 */
char si_day_area[1860][10];     /* area for assigned days
/* entry is described with
/* XPAND=ENTDATE */
};

```

```

/* List-of-holidays */
struct CALENDAR_out_lohol_md1 {

```

<< Eintrag für Feiertags-Liste; enthält die Schnittstellenversion, die Anzahl aller, der >>  
 << ausgewählten und der ausgegebenen Feiertage sowie den Bereich der >>  
 << Feiertags-Liste >>

```

unsigned short loh_version;     /* version of interface */
unsigned short loh_available_holiday; /* number of holidays in
/* calendar */
unsigned short loh_selected_holiday; /* number of selected
/* holidays */
unsigned short loh_given_holiday; /* number of given holidays */
char loh_holiday_area[1024][34]; /* area for holidays entry is
/* described with
/* XPAND=ENTHOL */
};

```

```

/* holiday-information */
struct CALENDR_out_holinf_md1 {

<< Eintrag für Feiertag mit zugeordneten Kalendertagen; enthält die Schnittstellen- >>
<< version, die Anzahl der zugeordneten, der ausgewählten und der ausgegebenen >>
<< Kalendertage, den Feiertagsnamen, den Typ, den Aktivierungszustand sowie den >>
<< Bereich für den Eintrag der zugeordneten Kalendertage >>

    unsigned short hi_version;          /* version of interface */
    unsigned short hi_assigned_days;    /* number of assigned days */
    unsigned short hi_selected_days;    /* number of selected days */
    unsigned short hi_given_days;       /* number of given days */

    char hi_holiday_name[30];          /* holiday-name */
    char hi_holiday_type[1];           /* holiday-type (C=cyclic,
    /* N=non cyclic) */
    char hi_assigned[1];                /* holiday is assigned
    /* (Y=yes, N=no) */
    char hi_day_area[1860][10];         /* area for assigned days
    /* entry is described with
    /* XPAND=ENTDATE */
};

/* Next-SYMDAT-time */
struct CALENDR_out_nestm_md1 {

<<Eintrag für das Datum und die Uhrzeit, die in der betrachteten >>
<<Menge von SYMDATs am nächsten nach einem angegebenen Zeitpunkt >>
<<folgen, sowie für den Namen des zugehörigen SYMDAT. >>

    unsigned short nst_version;         /* interface version */
    char nst_date[10];                  /* next date */
    char nst_time[8];                   /* next time */
    char nst_name[20];                  /* SYMDAT name */
};

#endif /* _CALENDR_H */

```





---

## 5 Kommandoschnittstelle

### SHOW-CALENDAR

#### Informationen aus einer Kalenderdatei anfordern

**Anwendungsbereich:** UTILITIES

**Privilegierung:** STD-PROCESSING

#### Funktionsbeschreibung

Das Kommando SHOW-CALENDAR informiert über die Kalenderdaten aus der angegebenen Kalenderdatei.

Standardmäßig (SELECT=\*TODAY) werden folgende Informationen über den aktuellen Tag ausgegeben: der Name der Kalenderdatei, das aktuelle Datum, die Bezeichnung des Wochentags (Montag...Sonntag), das Attribut des aktuellen Tags (freier Tag / Arbeitstag), die Anzahl der zugehörigen SYMDAT (symbolische Datumsangaben), die Arbeitszeit, gegebenenfalls der Feiertagsname, die Namen der zugehörigen SYMDAT mit zugeordneten Zeitangaben.

Mit SELECT=\*BASIC-INFORMATION werden die Basis-Informationen der Kalenderdatei zur Ausgabe angefordert: Name des Kalenders, Kalendergrenzen und Definitionen der Standard-Arbeitswoche. In der Standard-Arbeitswoche sind das Attribut und die Arbeitszeit für die Wochentage festgelegt.

Mit SELECT=\*DATE werden die Tagesinformationen über einen oder mehrere Tage angefordert: das Datum, die Wochentagsbezeichnung, das Attribut, die Anzahl der zugeordneten SYMDAT, die Arbeitszeit, gegebenenfalls der Feiertagsname, über spezielle Angabe auch die Namen der zugehörigen SYMDAT mit zugeordneten Zeitangaben.

Mit SELECT=\*SYMBOLIC-DATE werden Informationen über SYMDAT zur Ausgabe angefordert. SYMDAT sind symbolische Datumsangaben, unter deren Namen mehrere Kalendertage zusammengefasst werden (vgl. Beschreibung auf Seite 14ff).

Mit SELECT=\*HOLIDAY werden Informationen über Feiertage zur Ausgabe angefordert. Die Ausgabe kann nach SYSOUT oder SYSLST gelenkt werden. Das Kommando unterstützt die strukturierte Ausgabe in S-Variable, eine Beschreibung dazu finden Sie im Handbuch „Kommandos Band 4, Ausgabe in S-Variablen“ [1])

## Format

## SHOW-CALENDAR

**CALENDAR-NAME** = <full-filename 1..54>

,**SELECT** = \*TODAY / \***BASIC-INFORMATION** / \***DATE(...)** / \***SYMBOLIC-DATE(...)** / \***HOLIDAY(...)**

\***DATE(...)**

**FROM** = \***TODAY** / \***FIRST-CALENDAR-DATE** / <date 8..10>

**TO** = \***SAME** / \***TODAY** / \***LAST-CALENDAR-DATE** / <date 8..10> / \***BY-NUMBER-OF-DAYS(...)**

        \***BY-NUMBER-OF-DAYS(...)**

**NUMBER-OF-DAYS** = <integer 1..1827>

**ASSIGNED-SYM-DATE** = \***NONE** / \***ALL(...)** /

        <full-filename 1..20 without-cat-user-gen-vers with-wild>(...)

        \***ALL(...)**

**ORDER-WITHIN-DAY** = \***BY-TIME** / \***BY-SYMBOLIC-DATE**

        <full-filename 1..20 without-cat-user-gen-vers with-wild>(...)

**ORDER-WITHIN-DAY** = \***BY-TIME** / \***BY-SYMBOLIC-DATE**

\***SYMBOLIC-DATE(...)**

**FROM** = \***FIRST-SYMBOLIC-DATE** / <full-filename 1..20 without-cat-user-gen-vers with-wild>

**TO** = \***SAME** / \***LAST-SYMBOLIC-DATE** /

        <full-filename 1..20 without-cat-user-gen-vers with-wild> /

        \***BY-NUMBER-OF-SYMBOLIC-DATES(...)**

        \***BY-NUMBER-OF-SYMBOLIC-DATES(...)**

**NUMBER-OF-SYM-DATES** = <integer 1..4096>

**ASSIGNED-DATES** = \***NO** / \***ALL** / \***NEXT-DATE** / \***INTERVAL(...)**

        \***INTERVAL(...)**

**FROM** = \***TODAY** / \***FIRST-ASSIGNED-DATE** / <date 8..10>

**TO** = \***SAME** / \***TODAY** / \***LAST-ASSIGNED-DATE** / <date 8..10> /

                \***BY-NUMBER-OF-DAYS(...)**

                \***BY-NUMBER-OF-DAYS(...)**

**NUMBER-OF-DAYS** = <integer 1..1827>

Fortsetzung ➡



```

*HOLIDAY(...)
  |
  | FROM = *FIRST-HOLIDAY / <full-filename 1..30 without-cat-user-gen-vers with-wild>
  |
  | ,TO = *SAME / <full-filename 1..30 without-cat-user-gen-vers with-wild> /
  |   *LAST-HOLIDAY / *BY-NUMBER-OF-HOLIDAYS(...)
  |
  |   *BY-NUMBER-OF-HOLIDAYS(...)
  |     |
  |     | NUMBER-OF-HOLIDAYS = <integer 1..1024>
  |
  | ,ASSIGNED-DATES = *NO / *YES
  |
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
  |
  | *SYSLST(...)
  |   |
  |   | SYSLST-NUMBER = *STD / <integer 1..99>
  |

```

## Operandenbeschreibung

**CALENDAR-NAME = <full-filename 1..54>**

Name der Kalenderdatei, aus der Informationen zur Ausgabe angefordert werden.

**SELECT = \*TODAY / \*BASIC-INFORMATION / \*DATE(...) / \*SYMBOLIC-DATE(...) / \*HOLIDAY(...)**

Gibt an, welche Informationen aus der Kalenderdatei ausgegeben werden sollen.

**SELECT = \*TODAY**

Gibt die Informationen für den aktuellen Tag aus:

- den Namen der Kalenderdatei
- das aktuelle Datum
- die Bezeichnung des Wochentages
- das Attribut: W (Work) = Arbeitstag oder F (Free) = freier Tag
- die Anzahl der zugehörigen SYMDAT
- die Arbeitszeit (Beginn und Ende)
- gegebenenfalls den Namen des Feiertags
- eine Liste mit den Namen der zugehörigen SYMDAT und den diesen zugeordneten Zeitangaben (sortiert nach Zeiteinträgen)

**SELECT = \*BASIC-INFORMATION**

Gibt die Basis-Informationen des Kalenders aus:

- den Namen der Kalenderdatei
- die Kalendergrenzen
- die Tage der Standard-Arbeitswoche mit Attributen (Arbeitstag / freier Tag).

**SELECT = \*DATE(...)**

Gibt Tagesinformationen für einen Bereich von Tagen aus. Der gewünschte Bereich wird über die untergeordneten Operanden FROM und TO ausgewählt. Für jeden ausgewählten Tag wird ausgegeben:

- das Datum
- die Bezeichnung des Wochentages
- das Attribut: W (Work) = Arbeitstag oder F (Free) = freier Tag
- die Anzahl der zugehörigen SYMDAT
- die Arbeitszeit (Beginn und Ende)
- gegebenenfalls den Namen des Feiertags

Über den untergeordneten Operanden ASSIGNED-SYM-DATE kann zusätzlich eine Liste mit den Namen der zugehörigen SYMDAT und den diesen zugeordneten Zeiten angefordert werden.

**FROM = \*TODAY / \*FIRST-CALENDAR-DATE / <date 8..10>**

Bestimmt den ersten Tag, ab dem Informationen ausgegeben werden sollen.

**FROM = \*TODAY**

Die Ausgabe beginnt mit dem aktuellen Tag.

**FROM = \*FIRST-CALENDAR-DATE**

Die Ausgabe beginnt mit dem ersten Tag in der Kalenderdatei.

**FROM = <date 8..10>**

Die Ausgabe beginnt mit dem angegebenen Datum.

**TO = \*SAME / \*TODAY / \*LAST-CALENDAR-DATE / <date 8..10> / \*BY-NUMBER-OF-DAYS(...)**

Bestimmt den letzten Tag, bis zu dem Informationen ausgegeben werden sollen.

**TO = \*SAME**

Es werden nur Informationen über den bei FROM angegebenen Tag ausgegeben.

**TO = \*TODAY**

Die Ausgabe endet mit dem aktuellen Tag.

**TO = \*LAST-CALENDAR-DATE**

Die Ausgabe endet mit dem letzten Tag in der Kalenderdatei.

**TO = \*BY-NUMBER-OF-DAYS(...)**

Bestimmt die Größe des gewünschten Bereichs in Tagen, beginnend mit dem bei FROM angegebenen Tag.

**NUMBER-OF-DAYS = <integer 1..1827>**

Anzahl der Tage.

**ASSIGNED-SYM-DATE = \*NONE / \*ALL(...) /**

**<full-filename 1..20 without-cat-user-gen-vers with-wild>(…)**

Gibt an, ob zusätzlich zu der Ausgabe von Kalendertagen die zugeordneten SYMDAT ausgegeben werden.

Voreingestellt ist \*NONE, d.h. es wird keine zusätzliche Liste ausgegeben.

**ASSIGNED-SYM-DATE = \*ALL(...)**

Es werden zusätzlich alle Namen der den angegebenen Kalendertagen zugeordneten SYMDAT ausgegeben.

**ORDER-WITHIN-DAY = \*BY-TIME / \*BY-SYMBOLIC-DATE**

Bestimmt das Sortierkriterium. Die Ausgabe wird nach der zugeordneten Uhrzeit (\*BY-TIME; Voreinstellung) oder alphabetisch nach SYMDAT-Namen (\*BY-SYMBOLIC-DATE) sortiert.

**ASSIGNED-SYM-DATE =**

**<full-filename 1..20 without-cat-user-gen-vers with-wild>(…)**

Es wird zusätzlich das angegebene SYMDAT bzw. die über die Musterzeichenfolge benannte Gruppe von SYMDAT ausgegeben. (Über eine Musterzeichenfolge können mehrere SYMDAT mit einer Angabe ausgewählt werden, wobei nur das Musterzeichen \* am Ende für beliebige Zeichenfolge erlaubt ist.)

**ORDER-WITHIN-DAY = \*BY-TIME / \*BY-SYMBOLIC-DATE**

Bestimmt das Sortierkriterium. Die Ausgabe kann nach der zugeordneten Uhrzeit (\*BY-TIME; Voreinstellung) oder alphabetisch nach SYMDAT-Namen (\*BY-SYMBOLIC-DATE) sortiert werden.

**SELECT = \*SYMBOLIC-DATE(...)**

Gibt Informationen über die im Kalender definierten SYMDAT aus. Der Umfang der Ausgabe wird über die untergeordneten Operanden FROM und TO festgelegt. Über den untergeordneten Operanden ASSIGNED-DATES kann die zusätzliche Ausgabe der zugeordneten Tage veranlasst werden.

**FROM = \*FIRST-SYMBOLIC-DATE /**

**<full-filename 1..20 without-cat-user-gen-vers with-wild>**

Gibt an, ab welchem SYMDAT die Ausgabe erfolgen soll.

**FROM = \*FIRST-SYMBOLIC-DATE**

Die Ausgabe beginnt mit dem in alphabetischer Reihenfolge ersten SYMDAT.

**FROM = <full-filename 1..20 without-cat-user-gen-vers with-wild>**

Die Ausgabe beginnt mit dem angegebenen SYMDAT bzw. mit der Gruppe von SYMDAT (in alphabetischer Reihenfolge), die durch die Musterzeichenfolge benannt wird. (Über eine Musterzeichenfolge können mehrere SYMDAT ausgewählt werden, wobei nur das Musterzeichen \* am Ende für beliebige Zeichenfolge erlaubt ist.)

**TO = \*SAME / \*LAST-SYMBOLIC-DATE /**  
**<full-filename 1..20 without-cat-user-gen-vers with-wild> /**  
**\*BY-NUMBER-OF-SYMBOLIC-DATES(...)**

Gibt an, bis zu welchem SYMDAT die Ausgabe erfolgen soll.

**TO = \*SAME**

Es gilt dieselbe Angabe wie im Operanden FROM.

**TO = \*LAST-SYMBOLIC-DATE**

Die Ausgabe endet mit dem SYMDAT, das in alphabetischer Reihenfolge an letzter Stelle steht.

**TO = <full-filename 1..20 without-cat-user-gen-vers with-wild>**

Die Ausgabe endet mit dem angegebenen SYMDAT bzw. der Gruppe von SYMDAT (in alphabetischer Reihenfolge), die durch die Musterzeichenfolge benannt wird. (Über eine Musterzeichenfolge können mehrere SYMDAT ausgewählt werden, wobei nur das Musterzeichen \* am Ende für beliebige Zeichenfolge erlaubt ist.)

**TO = \*BY-NUMBER-OF-SYMBOLIC-DATES(...)**

Bestimmt die Anzahl der auszugebenden SYMDAT, beginnend mit dem bei FROM angegeben SYMDAT.

**NUMBER-OF-SYM-DATES = <integer 1..4096>**

Anzahl der SYMDAT.

**ASSIGNED-DATES = \*NO / \*ALL / \*NEXT-DATE / \*INTERVAL(...)**

Gibt an, ob zusätzlich die einem ausgewählten SYMDAT zugeordneten Kalendertage ausgegeben werden sollen.

Voreingestellt ist \*NO, d.h. es wird keine zusätzliche Liste ausgegeben.

**ASSIGNED-DATES = \*ALL**

Es werden zusätzlich alle Kalendertage, die dem ausgewählten SYMDAT zugeordnet sind, ausgegeben.

**ASSIGNED-DATES = \*NEXT-DATE**

Es werden zusätzlich alle Kalendertage, die dem ausgewählten SYMDAT zugeordnet sind, ab dem aktuellen Datum ausgegeben.

**ASSIGNED-DATES = \*INTERVAL(...)**

Bestimmt einen Bereich von aufzulistenden Tagen.

**FROM = \*TODAY / \*FIRST-ASSIGNED-DATE / <date 8..10>**

Erster Tag im Bereich der aufzulistenden Tage.

**FROM = \*TODAY**

Der Bereich beginnt mit dem aktuellen Tag.

**FROM = \*FIRST-ASSIGNED-DATE**

Der Bereich beginnt mit dem ersten zugeordneten Tag.

**FROM = <date 8..10>**

Der Bereich beginnt mit dem angegebenen Tag.

**TO = \*SAME / \*TODAY / \*LAST-ASSIGNED-DATE / <date 8..10> / \*BY-NUMBER-OF-DAYS(...)**

Letzter Tag im Bereich der aufzulistenden Tage.

**TO = \*SAME**

Es gilt dieselbe Angabe wie im Operanden FROM.

**TO = \*TODAY**

Der Bereich endet mit dem aktuellen Tag.

**TO = \*LAST-ASSIGNED-DATE**

Der Bereich endet mit dem letzten zugeordneten Tag.

**TO = <date 8..10>**

Der Bereich endet mit dem angegebenen Tag.

**TO = \*BY-NUMBER-OF-DAYS(...)**

Bestimmt die Anzahl der aufzulistenden Tage, beginnend mit dem bei FROM angegebenen Tag.

**NUMBER-OF-DAYS = <integer 1..1827>**

Anzahl der Tage.

**SELECT = \*HOLIDAY(...)**

Gibt Informationen über Feiertage aus. Ausgegeben werden die Namen der Feiertage und ihre Eigenschaften (Typ: zyklisch / nicht-zyklisch; Aktivierungszustand: ja / nein). Der Umfang der Ausgabe wird über die untergeordneten Operanden FROM und TO festgelegt. Über den untergeordneten Operanden ASSIGNED-DATES kann zusätzlich eine Auflistung der zugeordneten Kalendertage angefordert werden.

**FROM = \*FIRST-HOLIDAY /**

**<full-filename 1..30 without-cat-user-gen-vers with-wild>**

Gibt an, mit welchem Feiertag die Ausgabe beginnen soll.

**FROM = \*FIRST-HOLIDAY**

Die Ausgabe beginnt mit dem ersten Feiertag in der Kalenderdatei.

**FROM = <full-filename 1..30 without-cat-user-gen-vers with-wild>**

Die Ausgabe beginnt mit dem angegebenen Feiertag bzw. der Gruppe von Feiertagen (in alphabetischer Reihenfolge), die durch die Musterzeichenfolge benannt wird. (Über eine Musterzeichenfolge können mehrere Feiertage ausgewählt werden, wobei nur das Musterzeichen \* am Ende für beliebige Zeichenfolge erlaubt ist.)

**TO = \*SAME / <full-filename 1..30 without-cat-user-gen-vers with-wild> / \*LAST-HOLIDAY / \*BY-NUMBER-OF-HOLIDAYS(...)**

Gibt an, bis zu welchem Feiertag ausgegeben werden soll.

**TO = \*SAME**

Es gilt dieselbe Angabe wie im Operanden FROM.

**TO = \*LAST-HOLIDAY**

Die Ausgabe endet mit dem letzten Feiertag in der Kalenderdatei.

**TO = <full-filename 1..30 without-cat-user-gen-vers with-wild>**

Die Ausgabe endet mit dem angegebenen Feiertag bzw. der Gruppe von Feiertagen (in alphabetischer Reihenfolge), die durch die Musterzeichenfolge benannt wird. (Über eine Musterzeichenfolge können mehrere Feiertage ausgewählt werden, wobei nur das Musterzeichen \* am Ende für beliebige Zeichenfolge erlaubt ist.)

**TO = \*BY-NUMBER-OF-HOLIDAYS(...)**

Bestimmt die Anzahl der auszugebenden Feiertage, beginnend mit dem bei FROM angegebenen Feiertag.

**NUMBER-OF-HOLIDAYS = <integer 1..1024>**

Anzahl der auszugebenden Tage.

**ASSIGNED-DATES = \*NO / \*YES**

Gibt an, ob zusätzlich alle Kalendertage aufgelistet werden, die auf ausgewählte Feiertage fallen.

Voreingestellt ist \*NO, d.h. es wird keine zusätzliche Liste ausgegeben.

**OUTPUT = \*SYSOUT / list-poss(2): \*SYSOUT / \*SYSLST(...)**

Gibt an, ob die Ausgabe nach SYSOUT oder SYSLST erfolgen soll.

Voreingestellt ist \*SYSOUT. Die gleichzeitige Ausgabe nach SYSLST und SYSOUT ist möglich (Listenangabe).

**OUTPUT = \*SYSLST(...)**

Die Ausgabe erfolgt nach SYSLST.

**SYSLST-NUMBER = \*STD / <integer 1..99>**

Nummer der SYSLST-Datei.

Voreingestellt ist STD, d.h. die Ausgabe erfolgt nach SYSLST.

Bei Angabe einer Nummer ist darauf zu achten, dass der entsprechenden SYSLST-Datei (SYSLST01 bis SYSLST99) eine katalogisierte Datei zugewiesen ist (vgl. Kommando ASSIGN-SYSLST).

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando erfolgreich ausgeführt

## Ausgabeformate

Über den Operanden OUTPUT kann die gewünschte Ausgabe nach SYSOUT und / oder nach SYSLST gelenkt werden. Der Anwender hat die Möglichkeit, eine deutschsprachige Oberfläche einzustellen. Die nachfolgend beschriebenen Layouts der englischen Oberfläche sind für SYSOUT und SYSLST identisch.

SHOW-CALENDAR werk.kalender

```

%-----
%
%                               INFORMATION ABOUT CURRENT DAY
%-----
% CALENDAR NAME:   :1OSN:$QM211.WERK.KALENDER
%
%-----
%   DATE      DAY ATTR #SYM  WORKING HOURS HOLIDAY
%-----
% 1995-03-24  FRI   W     0  08:00-16:30
%-----
%
%-----

```

SHOW-CALENDAR werk.kalender,SELECT=\*BASIC-INFORMATION

```

%-----
%
%                               BASIC INFORMATION
%-----
% CALENDAR NAME:   :1OSN:$QM211.WERK.KALENDER
%
%-----
% CALENDAR LIMITS           START : 1994-01-01
%                               END  : 1996-12-31
%-----
% STANDARD WEEK
%                               WORKING HOURS
%                               START  END
%                               (HH:MM) (HH:MM)
%-----
%   MON           W           08:00 - 16:30
%   TUE           W           08:00 - 16:30
%   WED           W           08:00 - 16:30
%   THU           W           08:30 - 16:30
%   FRI           W           08:00 - 16:30
%   SAT           F           00:00 - 00:00
%   SUN           F           00:00 - 00:00
%-----
%
%-----

```

SHOW-CALENDAR werk.kalender,SELECT=\*DATE(FROM=\*TODAY,TO=1995-03-30)

```

%-----
%                               LIST OF DAYS
%-----
% CALENDAR NAME:   :IOSN:$QM211.WERK.KALENDER
%-----
%   DATE      DAY  ATTR #SYM  WORKING HOURS HOLIDAY
%-----
% 1995-03-24  FRI   W     0  08:00-16:30
% 1995-03-25  SAT   F     0  00:00-00:00
% 1995-03-26  SUN   F     0  00:00-00:00
% 1995-03-27  MON   W     0  08:00-16:30
% 1995-03-28  TUE   W     0  08:00-16:30
% 1995-03-29  WED   W     0  08:00-16:30
% 1995-03-30  THU   W     1  08:30-16:30
%-----
    
```

SHOW-CALENDAR werk.kalender,SELECT=\*DATE(FROM=1995-03-29,  
TO=1995-03-31,ASSIGNED-SYM-DATE=\*ALL)

```

%-----
%                               DAY INFORMATION
%-----
% CALENDAR NAME:   :IOSN:$QM211.WERK.KALENDER
%-----
% 1995-03-29  WED   W     0  08:00-16:30
%-----
%   DATE      DAY  ATTR #SYM  WORKING HOURS HOLIDAY
%-----
% 1995-03-31  THU   W     1  08:30-16:30
%-----
% SYMBOLIC DATE ----- TIME ----- SYMBOLIC DATE ----- TIME -----
% FRIDAYFORUM                13:30:00
%-----
%   DATE      DAY  ATTR #SYM  WORKING HOURS HOLIDAY
%-----
% 1995-03-31  FRI   W     1  08:00-16:30
%-----
% SYMBOLIC DATE ----- TIME ----- SYMBOLIC DATE ----- TIME -----
% ULTIMO                16:00:00
%-----
    
```



```
SHOW-CALENDAR werk.kalender,SELECT=*SYMBOLIC-DATE
(FROM=*FIRST-SYMBOLIC-DATE,TO=*LAST-SYMBOLIC-DATE)
```

```
%-----
%
%                               LIST OF SYMBOLIC DATES
%-----
% CALENDAR NAME:   :10SN:$QM211.WERK.KALENDER
%
%-----
% SYMBOLIC DATE           TIME      TYPE   CYCLTYP  CYCLVAL  CYCLALT
%-----
% FRIDAYFORUM            13:30:00   C     WEEK           1  SKIP
% ULTIMO                  16:00:00   C     MONTH          1  BEFORE
%-----
```

```
/SHOW-CALENDAR werk.kalender,SELECT=*SYMBOLIC-DATE
(FROM=*FIRST-SYMBOLIC-DATE,TO=*LAST-SYMBOLIC-DATE,
ASSIGNED-DATES=*INTERVAL(FROM=*TODAY,TO=1995-03-31))
```

```
%-----
%                               SYMBOLIC DATE INFORMATION
%-----
% CALENDAR NAME:   :10SN:$QM211.WERK.KALENDER
%
%-----
% SYMBOLIC DATE           TIME      TYPE   CYCLTYP  CYCLVAL  CYCLALT
%-----
% FRIDAYFORUM            13:30:00   C     WEEK           1  SKIP
%
%-----
%                               ASSIGNED DATES
%-----
% 1995-03-10  1995-03-17  1995-03-24  1995-03-31
%-----
% SYMBOLIC DATE           TIME      TYPE   CYCLTYP  CYCLVAL  CYCLALT
%-----
% ULTIMO                  16:00:00   C     MONTH          1  BEFORE
%
%-----
%                               ASSIGNED DATES
%-----
% 1995-03-31
%-----
```

SHOW-CALENDAR werk.kalender, SELECT=\*HOLIDAY(FROM=CHRISTMAS,  
TO=EASTER)

```

%-----
%
%                               LIST OF HOLIDAYS
%-----
% CALENDAR NAME:   :10SN:$QM211.WERK.KALENDER
%-----
% HOLIDAY NAME           TYPE    ACTIVE
%-----
% CHRISTMAS              C      Y
% EARLYMAYDAY           C      Y
% EASTER                 N      Y
%-----

```

SHOW-CALENDAR werk.kalender, SELECT = \*HOLIDAY(FROM=CHRISTMAS,  
TO=EASTER, ASSIGNED-DATES=\*YES)

```

%-----
%                               HOLIDAY INFORMATION
%-----
% CALENDAR NAME:   :10SN:$QM211.WERK.KALENDER
%-----
% HOLIDAY NAME           TYPE    ACTIVE    DATE
%-----
% CHRISTMAS              C      Y          ****-12-24
%-----
%                               ASSIGNED DATES
%-----
% 1995-12-24
%-----
% HOLIDAY NAME           TYPE    ACTIVE    DATE
%-----
% EARLYMAYDAY           C      Y          ****-05-01
%-----
%                               ASSIGNED DATES
%-----
% 1995-05-01
%-----
% HOLIDAY NAME           TYPE    ACTIVE    DATE
%-----
% EASTER                 N      Y
%-----
%                               ASSIGNED DATES
%-----
% 1995-04-16
%-----

```

**Beschreibung der Ausgabefelder**

ACTIVE	Aktivierungszustand des Feiertags: Y = aktiviert, N = nicht aktiviert
ASSIGNED DATES	Datumsangaben für die Kalendertage, die einem SYMDAT oder einem Feiertag zugeordnet sind; Ausgabe in der Form yyyy-mm-dd
ATTR	Attribut des Wochentags: W (Work) = Arbeitstag, F (Free) = freier Tag
CALENDAR LIMITS	Kalendergrenzen:
START	Erstes Datum im Kalender (untere Kalendergrenze); Ausgabe in der Form yyyy-mm-dd
END	Letztes Datum im Kalender (obere Kalendergrenze); Ausgabe in der Form yyyy-mm-dd
CALENDAR NAME:	Name der Kalenderdatei
CYCLALT	Alternative für den Fall, dass ein berechneter SYMDAT-Eintrag ein freier Tag ist; Mögliche Werte: BEFORE = davor eintragen, AFTER = danach eintragen, SKIP = Tag überspringen, ON = unbedingt eintragen
CYCLTYP	Zyklus-Art des SYMDAT: DAY (Tag), WORKDAY (Arbeitstag), MONTH (Monat), WEEK (Woche)
CYCLVAL	Zyklus-Wert
DATE	Datum; Ausgabe in der Form yyyy-mm-dd; für zyklische Feiertage Ausgabe in der Form: ****-mm-dd
DAY	Bezeichnung des Wochentags: MON (Montag), TUE (Dienstag), WED (Mittwoch), THU (Donnerstag), FRI (Freitag), SAT (Samstag), SUN (Sonntag)
HOLIDAY	Name des zugehörigen Feiertags
HOLIDAY NAME	Name des Feiertags; max. 30 Zeichen
SYMBOLIC DATE	Name eines SYMDAT; max. 20 Zeichen
STANDARD WEEK	Definitionen der Standard-Arbeitswoche beinhalten folgende Informationen: DAY, ATTR und WORKING HOURS (Beschreibungen siehe dort)
TIME	Zeiteintrag, der dem SYMDAT zugeordnet ist. Ausgabe in der Form hh:mm:ss
TYPE	Typ des SYMDATs oder Feiertags: C (cyclic) = zyklisch, N (non cyclic) = nicht-zyklisch

WORKING HOURS	Beginn und Ende der Arbeitszeit; Ausgabe in der Form hh:mm-hh:mm
#SYM	Anzahl der dem Tag zugeordneten SYMDAT

---

## Literatur

Die Handbücher sind online unter <http://manuals.ts.fujitsu.com> zu finden oder in gedruckter Form gegen gesondertes Entgelt unter <http://manualshop.ts.fujitsu.com> zu bestellen.

- [1] **BS2000/OSD-BC**  
Kommandos  
Benutzerhandbuch
- [2] **JV (BS2000/OSD)**  
Jobvariablen  
Benutzerhandbuch



---

# Stichwörter

## A

- Aktivierung von Feiertagen [17](#)
- Ändern
  - Basis-Information (Makro) [127](#)
  - Kalendertag (Makro) [133](#)
  - nicht-zyklischen Feiertag (Makro) [142](#)
  - nicht-zyklisches SYMDAT (Makro) [146](#)
  - zyklischen Feiertag [131](#)
- Arbeitstag-Zyklus
  - SYMDAT [16](#)
- Arbeitszeit [12](#)
  - Beginnzeit [12](#)
  - Endezeit [12](#)
- Attribut eines Wochentags [12](#)
- Attribut-Regel [13](#)
- Aufbau der Feiertagdatei [18](#)
- Aufbau eines Kalenders [11](#)
- Ausgeben
  - Basis-Information (Makro) [156](#)
  - Feiertag (Makro) [168](#)
  - Feiertage (Makro) [178](#)
  - Kalendertag mit Feiertagen (Makro) [159](#)
  - Kalendertag mit SYMDATs (Makro) [163](#)
  - Kalendertage (Makro) [173](#)
  - SYMDAT (Makro) [187](#)
  - SYMDATs (Makro) [182](#)

## B

- Basis-Information [11](#)
  - ändern (Makro) [127](#)
  - ausgeben (Makro) [156](#)
- Beenden
  - Kalender-Editor [25](#)
- Beginnzeit
  - Arbeitszeit [12](#)

Blätter-Funktion [30](#)

## C

- C-Headerfile [208](#)
- Control-Block (Makro) [77](#)

## D

- Dateiattribute (Kalenderdatei) [20](#)
- Datenübertragung [32](#)
- Datumsangaben
  - symbolisch [7](#)
- Deaktivieren Feiertag (Makro) [121](#)
- Dialogschnittstelle [7, 23](#)
- DSECT [197](#)

## E

- Endezeit
  - Arbeitszeit [12](#)
- Erzeugen
  - Kalender (Makro) [102](#)
  - nicht-zyklisches SYMDAT (Makro) [117](#)

## F

- Feiertag [17](#)
  - Namenskonvention [17](#)
  - nicht-zyklisch [17](#)
  - zyklisch [17](#)
- Feiertag aktivieren [17](#)
- Feiertag ändern (Makro) [131](#)
- Feiertag ausgeben (Makro) [168](#)
- Feiertag deaktivieren (Makro) [121](#)
- Feiertag-Datei [18](#)
  - Aufbau [18](#)
- Feiertage ausgeben (Makro) [178](#)
- Free-Day-Rule [15](#)

### Funktion

- Blättern 30
- Positionieren 30
- Funktionstasten (Übersicht) 33
- Funktionsübersicht (Makro) 97

### G

- GOTO-Angabe 30

### I

- Informationsstruktur 11
- Initialwerte
  - Kalendertage 12
- Installation 20

### J

- Jobvariable
  - Programm überwachen 26

### K

#### Kalender

- erzeugen (Makro) 102
- Obergrenze 11
- öffnen (Makro) 78, 151
- schließen (Makro) 100
- sichern (Makro) 154
- Untergrenze 11
- Kalender-Editor 23
  - Basis-Informationen ausgeben 43
  - Bedienung 24
  - Beenden 25
  - Blätter-Funktion 30
  - Eingabeauswertung 31
  - Eingabeprüfung 31
  - Feiertage auflisten 62
  - Feiertags-Informationen ausgeben 66
  - Funktionen 29
  - Funktionsauswahl 38
  - Funktionstasten 33
  - Funktionsübersicht 23, 24
  - Hauptmenü 34
  - Kalendertage auflisten 47
  - Maske B020 43
  - Maske C000 34

- Maske C010 38
- Maske D010 47
- Maske D020 50
- Maske H010 62
- Maske H020 66
- Maske S010 54
- Maske S020 57
- Maskenaufbau 28
- Maskenbereiche 28
- Maskenformat 28
- Maskenübersicht 27
- Objektauswahl 31
- Open-Modus 34
- Positionieren 30
- Starten 25
- SYMDAT-Informationen ausgeben 57
- SYMDAT-Liste ausgeben 54
- Tagesinformationen bearbeiten 50
- Unterbrechen 25

- Kalender-Obergrenze 11
- Kalender-Untergrenze 11
- Kalenderaufbau 11
- Kalenderdatei 19
  - Dateiattribute 20
  - Schutzmechanismen 19
- Kalenderdatei lesen (SHOW-CALENDAR) 221
- Kalenderdaten 7
- Kalendergrenzen 11, 12
  - verschieben 11
- Kalendergrenzen verschieben 11
- Kalendertag ändern (Makro) 133
- Kalendertag mit Feiertagen ausgeben (Makro) 159
- Kalendertag mit SYMDATs ausgeben (Makro) 163
- Kalendertage 13
  - ausgeben (Makro) 173
  - Initialwerte 12
- Klassifizierung der Returncodes (Subcode 1) 87
- Kommando
  - SHOW-CALENDAR 221
- Kommando-Returncode 228
- Kommandoschnittstelle 7, 221



**L**

Layout-Übersicht [84](#)  
 Layouts erzeugen (Makro) [76](#)  
 Lesen Kalenderdatei (Kommando) [221](#)  
 Löschen SYMDAT (Makro) [125](#)

**M**

Maincode (Returncode) [86](#)

**Makro**

  Beispiel [90](#)  
 Control-Block [77](#)  
 Funktionsübersicht [97](#)  
 Kalender öffnen [78](#)  
 Layout-Übersicht [84](#)  
 Layouts erzeugen [76](#)  
 Performance [71](#)

**Makro-Funktionsaufruf**

  Basis-Information ändern [127](#)  
 Basis-Information ausgeben [156](#)  
 Feiertag deaktivieren [121](#)  
 Feiertag mit Kalendertagen ausgeben [168](#)  
 Kalender erzeugen [102](#)  
 Kalender öffnen [151](#)  
 Kalender schließen [100](#)  
 Kalender sichern [154](#)  
 Kalendertag ändern [133](#)  
 Kalendertag mit Feiertagen ausgeben [159](#)  
 Kalendertag mit SYMDAT ausgeben [163](#)  
 Liste von Feiertagen ausgeben [178](#)  
 Liste von Kalendertagen ausgeben [173](#)  
 Liste von SYMDAT ausgeben [182](#)  
 Nächsten Zeitpunkt zu SYMDATs  
   ausgeben [192](#)  
 Nicht-zyklischen Feiertag ändern [142](#)  
 Nicht-zyklisches SYMDAT ändern [146](#)  
 Nicht-zyklisches SYMDAT erzeugen [117](#)  
 SYMDAT löschen [125](#)  
 SYMDAT mit Kalendertagen ausgeben [187](#)  
 Zyklischen Feiertag ändern [131](#)  
 Zyklisches SYMDAT ändern [137](#)

Makro-Syntax [202](#)

Makroaufrufformat [202](#)

**Maske**

  Aufbau [28](#)  
 Bereichsbeschreibung [28](#)  
 Format [28](#)  
 Verlassen [29](#)

**Maskenübersicht**

  Kalender-Editor [27](#)

Meldungsdatei [20](#)

Metasyntax des Makros [203](#), [204](#)

MF-Operand [205](#), [206](#), [207](#)

**Monats-Zyklus**

  SYMDAT [16](#)

**N****Namenskonvention**

  Feiertag [17](#)  
 SYMDAT [14](#)

Nicht-zyklischen Feiertag ändern (Makro) [142](#)

Nicht-zyklischer Feiertag [17](#)

Nicht-zyklisches SYMDAT [15](#)

Nicht-zyklisches SYMDAT ändern (Makro) [146](#)

Nicht-zyklisches SYMDAT erzeugen (Makro) [117](#)

**O**

Öffnen Kalender (Makro) [151](#)

**P**

Parameterliste [197](#)

**Programm**

  überwachen (Jobvariable) [26](#)

Programmschnittstelle [7](#), [71](#)

  Formatübersicht [73](#)

**R**

Readme-Datei [9](#)

Returncode (Kommando) [228](#)

**Returncodes**

  Übersicht [87](#)

Returncodes (Makro) [86](#)

  Ausnahmen [87](#)

  Klassifizierung [87](#)

  Übersicht [88](#)

Rückinformation [86](#)

### S

- S-Variable [7](#)
- Schließen Kalender (Makro) [100](#)
- Schnittstellen [7](#)
- SDF-P-Prozedur [7](#)
- SHOW-CALENDAR-Kommando [221](#)
  - Ausgabeformate [229](#)
- Sichern Kalender (Makro) [154](#)
- Standard-Arbeitswoche [12](#)
- Standard-Header [86](#)
- Start-Datum
  - SYMDAT [16](#)
- Starten
  - Kalender-Editor [25](#)
- Subcode 1 (Returncode) [86](#)
  - Bedeutung [87](#)
- Subcode 2 (Returncode) [86](#)
- Subsystem-Deklaration [20](#)
- Symbolische Datumsangabe [14](#)
- SYMDAT [14](#)
  - Arbeitstag-Zyklus [16](#)
  - Monats-Zyklus [16](#)
  - Namenskonvention [14](#)
  - nicht-zyklisch [15](#)
  - Start-Datum [16](#)
  - Tag-Zyklus [16](#)
  - Wochen-Zyklus [16](#)
  - Zeitangabe [14](#)
  - zyklisch [15](#)
  - Zyklus-Wert [15](#)
  - Zyklusart [15](#)
  - Zyklusarten [15](#)
- SYMDAT ausgeben (Makro) [187, 192](#)
- SYMDAT löschen (Makro) [125](#)
- SYMDATs ausgeben (Makro) [182](#)
- Syntaxdatei [20](#)

### T

- Tag-Zyklus
  - SYMDAT [16](#)

### U

- Unterbrechen
  - Kalender-Editor [25](#)

- Untergrenze
  - Kalender [11](#)

### W

- Wochen-Zyklus
  - SYMDAT [16](#)
- Wochentag [12](#)
  - Attribut [12](#)

### Z

- Zeitangabe (SYMDAT) [14](#)
- Zyklischer Feiertag [17](#)
- Zyklisches SYMDAT [14](#)
- Zyklus
  - SYMDAT [15](#)
- Zyklus-Wert
  - SYMDAT [15](#)
- Zyklusart
  - SYMDAT [15](#)