



Produktbeschreibung

SWT01 Release 3.10



1 Inhalt

1	INHALT	2
2	ÜBERSICHT	3
2.1	Trägersysteme	3
2.2	Zielumgebungen.....	3
2.3	COBOL- und PL/1-Sprachlevel.....	3
2.4	Unterstützte Datenzugriffstechniken	3
3	ZIELE DER SWT01-ENTWICKLUNG	4
4	ABLAUF BEI DER PROGRAMMENTWICKLUNG MIT SWT01.....	5
5	ÜBERSICHT ÜBER DIE SWT01-SYNTAX	6
5.1	Beispiel für ein einfaches SWT01-Programm.....	6
6	DER AUFBAU EINES VOM SWT01 GENERIERTEN PROGRAMMS	8
6.1	Regeln für den Aufbau eines SWT01-Programms.....	8
6.2	Programmmodule (COBOL Sections)	8
6.3	Ablauf einer Phase in einem SWT01-Programm.....	8
6.4	Ablauf einer Phase mit zweistufigem Gruppenwechsel	9
6.5	Phase mit mehreren steuernden Eingabedateien	10
6.6	Hierarchische Phasenabläufe	10
7	KUNDENSPEZIFISCHE STANDARDS.....	11
8	BASISVERSION UND ERWEITERUNGEN.....	12



2 Übersicht

Dieses Handbuch beschreibt in Kurzform den Leistungsumfang des COBOL- und PL/1-Programmgenerators SWT01.

Mit Hilfe des SWT01 werden aus einem vom Anwendungsprogrammierer erstellten SWT01-Quellcode sofort ablauffähige Batch- oder Online-Programme generiert.

2.1 Trägersysteme

SWT01 ist für die folgenden Trägersysteme verfügbar:

- ❑ IBM OS/390 und z/OS
- ❑ IBM VM/ESA und z/VM
- ❑ Siemens BS2000/OSD
- ❑ Microsoft Windows NT, 2000 oder XP.

2.2 Zielumgebungen

Mit dem SWT01 erstellte Programme sind in allen Umgebungen nutzbar, in denen auch normale COBOL-Programme nutzbar sind. Insbesondere werden unterstützt:

- ❑ OS390 und z/OS Batch-Programme
- ❑ Batch-Programme mit Zugriffen auf IMS/DB und DB/2-Datenbanken
- ❑ Online-Programme für IMS/TM und CICS
- ❑ Native BS2000-Programme
- ❑ Online-Programme für UTM unter BS2000

2.3 COBOL- und PL/1-Sprachlevel

SWT01 generiert Programmcode für folgende COBOL- bzw. PL/1-Sprachlevel:

- ❑ IBM VS COBOL (ANS-74)
- ❑ IBM VS COBOL II Rel.2 (ANS-78)
- ❑ IBM VS COBOL II Rel.3 oder höher
- ❑ IBM COBOL/370 (COBOL370)
- ❑ IBM COBOL for OS390 & VM
- ❑ IBM Enterprise COBOL
- ❑ Siemens COBOL85
- ❑ PL/1 für MVS & VM

2.4 Unterstützte Datenzugriffstechniken

SWT01 generiert Programmcode zum Zugriff auf folgende Datenbestände:

- ❑ Sequentielle Non-VSAM-Dateien
- ❑ VSAM-ESDS, -RRDS, -KSDS
- ❑ IMS/DB-Datenbanken
- ❑ SQL-Datenbanken (DB/2, SESAM/ESQL)
- ❑ ADABAS



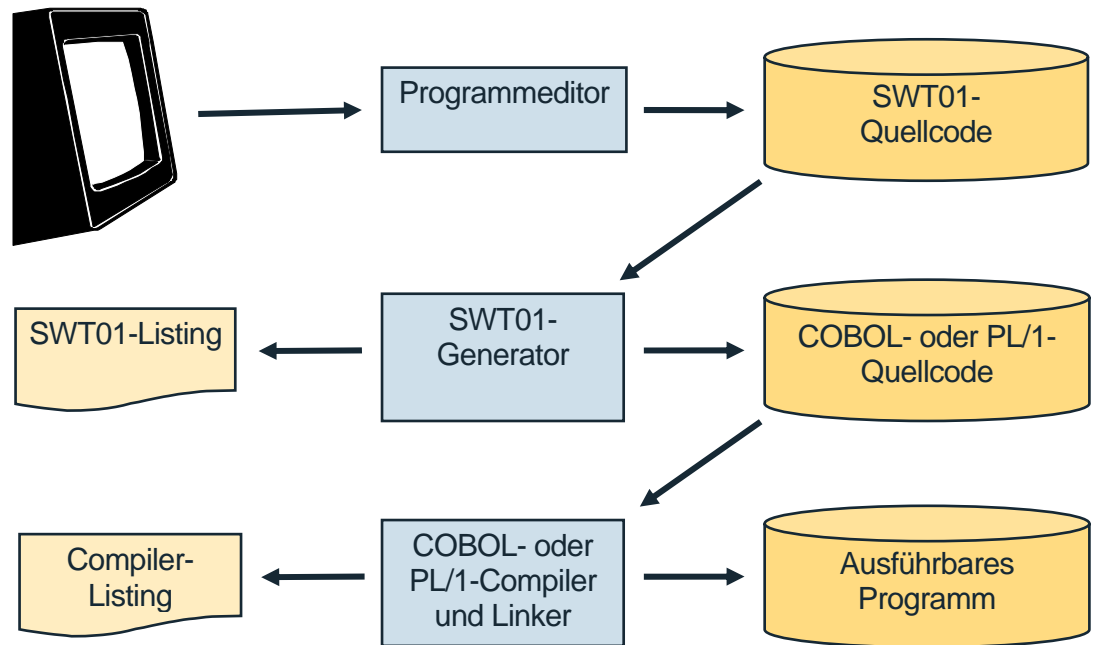
3 Ziele der SWT01-Entwicklung

Durch die Programmentwicklung mit SWT01 werden die folgenden Ziele unterstützt:

- ❑ Methodische Programmentwicklung
- ❑ strukturierter und modularer Programm-Aufbau und gleichartige Behandlung von Batch - und Online - Programmen
- ❑ gleichartige Behandlung von konventionell organisierten Datenbeständen und Datenbanken
- ❑ Portabilität des erzeugten Programmcodes
- ❑ große Änderungs- und Wartungsfreundlichkeit
- ❑ gute Lesbarkeit des generierten Codes
- ❑ gute automatische Dokumentation
- ❑ gemeinsamer Sprachgebrauch bei allen Entwicklern
- ❑ Unterstützung kommerzieller Konstrukte durch den Generator. Dazu gehören z.B. Gruppenwechsel oder Behandlung mehrerer steuernder Eingabedatenbestände durch den Generator

4 Ablauf bei der Programmentwicklung mit SWT01

Der folgende Ablauf beschreibt schematisch den Ablauf bei der Programmentwicklung mit SWT01:



- ❑ Ein Anwendungsentwickler erfasst über einen beliebigen Programm- oder Texteditor den SWT01 Quellcode.
- ❑ Der SWT01 Quellcode wird vom SWT01-Generator umgesetzt in einen reinen COBOL- oder PL/1-Quellcode. Dabei wird ein SWT01-Generatorlisting erzeugt.
- ❑ Der COBOL- oder PL/1-Quellcode wird, wie in der Programmentwicklung ohne den SWT01-Generator, über den Compiler und Linker umgesetzt in ein ausführbares Programm.



5 Übersicht über die SWT01-Syntax

SWT01 unterstützt verschiedene Kommandos, mit denen der Entwickler sein Programm beschreibt. Dabei wird unterschieden zwischen

- strukturbildenden Kommandos, mit denen die Programmstruktur beschrieben wird und
- Benutzereinfügungen

Aus den strukturbildenden Kommandos generiert SWT01 einen syntaktisch korrekten Programmrahmen, der bereits ein ablauffähiges Programm darstellt.

Über die Benutzereinfügungen kann dann der Entwickler gezielt an jeder beliebigen Stelle im Programmrahmen nativen COBOL- oder PL/1-Code einfügen. Diese Einfügungen stellen dann die eigentliche Programmlogik dar.

5.1 Beispiel für ein einfaches SWT01-Programm

Der folgende Code stellt bereits ein einfaches SWT01-Programm dar:

```
./ OPTIONS OUTPUT(PRINT(INPUT OUTPUT)).  
./ PROGRAM TESTPGM W.RUPPRATH GENERATE(DISPLAY).  
  
./ PHASE S05 LINKAGE-TYPE(STANDARD)  
./ GROUP ( 90 L-FILE )  
          ( 05 FILIALE PICTURE('9(4) COMP-3') ).  
./ FILE STEUER DESCRIPTION ( ORGANIZATION ( ESDS ) ).  
  
./ USER STEUER RECORD.  
05 SATZ.  
   10 ABC-FILIALE PIC 9(4) COMP-3.  
*   .... usw ...  
./ FILE STEUER USAGE ( FILE-PRIORITY ( 10 )  
                       FUNCTION ( INPUT )  
                       CONTROL-ELEMENTS ((05 ABC-FILIALE)) ).  
./ USER G05-NEUE-FILIALE.  
* Programmcode für die Gruppeneröffnung auf Stufe 05
```

Die mit ./ beginnenden Zeilen stellen die SWT01-Kommandos dar.

- Das OPTIONS-Kommando beschreibt Meta-Informationen für den Generator, in diesem Fall den Umfang des auszugebenden Listings.
- Das PROGRAM-Kommando beschreibt Eigenschaften des zu generierenden Programms. Hier sind dies der Programmname, der Autor und eine Option, in welchem Umfang DISPLAY-Kommandos in das COBOL-Programm generiert werden sollen.
- Das PHASE-Kommando beschreibt Eigenschaften einer Programmphase. Jedes SWT01-Programm besteht aus mindestens einer Phase, die eine in sich geschlossene Verarbeitungslogik enthält. Phasen können seriell hintereinander oder hierarchisch strukturiert angeordnet sein.
- Das GROUP-Kommando beschreibt die einzelnen Level eines zu generierenden Gruppenwechsels. Die Stufe 90 (L-FILE) beschreibt hier den gesamten Datenbestand, die Stufe 05 (FILIALE) beschreibt eine untergeordnete Gruppenstufe.



- ❑ Das FILE DESCRIPTION-Kommando beschreibt die Eigenschaften einer zu verarbeitenden Datei, in diesem Fall eines VSAM ESDS.
- ❑ Das USER RECORD-Kommando ist eine Benutzereinfügung und beschreibt den Aufbau eines Datensatzes in der Datei STEUER.
- ❑ Das FILE USAGE-Kommando bewirkt die Generierung eines Zugriffsmoduls für das Lesen der Datei STEUER. Der Parameter FILE-PRIORITY zeigt dem Generator an, dass es sich um eine steuernde Eingabedatei handelt. Durch CONTROL-ELEMENTS wird beschrieben, welche Felder des Eingabedatensatzes die für den Gruppenwechsel signifikanten Werte enthalten.
- ❑ Das USER G05-Kommando ist eine Benutzereinfügung, mit dem der Entwickler in der Gruppeneröffnungsroutine der Stufe 05 beliebigen Code einfügen kann.



6 Der Aufbau eines vom SWT01 generierten Programms

6.1 Regeln für den Aufbau eines SWT01-Programms

- ❑ Jedes generierte Programm besteht aus mindestens einer Phase.
- ❑ Jede Phase besteht aus mehreren COBOL-Sections bzw. PL/1-PROCs.
- ❑ Sofern das Programm mehr als eine seriell ablaufende Phase beinhaltet, wird zusätzlich eine Steuerphase, die die Ablauffolge aller anderen Phasen steuert, generiert (Phase 00).
- ❑ Die Phasen können sequentiell oder hierarchisch durchlaufen werden.
Phasen können auch als Eingabe- oder Ausgabemodule eines COBOL SORT- oder MERGE verwendet werden.

6.2 Programmmodule (COBOL Sections)

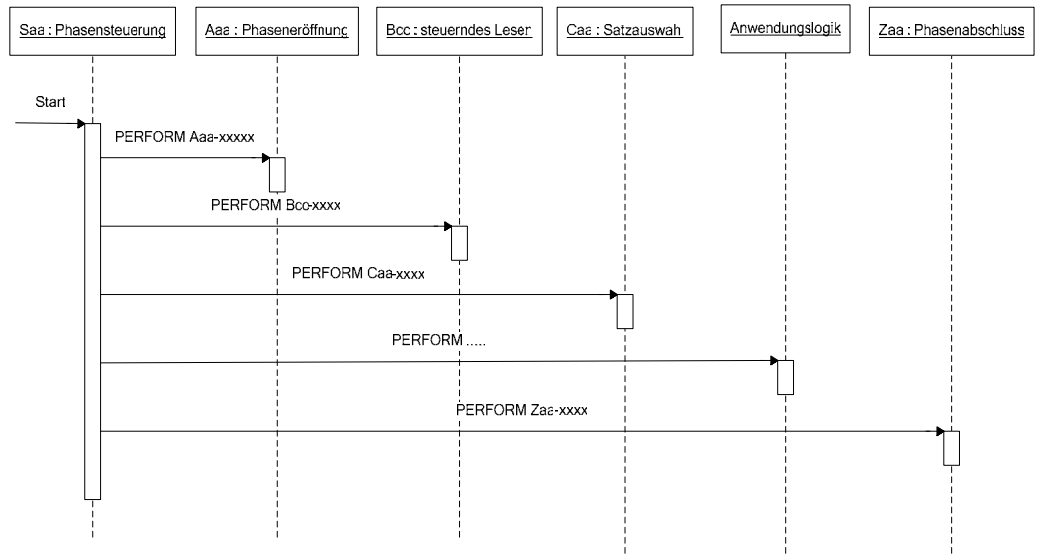
Phasen und alle untergeordneten SWT01-Module wie z.B. Einzelsatzverarbeitungen werden in Form von COBOL-Sections bzw. PL/1-Prozeduren generiert. Der SWT01-Generator kennt dabei die folgenden Modultypen:

- Saa - Die Steuerungslogik für Phase aa
- Aaa - Die Phaseneröffnungsroutine für Phase aa. Hier werden z.B. die steuernden Eingabedatenbestände der Phase geöffnet.
- Zaa - Die Phasenabschlussroutine für Phase aa. Hier werden die steuernden Datenbestände der Phase wieder geschlossen.
- Bcc - Lesen der steuernden Eingabedatei mit Priorität (FILE-PRIORITY) cc
- Caa - Satzauswahl und Prüfung auf Gruppenwechsel für Phase aa
- Dcc - Einzelsatzprüfung für die steuernde Eingabedatei mit Priorität cc
- Gbb - Gruppeneröffnungsroutine für Gruppenstufe bb
- Ebb - Gruppenabschlussroutine für Gruppenstufe bb
- Fcc - Einzelsatzverarbeitung der steuernden Eingabedatei mit Priorität cc
- Inn - Lesen der nicht steuernden Eingabedatei mit Dateinummer nn
- Wnn - Schreiben der Ausgabedatei mit Dateinummer nn
- Rnn - Update der Ein-/Ausgabedatei mit Dateinummer nn
- Vnn - Sonstige Ein-/Ausgaberroutinen für Dateinummer nn
- Xhh - Fehlerbehandlungsroutinen.
- Unn - Benutzereinfügungen.

Diese Modultypen finden sich in jedem vom SWT01 generierten Programm wieder, so dass eine hohe Wiedererkennung der Programmstruktur gewährleistet ist, und zwar auch für unbekannte Programme.

6.3 Ablauf einer Phase in einem SWT01-Programm

Der Ablauf einer Phase in einem SWT01-Programm ist immer gleich und wird durch das folgende Sequenzdiagramm beschrieben:



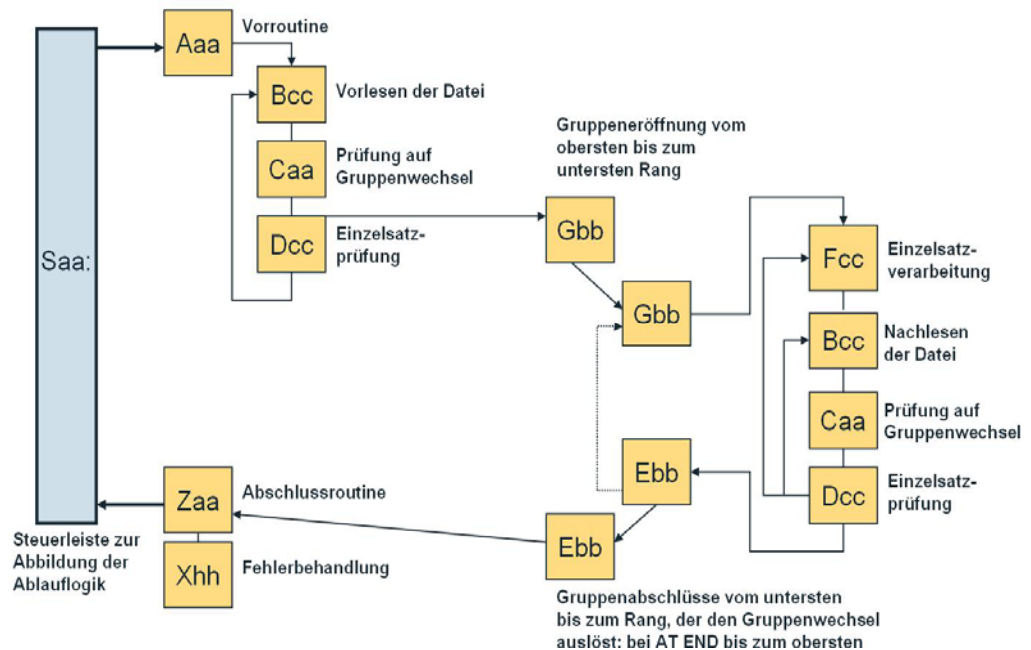
Die im Diagramm aufgeführten Module Saa, Aaa, Bcc, Caa und Zaa sind COBOL SECTIONS oder PL/1-PROCs.

Die Anwendungslogik ist dabei abhängig von der Art des Programms. Die einfachste Form kann dabei eine Einzelsatzverarbeitung einer einzigen steuernden Eingabedatei sein.

In komplizierten Phasen können viele steuernde Eingabedateien behandelt werden, mehrstufige Gruppenwechsel und Paritätsprüfungen auf den steuernden Eingabedateien. Dazu werden in den folgenden Abschnitten einige Beispiele gegeben.

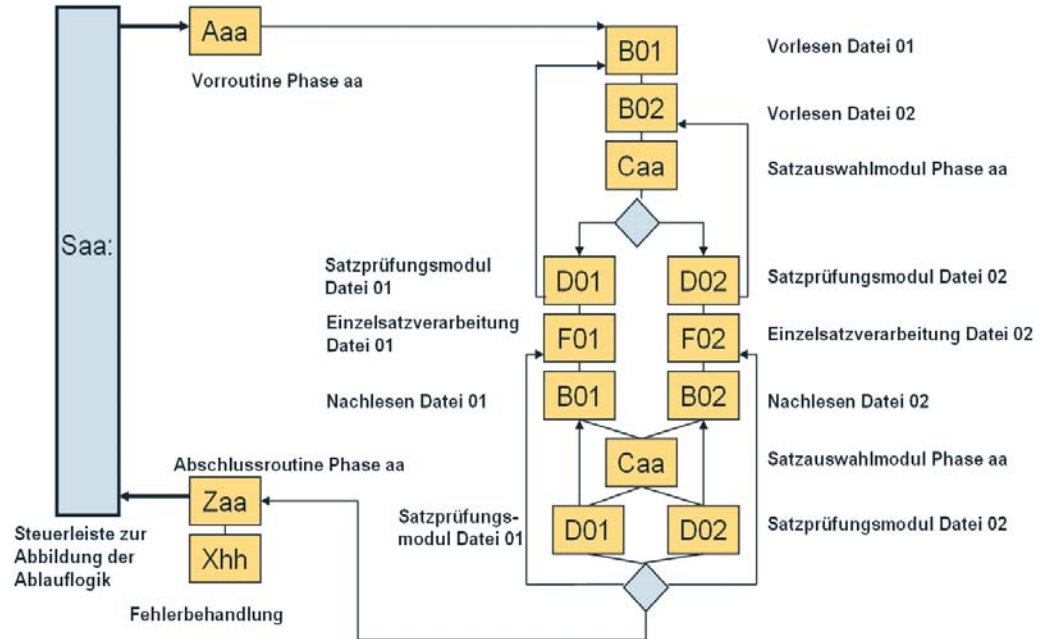
6.4 Ablauf einer Phase mit zweistufigem Gruppenwechsel

Der folgende Ablauf stellt den Programmfluss in einer Phase mit zweistufigem Gruppenwechsel und einer einzelnen steuernden Eingabedatei dar:



6.5 Phase mit mehreren steuernden Eingabedateien

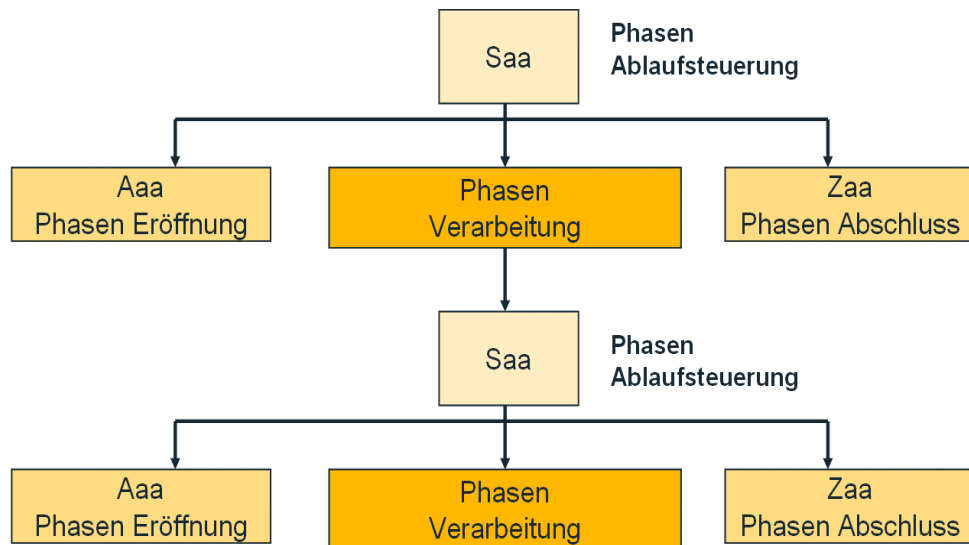
Der folgende Ablauf stellt den Programmfluss in einer Phase mit mehreren steuernden Eingabedateien dar.



Selbstverständlich sind in einer Phase auch mehrere steuernde Eingabedateien und mehrstufige Gruppenwechsel kombinierbar.

6.6 Hierarchische Phasenabläufe

Sollen in einem SWT01-Programm zwei Phasen hierarchisch angeordnet werden, so ergibt sich der folgende Ablauf:



Auch hier können natürlich mehrstufige und komplexere Hierarchien aufgebaut werden.



7 Kundenspezifische Standards

Da sich in jeder Entwicklungsabteilung unterschiedliche Standards entwickelt haben, wird jedes ausgelieferte SWT01 entsprechend dem Wunsch des Kunden eigens für diesen generiert.

Dabei werden die Installationsvariablen (soweit benötigt) besetzt und die Elemente der Basisversion mit den SWT01-Zusätzen verbunden. Auf Wunsch können auch für alle Parameter andere Standardwerte eingestellt werden, als in diesem Handbuch beschrieben.

Über Installationsvariablen kann z.B. gesteuert werden, welcher COBOL-Sprachlevel als Standard unterstützt wird (wenn im PROGRAM-Kommando nichts angegeben ist) oder ob Unterprogramme statisch oder dynamisch aufgerufen werden sollen.

Auch können z.B. Aufrufe an kundenspezifische Fehlerbehandlungsroutinen in die Xhh-Module generiert werden.



8 Basisversion und Erweiterungen

Damit nur die tatsächlich beim Anwender benötigten Elemente des SWT01 zur Auslieferung gelangen (und damit in Rechnung gestellt werden), ist die Unterscheidung zwischen SWT01-Basisversion und Erweiterungen notwendig.

Die Basisversion unterstützt folgendes:

- ❑ Programmablauf-Steuerung ('S' und 'C' Module),
- ❑ Programm- und Phasen-Eröffnung- sowie -Abschluß ('A', 'X03' und 'Z' Module),
- ❑ Gruppen-Eröffnung- sowie -Abschluß ('G' und 'E' Module),
- ❑ Einzelsatz-Verarbeitung ('D' und 'F' Module),
- ❑ konventionelles I/O für sequentielle sowie VSAM- und ISAM-Dateien ('B', 'I', 'R', 'V' und 'W' Module),
- ❑ Statuscodeüberprüfungen nach durchgeführten I/O
- ❑ Trace (X01).
- ❑ Dokumentation der generierten Programmstruktur in Kommentaren im Kopf des generierten Programms

Die Erweiterungen behandeln das I/O für unterschiedliche Datenbanksysteme (IMS/DB, DB/2, SESAM/SQL) oder die Generierung von Ablaufsteuerungen für Online-Programme unter den Transaktionsmonitoren IMS, CICS und UTM.

Dabei werden unterstützt:

- ❑ das eigentliche I/O für ein Datenbanksystem
- ❑ die Behandlung von I/O-Aufrufen zur an den Transaktionsmonitor
- ❑ die Einhaltung von Konventionen, die durch ein Datenbanksystem und/oder einen Transaktionsmonitor vorgegeben sind (syntaktisch, z.B. benötigte Parameter etc.),
- ❑ die Generierung von Masken für Kontrollblöcke, sofern sie im Programm benötigt werden (z.B. PCB-Masken),
- ❑ die Status- und Fehlerbehandlung nach durchgeführtem I/O
- ❑ spezielle Interfaces zum Datenbanksystem oder Transaktionsmonitor (z.B. SWTTDLI)
- ❑ Zyklus-Modelle (z.B. MODEL-Parameter im PROGRAM Command).