# GCOS-8 COBOL Workbench Ô

# Reference Guide

Version 1.2.00
Revision: November 1994

# REFERENCE GUIDE

B & C Solution S.A.R.L, has made every effort to ensure that this manual is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time.

The Software described in this document is supplied under a license and may be used or copied only in accordance with the terms of such license, and in particular any warranty of fitness of B & C Solutions products for any particular purpose is expressly excluded and in no event will B & C Solution be liable for any consequential loss.

B & C SOLUTION S.A.R.L
B.P.1, GRAND RUE
F-30210 COLLIAS, FRANCE
TEL. +33.66.22.05.84
FAX. +33.66.22.07.18

# Table of Contents

# Chapter 1 INTRODUCTION

## Preface

GCOS-8 COBOL Workbench (G8WB) from B & C Solution is a menu driven integrated BULL GCOS-8, IDS-II, INTEREL, DMIV-TP, TP8 and batch emulation and development system for IBM and IBM compatible micro computer systems. This manual is intended to document how the various G8WB facilities may be utilized to develop, maintain and prototype GCOS-8 application systems targeted for production operation on a GCOS-8 Mainframe computer.

## Intended Audience

This manual is not intended to provide education or techniques for developing or coding GCOS-8 application systems. It is intended for use by programmers and analysts knowledgeable regarding the design and coding of GCOS-8 application systems.

## Related Publications

In describing the integration, level of support and limitations of the G8WB system, this manual will only contain information related to the use of G8WB. For a comprehensive list and definition of COBOL language, IDS-II and INTEREL Database Manager, DMIV-TP and TP8 Transaction Processor, Forms Software, etc. and Workbench features and facilities, please refer to the manuals available from BULL S.A, Micro Focus Limited and other parties.

This manual is intended for reference use and is accompanied by three other manuals. The Tutorial Guide, which is intended for overview and initial training, The User Guide, which is intended as a supplement to the Reference guide and the Technical Guide, which is intended for the TP/DB administrator.

## Operation Environment & Requirements

### Hardware

G8WB was designed within the concept and environment of Micro Focus Workbench (MFWB) and will operate on any micro computers supported by MFWB. In general MFWB will operate on IBM PC/AT, 3270 PC/AT and PS/2 series of micro computers, and most compatibles. Please refer to Micro Focus Workbench documentation for exact details.

G8WB was designed for optimum functionality combined with dynamic memory capabilities and its memory requirements are closely related to the application systems requirements. Configuration parameters with substantial impact on memory requirements are: IDS-II schema and subschema size, DMIV-TP and TP8 system parameters (Constant-, TX-storage etc.), program/TPR size.

Using MS-DOS or PC-DOS the Micro Focus XM Extended Memory facility is required and a minimum of 4MB.

If Disk Cache device drivers are used, they must allocate buffers in extended memory and coordinate memory use with Micro Focus XM. Please refer to MFWB documentation for details.

Using OS/2 a minimum of 6MB is required.

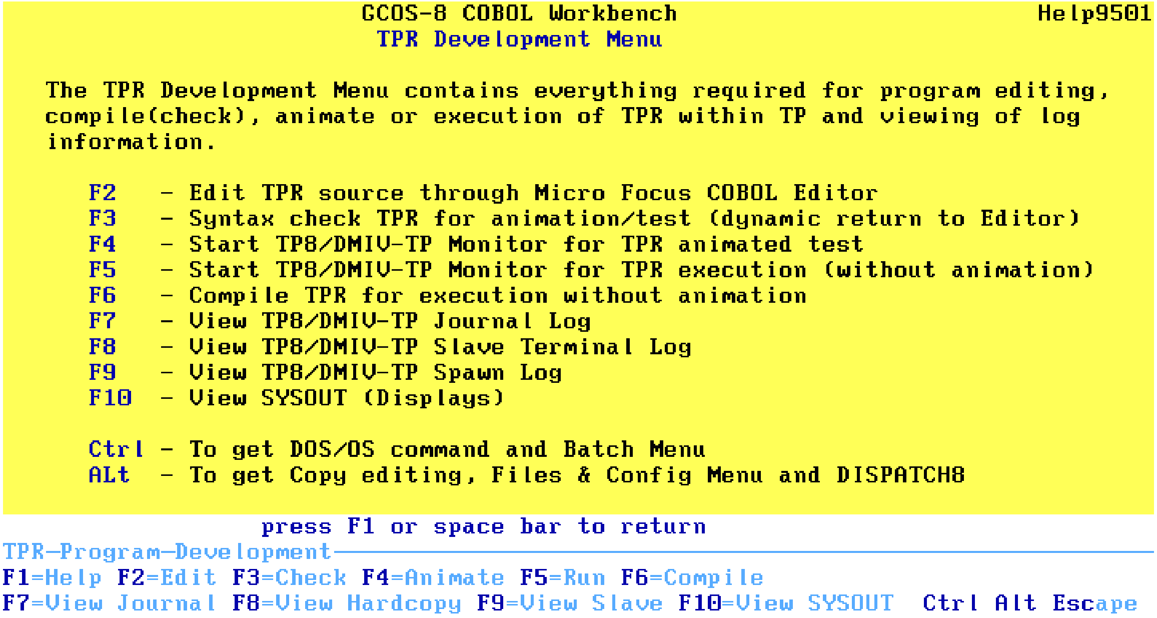
## Software

G8WB requires MFWB v3.2 (minimum 3.2.24), which in turn requires OS/2 1.2 or DOS 3.1 or greater. Please refer to MFWB documentation for details.

G8WB may be installed to operate on a single workstation or to operate in a Server & Workstation network environment. Server & Workstation operation requires standard Local Area Network software, i.e NETWARE, LAN MANAGER etc.

# Chapter 2 TP DEVELOPMENT MENU

The TP development menu is your place of work for checking, editing and testing of TP application programs.



```
                         GCOS-8 COBOL Workbench                         Help9501
                          TPR Development Menu

   The TPR Development Menu contains everything required for program editing,
   compile(check), animate or execution of TPR within TP and viewing of log
   information.

      F2    - Edit TPR source through Micro Focus COBOL Editor
      F3    - Syntax check TPR for animation/test (dynamic return to Editor)
      F4    - Start TP8/DMIV-TP Monitor for TPR animated test
      F5    - Start TP8/DMIV-TP Monitor for TPR execution (without animation)
      F6    - Compile TPR for execution without animation
      F7    - View TP8/DMIV-TP Journal Log
      F8    - View TP8/DMIV-TP Slave Terminal Log
      F9    - View TP8/DMIV-TP Spawn Log
      F10   - View SYSOUT (Displays)

      Ctrl  - To get DOS/OS command and Batch Menu
      ALt   - To get Copy editing, Files & Config Menu and DISPATCH8

                  press F1 or space bar to return
TPR-Program-Development---------------------------------------------------------
F1=Help F2=Edit F3=Check F4=Animate F5=Run F6=Compile
F7=View Journal F8=View Hardcopy F9=View Slave F10=View SYSOUT  Ctrl Alt Escape
```

## Developing

The compile process for G8WB is accomplished is two steps:

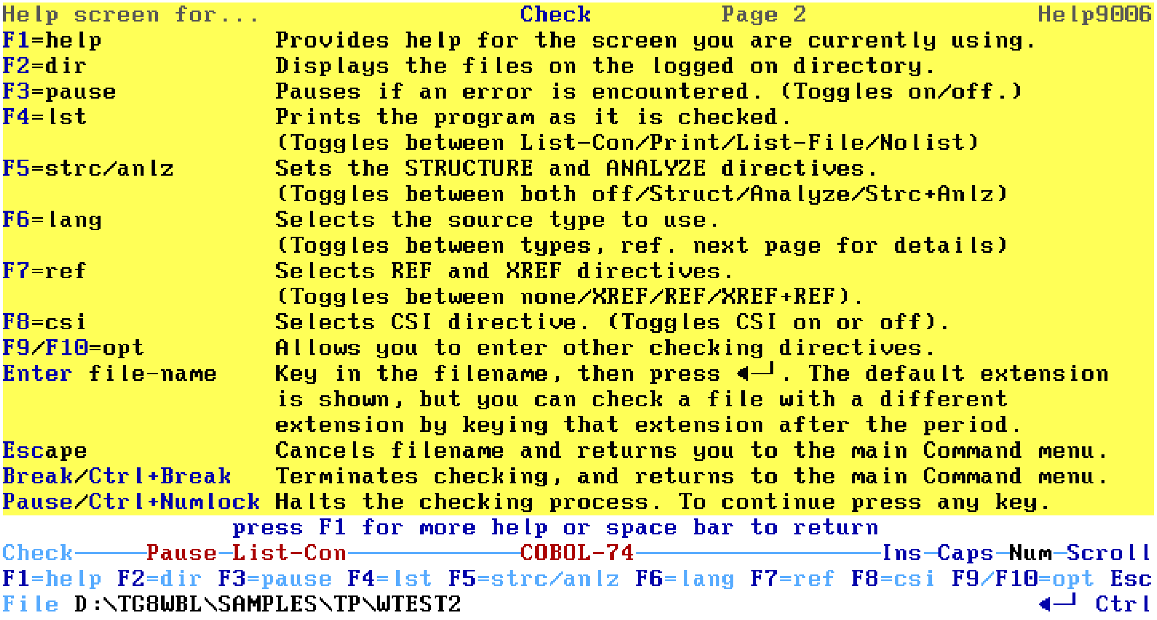
- Checking
- Generating

The checking process represents the traditional COBOL compilation up to the point of object code generation. Once checked, a program may be execution both with or without animation, from the .INT version generated by the MFWB Checker.

### Checking

Checking menu may be entered through F3-Checking from the TP Development menu. Checking menu provides a number of toggle switches, which may be changed by pressing the appropriate function key. Toggle switch settings are remembered as long as you do not exit G8WB. Initial toggle switch settings may be configured, ref. Configuration chapter for details.

Following illustrate the Checker menu with F1-Help enabled:



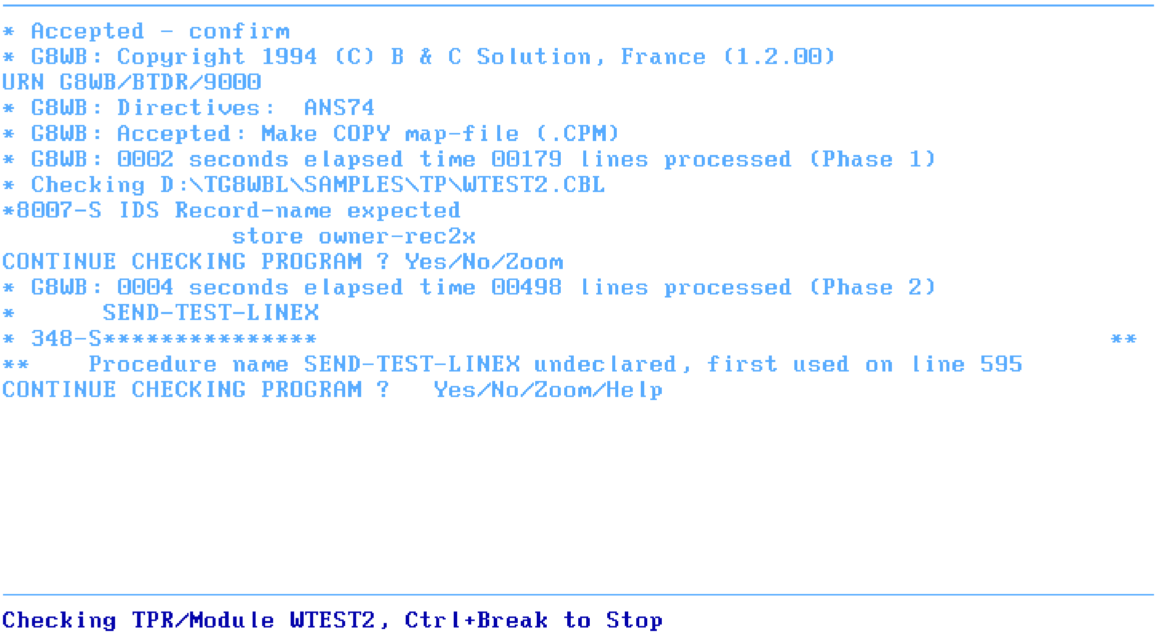
```
Help screen for...                  Check         Page 2                 Help9006
F1=help            Provides help for the screen you are currently using.
F2=dir             Displays the files on the logged on directory.
F3=pause           Pauses if an error is encountered. (Toggles on/off.)
F4=lst             Prints the program as it is checked.
                   (Toggles between List-Con/Print/List-File/Nolist)
F5=strc/anlz       Sets the STRUCTURE and ANALYZE directives.
                   (Toggles between both off/Struct/Analyze/Strc+Anlz)
F6=lang            Selects the source type to use.
                   (Toggles between types, ref. next page for details)
F7=ref             Selects REF and XREF directives.
                   (Toggles between none/XREF/REF/XREF+REF).
F8=csi             Selects CSI directive. (Toggles CSI on or off).
F9/F10=opt         Allows you to enter other checking directives.
Enter file-name    Key in the filename, then press ◄┘. The default extension
                   is shown, but you can check a file with a different
                   extension by keying that extension after the period.
Escape             Cancels filename and returns you to the main Command menu.
Break/Ctrl+Break   Terminates checking, and returns to the main Command menu.
Pause/Ctrl+Numlock Halts the checking process. To continue press any key.
                press F1 for more help or space bar to return
Check─────Pause─List-Con────────────COBOL-74─────────────────Ins─Caps─Num─Scroll
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
File D:\TG8WBL\SAMPLES\TP\WTEST2                                          ◄┘ Ctrl
```

Please refer to MFWB documentation for details on toggle switches.

## Error handling

Once invoked, Checker and G8WB Preprocessor will output error messages with associated source lines in the checking window. G8WB information lines document the source language and type, the number of lines checked and the two checking phases. Phase 1 is G8WB preprocessing and Phase 2 is checking of preprocessed program through COBOL/2.

Checker will normally stop and prompt for continuation, when errors are detected, i.e. Pause toggle set. Following illustrate a checking with two errors detected.



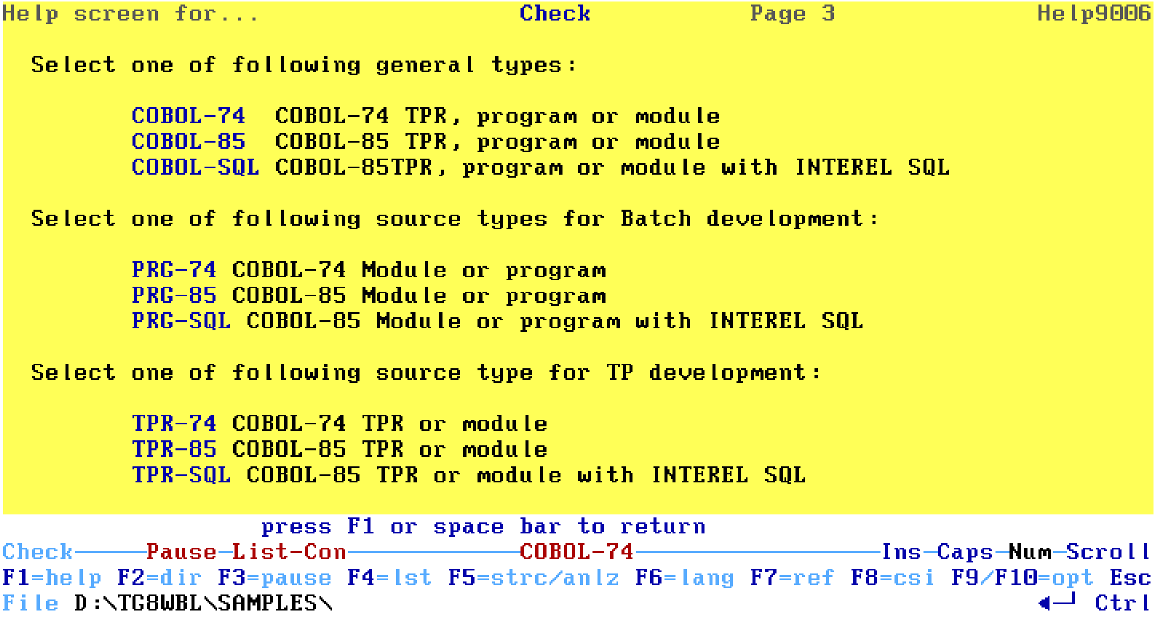
```
* Accepted - confirm
* G8WB: Copyright 1994 (C) B & C Solution, France (1.2.00)
URN G8WB/BTDR/9000
* G8WB: Directives:  ANS74
* G8WB: Accepted: Make COPY map-file (.CPM)
* G8WB: 0002 seconds elapsed time 00179 lines processed (Phase 1)
* Checking D:\TG8WBL\SAMPLES\TP\WTEST2.CBL
*8007-S IDS Record-name expected
                store owner-rec2x
CONTINUE CHECKING PROGRAM ? Yes/No/Zoom
* G8WB: 0004 seconds elapsed time 00498 lines processed (Phase 2)
*      SEND-TEST-LINEX
* 348-S***************                                                     **
**     Procedure name SEND-TEST-LINEX undeclared, first used on line 595
CONTINUE CHECKING PROGRAM ?   Yes/No/Zoom/Help




Checking TPR/Module WTEST2, Ctrl+Break to Stop
```

When prompted, you may use NO to stop checking, YES to continue until next error or ZOOM to finish checking without further prompting. Once checking completes, Editor is automatically invoked when errors are detected, ref. below.

*Note: Preprocessor and Checker issue similar prompts, however your answer is only recognized by the issuer. As a result you may have to ZOOM twice, if both Preprocessor and Checker errors are detected. Pause toggle is only recognized by Checker, so Preprocessor may still prompt you even though Pause toggle is unset.*

## Language

F6-Language may be used to specify the COBOL language dialect to be used during checking.



```
Help screen for...                    Check           Page 3              Help9006

  Select one of following general types:

         COBOL-74  COBOL-74 TPR, program or module
         COBOL-85  COBOL-85 TPR, program or module
         COBOL-SQL COBOL-85TPR, program or module with INTEREL SQL

  Select one of following source types for Batch development:

         PRG-74 COBOL-74 Module or program
         PRG-85 COBOL-85 Module or program
         PRG-SQL COBOL-85 Module or program with INTEREL SQL

  Select one of following source type for TP development:

         TPR-74 COBOL-74 TPR or module
         TPR-85 COBOL-85 TPR or module
         TPR-SQL COBOL-85 TPR or module with INTEREL SQL

                  press F1 or space bar to return
Check─────Pause─List─Con────────────COBOL-74─────────────────Ins─Caps─Num─Scroll
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
File D:\TG8WBL\SAMPLES\                                                  ◄┘ Ctrl
```

Since Micro Focus COBOL/2 compiler system does not directly support GCOS-8 specific COBOL-74 and COBOL-85 language adaptions, G8WB will invoke a GCOS-8 specific preprocessor during checking. The G8WB Preprocessor is, however, integrated with the Checker, so the checking process will be viewed as one single activity.

G8WB Preprocessor will include various verification code in the checked source to assist you in your testing efforts. As an example, the TPR calling arguments are verified for proper linkage sizing, before you application TPR enter execution.

### COBOL-74

May be used for any type of COBOL-74 source, i.e. module, program, TPR.

### COBOL-85

May be used for any type of COBOL-85 source, i.e. module, program, TPR.

### COBOL-SQL

May be used for any type of COBOL-85 source with embedded SQL (INTEREL), i.e. module, program, TPR. COBOL-SQL will trigger SQL preprocessing through XDB Preprocessor.

### TPR-74

Like COBOL-74, however Preprocessing will enforce additional TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

### TPR-85

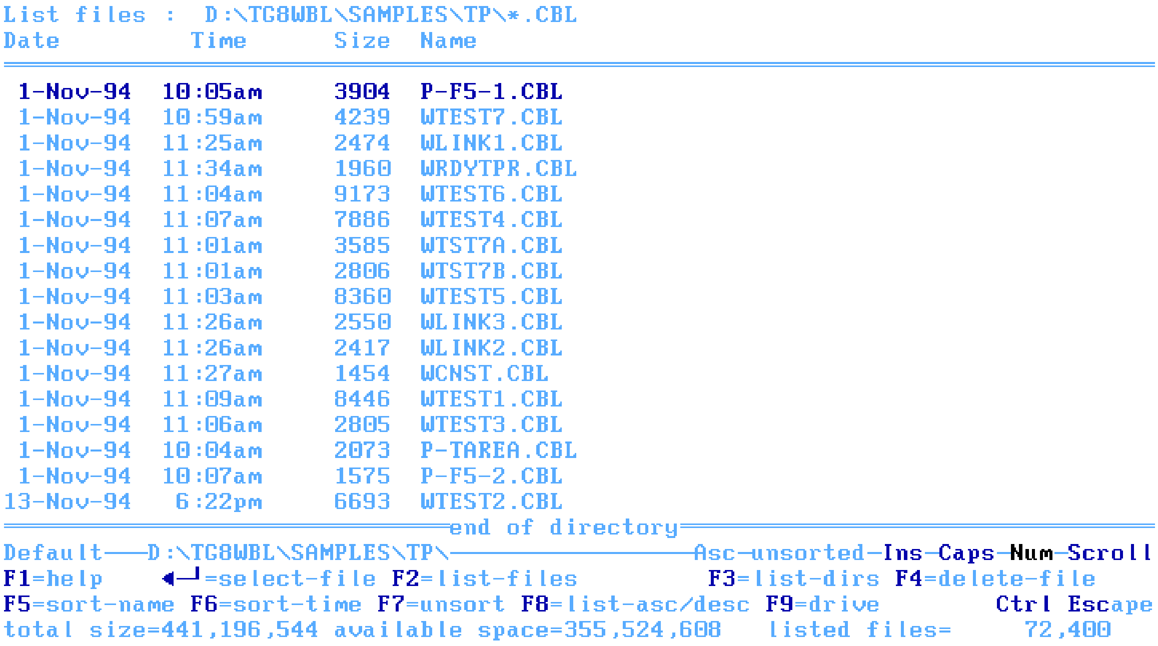
Like COBOL-85, however Preprocessing will enforce additional TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

### TPR-SQL

Like COBOL-SQL, however Preprocessing will enforce additional TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

## Directory

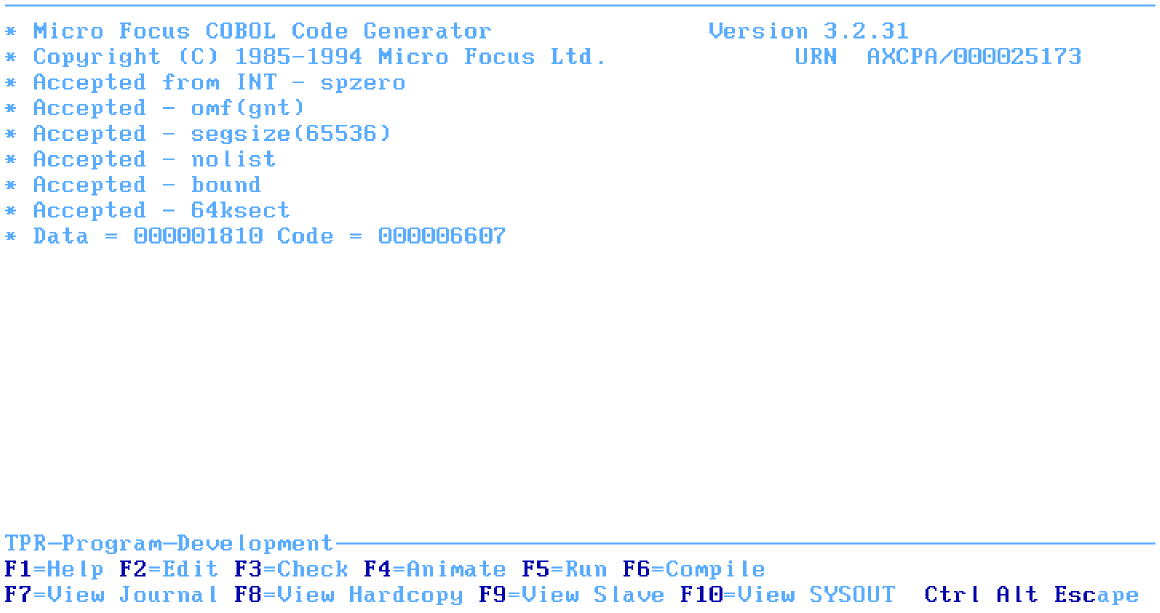
The F2-Dir function allows you to retrieve source components through directory browsing. The selected source will become default for Checker, Editor, Animator and Generator.



```
List files :  D:\TG8WBL\SAMPLES\TP\*.CBL
Date          Time       Size  Name
══════════════════════════════════════════════════════════════════════════════

 1-Nov-94  10:05am      3904  P-F5-1.CBL
 1-Nov-94  10:59am      4239  WTEST7.CBL
 1-Nov-94  11:25am      2474  WLINK1.CBL
 1-Nov-94  11:34am      1960  WRDYTPR.CBL
 1-Nov-94  11:04am      9173  WTEST6.CBL
 1-Nov-94  11:07am      7886  WTEST4.CBL
 1-Nov-94  11:01am      3585  WTST7A.CBL
 1-Nov-94  11:01am      2806  WTST7B.CBL
 1-Nov-94  11:03am      8360  WTEST5.CBL
 1-Nov-94  11:26am      2550  WLINK3.CBL
 1-Nov-94  11:26am      2417  WLINK2.CBL
 1-Nov-94  11:27am      1454  WCNST.CBL
 1-Nov-94  11:09am      8446  WTEST1.CBL
 1-Nov-94  11:06am      2805  WTEST3.CBL
 1-Nov-94  10:04am      2073  P-TAREA.CBL
 1-Nov-94  10:07am      1575  P-F5-2.CBL
13-Nov-94   6:22pm      6693  WTEST2.CBL
══════════════════════════════end of directory═════════════════════════════════
Default──D:\TG8WBL\SAMPLES\TP\────────────────Asc─unsorted─Ins─Caps─Num─Scroll
F1=help     ◄┘=select-file F2=list-files         F3=list-dirs F4=delete-file
F5=sort-name F6=sort-time F7=unsort F8=list-asc/desc F9=drive        Ctrl Escape
total size=441,196,544 available space=355,524,608   listed files=     72,400
```

## Generating

Once the testing activity has been completed for an application component, you may generate your module, program or TPR into the more efficient GNT object format. Besides being more efficient, GNT's also prevent animation of the source. You should thus use GNT format for application components, where you have completed testing, i.e. only components under test are animated.

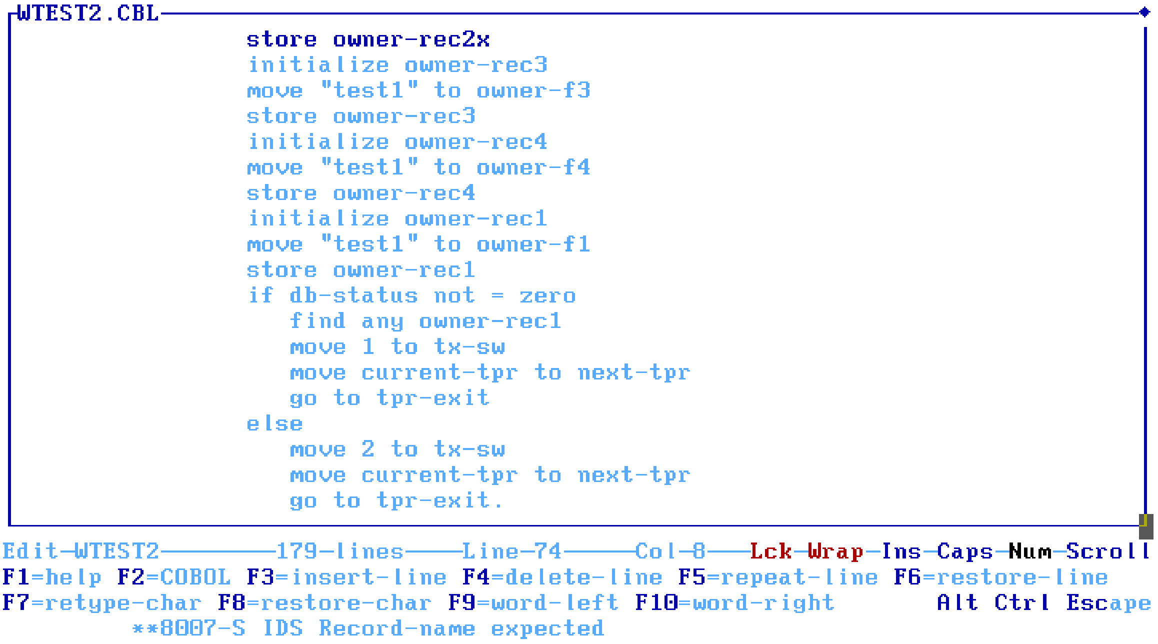


```
* Micro Focus COBOL Code Generator              Version 3.2.31
* Copyright (C) 1985-1994 Micro Focus Ltd.            URN  AXCPA/000025173
* Accepted from INT - spzero
* Accepted - omf(gnt)
* Accepted - segsize(65536)
* Accepted - nolist
* Accepted - bound
* Accepted - 64ksect
* Data = 000001810 Code = 000006607








TPR-Program-Development-------------------------------------------------------
F1=Help F2=Edit F3=Check F4=Animate F5=Run F6=Compile
F7=View Journal F8=View Hardcopy F9=View Slave F10=View SYSOUT  Ctrl Alt Escape
```

## Editing

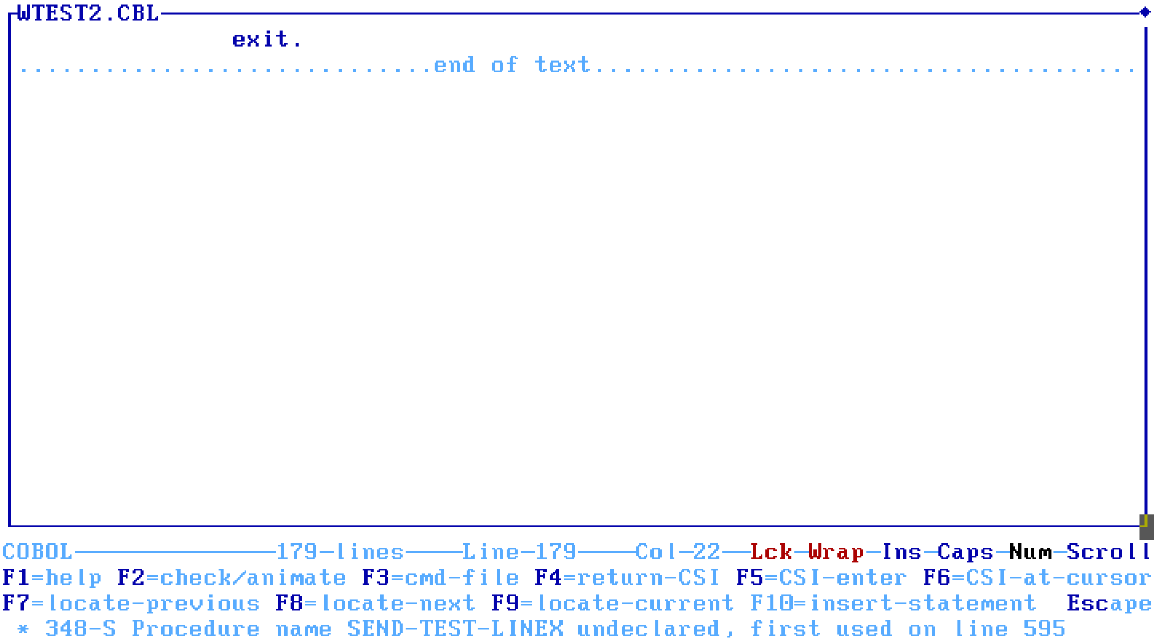
This documentation assume you are utilizing the MFWB Editor, however other editors may be used also. Please refer to Configuration chapter for details.

As mentioned above, Checker will automatically invoke Editor when errors are detected. Normally the source program you are checking will be loaded, however if errors occur in COPY members, both the source program and the COPY member will be loaded. Following illustrate Editor after the checking example illustrated above:



```
WTEST2.CBL------------------------------------------------------------------
                 store owner-rec2x
                 initialize owner-rec3
                 move "test1" to owner-f3
                 store owner-rec3
                 initialize owner-rec4
                 move "test1" to owner-f4
                 store owner-rec4
                 initialize owner-rec1
                 move "test1" to owner-f1
                 store owner-rec1
                 if db-status not = zero
                    find any owner-rec1
                    move 1 to tx-sw
                    move current-tpr to next-tpr
                    go to tpr-exit
                 else
                    move 2 to tx-sw
                    move current-tpr to next-tpr
                    go to tpr-exit.
Edit-WTEST2--------179-lines----Line-74-----Col-8---Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right        Alt Ctrl Escape
         **8007-S IDS Record-name expected
```

Editor will automatically position to the source line where the first error is detected and give you the error message at the bottom of the screen. After correcting your source, you may continue positioning to the next error through F2-COBOL and F8-Locate-next.



```
WTEST2.CBL
                    exit.
.............................end of text...................................
COBOL──────────────179-lines────Line-179────Col-22──Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=check/animate F3=cmd-file F4=return-CSI F5=CSI-enter F6=CSI-at-cursor
F7=locate-previous F8=locate-next F9=locate-current F10=insert-statement  Escape
 * 348-S Procedure name SEND-TEST-LINEX undeclared, first used on line 595
```

Once you have completed error browsing and correction of source, you need to Escape out of F2-COBOL, and then resave your source through ALT-F4-Save-file.

When COPY members are involved, error browsing will automatically open and close COPY members as you browse through the errors. Once you have made your source correction, you need to resave your COPY member like you do with your source. When you Escape out of F2-COBOL, i.e. to resave, you may return and continue error browsing again through F2-COBOL. If you finish up in a COPY member, then Escape will take you back to the main source.

Once you have completed correction of source, you may use Escape to return to the TP development menu, where you can request checking again.

If you have customized the Editor, ref. User Guide for details, you may also stay within Editor and start checking through F2-COBOL and F2-Check/animate. You cannot start TP animation within Editor since TP Monitor needs to be loaded first, however, you may start animation to insert break-points in the TPR or module and then exit.

## Testing

When you initiate testing in a TP environment, no specific program is specified. Instead a TP Monitor is loaded, like on GCOS-8. When you request Animation, it is thus not until  you start to execute transactions, that Animation will commence.

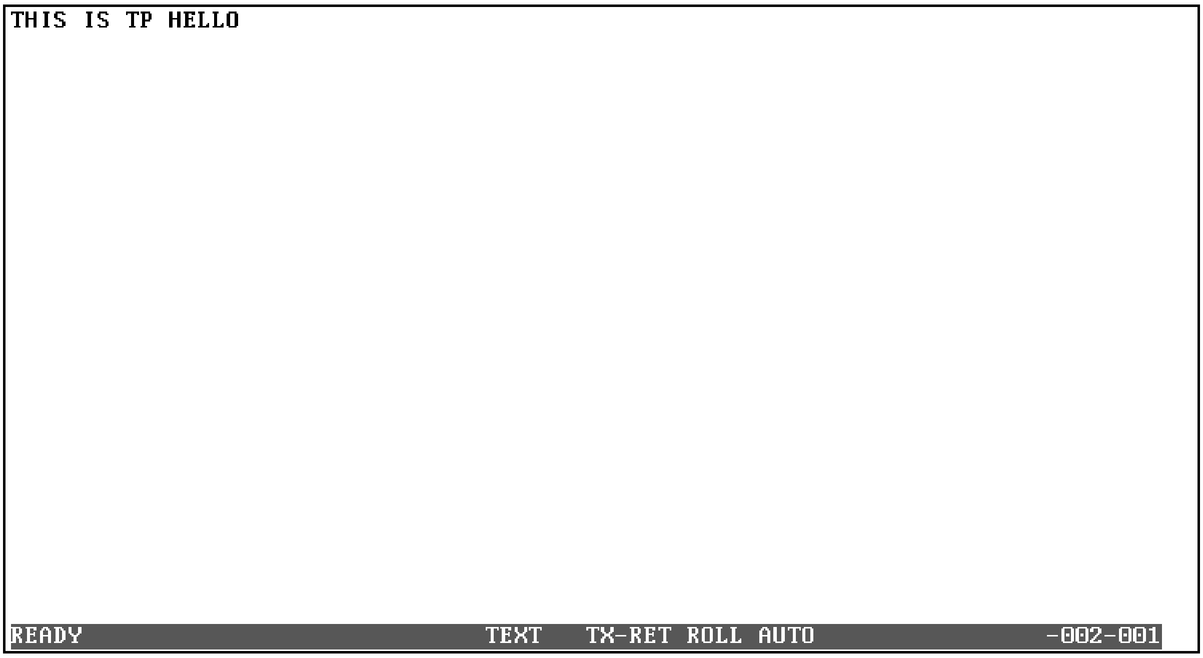
With DMIV-TP and TP8 you have various ways of logging on to the TP Monitor, and you may or may not explicitly identify a Logical-Id (LID). With G8WB you may configure whether you want to logon with an explicit Logical-Id, i.e. prompted for LID, or logon with an implicit Logical-ID. Please refer to Configuration chapter for details.

G8WB does not directly support the DMIV-TP/TP8 Logon transaction feature, however, you may configure whether you want to automatically execute a transaction during logon to TP. Please refer to Configuration chapter for details.

During TP Monitor startup, the various initialization and configuration steps are documented. If any errors are detected, you will receive a prompt, otherwise G8WB will enter terminal emulation. Following illustrate a TP start-up sequence:

```
GCOS-8 COBOL Workbench - TP Monitor (1.2.00)
URN G8WB/BTDM/9000
Defaults configuration ...
TP8 Workstation Configuration ...
TP-Executive Call Interface - Initialized
TP8 Workstation Initialize
CXPP User Procedure Initialize
CXAP User Procedure Initialize
FORMAT Forms Manager Initialize
FORMAT User Exit initialized
DISPATCH8 Print Interface Manager Initialize
DBMS IDS-II Initialize
Subschema Domain Loading ...
Environment Loading ...
Library open ...
Resident TPR Loading ...
VIP77XX Emulation Initialized
```

Once in terminal emulation mode, you may enter Message-Id's/Commands as you are familiar with from the GCOS-8 environment. Following illustrate VIP78XX terminal emulation.

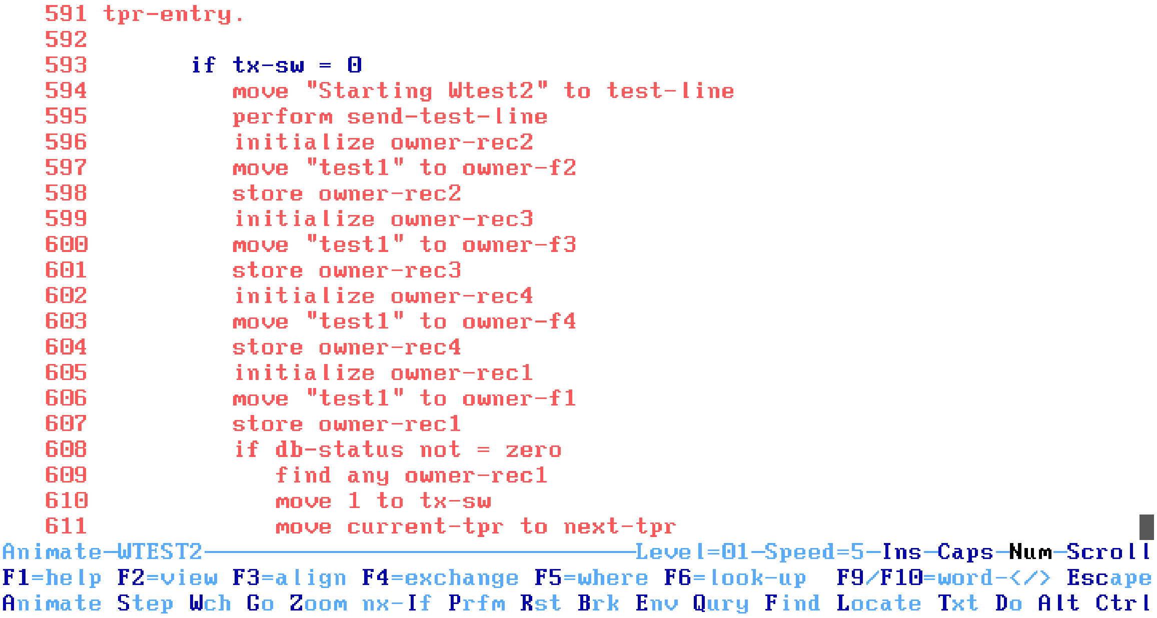


```
THIS IS TP HELLO

READY                                  TEXT   TX-RET ROLL AUTO                -002-001
```

## Animating

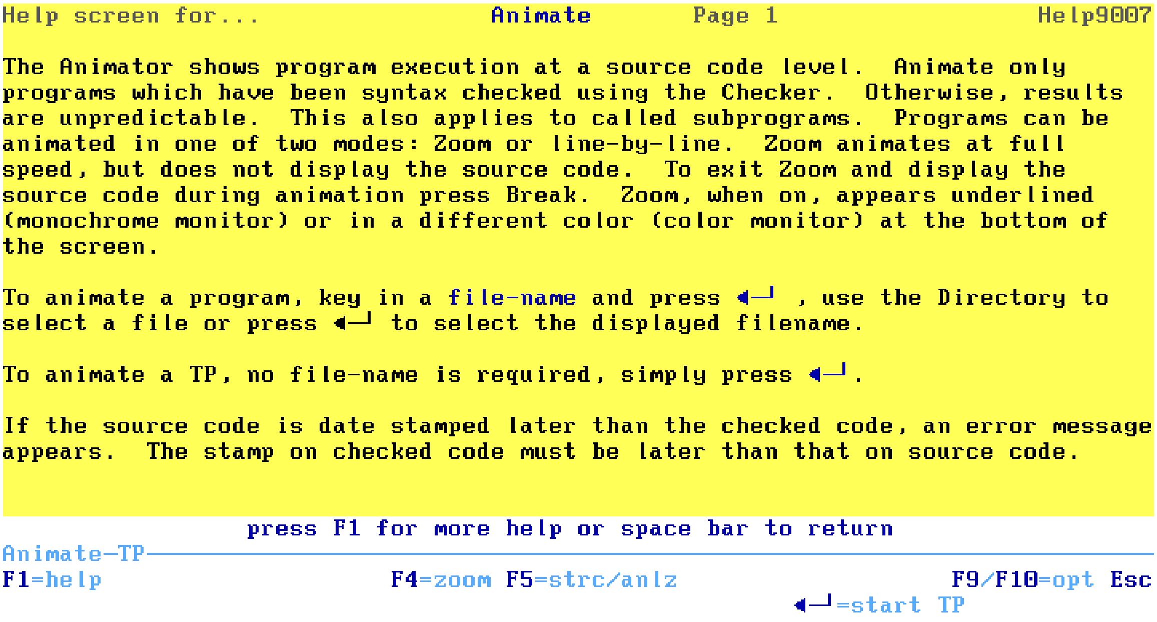
Program testing is normally performed through Animator, which enable you to visually check correct program execution and allows you to fully control and verify execution of your application. Please refer to MFWB documentation for details on Animator features.

Animation will commence when you reach your application TPR, after you have started the appropriate transaction. You may start any TPR by giving the TPR name as Message-Id or Command, or use a configured Message-Id or Command. Following illustrate the animator screen:



```
 tpr-entry.

        if tx-sw = 0
           move "Starting Wtest2" to test-line
           perform send-test-line
           initialize owner-rec2
           move "test1" to owner-f2
           store owner-rec2
           initialize owner-rec3
           move "test1" to owner-f3
           store owner-rec3
           initialize owner-rec4
           move "test1" to owner-f4
           store owner-rec4
           initialize owner-rec1
           move "test1" to owner-f1
           store owner-rec1
           if db-status not = zero
              find any owner-rec1
              move 1 to tx-sw
              move current-tpr to next-tpr
Animate-WTEST2------------------------------Level=01-Speed=5-Ins-Caps-Num-Scroll
F1=help F2=view F3=align F4=exchange F5=where F6=look-up  F9/F10=word-</> Escape
Animate Step Wch Go Zoom nx-If Prfm Rst Brk Env Qury Find Locate Txt Do Alt Ctrl
```

Before starting animation, you may select whether you want to begin in Animate or Zoom mode.



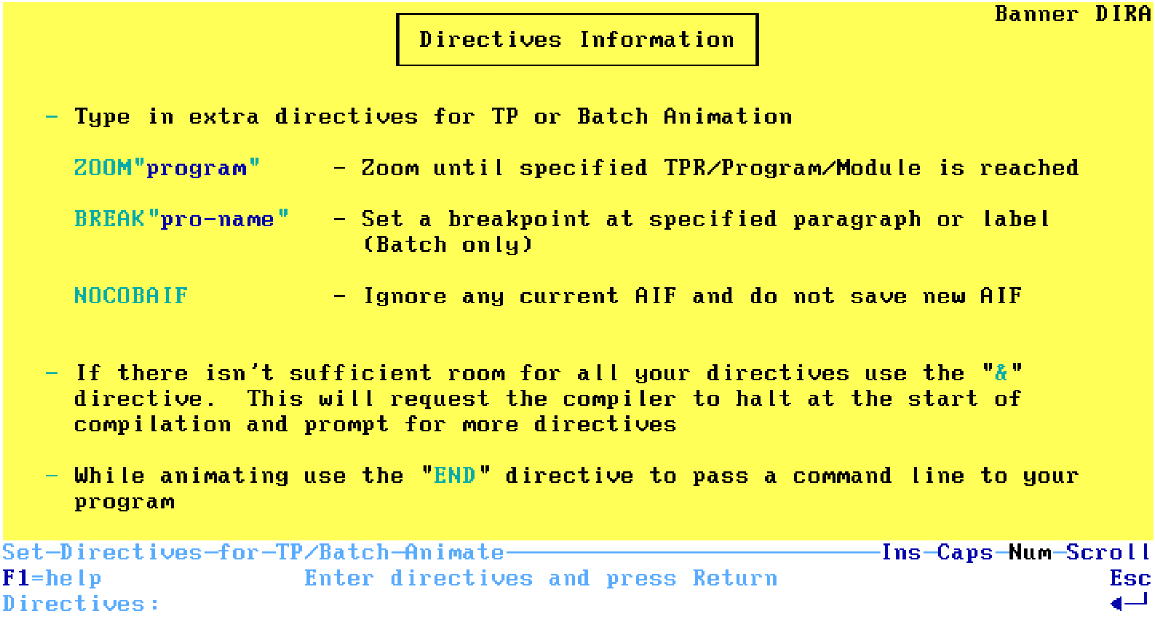
```
Help screen for...                  Animate        Page 1                 Help9007

The Animator shows program execution at a source code level.  Animate only
programs which have been syntax checked using the Checker.  Otherwise, results
are unpredictable.  This also applies to called subprograms.  Programs can be
animated in one of two modes: Zoom or line-by-line.  Zoom animates at full
speed, but does not display the source code.  To exit Zoom and display the
source code during animation press Break.  Zoom, when on, appears underlined
(monochrome monitor) or in a different color (color monitor) at the bottom of
the screen.

To animate a program, key in a file-name and press ◄┘ , use the Directory to
select a file or press ◄┘ to select the displayed filename.

To animate a TP, no file-name is required, simply press ◄┘.

If the source code is date stamped later than the checked code, an error message
appears.  The stamp on checked code must be later than that on source code.


                 press F1 for more help or space bar to return
Animate-TP----------------------------------------------------------------------
F1=help                    F4=zoom F5=strc/anlz                     F9/F10=opt Esc
                                                         ◄┘=start TP
```

If F4-Zoom is selected, no visual animation will take place until:

- You enter Ctrl+Break to stop zooming
- You reach a previously defined Break-point

Otherwise animation will begin in the first statement of your first application TPR.

You also may define an implied break-point through F9/F10-Opt function, so you are able to zoom to a specific application TPR.



```
                                                               Banner DIRA
                           Directives Information

   - Type in extra directives for TP or Batch Animation

     ZOOM"program"       - Zoom until specified TPR/Program/Module is reached

     BREAK"pro-name"     - Set a breakpoint at specified paragraph or label
                           (Batch only)

     NOCOBAIF            - Ignore any current AIF and do not save new AIF


   - If there isn't sufficient room for all your directives use the "&"
     directive.  This will request the compiler to halt at the start of
     compilation and prompt for more directives

   - While animating use the "END" directive to pass a command line to your
     program
Set-Directives-for-TP/Batch-Animate--------------------------Ins-Caps-Num-Scroll
F1=help                Enter directives and press Return                   Esc
Directives:                                                                 ◄┘
```

*Note: If you QUIT the Animator, you are stopping TP Monitor also.*

## Running

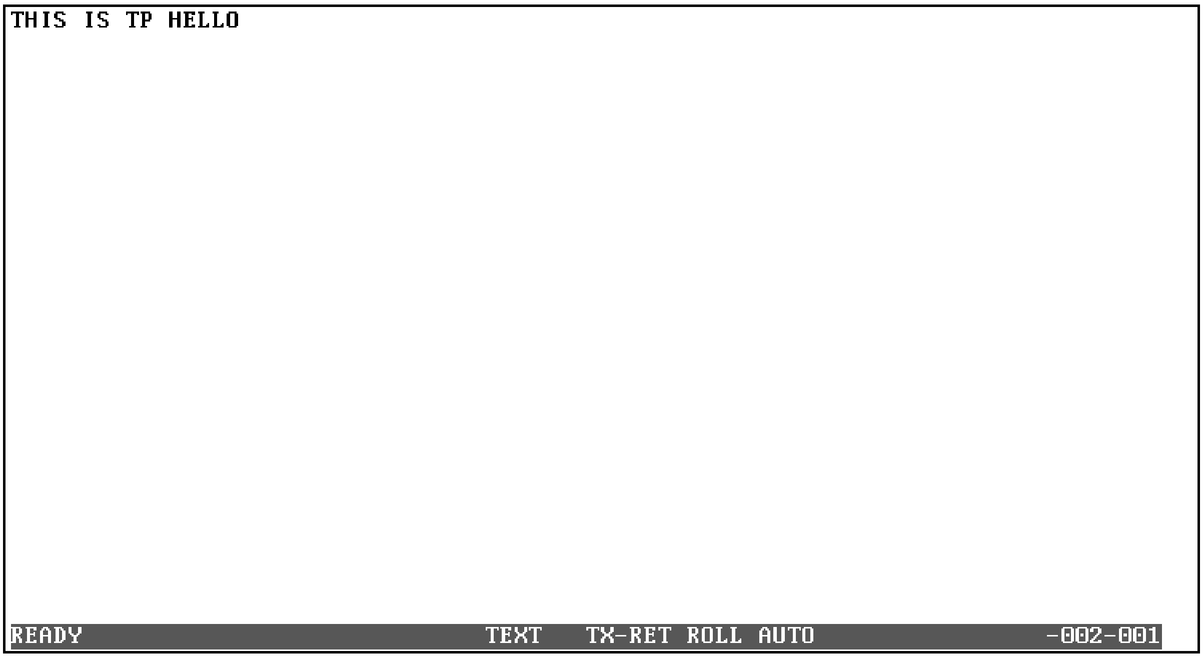
F5-Run function may be used to execute your application TPR's under the TP Monitor. TPR's and modules may be in either INT or GNT object format, however, GNT format provides the most efficient execution.

# Operating

## Terminal Emulation

G8WB may be configured for emulation of various terminal types. Such configuration must be set according to application requirements, ex. terminal control-sequences embedded in application, or according to Forms functionality required, ex. TPFFwith VIP78XX. Please refer to Configuration chapter for details.

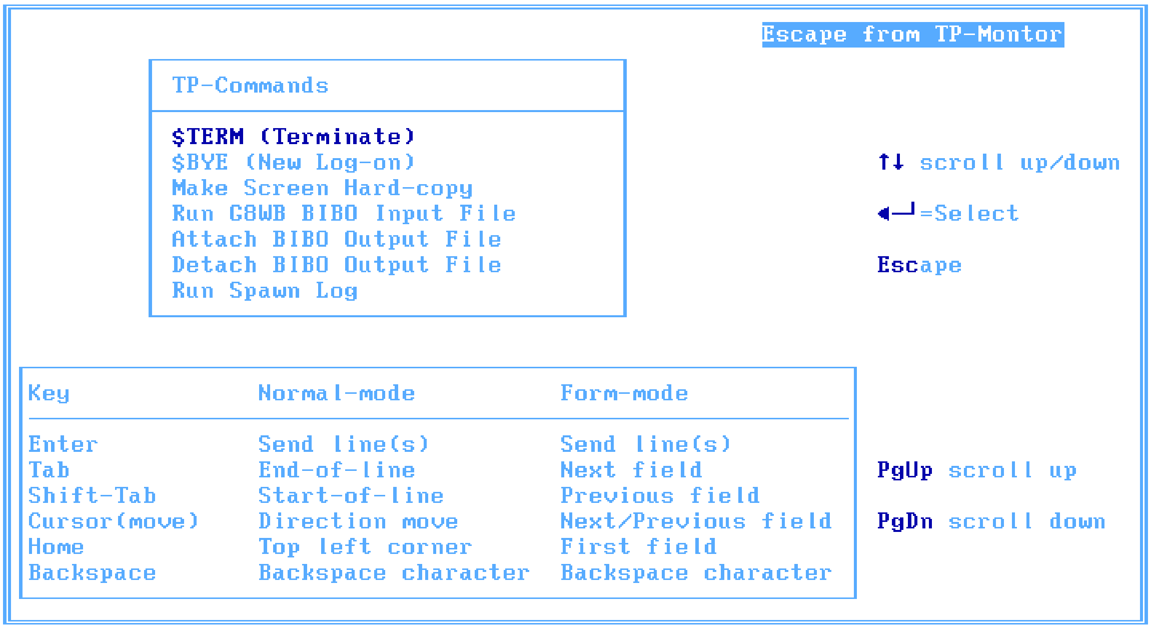
The terminal emulation may not include all features, and keyboard mapping and visual presentation may also be different from a real terminal or other emulators. Following illustrate VIP78XX terminal emulation.



```
THIS IS TP HELLO








READY                         TEXT   TX-RET ROLL AUTO             -002-001
```

## Escape

You may request Escape from any terminal emulators through the Escape key. Escape may be requested at any time while terminal emulator is active, i.e. Input mode, and the original screen is restored when you return from Escape.



```
                                                   Escape from TP-Montor

          TP-Commands

          $TERM (Terminate)
          $BYE (New Log-on)                                   ↑↓ scroll up/down
          Make Screen Hard-copy
          Run G8WB BIBO Input File                            ◄┘=Select
          Attach BIBO Output File
          Detach BIBO Output File                             Escape
          Run Spawn Log


  Key              Normal-mode           Form-mode

  Enter            Send line(s)          Send line(s)
  Tab              End-of-line           Next field                PgUp scroll up
  Shift-Tab        Start-of-line         Previous field
  Cursor(move)     Direction move        Next/Previous field       PgDn scroll down
  Home             Top left corner       First field
  Backspace        Backspace character   Backspace character
```

## Key Mapping

Escape will document all special key mapping for the currently active terminal emulator. You may use PgUp/PgDn keys to browse through key mapping documentation.

## TP Commands

Escape will also document the various TP commands available. You may use ArrowUp/Down keys to browse through TP commands, and Enter key to request execution of a TP command.

### $TERM

May be used to terminate TP Monitor. $TERM may also be typed during terminal emulation.

### $BYE

May be used to log off from TP Monitor and log on again with a new Logical-Id. $BYE may also be typed during terminal emulation.

### $ABORT

If your transaction is in Conversation-mode, you may use $ABORT to abort the current transaction and return to Command-mode.

*Note: When in Conversation-mode, Escape will only offer you a limited number of TP commands.*

### Make hardcopy

May be used to log a hardcopy of the screen (your current application screen) into the hardcopy log file.

### Run BIBO

May be used to start execution of transactions from a BIBO file, ref. below for details.

### Attach BIBO

May be used to attach a BIBO output file and start recording input and output messages from your TP test session, ref. below for details.

### Detach BIBO

May be used to detach a BIBO output file and stop recording input and output messages from your TP test session, ref. below for details.

### Run Spawn

May be used to start execution of transactions in the Spawn log file, ref. below for details.

### Run GCOS-8 BIBO

May be used to start execution of transactions from a GCOS-8 BIBO file, ref. below for details.

### Edit file

May be used to edit a file through MF Editor, i.e. BIBO or other files.

### View Journal

May be used to view the TP Journal through MF Editor.

### View Spawn

May be used to view the Spawn log through MF Editor.

### $*$LC

$*$LC ON may be used to set terminal emulator in upper- and lowercase mode.

$*$LC OFF may be used to set terminal terminate emulator in uppercase mode, i.e. anything you type is converted to uppercase.

$*$LC must be typed during terminal emulation.

### $*$DIS

May be used to terminate TP Monitor.

$*$DIS must be typed during terminal emulation.

## Database

### Commitment

TP Monitor will request database commitment at End-of-Phase and End-of-Transaction, just like on GCOS-8. Also, if a transaction abort, an explicit database rollback is requested. If you QUIT Animator in the middle of a transaction, this is equivalent to a transaction abort.

For INTEREL database integrity control is default, i.e  Backward logging enabled, and all database updates within a transaction phase are thus either committed or rolled back.

For IDS-II, database integrity control is optional/configurable, ref. Configuration chapter for details.

If you have not enabled Integrity control for IDS-II, all database updates are permanent, regardless of whether transaction abort or complete normally.

If you have Integrity control enabled for IDS-II, all database updates are deferred until a commitment is taken, and all database updates within a transaction phase are thus either committed or rolled back.

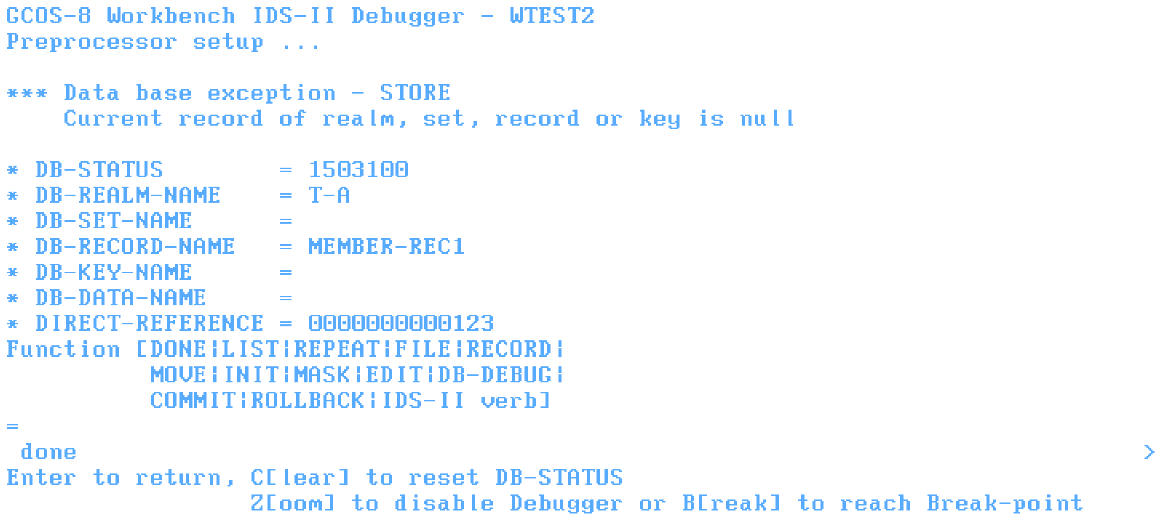
Since you are able to alter the processing sequence via Animator, i.e. restart from the beginning of the TPR, you may want to control database commitment/rollback dynamically. Such a request for commitment/rollback is done via the Animator DO function as follows:

| | |
|---|---|
| **CALL "DB_CMIT"** | force an immediate database commitment |
| **CALL "DB_RBCK"** | force an immediate database rollback |

### Debugger

The dynamic IDS-II Database Debugger, ref. IDS-II Database chapter for details, may be configured so unexpected DB-STATUS values automatically invoke Dynamic Debugger.

Following illustrate Dynamic Debugger invoked due to a “1503100” DB-STATUS:



```
GCOS-8 Workbench IDS-II Debugger - WTEST2
Preprocessor setup ...

*** Data base exception - STORE
    Current record of realm, set, record or key is null

* DB-STATUS         = 1503100
* DB-REALM-NAME     = T-A
* DB-SET-NAME       =
* DB-RECORD-NAME    = MEMBER-REC1
* DB-KEY-NAME       =
* DB-DATA-NAME      =
* DIRECT-REFERENCE = 0000000000123
Function [DONE¦LIST¦REPEAT¦FILE¦RECORD¦
          MOVE¦INIT¦MASK¦EDIT¦DB-DEBUG¦
          COMMIT¦ROLLBACK¦IDS-II verb]
=
 done                                                                          >
Enter to return, C[lear] to reset DB-STATUS
                 Z[oom] to disable Debugger or B[reak] to reach Break-point
```

Once Debugger is invoked, you have the complete Interactive IDS-II Utlity functionality available to support debugging and corrective measures, i.e. establish a missing currency before return to application program.

When Debugger is terminated through Done, you are given various options before returning to the origination application program:

- Clear (reset db-status, only appear if non zero)
- Return (return to execution statement following DML verb)
- Zoom (disable dynamic debugger and return)
- Break (force return to a break-point if animating)

*Note: If you use DB-EXCEPTION procedures and you intend to restart the DML verb, make sure you clear DB-STATUS before returning, otherwise your DB-EXCEPTION procedure may cause your program to abort or stop.*

*Note: Via Break you will return to the origination application program, where you may restart you program anywhere (including re-execution of failed DML verb), using the standard Restart feature of the ANIMATOR.*

During animation, you cannot execute a DML verb via the DO feature of Animator, because a DML verb is not a COBOL statement. You may, however, invoke Dynamic Debugger through the DO feature, and thus execute any DML verb once you are into the Dynamic Debugger. The DO statement to invoke Dynamic Debugger is:

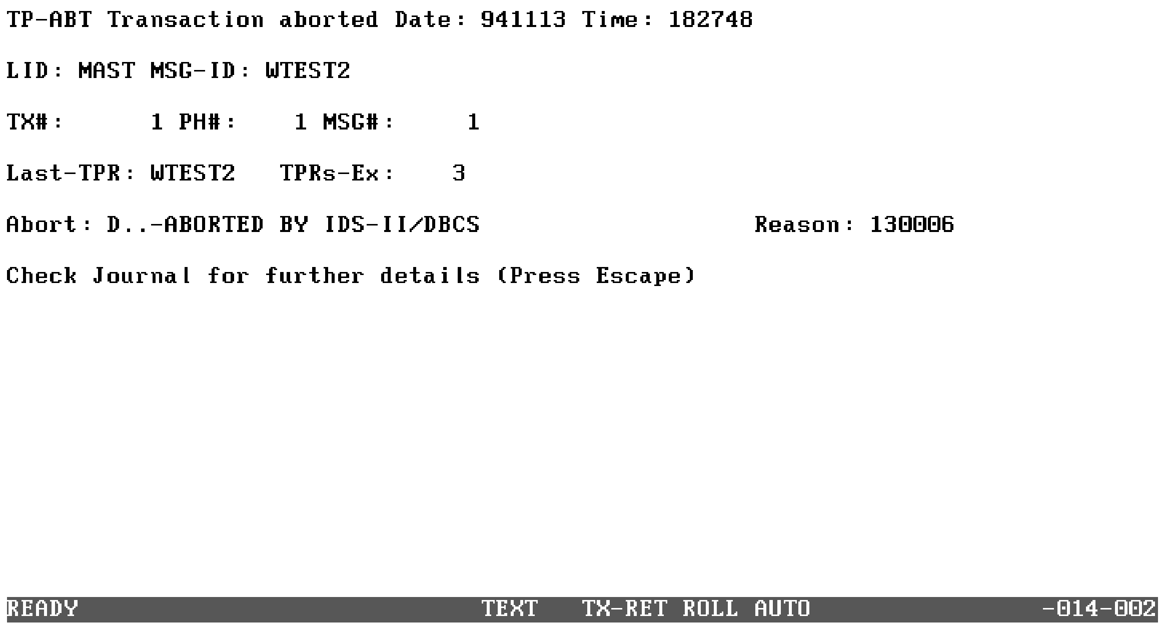
**CALL “DB_<subschema-name>“**

where <subschema-name> is the name of the subschema currently being used.

## TX Abort

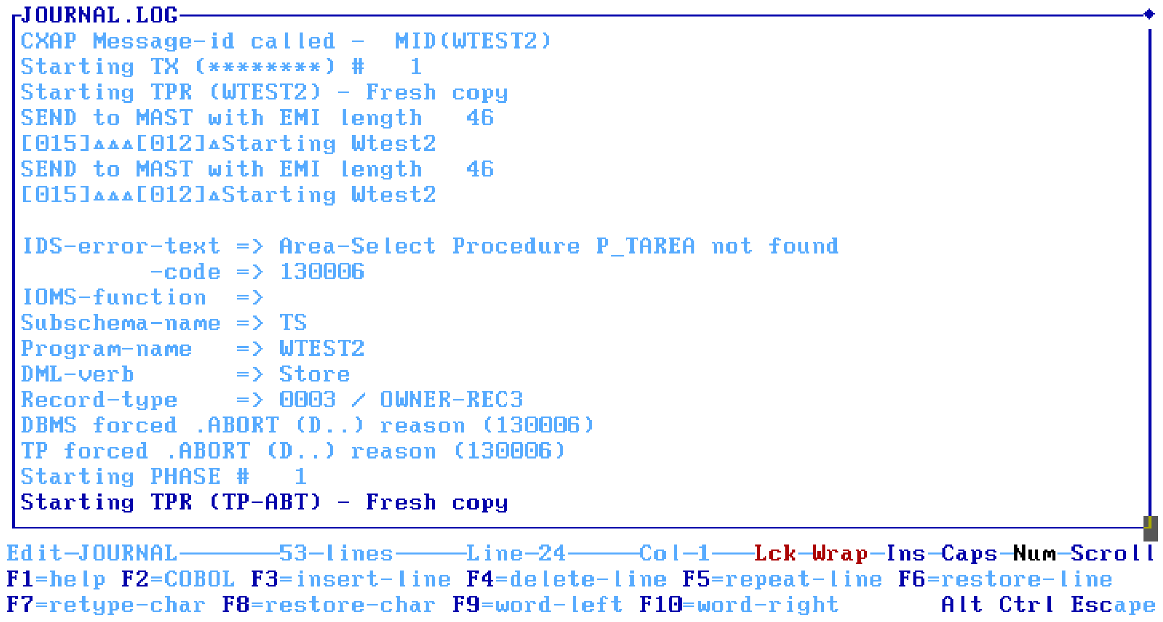
If your transaction abort, the standard TP-ABT abort TPR is invoked, unless you have otherwise configured or enabled your own abort TPR. G8WB will generally use DMIV-TP/TP8 compatible abort codes, and TP-ABT will provide a short explanatory message for each of the standard abort codes.

For system generated abort codes, i.e. W8E, ISE and D<xx>, you need to consult the Journal Log to obtain more detailed information. Following illustrate a fatal DBMS abort and the associated information in the Journal Log.



```
TP-ABT Transaction aborted Date: 941113 Time: 182748

LID: MAST MSG-ID: WTEST2

TX#:       1 PH#:     1 MSG#:     1

Last-TPR: WTEST2   TPRs-Ex:    3

Abort: D..-ABORTED BY IDS-II/DBCS                   Reason: 130006

Check Journal for further details (Press Escape)
```

```
READY                           TEXT   TX-RET ROLL AUTO               -014-002
```



```
JOURNAL.LOG
CXAP Message-id called -  MID(WTEST2)
Starting TX (********) #   1
Starting TPR (WTEST2) - Fresh copy
SEND to MAST with EMI length   46
[015]▲▲▲[012]▲Starting Wtest2
SEND to MAST with EMI length   46
[015]▲▲▲[012]▲Starting Wtest2

IDS-error-text => Area-Select Procedure P_TAREA not found
         -code => 130006
IOMS-function  =>
Subschema-name => TS
Program-name   => WTEST2
DML-verb       => Store
Record-type    => 0003 / OWNER-REC3
DBMS forced .ABORT (D..) reason (130006)
TP forced .ABORT (D..) reason (130006)
Starting PHASE #   1
Starting TPR (TP-ABT) - Fresh copy
```

```
Edit—JOURNAL———————53—lines—————Line—24—————Col—1———Lck—Wrap—Ins—Caps—Num—Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

## TP Journal

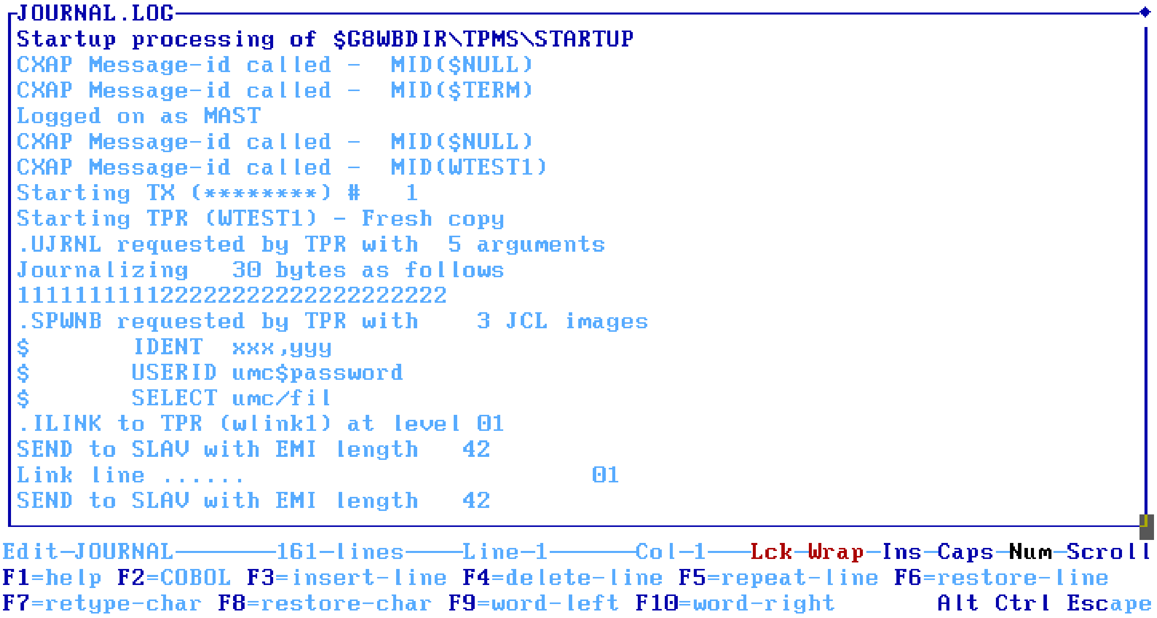
The TP Monitor will continuously write information to a Journal Log file. Although similar in principle to the DMIV-TP or TP8 Journal files, the G8WB journal log holds information in plain text, and it may easily be viewed during or after a test session.

The extent of information may be configured through the Configuration Menu. Three basic types of information are provided:

- Trace of Executive sequence, i.e. Start TX, Start TPR etc
- Trace of Executive interface calls, i.e. Send, .FREER etc.
- Trace of Communication buffers

DBMS debug information, when enabled, will also be logged to G8WB journal.

Following example illustrates the content of the Journal Log.

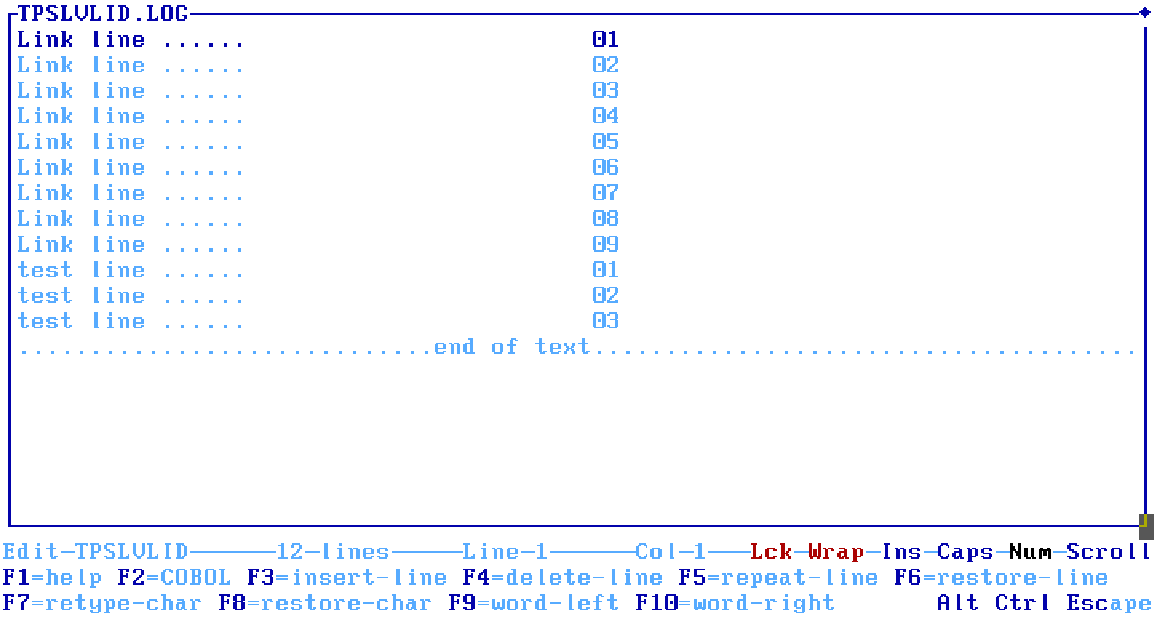


```
JOURNAL.LOG
Startup processing of $G8WBDIR\TPMS\STARTUP
CXAP Message-id called -  MID($NULL)
CXAP Message-id called -  MID($TERM)
Logged on as MAST
CXAP Message-id called -  MID($NULL)
CXAP Message-id called -  MID(WTEST1)
Starting TX (********) #    1
Starting TPR (WTEST1) - Fresh copy
.UJRNL requested by TPR with  5 arguments
Journalizing   30 bytes as follows
111111111122222222222222222222
.SPWNB requested by TPR with    3 JCL images
$        IDENT   xxx,yyy
$        USERID umc$password
$        SELECT umc/fil
.ILINK to TPR (wlink1) at level 01
SEND to SLAV with EMI length   42
Link line ......                       01
SEND to SLAV with EMI length   42
Edit─JOURNAL─────────161─lines─────Line─1───────Col─1───Lck─Wrap─Ins─Caps─Num─Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

## TP Slave Messages

The DMIV-TP and TP8 concept of a Slave Terminal is emulated by G8WB with the Slave Log file. Any send operation to the slave will be directed to the Slave Log file, which may easily be viewed or printed after a test session.

Following example illustrate the content of the Slave Log file.



```
TPSLVLID.LOG
Link line ......                       01
Link line ......                       02
Link line ......                       03
Link line ......                       04
Link line ......                       05
Link line ......                       06
Link line ......                       07
Link line ......                       08
Link line ......                       09
test line ......                       01
test line ......                       02
test line ......                       03
.............................end of text.....................................
Edit─TPSLVLID────────12─lines──────Line─1───────Col─1───Lck─Wrap─Ins─Caps─Num─Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

## TP Receive-only Printer Messages

The DMIV-TP and TP8 concept of Receive-Only printers is emulated by G8WB with the ROP Log file. Any send operation to a ROP printer will be logged to the ROP Log file. If destinations are involved, the messages will be intermixed and you will not be able to determine the individual destinations from the

ROP Log file, however the Journal Log file will indicate this. The ROP Log file may easily be viewed or printed after a test session.

## TP Spawn Messages

The DMIV-TP and TP8 concept of Spawn Destinations and messages is emulated by G8WB with the Spawn Log file. Any send operation to a Spawn destination or .SWTM request will be logged to the SPAWN Log file, except for .SWTM to originator, which is treated as next transaction (automatic execution).

The TP Monitor emulates a single-user environment and is thus not able to automatically execute spawned transactions. Once you have finished your current transaction, you may however execute transactions from the Spawn Log. The Spawn Log is treated like a BIBO file, ref. below, and each individual transaction will be executed under the spawn destination LID. You may also view the Spawn Log during or after a test session.

*Note: G8WB does not provide direct support for Background-Drivers, i.e. spawned transaction with sleep. However, once such a driver transaction is spawned, you are able to run it using the Spawn Log. Sleep requests are ignored, so the transaction will execute probably indefinitely, unless you provide some way of stopping it, so you are able to return to normal transaction processing.*

## Hardcopy

The Hardcopy Log feature is an alternative or a complement to printed hard copies. During a test session, you may request a hardcopy of the current application. These hard copies, which consist of a Form-feed followed by an image of the 24 terminal lines, are written to a file named HARDCOPY.LOG in your current path. The Hardcopy Log may be viewed or printed after a test session.

## BIBO

The BIBO feature is based on BIBO text files with a pre-defined format. BIBO files may either be generated though Recording, Extract from GCOS-8, Transaction spawn or created manually. BIBO files may contain both input and output message, however when you run a BIBO file, only input messages are used.

### Input Message format

The format of an input message is as follows:

**<header><detail>[<new-line><detail>]<trailer>**

<header> identifies the type of input, i.e. Initial or Conversational, and for initial type, the origination Logical-Id.

**[I,IIII]**

**[R]**

<trailer> identifies the end of the message.

**[<]**

<detail> is the actual input message content, specified as a string of of ASCII TEXT characters. Non-TEXT characters must be specified through it’s OCTAL values as follows:

**[nnn]**

i.e. [010] is HT, [003] is ETX.

<new-line> is used if the input message exceed 160 characters, i.e a text line. This condition must be signaled at the end of the text line as follows:

**[>]**

This condition must also be signaled at the beginning of the next text line as follows:

**[C]**

<new-line> may be specified anywhere within <detail>. The 160 characters line length designate when a <new-line> is mandatory. Following illustrate examples of input messages:

**[I,MAST]mid[003] [<]**

**[R]field-1[010]field-2[010][010][003] [<]**

**[R]field-1[010]field-2[>]**

**[C][010][010][003] [<]**

## Output Message format

The format of an output message is as follows:

**<header><detail>[<new-line><detail>]<trailer>**

<header> identifies the start of the output message and the destination Logical-Id.

**[O,llll]**

<trailer> identifies the end of the message.

**[<]**

<detail> is the actual output message content, specified as a string of of ASCII TEXT characters. Non-TEXT characters are specified through it’s OCTAL values as follows:

**[nnn]**

i.e. [033] is ESCAPE, [003] is ETX.

<new-line> is used if the output message exceed 160 characters, i.e a text line. This condition is signaled at the end of the text line as follows:

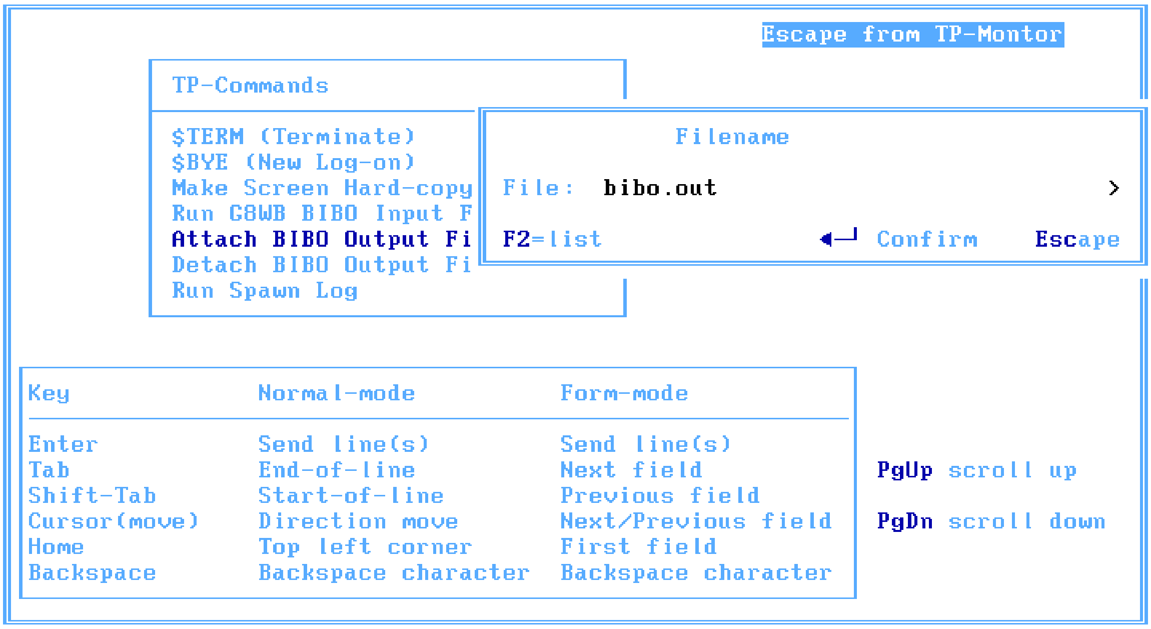
**[>]**

This condition is also signaled at the beginning of the next text line as follows:

**[C]**

## Recording

When you attach a BIBO output file, all input and output messages from your test session are recorded on to the attached file, until you detach BIBO output file or terminate your test session.



```
                                                 Escape from TP-Montor

          TP-Commands

          $TERM (Terminate)       Filename
          $BYE (New Log-on)
          Make Screen Hard-copy  File:  bibo.out                            >
          Run G8WB BIBO Input F
          Attach BIBO Output Fi  F2=list                   ◄┘ Confirm   Escape
          Detach BIBO Output Fi
          Run Spawn Log

Key              Normal-mode           Form-mode

Enter            Send line(s)          Send line(s)
Tab              End-of-line           Next field              PgUp scroll up
Shift-Tab        Start-of-line         Previous field
Cursor(move)     Direction move        Next/Previous field     PgDn scroll down
Home             Top left corner       First field
Backspace        Backspace character   Backspace character
```

The BIBO output file may subsequently be used as a BIBO input file, thus allowing you to re-process a test suite of transactions after correction your application programs.

If you attach another BIBO output file during re-process, then you are able to compare the two test suites via MF DIFF utility. Such comparison will document any changes in output messages between the two executions of the test suite.

## Running

Running a BIBO input file will suspend interactive terminal emulation and enter into non-interactive execution of all transactions of the BIBO input file. Each BIBO tranaction will be executed under the originator Logical-Id specified in each input message. If BIBO input messages get out of synchronization, i.e. Initial input message when in conversational-mode or reverse, TP Monitor will force a $ABORT as required, to resynchronize. Synchronization errors are documented in the Journal Log.

## Spawn

Spawn Log is treated like a BIBO input file.

## GCOS-8

G8WB provides host utilities to generate BIBO input files from DMIV-TP or TP8 journals, ref. User Guide for details.

Running a GCOS-8 BIBO input file differ from a normal BIBO input files in the following areas.

### Spawn to originator

.SWTM to originator is ignored, because the spawned transaction input message will immediately followed current transaction in the BIBO input file.

### Character-Set-Adjust

No Character-Set-Adjust, if applicable, has been done to the GCOS-8 BIBO input file. As a result input and output messages will contain host representation of national characters. This may or may not represent a problem for your application, i.e. national characters used in key data would result in failure, because your database and application will work with G8WB representation.

For output message logged, i.e. BIBO output file attached, a reverse character-set-adjust will take place, i.e. host representation used, so you are better able to compare output from GCOS-8 with output from G8WB.

### Seuquencing

When building GCOS-8 BIBO files from DMIV-TP or TP8 journals, you may specify how messages are to be sequenced. A a minimum, all messages pertaining to a conversational transaction must be sequenced together, so they are executed in the proper sequence. If journal contain messages from multiple logical-id's, you may either sequence according to time of execution or according to logical-id and time of execution, i.e. all transactions for one logical-id is grouped together. Grouping within logical-id is relevant if you require a log-on for each lid.

*Note: To successfully re-process a GCOS-8 BIBO file, you need to consider sequencing and take special care that transaction phases between logical-id are NOT dependent, i.e. updates in one TX phase is used in another TX phase run semi-concurrently.*

## ALT Support Menu

Pressing ALT key from TP Development Menu will provide you with an alternate menu of support functions. As soon as you release the ALT key, you return to the normal menu again.

ALT Support Menu is described in the Batch Development chapter.

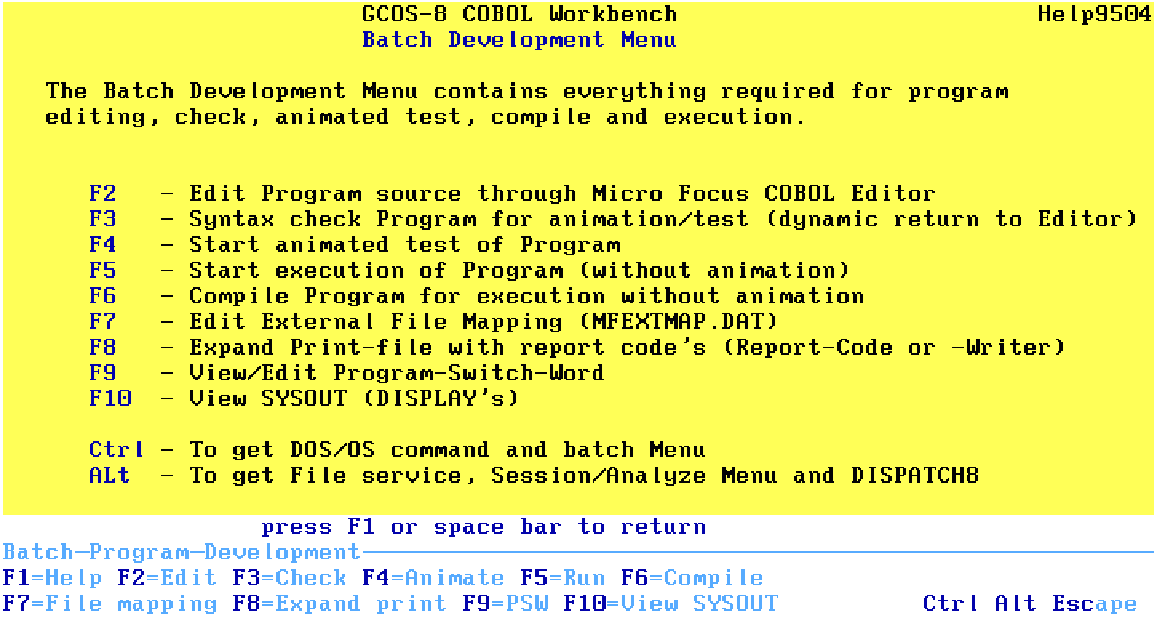
## CTRL Support Menu

Pressing CTRL key from TP Development Menu will provide you with an alternate menu of support functions. As soon as you release the CTRL key, you return to the normal menu again.

CTRL Support Menu is described in the Batch Development chapter.

# Chapter 3 BATCH DEVELOPMENT MENU

The Batch development menu is your place of work for checking, editing and testing of batch application programs.



```
                            GCOS-8 COBOL Workbench                         Help9504
                            Batch Development Menu

   The Batch Development Menu contains everything required for program
   editing, check, animated test, compile and execution.


      F2    - Edit Program source through Micro Focus COBOL Editor
      F3    - Syntax check Program for animation/test (dynamic return to Editor)
      F4    - Start animated test of Program
      F5    - Start execution of Program (without animation)
      F6    - Compile Program for execution without animation
      F7    - Edit External File Mapping (MFEXTMAP.DAT)
      F8    - Expand Print-file with report code's (Report-Code or -Writer)
      F9    - View/Edit Program-Switch-Word
      F10   - View SYSOUT (DISPLAY's)

      Ctrl  - To get DOS/OS command and batch Menu
      ALt   - To get File service, Session/Analyze Menu and DISPATCH8

                  press F1 or space bar to return
Batch-Program-Development---------------------------------------------------------
F1=Help F2=Edit F3=Check F4=Animate F5=Run F6=Compile
F7=File mapping F8=Expand print F9=PSW F10=View SYSOUT          Ctrl Alt Escape
```

## Developing

The compile process for G8WB is accomplished is two steps:

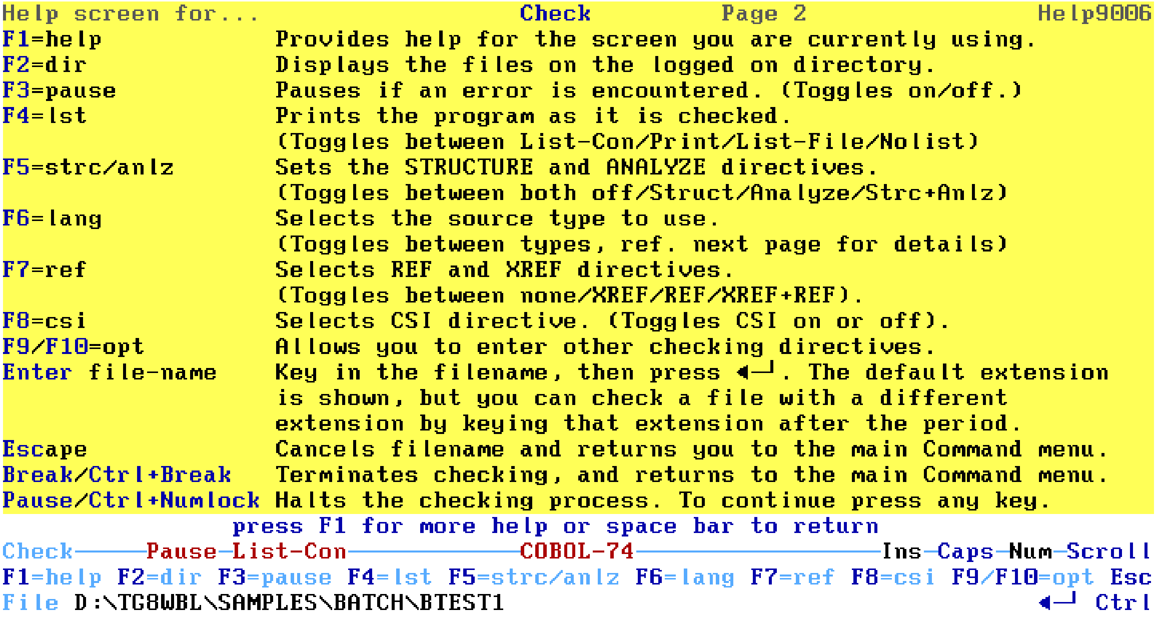
- Checking

- Generating

The checking process represents the traditional COBOL compilation up to the point of object code generation. Once checked, a program may be execution both with or without animation, from the .INT version generated by the MFWB Checker.

### Checking

Checking menu may be entered through F3-Checking from the Batch Development menu. Checking menu provides a number of toggle switches, which may be changed by pressing the appropriate function key. Toggle switch settings are remembered as long as you do not exit G8WB. Initial toggle switch settings may be configured, ref. Configuration chapter for details. Following illustrate use the Checker menu with F1-Help enabled:



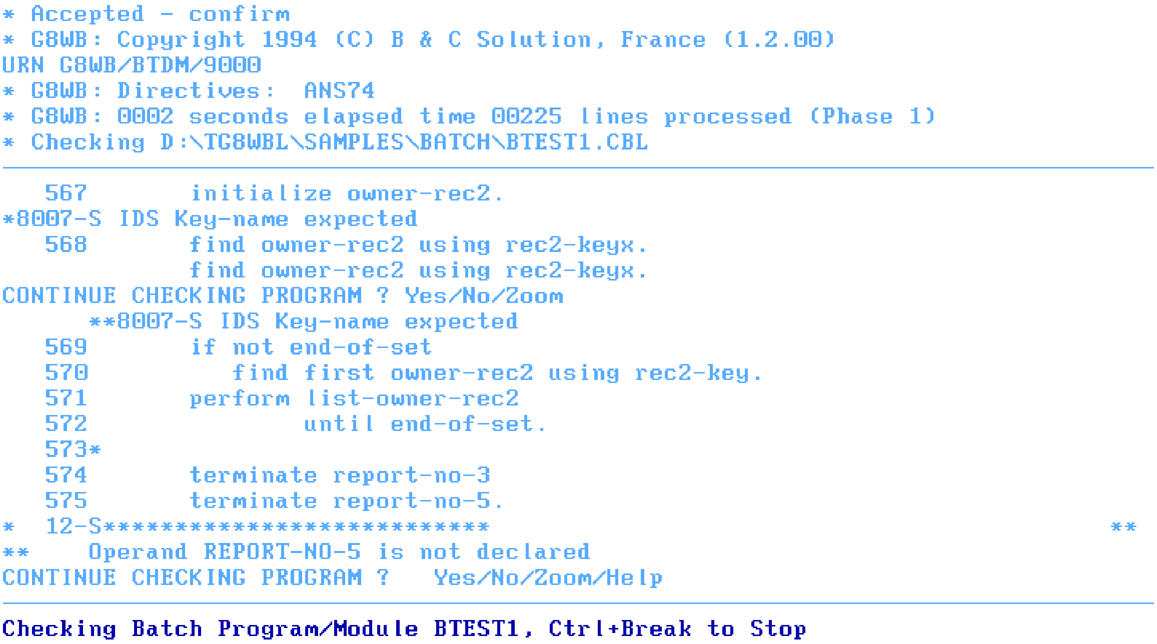
```
Help screen for...                    Check          Page 2                Help9006
F1=help              Provides help for the screen you are currently using.
F2=dir               Displays the files on the logged on directory.
F3=pause             Pauses if an error is encountered. (Toggles on/off.)
F4=lst               Prints the program as it is checked.
                     (Toggles between List-Con/Print/List-File/Nolist)
F5=strc/anlz         Sets the STRUCTURE and ANALYZE directives.
                     (Toggles between both off/Struct/Analyze/Strc+Anlz)
F6=lang              Selects the source type to use.
                     (Toggles between types, ref. next page for details)
F7=ref               Selects REF and XREF directives.
                     (Toggles between none/XREF/REF/XREF+REF).
F8=csi               Selects CSI directive. (Toggles CSI on or off).
F9/F10=opt           Allows you to enter other checking directives.
Enter file-name      Key in the filename, then press ◄┘. The default extension
                     is shown, but you can check a file with a different
                     extension by keying that extension after the period.
Escape               Cancels filename and returns you to the main Command menu.
Break/Ctrl+Break     Terminates checking, and returns to the main Command menu.
Pause/Ctrl+Numlock Halts the checking process. To continue press any key.
                  press F1 for more help or space bar to return
Check─────Pause-List-Con────────────COBOL-74─────────────────Ins-Caps-Num-Scroll
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
File D:\TG8WBL\SAMPLES\BATCH\BTEST1                                      ◄┘ Ctrl
```

Please refer to MFWB documentation for details on toggle switches.

## Error handling

Once invoked, Checker and G8WB Preprocessor will output error messages with associated source lines in the checking window. G8WB information lines document the source language and type, the number of lines checked and the two checking phases. Phase 1 is G8WB preprocessing and Phase 2 is checking of preprocessed program through COBOL/2.

Checker will normally stop and prompt for continuation, when errors are detected, i.e. Pause toggle set. Following illustrate a checking with two errors detected.



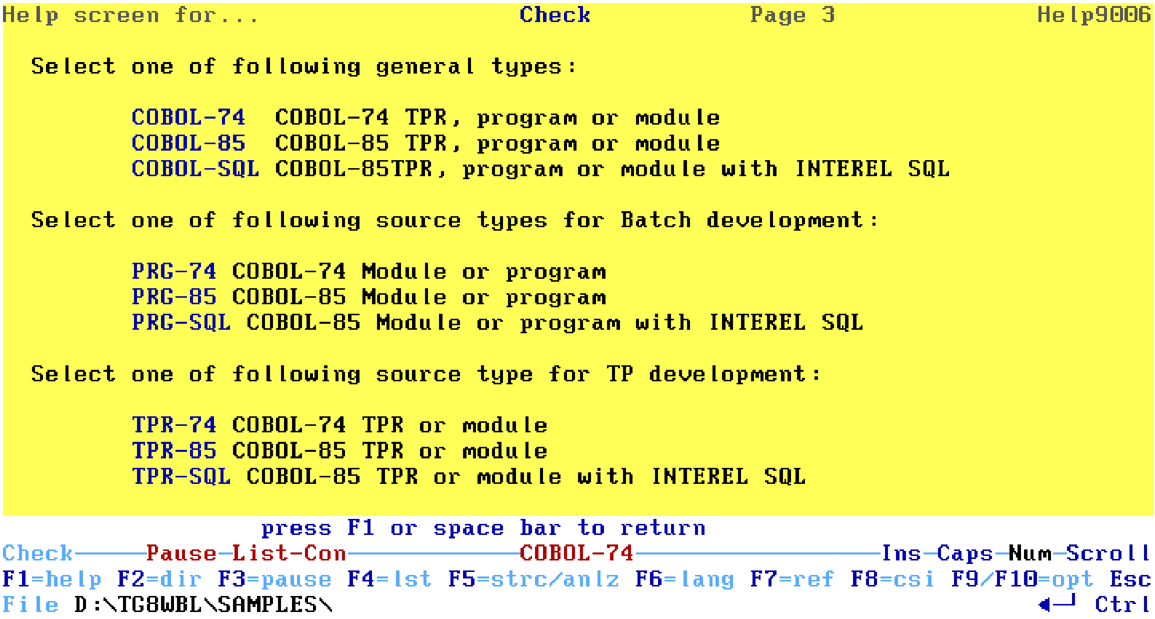
```
* Accepted - confirm
* G8WB: Copyright 1994 (C) B & C Solution, France (1.2.00)
URN G8WB/BTDM/9000
* G8WB: Directives:  ANS74
* G8WB: 0002 seconds elapsed time 00225 lines processed (Phase 1)
* Checking D:\TG8WBL\SAMPLES\BATCH\BTEST1.CBL
────────────────────────────────────────────────────────────────────────────────
   567        initialize owner-rec2.
*8007-S IDS Key-name expected
   568        find owner-rec2 using rec2-keyx.
              find owner-rec2 using rec2-keyx.
CONTINUE CHECKING PROGRAM ? Yes/No/Zoom
      **8007-S IDS Key-name expected
   569        if not end-of-set
   570           find first owner-rec2 using rec2-key.
   571        perform list-owner-rec2
   572                until end-of-set.
   573*
   574        terminate report-no-3
   575        terminate report-no-5.
*  12-S***************************                                            **
**    Operand REPORT-NO-5 is not declared
CONTINUE CHECKING PROGRAM ?   Yes/No/Zoom/Help
────────────────────────────────────────────────────────────────────────────────
Checking Batch Program/Module BTEST1, Ctrl+Break to Stop
```

When prompted, you may use NO to stop checking, YES to continue until next error or ZOOM to finish checking without further prompting. Once checking completes, Editor is automatically invoked when errors are detected, ref. below.

*Note: Preprocessor and Checker issue similar prompts, however your answer is only recognized by the issuer. As a result you may have to ZOOM twice, if both Preprocessor and Checker errors are* *detected. Pause toggle is only recognized by Checker, so Preprocessor may still prompt you even though Pause toggle is unset.*

## Language

F6-Language may be used to specify the COBOL language dialect to be used during checking.



```
Help screen for...                    Check           Page 3              Help9006

  Select one of following general types:

         COBOL-74  COBOL-74 TPR, program or module
         COBOL-85  COBOL-85 TPR, program or module
         COBOL-SQL COBOL-85TPR, program or module with INTEREL SQL

  Select one of following source types for Batch development:

         PRG-74 COBOL-74 Module or program
         PRG-85 COBOL-85 Module or program
         PRG-SQL COBOL-85 Module or program with INTEREL SQL

  Select one of following source type for TP development:

         TPR-74 COBOL-74 TPR or module
         TPR-85 COBOL-85 TPR or module
         TPR-SQL COBOL-85 TPR or module with INTEREL SQL

                    press F1 or space bar to return
Check─────Pause─List─Con────────────COBOL-74─────────────────Ins─Caps─Num─Scroll
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
File D:\TG8WBL\SAMPLES\                                                  ◄┘ Ctrl
```

Since Micro Focus COBOL/2 compiler system does not directly support GCOS-8 specific COBOL-74 and COBOL-85 language adaptions, G8WB will invoke a GCOS-8 specific preprocessor during checking. The G8WB Preprocessor is, however, integrated with the Checker, so the checking process will be viewed as one single activity.

### COBOL-74

May be used for any type of COBOL-74 source, i.e. module, program, TPR.

### COBOL-85

May be used for any type of COBOL-85 source, i.e. module, program, TPR.

### COBOL-SQL

May be used for any type of COBOL-85 source with embedded SQL (INTEREL), i.e. module, program, TPR. COBOL-SQL will trigger SQL preprocessing through XDB Preprocessor.

### PRG-74

Like COBOL-74, however Preprocessing will enforce additional non-TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

### PRG-85

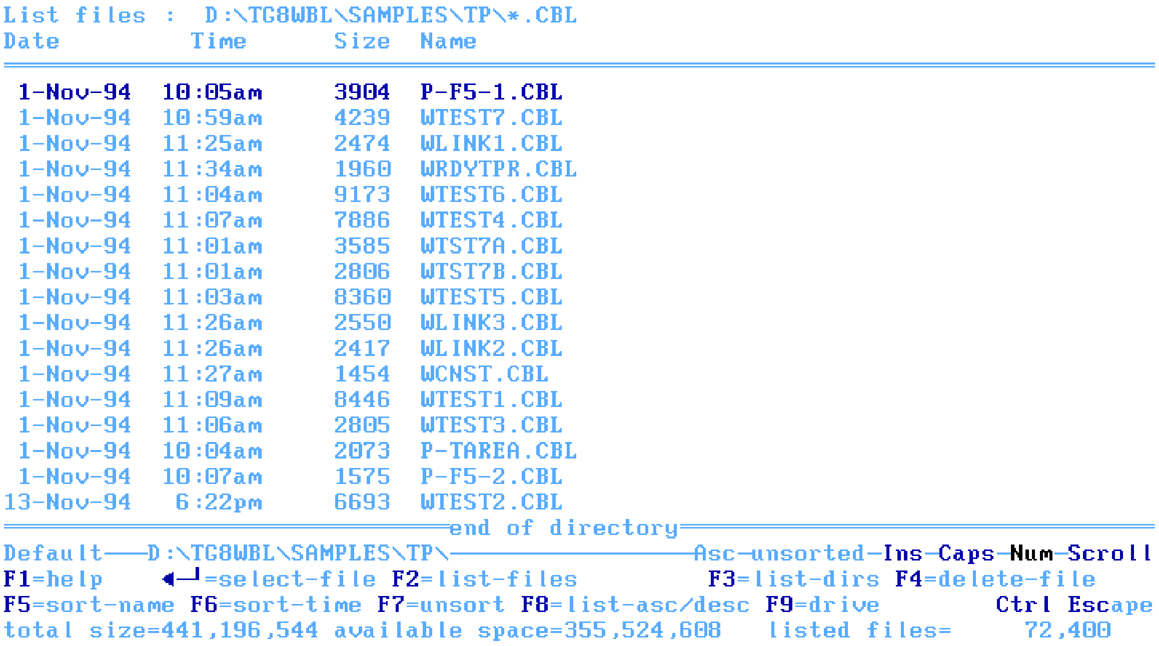
Like COBOL-85, however Preprocessing will enforce additional non-TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

### PRG-SQL

Like COBOL-SQL, however Preprocessing will enforce additional non-TPR specific syntax checking, which will cause flagging during checking, as opposed to failure during testing.

## Directory

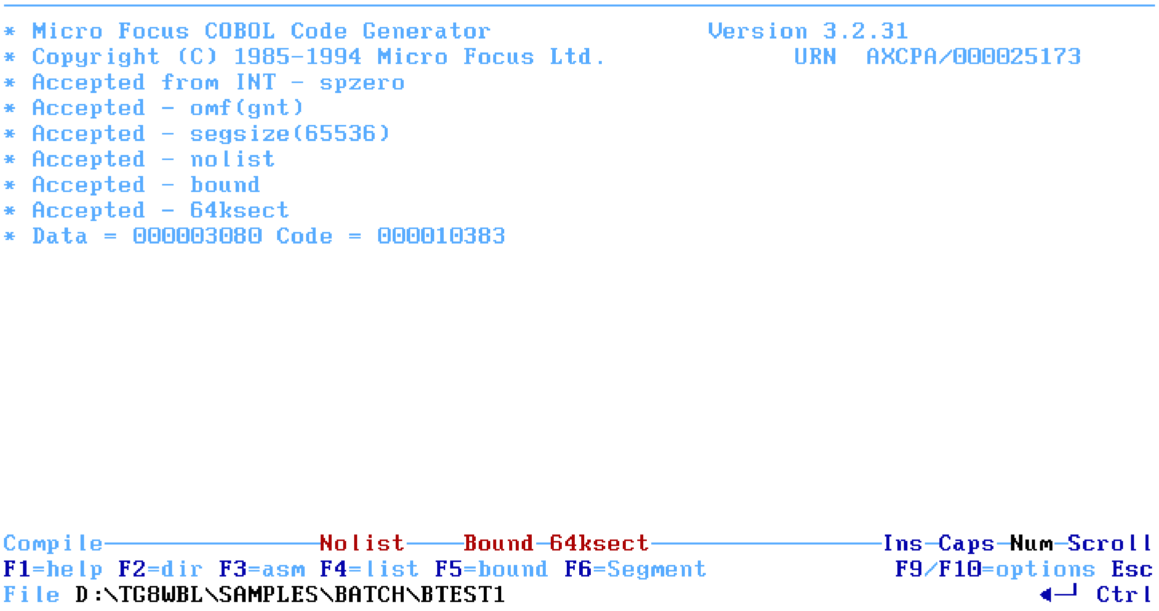
The F2-Dir function allows you to retrieve source components through directory browsing. The selected source will become default for Checker, Editor, Animator and Generator.



```
List files :  D:\TG8WBL\SAMPLES\TP\*.CBL
Date          Time       Size  Name
══════════════════════════════════════════════════════════════════════════════════

 1-Nov-94   10:05am      3904  P-F5-1.CBL
 1-Nov-94   10:59am      4239  WTEST7.CBL
 1-Nov-94   11:25am      2474  WLINK1.CBL
 1-Nov-94   11:34am      1960  WRDYTPR.CBL
 1-Nov-94   11:04am      9173  WTEST6.CBL
 1-Nov-94   11:07am      7886  WTEST4.CBL
 1-Nov-94   11:01am      3585  WTST7A.CBL
 1-Nov-94   11:01am      2806  WTST7B.CBL
 1-Nov-94   11:03am      8360  WTEST5.CBL
 1-Nov-94   11:26am      2550  WLINK3.CBL
 1-Nov-94   11:26am      2417  WLINK2.CBL
 1-Nov-94   11:27am      1454  WCNST.CBL
 1-Nov-94   11:09am      8446  WTEST1.CBL
 1-Nov-94   11:06am      2805  WTEST3.CBL
 1-Nov-94   10:04am      2073  P-TAREA.CBL
 1-Nov-94   10:07am      1575  P-F5-2.CBL
13-Nov-94    6:22pm      6693  WTEST2.CBL
═══════════════════════════════end of directory═══════════════════════════════════
Default──D:\TG8WBL\SAMPLES\TP\─────────────────Asc─unsorted─Ins─Caps─Num─Scroll
F1=help    ◄┘=select-file F2=list-files         F3=list-dirs F4=delete-file
F5=sort-name F6=sort-time F7=unsort F8=list-asc/desc F9=drive        Ctrl Escape
total size=441,196,544 available space=355,524,608   listed files=     72,400
```

## Generating

Once the testing activity has been completed for an application component, you may generate your module, program or TPR into the more efficient GNT object format. Besides being more efficient, GNT's also prevent animation of the source. You should thus use GNT format for application components, where you have completed testing, i.e. only components under test are animated.

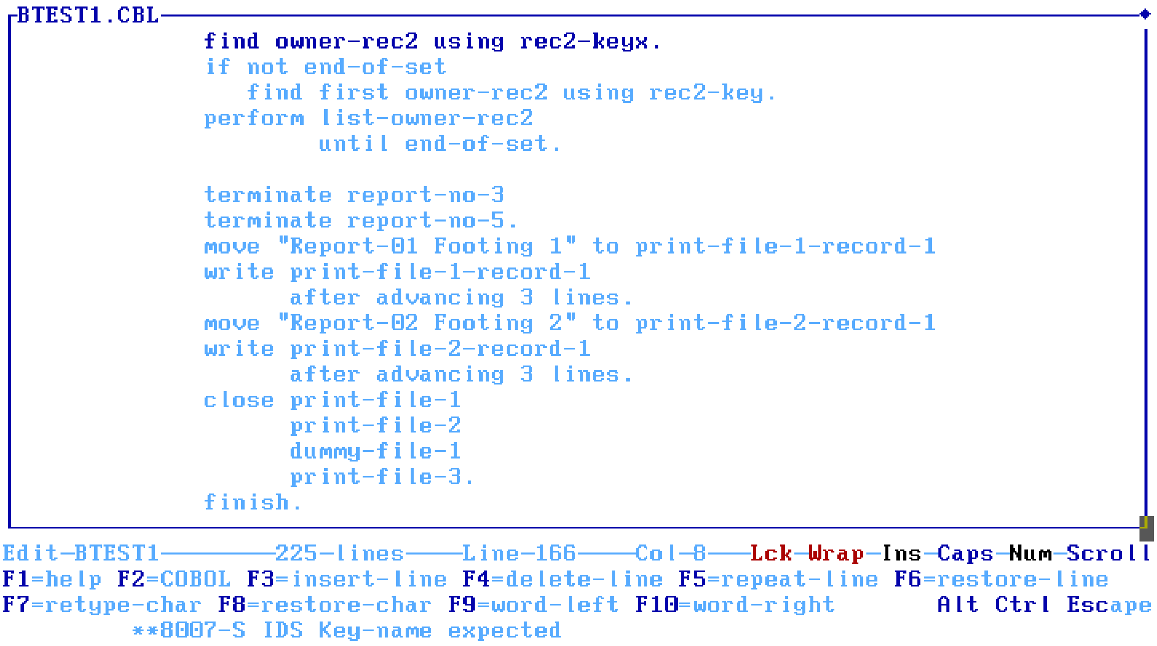


```
──────────────────────────────────────────────────────────────────────────────────
* Micro Focus COBOL Code Generator                 Version 3.2.31
* Copyright (C) 1985-1994 Micro Focus Ltd.               URN   AXCPA/000025173
* Accepted from INT - spzero
* Accepted - omf(gnt)
* Accepted - segsize(65536)
* Accepted - nolist
* Accepted - bound
* Accepted - 64ksect
* Data = 000003080 Code = 000010383








Compile───────────────Nolist────Bound─64ksect────────────────Ins─Caps─Num─Scroll
F1=help F2=dir F3=asm F4=list F5=bound F6=Segment               F9/F10=options Esc
File D:\TG8WBL\SAMPLES\BATCH\BTEST1                                        ◄┘ Ctrl
```

## Editing

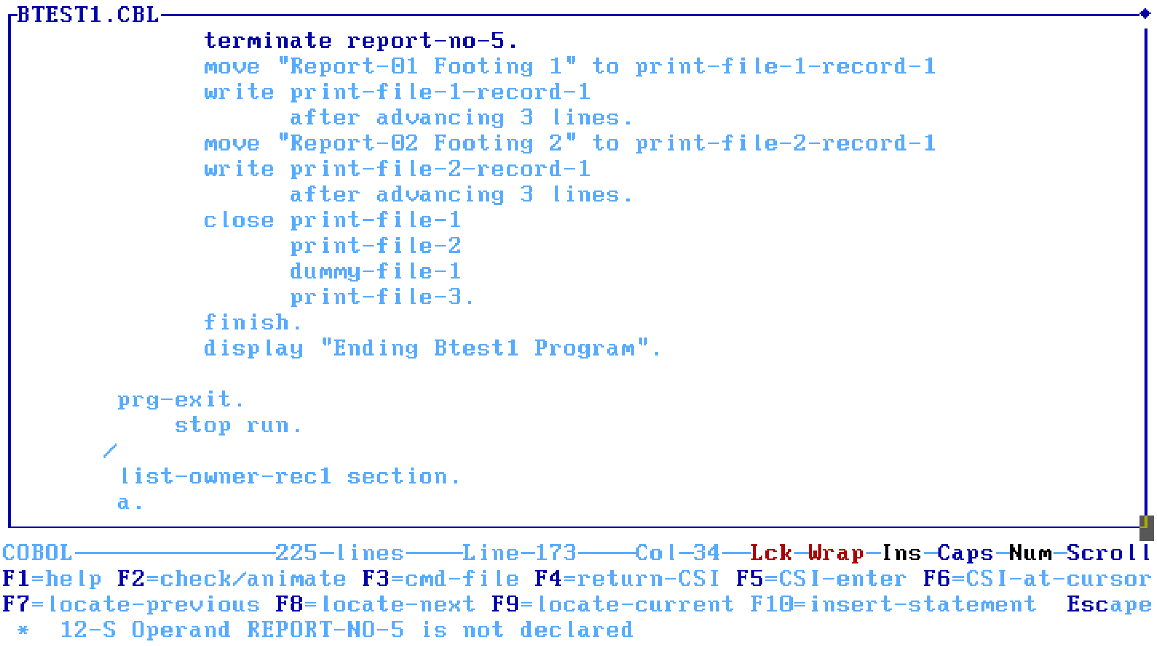
This documentation assume you are utilizing the MFWB Editor, however other editors may be used also. Please refer to Configuration chapter for details.

As mentioned above, Checker will automatically invoke Editor when errors are detected. Normally the source program you are checking will be loaded, however if errors occur in COPY members, both the source program and the COPY member will be loaded. Following illustrate Editor after the checking example illustrated above:



```
BTEST1.CBL
              find owner-rec2 using rec2-keyx.
              if not end-of-set
                 find first owner-rec2 using rec2-key.
              perform list-owner-rec2
                      until end-of-set.

              terminate report-no-3
              terminate report-no-5.
              move "Report-01 Footing 1" to print-file-1-record-1
              write print-file-1-record-1
                    after advancing 3 lines.
              move "Report-02 Footing 2" to print-file-2-record-1
              write print-file-2-record-1
                    after advancing 3 lines.
              close print-file-1
                    print-file-2
                    dummy-file-1
                    print-file-3.
              finish.
Edit-BTEST1--------225-lines-----Line-166-----Col-8----Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right        Alt Ctrl Escape
         **8007-S IDS Key-name expected
```

Editor will automatically position to the source line where the first error is detected and give you the error message at the bottom of the screen. After correcting your source, you may continue positioning to the next error through F2-COBOL and F8-Locate-next.



```
BTEST1.CBL
              terminate report-no-5.
              move "Report-01 Footing 1" to print-file-1-record-1
              write print-file-1-record-1
                    after advancing 3 lines.
              move "Report-02 Footing 2" to print-file-2-record-1
              write print-file-2-record-1
                    after advancing 3 lines.
              close print-file-1
                    print-file-2
                    dummy-file-1
                    print-file-3.
              finish.
              display "Ending Btest1 Program".

        prg-exit.
            stop run.
       /
        list-owner-rec1 section.
        a.
COBOL--------------225-lines-----Line-173-----Col-34---Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=check/animate F3=cmd-file F4=return-CSI F5=CSI-enter F6=CSI-at-cursor
F7=locate-previous F8=locate-next F9=locate-current F10=insert-statement  Escape
 *  12-S Operand REPORT-NO-5 is not declared
```

Once you have completed error browsing and correction of source, you need to Escape out of F2-COBOL, and then resave your source through ALT-F4-Save-file.

When COPY members are involved, error browsing will automatically open and close COPY members as you browse through the errors. Once you have made your source correction, you need to resave your COPY member like you do with your source. When you Escape out of F2-COBOL, i.e. to resave, you may return and continue error browsing again through F2-COBOL. If you finish up in a COPY member, then Escape will take you back to the main source.

Once you have completed correction of source, you may use Escape to return to the Batch development menu, where you can request checking again.

If you have customized the Editor, ref. User Guide for details, you may also stay within Editor and start checking and/or animation through F2-COBOL and F2-Check/animate.

# Testing

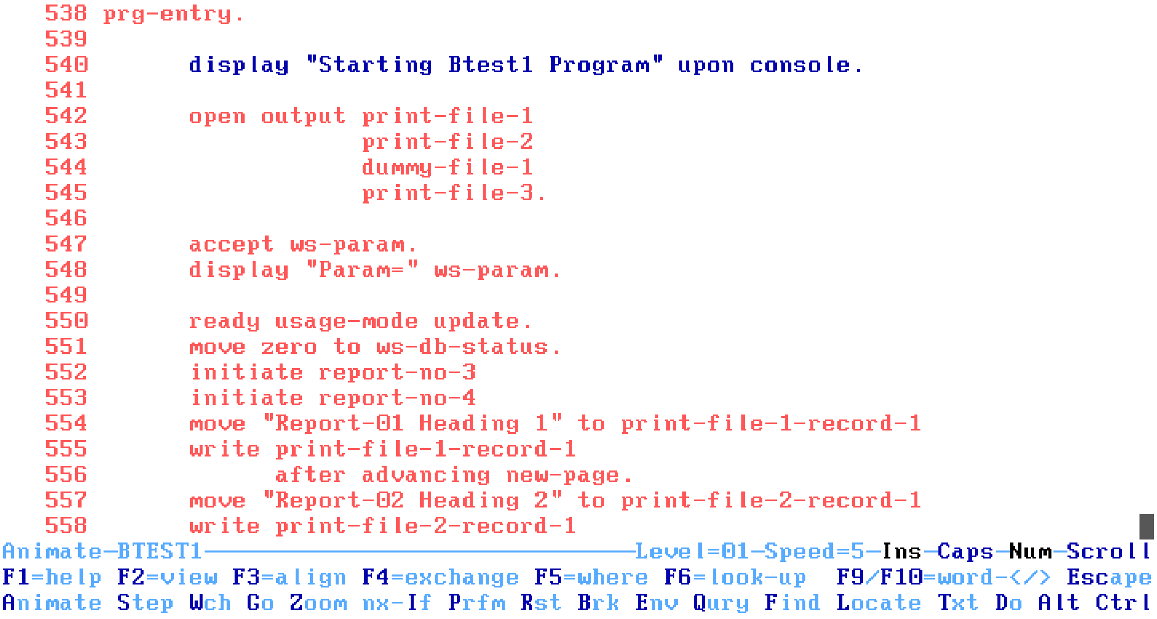
In the GCOS-8 environment you are required to build JCL and submit a job to actually test a program. With G8WB, no JCL and submission is required,  and you may start a test from the Batch Development menu as soon as you have successfully checked your application program. G8WB does not require any linking, thus programs and modules may be animated as soon as they are successfully checked.

G8WB will allocate IDS-II and INTEREL databases automatically. Conventional files are allocated through File Mapping, which correspond to JCL file allocation on GCOS-8. Defaults file allocation is used, however you may be required to specify explicit file mapping rules, ref. File Mapping below.

## Animating

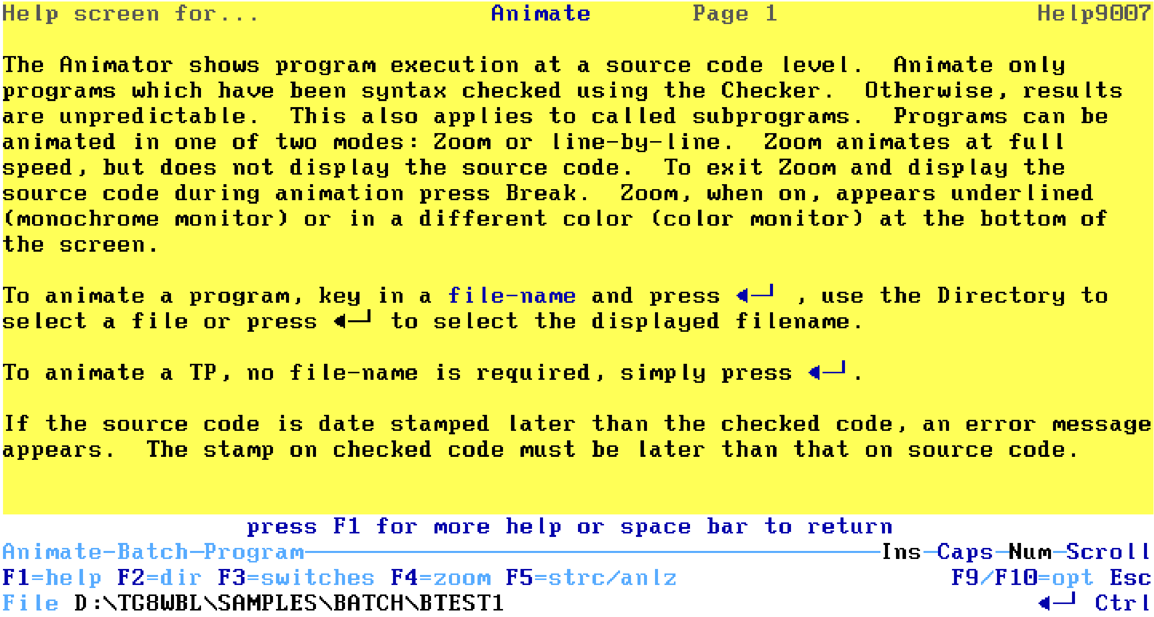
Program testing is normally performed through Animator, which enable you to visually check correct program execution and allows you to fully control and verify execution of your application. Please refer to MFWB documentation for details on Animator features.

If you have just completed checking of your program, F4-Animate will use this as default and you may simply start animation. Alternatively you may enter another program or locate another program through F2-Dir. Following illustrate the animator screen:



```
prg-entry.

      display "Starting Btest1 Program" upon console.

      open output print-file-1
                  print-file-2
                  dummy-file-1
                  print-file-3.

      accept ws-param.
      display "Param=" ws-param.

      ready usage-mode update.
      move zero to ws-db-status.
      initiate report-no-3
      initiate report-no-4
      move "Report-01 Heading 1" to print-file-1-record-1
      write print-file-1-record-1
            after advancing new-page.
      move "Report-02 Heading 2" to print-file-2-record-1
      write print-file-2-record-1
Animate-BTEST1------------------------------Level=01-Speed=5-Ins-Caps-Num-Scroll
F1=help F2=view F3=align F4=exchange F5=where F6=look-up  F9/F10=word-</> Escape
Animate Step Wch Go Zoom nx-If Prfm Rst Brk Env Qury Find Locate Txt Do Alt Ctrl
```

Before starting animation, you may select whether you want to begin in Animate or Zoom mode.



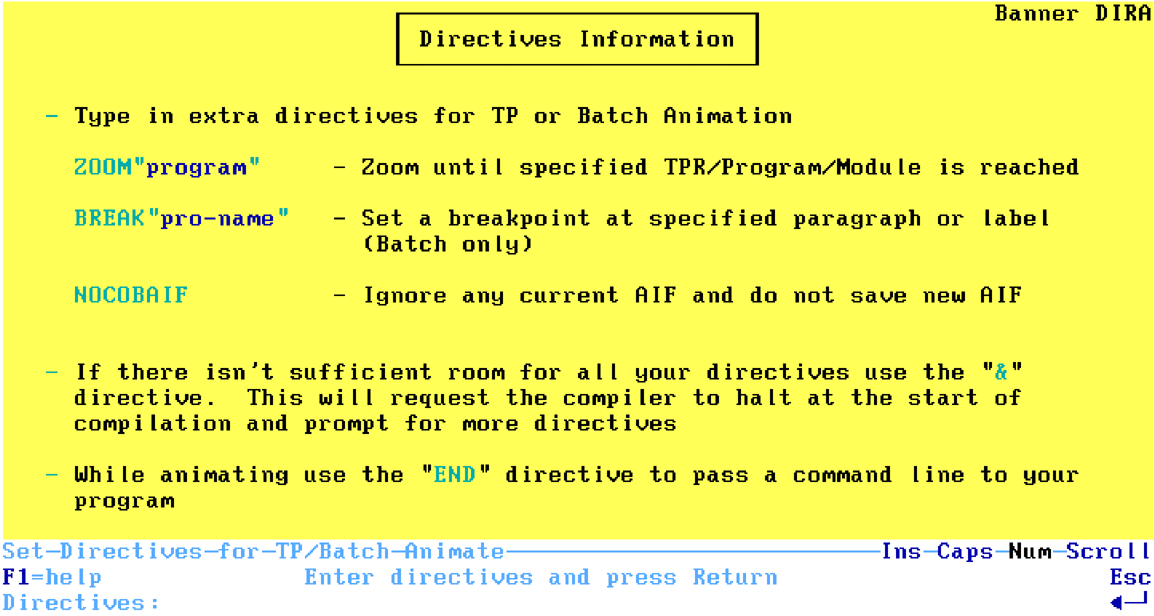
```
Help screen for...                 Animate       Page 1                Help9007

The Animator shows program execution at a source code level.  Animate only
programs which have been syntax checked using the Checker.  Otherwise, results
are unpredictable.  This also applies to called subprograms.  Programs can be
animated in one of two modes: Zoom or line-by-line.  Zoom animates at full
speed, but does not display the source code.  To exit Zoom and display the
source code during animation press Break.  Zoom, when on, appears underlined
(monochrome monitor) or in a different color (color monitor) at the bottom of
the screen.

To animate a program, key in a file-name and press ◄┘ , use the Directory to
select a file or press ◄┘ to select the displayed filename.

To animate a TP, no file-name is required, simply press ◄┘.

If the source code is date stamped later than the checked code, an error message
appears.  The stamp on checked code must be later than that on source code.


                 press F1 for more help or space bar to return
Animate-Batch-Program────────────────────────────────────────Ins-Caps-Num-Scroll
F1=help F2=dir F3=switches F4=zoom F5=strc/anlz                     F9/F10=opt Esc
File D:\TG8WBL\SAMPLES\BATCH\BTEST1                                    ◄┘ Ctrl
```

If F4-Zoom is selected, no visual animation will take place until:

- You enter Ctrl+Break to stop zooming
- You reach a previously defined Break-point

Otherwise animation will begin in the first statement of your application program.

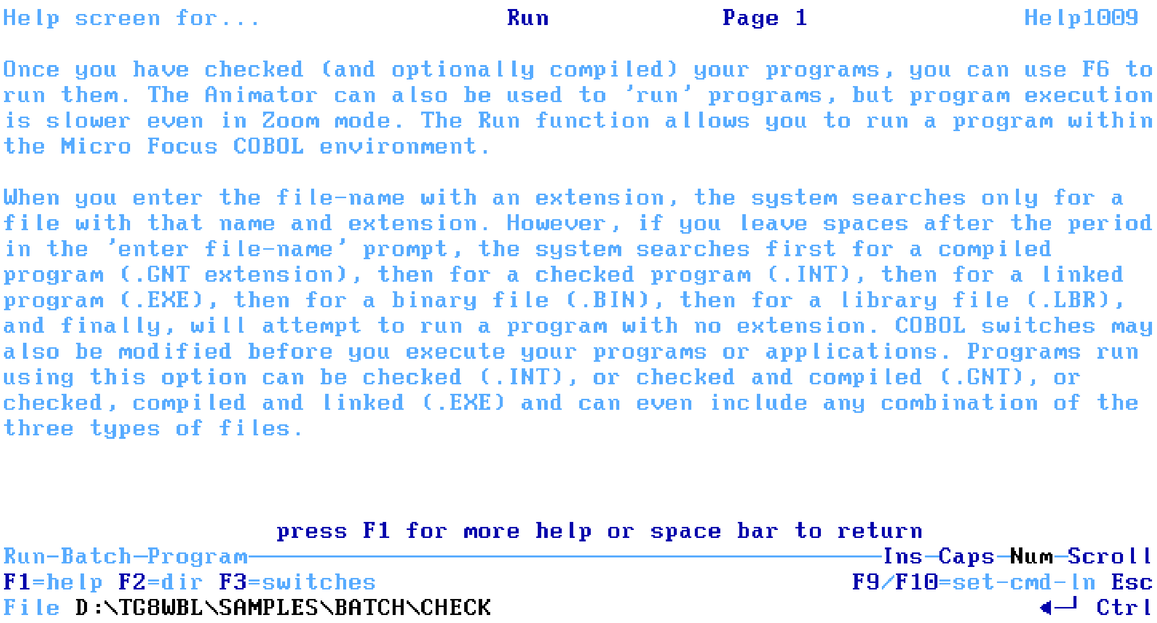
You also may define an implied break-point through F9/F10-Opt function, so you are able to zoom to a specific application program or a specific label within the main source program.



```
                                                                    Banner DIRA
                           ┌─────────────────────────┐
                           │  Directives Information │
                           └─────────────────────────┘

  - Type in extra directives for TP or Batch Animation

    ZOOM"program"       - Zoom until specified TPR/Program/Module is reached

    BREAK"pro-name"     - Set a breakpoint at specified paragraph or label
                          (Batch only)

    NOCOBAIF            - Ignore any current AIF and do not save new AIF


  - If there isn't sufficient room for all your directives use the "&"
    directive.  This will request the compiler to halt at the start of
    compilation and prompt for more directives

  - While animating use the "END" directive to pass a command line to your
    program

Set-Directives-for-TP/Batch-Animate──────────────────────────Ins-Caps-Num-Scroll
F1=help                 Enter directives and press Return                    Esc
Directives:                                                                   ◄┘
```

## Running

F5-Run function may be used to execute your application program. Programs and modules may be in either INT or GNT object format, however, GNT format provides the most efficient execution.



```
Help screen for...                  Run             Page 1                Help1009

Once you have checked (and optionally compiled) your programs, you can use F6 to
run them. The Animator can also be used to 'run' programs, but program execution
is slower even in Zoom mode. The Run function allows you to run a program within
the Micro Focus COBOL environment.

When you enter the file-name with an extension, the system searches only for a
file with that name and extension. However, if you leave spaces after the period
in the 'enter file-name' prompt, the system searches first for a compiled
program (.GNT extension), then for a checked program (.INT), then for a linked
program (.EXE), then for a binary file (.BIN), then for a library file (.LBR),
and finally, will attempt to run a program with no extension. COBOL switches may
also be modified before you execute your programs or applications. Programs run
using this option can be checked (.INT), or checked and compiled (.GNT), or
checked, compiled and linked (.EXE) and can even include any combination of the
three types of files.


                   press F1 for more help or space bar to return
Run-Batch-Program--------------------------------------------Ins-Caps-Num-Scroll
F1=help F2=dir F3=switches                                  F9/F10=set-cmd-ln Esc
File D:\TG8WBL\SAMPLES\BATCH\CHECK                                       ◄┘ Ctrl
```

## File Mapping

On GCOS-8, files would be allocated through JCL and each file is identified by a File-Code (FC). G8WB uses a similar concept, however as no JCL is involved, files are accessed either using default names or through a file mapping file. G8WB use the External & Program File Mapping feature of MWFB, please refer to MFWB for details.

G8WB will use the File-Code to logically identify a file, except for I* and P* files, which use SYSIN and SYSOUT as logical file names.

Following example of GCOS-8 JCL for program ABC is used to illustrate the difference in file allocation between GCOS-8 and G8WB:

```
$   PRMFL       IN,R,S,UMC/INPUT

$   FILE        OT,,10R

$   DATA        I*

$   SYSOUT P*,ORG
```

## Default assignement

G8WB will by default access file in your current path using the Logical-name. Following illustrate the default files accessed for above example:

**IN.**

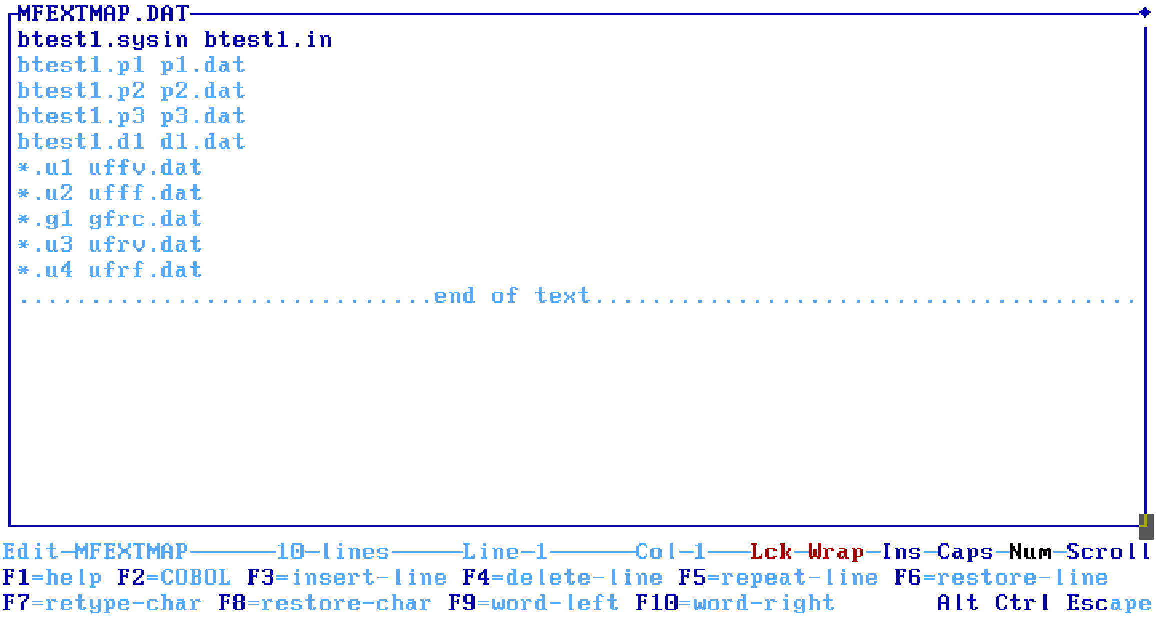
**OT.**

**SYSIN.**

**SYSOUT.**

## File Mapping

You may use a Mapping File to define explicit file allocation for all or some your files. The mapping file is a text file and you may use F7-File Mapping to invoke Editor for maintenance. Each line specify a mapping rule, which is quite similar to a GCOS-8 $ PRMFL allocation. The mapping file is always named MFEXTMAP.DAT. Following illustrate the format of a mapping file:



```
MFEXTMAP.DAT
btest1.sysin btest1.in
btest1.p1 p1.dat
btest1.p2 p2.dat
btest1.p3 p3.dat
btest1.d1 d1.dat
*.u1 uffv.dat
*.u2 ufff.dat
*.g1 gfrc.dat
*.u3 ufrv.dat
*.u4 ufrf.dat
...........................end of text.................................
```

```
Edit—MFEXTMAP————10—lines————Line—1————Col—1——Lck—Wrap—Ins—Caps—Num—Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

If the Mapping file does not exist, you need to create mapping line(s) and save it with the name MFEXTMAP.DAT.

Each mapping line consist of:

- Logical-name, Program.File-code or *.File-code
- Physical-name, File specification

Logical- and physical-name must be separated by one or more spaces. Mapping may either be specific to a program, i.e. Program.File-code, or generic for all programs, i.e. *.File-code.

Following illustrate program specific file mapping for above example:

```
ABC.IN INPUT.DAT
ABC.OT C:\ABC\OT.DAT
ABC.SYSIN ABC.IN
ABC.SYSOUT ABC.OUT
```

Following illustrate generic file mapping for above example:

```
*.IN INPUT.DAT
*.OT C:\OT.DAT
*.SYSIN SYSIN
*.SYSOUT SYSOUT
```

*Note: F7-File Mapping funtion will invoke Editor with MFEXTMAP.DAT in current path. You may change this behaviour if required, ref. Configuration chapter for details.*

## MFEXTMAP

The Mapping file is NOT related to any specific program or application and as such, the Mapping file will reflect ALL file allocations for your entire G8WB environment. You need to consider this when defining generic and/or specific mapping rules.

The External File Mapper does, however, provide the possibility to maintain multiple Mapping files as follows:

- MFEXTMAP.DAT in any path, with SET MFEXTMAP=<path>
- MFEXTMAP.DAT in current PATH
- MFEXTMAP.DAT in $COBDIR

External File Mapper will search for an MFEXTMAP.DAT, in above priority sequence, and open the MFEXTMAP.DAT file if found. If none is found, default file mapping is used.

*Note: MFEXTMAP.DAT file are not combined, i.e. once a Mapping file is found, ONLY this file is used for mapping.*

## Setting

You may use environment setting to perform file mapping. This file allocation technique is well suited for batch files, where you execute a complete test suite. Environment settings may be specified in your OS startup file, i.e. CONFIG.SYS or AUTOEXEC.BAT, or specified at any via the OS SET command. For GUI-mode you may specify settings via Project objects.

The environment name is the Logical-name and the environment value is the Physical-name.

**SET <logical-name>=<physical-name>**

Following illustrate program specific environment setting for above example:

```
SET ABC.IN=INPUT.DAT
SET ABC.OT=C:\ABC\OT.DAT
SET ABC.SYSIN=ABC.IN
SET ABC.SYSOUT=ABC.OUT
```

*Note: Environment setting must be given before you run the application, as opposed to GCOS-8 where JCL is given after the $ EXECUTE or $ PROGRAM.*

## Rules

File mapping rules are applied in the following priority order:

- environment setting
- mapping file, specific program
- mapping file, generic
- logical-name

## Optional files

Your application program may specify a file as Optional. At run-time, External File Mapper determine the presence of an optional file using the same file mapping rules as for non-optional files. A file is considered "not-present" when External File Mapper fails to allocate any file.

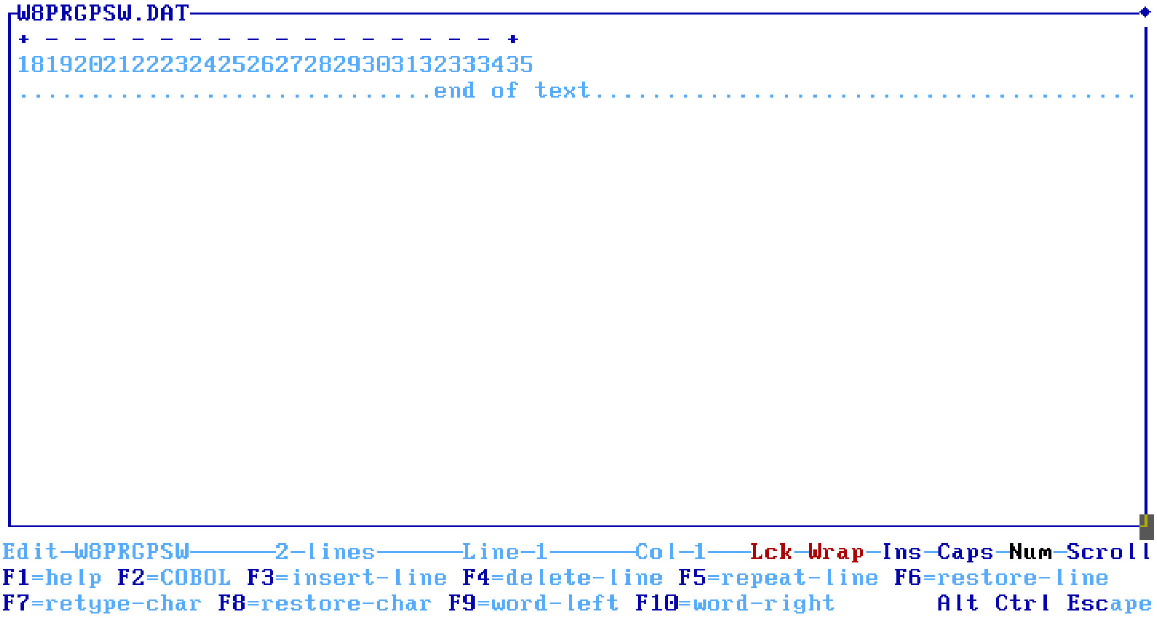
# Program-Switch-Word

GCOS-8 allow programs to communicate through the Program-Switch-Word (PSW). PSW settings may be used to control the execution flow of application programs. Application setting is normally limited to Bit 18-35.

G8WB support use of PSW and provide various features for setting of PSW during program test and execution of test suites.

## Edit

G8WB maintains PSW information is a file named W8PRGPSW.DAT in your current path. If this file is not present, all switches are unset by default. You may use the F9-PSW to edit or create the PSW file, which is a text file. Positional Set(+) and Unset(-) define the current PSW setting and a ruler line is provided below the setting. Following illustrate the format of the PSW file:



```
W8PRGPSW.DAT
+ - - - - - - - - - - - - - - - - +
181920212223242526272829303132333435
.............................end of text......................................
```

Edit—W8PRGPSW———2—lines———Line—1———Col—1——Lck—Wrap—Ins—Caps—Num—Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right Alt Ctrl Escape

If the PSW file does not exist, you need to create a setting line (ruler is optional) and save it with the name W8PRGPSW.DAT.

When you subsequently test your program, the PSW setting is established before entering your program.

## Passing

Any changes, SET switch ON/OFF, performed by your program(s) will be recorded in the PSW file. The PSW file may thus be used to pass PSW setting to other application programs.

*Note: If you re-test a program, which change PSW setting, you may need to edit the PSW file before re-testing.*

## Test suites

You may use batch files to run test suites in a non-interactive mode, similar to GCOS-8 batch jobs. G8WB provides two commands for setting and testing PSW within batch files.

Following illustrate PSW setting:

```
XM WB W8 PSWS +18          (Set switch 18)
XM WB W8 PSWS -99          (Reset all switches)
XM WB W8 PSWS +21 +35      (Set switch 21 & 35)
```

You should include a reset of all switches at the beginning of your test suite, to avoid use of any residue settings in the PSW file. The W8PRGPSW.DAT file is either updated or created.

Following illustrate PSW testing:

```
XM WB W8 TSWS 21       (Test switch 21)
IF ERRORLEVEL 1 GOTO label2
:label1                (Switch 21 OFF)
XM WB W8 RUNPRG program1
GOTO label3
:label2                (Switch 21 ON)
XM WB W8 RUNPRG program2
:label3
```

The OS batch commands IF and labels may be used to emulate GCOS-8 $ IF and $ label in your test suite. ERRORLEVEL is set to 1, if the PSW switch is Set.

## Print Files

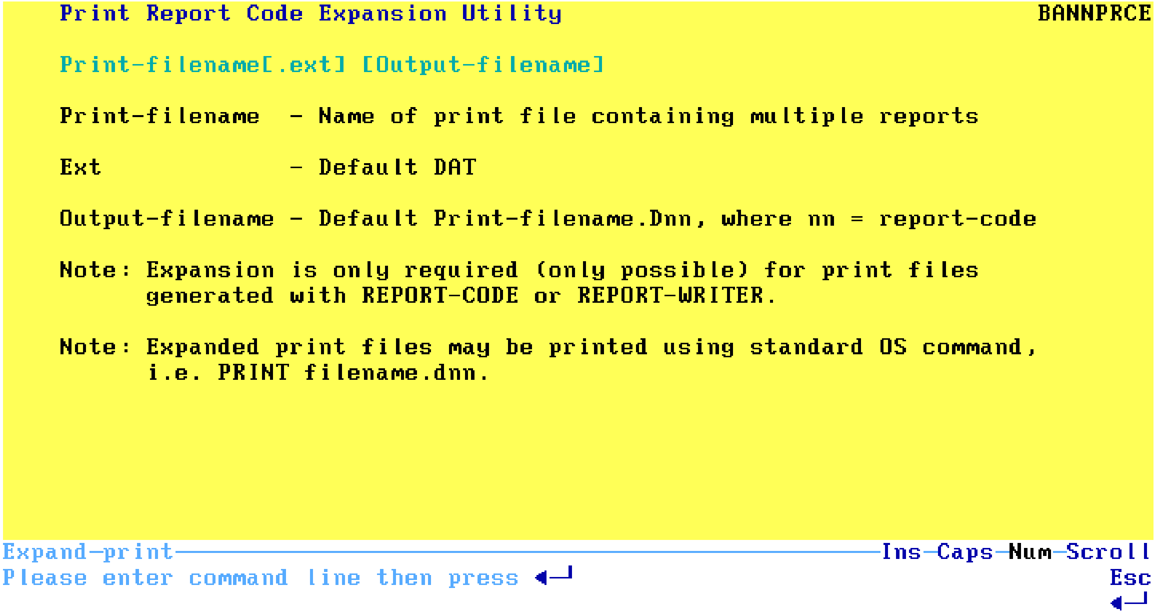
Print files are automatically formatted for PC environment printing. Although formatting is different from GCOS-8, it’s functionally equivalent and you are able to verify print formatting and content either through Editor or printing via OS PRINT command or other command.

*Note: You may need to setup your printer to support long print lines, i.e. 132 columns.*

### Expand Print

GCOS-8 allow multiple Report-code's on the same physical print file, either through the SELECT ... REPORT CODE syntax or through Report Writer. Separation of report codes is handled either by GCOS-8 SYSOUT or CONVER utility. G8WB supports the same functionality, however the Print file must be expanded, before the inidividual reports can be printed.

You may use F8-Expand Print to expand such multiple report print files. The expanded print files are assigned a D<nn> extension, where <nn> is equal to the report-code. Once expanded, the print files may be viewed through Editor or printed via OS PRINT command or other command.



```
   Print Report Code Expansion Utility                                 BANNPRCE

   Print-filename[.ext] [Output-filename]

   Print-filename  - Name of print file containing multiple reports

   Ext             - Default DAT

   Output-filename - Default Print-filename.Dnn, where nn = report-code

   Note: Expansion is only required (only possible) for print files
         generated with REPORT-CODE or REPORT-WRITER.

   Note: Expanded print files may be printed using standard OS command,
         i.e. PRINT filename.dnn.







Expand-print────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                     Esc
                                                                             ◄┘
```

Following illustrate Report-Code file expansion:

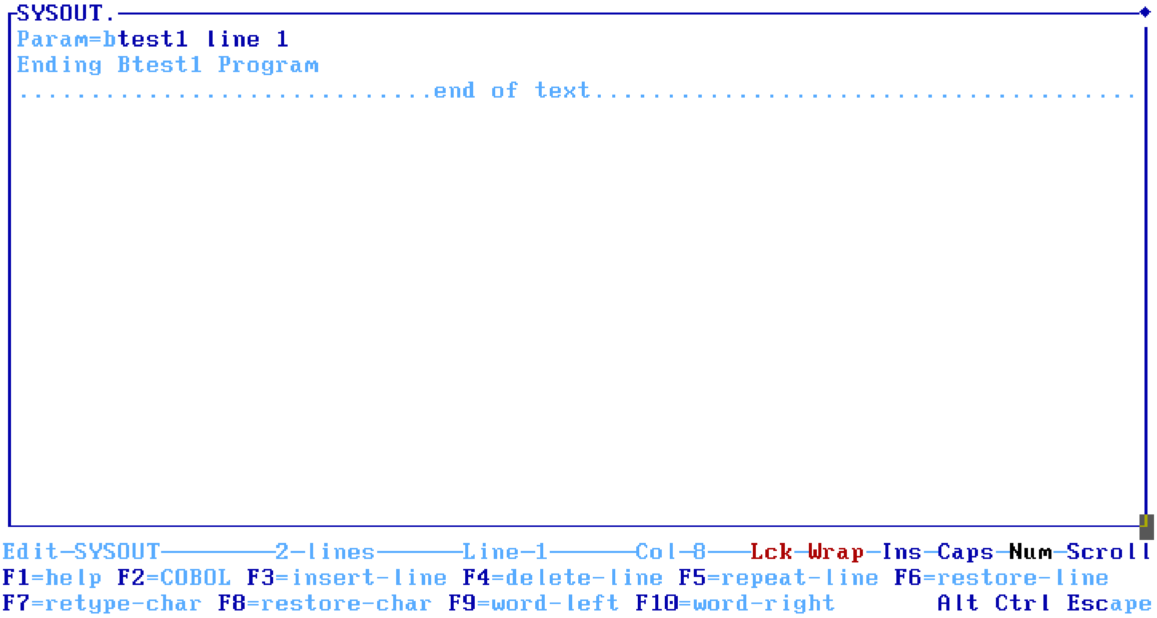
```
W8PRCEXP - Print Report Code Expansion Utility
Defaults: ext=DAT, generated ext=Drc
Print-filename[.ext]
BTEST1.PR1
Analyzing report format ...
Scanning for report codes ...
Expanding Report-Code file(s) ...
000061 records written to C:\G8WBV06\SAMPLES\BTEST1.D10
```

Following illustrate Report-Writer file expansion:

```
W8PRCEXP - Print Report Code Expansion Utility
Defaults: ext=DAT, generated ext=Drc
Print-filename[.ext]
BTEST1.PR3
Analyzing report format ...
Scanning for report codes ...
Expanding Report Writer file(s) ...
000096 records written to C:\G8WBV06\SAMPLES\BTEST1.D01
000048 records written to C:\G8WBV06\SAMPLES\BTEST1.D02
```

## SYSOUT

All DISPLAY's are collected on the SYSOUT file. The placement of the SYSOUT file may be controlled through File mapping. You may use F10-View SYSOUT to invoke Editor for browsing of the SYSOUT file. Following illustrate the content of a SYSOUT file:

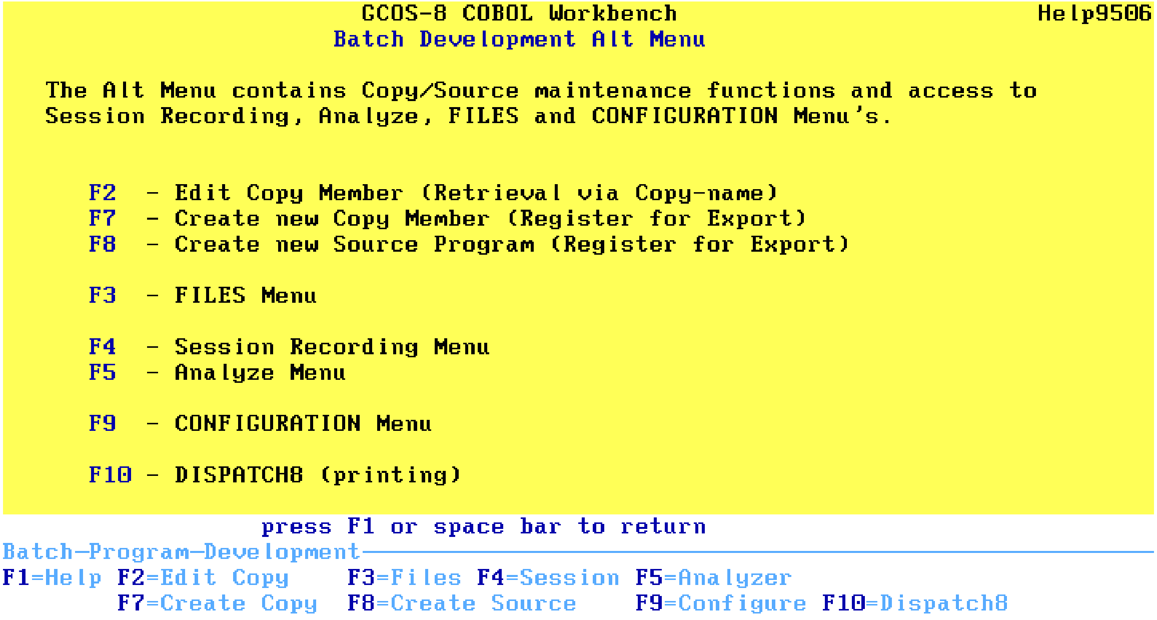


```
SYSOUT.
Param=btest1 line 1
Ending Btest1 Program
............................end of text....................................

Edit-SYSOUT--------2-lines------Line-1------Col-8---Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

*Note: F10-View SYSOUT funtion will invoke Editor with SYSOUT. in current path. You may change this behaviour if required, ref. Configuration chapter for details.*

## ALT Support Menu

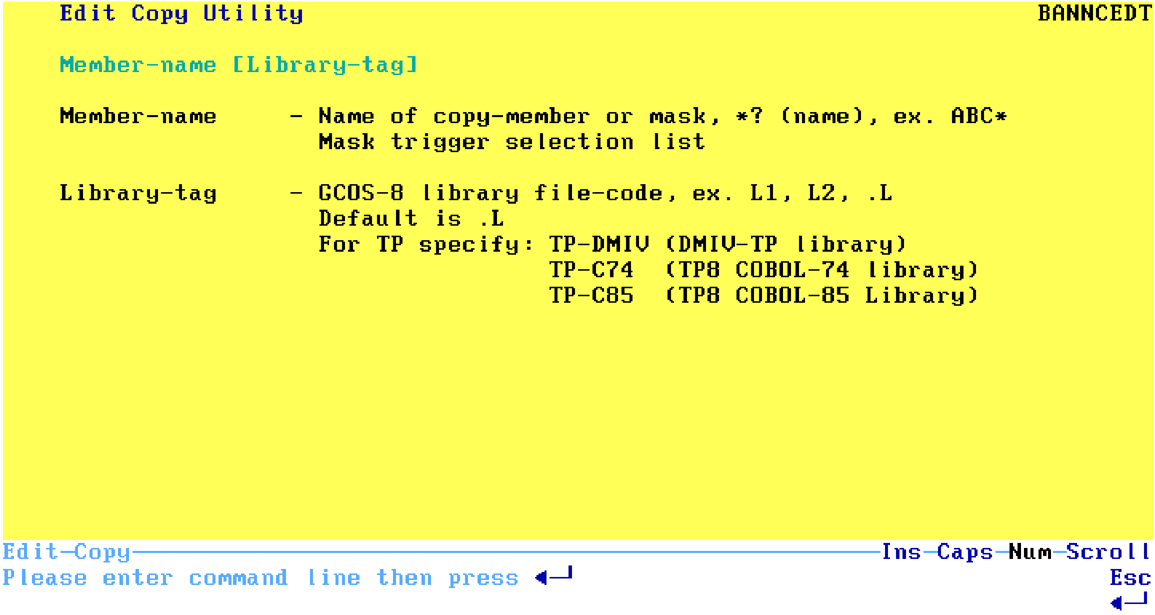
Pressing ALT key from Batch Development Menu will provide you with an alternate menu of support functions. As soon as you release the ALT key, you return to the normal menu again.



```
                         GCOS-8 COBOL Workbench                          Help9506
                       Batch Development Alt Menu

   The Alt Menu contains Copy/Source maintenance functions and access to
   Session Recording, Analyze, FILES and CONFIGURATION Menu's.


      F2   - Edit Copy Member (Retrieval via Copy-name)
      F7   - Create new Copy Member (Register for Export)
      F8   - Create new Source Program (Register for Export)

      F3   - FILES Menu

      F4   - Session Recording Menu
      F5   - Analyze Menu

      F9   - CONFIGURATION Menu

      F10  - DISPATCH8 (printing)

                  press F1 or space bar to return
Batch-Program-Development-------------------------------------------------------
F1=Help F2=Edit Copy    F3=Files F4=Session F5=Analyzer
        F7=Create Copy  F8=Create Source    F9=Configure F10=Dispatch8
```

## Edit Copy Member

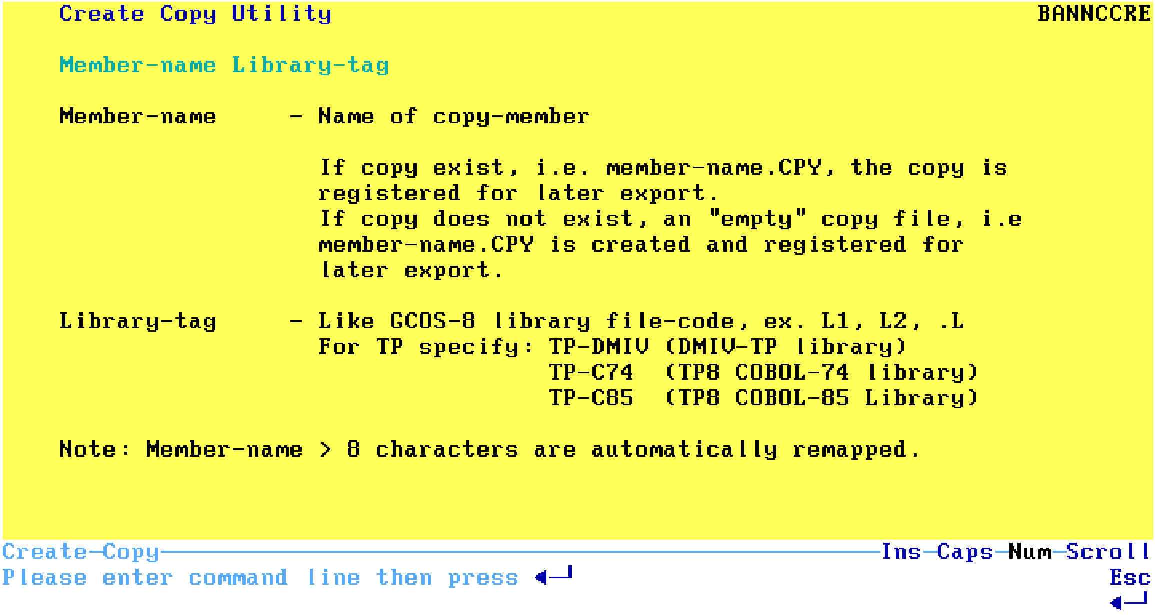
May be used to retrieve a copy member using the COBOL copy member name and library-tag. When multiple directories have been specified for the library-tag, the directories are search left-to-right until a version of the copy member is found. MF Editor is automatically invoked after retrieval and the changed copy member may simply be re-saved within the Editor.



```
    Edit Copy Utility                                                   BANNCEDT

    Member-name [Library-tag]

    Member-name      - Name of copy-member or mask, *? (name), ex. ABC*
                       Mask trigger selection list

    Library-tag      - GCOS-8 library file-code, ex. L1, L2, .L
                       Default is .L
                       For TP specify: TP-DMIV (DMIV-TP library)
                                       TP-C74  (TP8 COBOL-74 library)
                                       TP-C85  (TP8 COBOL-85 Library)









Edit-Copy────────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ↵                                       Esc
                                                                               ↵
```

## Create Copy Member

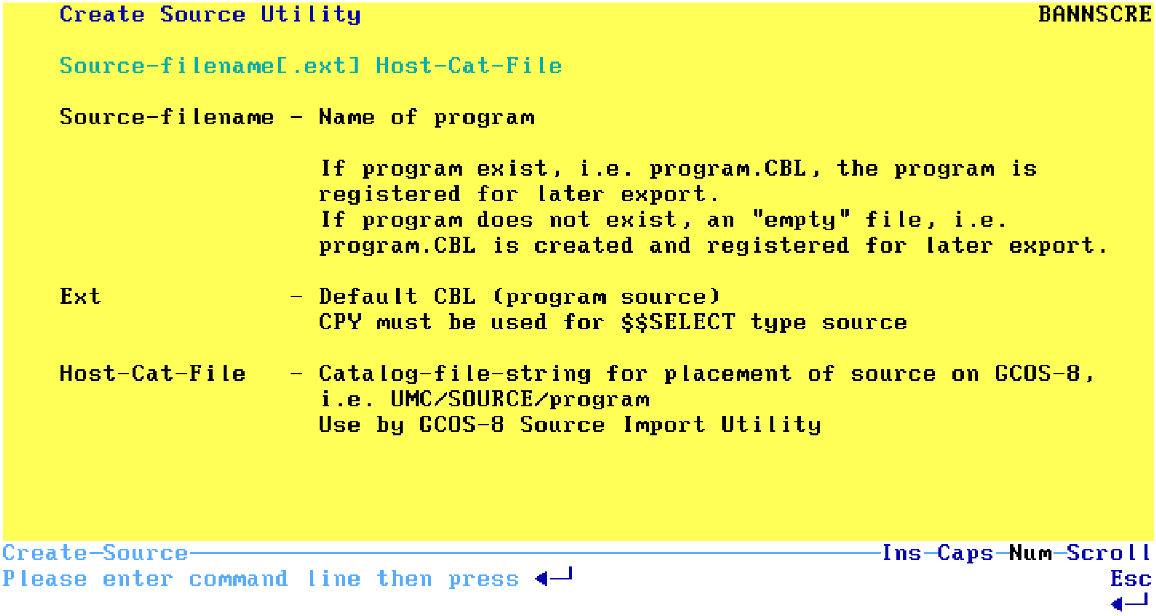
May be used to create and/or register a new copy member for subsequent export to GCOS-8. If you have already created the COPY, it is simply registered, otherwise an “empty” COPY file is created. New COPY are always created in the base directory for the Library-Tag. MF Editor is automatically invoked after retrieval and the changed copy member may simply be re-saved within the Editor.



```
    Create Copy Utility                                                 BANNCCRE

    Member-name Library-tag

    Member-name      - Name of copy-member

                       If copy exist, i.e. member-name.CPY, the copy is
                       registered for later export.
                       If copy does not exist, an "empty" copy file, i.e
                       member-name.CPY is created and registered for
                       later export.

    Library-tag      - Like GCOS-8 library file-code, ex. L1, L2, .L
                       For TP specify: TP-DMIV (DMIV-TP library)
                                       TP-C74  (TP8 COBOL-74 library)
                                       TP-C85  (TP8 COBOL-85 Library)

    Note: Member-name > 8 characters are automatically remapped.


Create-Copy──────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ↵                                       Esc
                                                                               ↵
```

## Create Source Program

May be used to  register a new source program for subsequent export to GCOS-8. The source program file must already exist. MF Editor is automatically invoked after retrieval and the changed source file may simply be re-saved within the Editor.



```
   Create Source Utility                                                   BANNSCRE

   Source-filename[.ext] Host-Cat-File

   Source-filename - Name of program

                     If program exist, i.e. program.CBL, the program is
                     registered for later export.
                     If program does not exist, an "empty" file, i.e.
                     program.CBL is created and registered for later export.

   Ext             - Default CBL (program source)
                     CPY must be used for $$SELECT type source

   Host-Cat-File   - Catalog-file-string for placement of source on GCOS-8,
                     i.e. UMC/SOURCE/program
                     Use by GCOS-8 Source Import Utility



Create-Source--------------------------------------------------Ins-Caps-Num-Scroll
Please enter command line then press ↵                                        Esc
                                                                                ↵
```

Host-Cat-File information (mandatory) will be used during subsequent import of the program source on GCOS-8.

*Note: $$SELECT copy members should be treated as an ordinary source file.*

## FILES Menu

May be used to enter the FILES menu directly from the Batch or TP Development menu.

## Configuration Menu

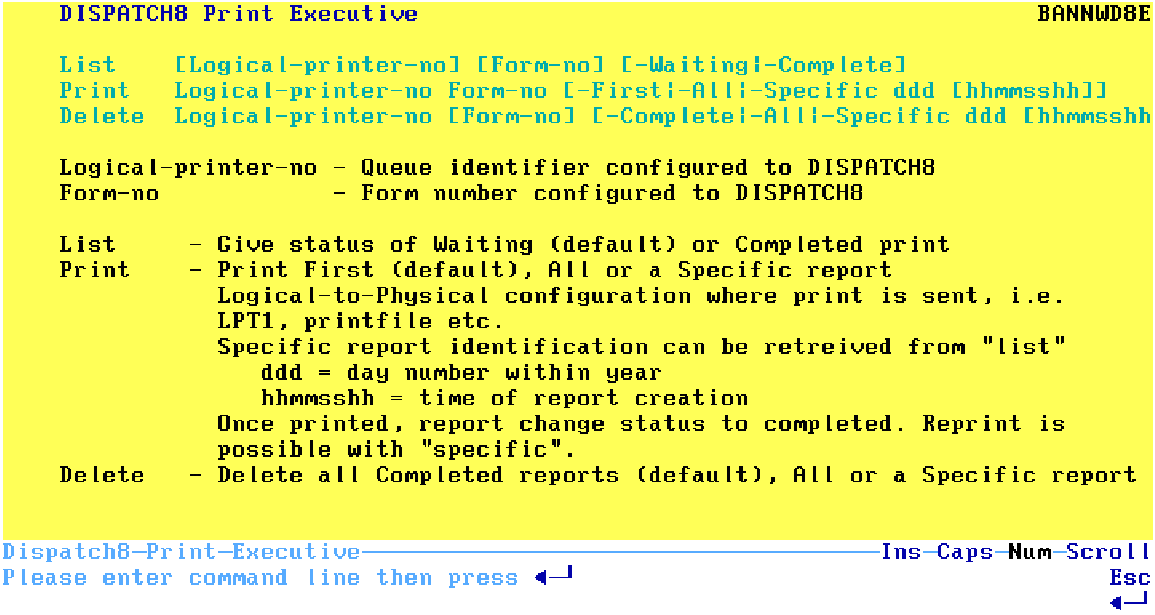
May be used to enter the CONFIGURATION menu directly from the Batch or TP Development menu.

## DISPATCH8

DISPATCH8 Distributed Printing System is fully supported by G8WB. Although integration is different, the functionality is equivalent and any customized DISPATCH8 application interface and print control modules, may be downloaded from GCOS-8 an integrated with G8WB. The GCOS-8 Administrative Services and Control Database has been replaced by a DISPATCH8 configuration file, ref. Configuration chapter for details.

Print from TP and Batch is automatically collected through the various DISPATCH8 application program interfaces. Collected reports are place under $G8WBFDIR and may thus be shared in a networked environment.

Print distribution is NOT automatic, and the DISPATCH8 Executive Utility (D8E) may be used to request printing as required. D8E may be invoked through ALT-F10 from the Batch or TP Development menu.



```
    DISPATCH8 Print Executive                                            BANNWD8E

    List     [Logical-printer-no] [Form-no] [-Waiting¦-Complete]
    Print    Logical-printer-no Form-no [-First¦-All¦-Specific ddd [hhmmsshh]]
    Delete   Logical-printer-no [Form-no] [-Complete¦-All¦-Specific ddd [hhmmsshh

    Logical-printer-no - Queue identifier configured to DISPATCH8
    Form-no            - Form number configured to DISPATCH8

    List      - Give status of Waiting (default) or Completed print
    Print     - Print First (default), All or a Specific report
                Logical-to-Physical configuration where print is sent, i.e.
                LPT1, printfile etc.
                Specific report identification can be retreived from "list"
                   ddd = day number within year
                   hhmmsshh = time of report creation
                Once printed, report change status to completed. Reprint is
                possible with "specific".
    Delete    - Delete all Completed reports (default), All or a Specific report

Dispatch8-Print-Executive─────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                                Esc
                                                                                        ◄┘
```

## List

List function may used to obtain a status of collected report files. Status may be reduced through optional specification of Logical-printer-no and Form-no arguments.

By default, only collected/unprinted (waiting) report are listed. -Complete argument may be used to request a status for printed (completed) reports.

## Print

Print function may used to request printing of one or more reports. Logical-printer-no specify which queue to print from and Form-no which type of form. Printing will be directed to the device (or file) designated in the DISPATCH8 configuration.

By default, only the first report is printed. -All argument may be used to request print of all waiting reports. -Specific argument may be used to request print of all a specific report or all reports within a specific day. With -Specific both waiting and completed reports are considered, i.e. re-print function.

*Note: Reports are not deleted after printing, but marked as complete.*

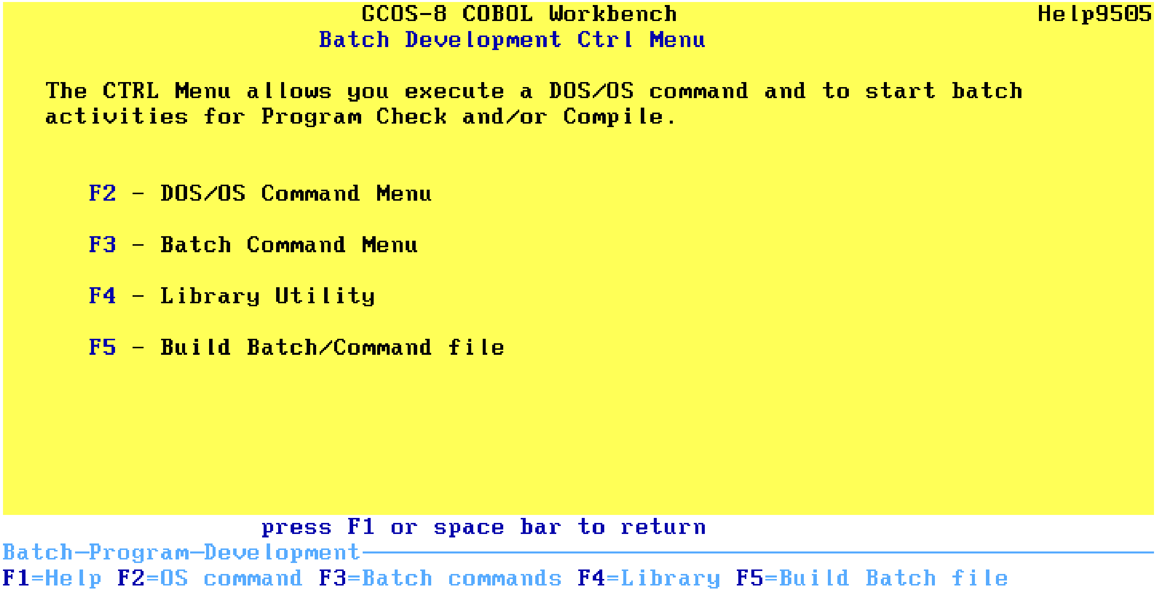
## Delete

Delete function may be used delete reports. Logical-printer-no specify which queue to delete from. Form-no may be used to limit deletion to a specific form type.

By default, all completed report are deleted. -All arguments may be used to request delete of both waiting and completed reports. -Specific argument may be used to delete a specific report or all reports within a specific day.

# CTRL Support Menu

Pressing CTRL key from Batch Development Menu will provide you with an alternate menu of support functions. As soon as you release the CTRL key, you return to the normal menu again.



```
                              GCOS-8 COBOL Workbench                         Help9505
                           Batch Development Ctrl Menu

  The CTRL Menu allows you execute a DOS/OS command and to start batch
  activities for Program Check and/or Compile.


     F2 - DOS/OS Command Menu

     F3 - Batch Command Menu

     F4 - Library Utility

     F5 - Build Batch/Command file





                  press F1 or space bar to return
Batch-Program-Development---------------------------------------------------------
F1=Help F2=OS command F3=Batch commands F4=Library F5=Build Batch file
```

## DOS Command

May be used to execute a DOS or OS/2 command, or start an OS command file.

## Library Utility

May be used to build and maintain libraries of programs. Typically you could place system programs and other programs, which are not tested within the current G8WB environment, into static library files. Such libraries may be allocated through G8WB Environment configuration. Please refer to MFWB documentation and Configuration chapter for details.

## Batch Menu

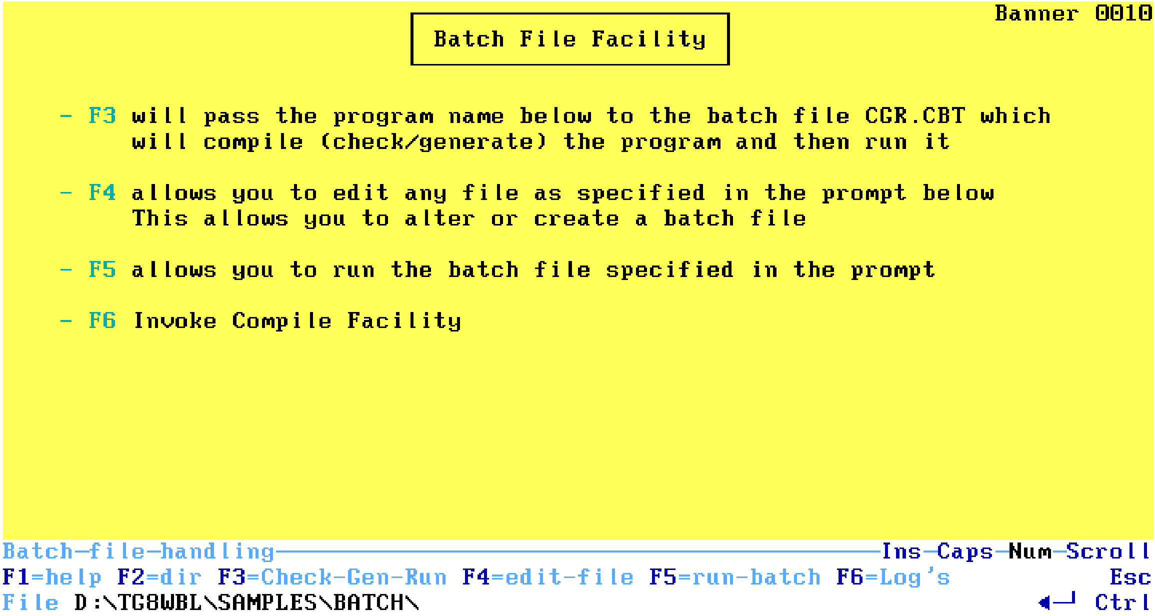
Most of the facilities and functions within G8WB workbench are also available through commands, which can be embedded in command files. Commands are typically used for bulk compiles and batch test suites.

A WB command file (CBT) is a collection of G8WB commands, which can be executed as a batch. In contrast to an OS command file, a WB command file does not allow conditional or cyclic constructs, and only G8WB commands are supported.

A WB command file may contain G8WB commands as follows:

**command [argument(s)]**

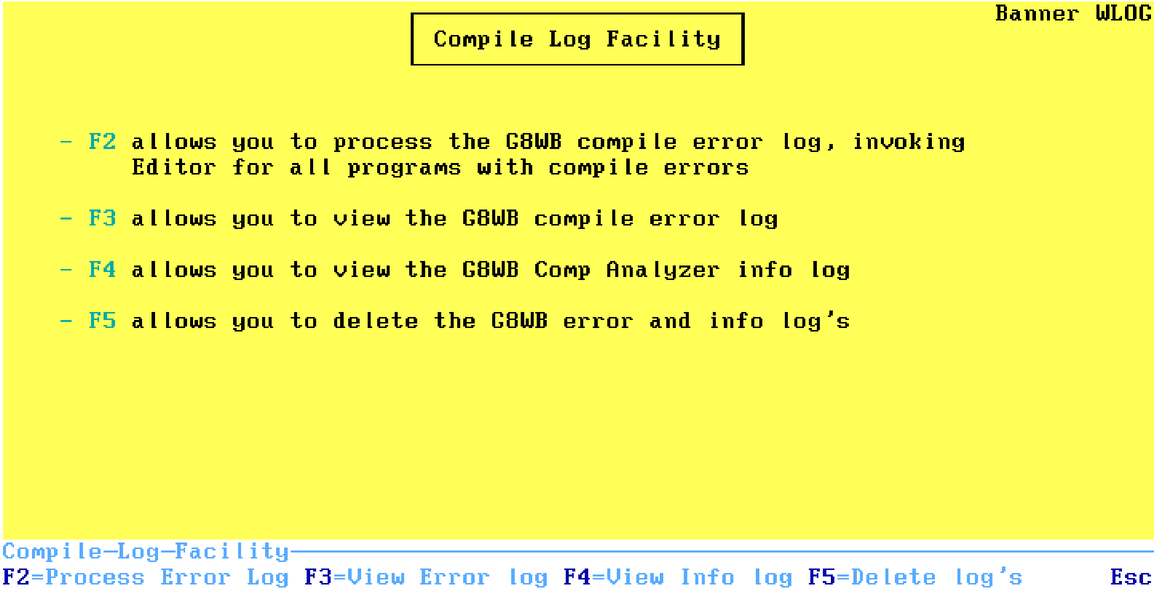
Batch File Facility may be used to locate, edit and launch WB command files. It also provides entry to the Compile Log Facility.



```
                                                                    Banner 0010
                        Batch File Facility

   - F3 will pass the program name below to the batch file CGR.CBT which
        will compile (check/generate) the program and then run it

   - F4 allows you to edit any file as specified in the prompt below
        This allows you to alter or create a batch file

   - F5 allows you to run the batch file specified in the prompt

   - F6 Invoke Compile Facility

Batch-file-handling──────────────────────────────────────────Ins-Caps-Num-Scroll
F1=help F2=dir F3=Check-Gen-Run F4=edit-file F5=run-batch F6=Log's          Esc
File D:\TG8WBL\SAMPLES\BATCH\                                            ◄┘ Ctrl
```

## Compile Log

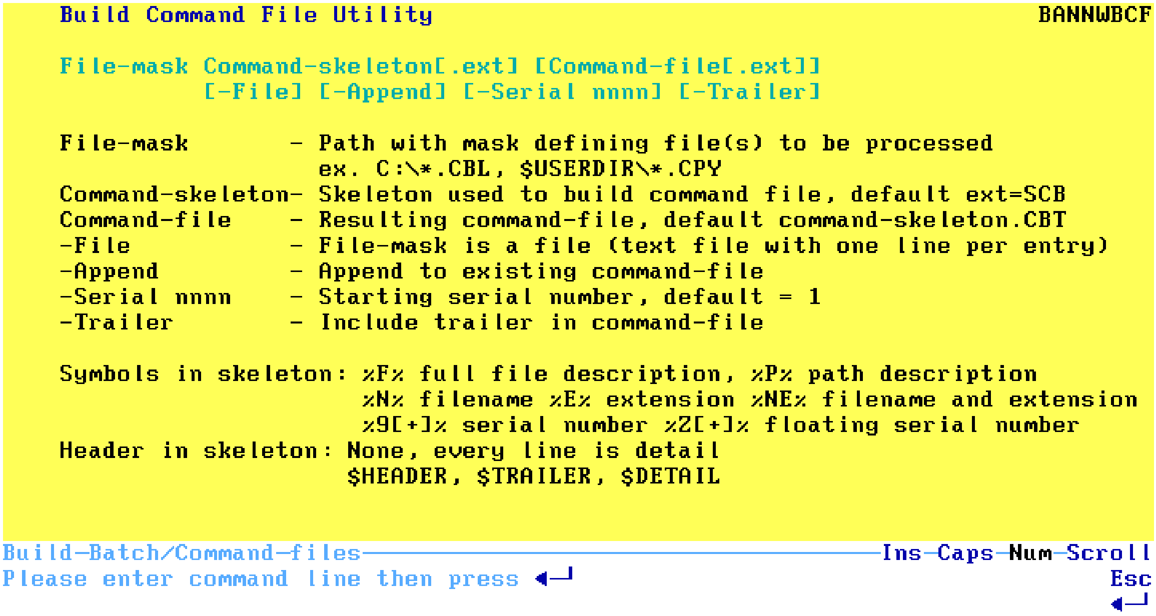
G8WB provides Compile Log features to assist you when you are working with bulk compilation, ref. below for details on Bulk Compilation. These features are available through F6-Compile Log Facility in the Batch File Facility menu.



```
                                                                    Banner WLOG
                        Compile Log Facility

   - F2 allows you to process the G8WB compile error log, invoking
        Editor for all programs with compile errors

   - F3 allows you to view the G8WB compile error log

   - F4 allows you to view the G8WB Comp Analyzer info log

   - F5 allows you to delete the G8WB error and info log's

Compile-Log-Facility────────────────────────────────────────────────────────────
F2=Process Error Log F3=View Error log F4=View Info log F5=Delete log's     Esc
```

## Build Command file

Build Command File utility (BCF) may be used to build bulk compile or other WB or OS command files. BCF works on the basis of a command skeleton and a dynamic file mask, which is used to generate a command file. The command file may subsequently be launched through Batch Fiel Facility Menu. Please refer to examples under Bulk Compile below.



```
    Build Command File Utility                                                BANNWBCF

    File-mask Command-skeleton[.ext] [Command-file[.ext]]
              [-File] [-Append] [-Serial nnnn] [-Trailer]

    File-mask         - Path with mask defining file(s) to be processed
                        ex. C:\*.CBL, $USERDIR\*.CPY
    Command-skeleton- Skeleton used to build command file, default ext=SCB
    Command-file      - Resulting command-file, default command-skeleton.CBT
    -File             - File-mask is a file (text file with one line per entry)
    -Append           - Append to existing command-file
    -Serial nnnn      - Starting serial number, default = 1
    -Trailer          - Include trailer in command-file

    Symbols in skeleton: %F% full file description, %P% path description
                         %N% filename %E% extension %NE% filename and extension
                         %9[+]% serial number %Z[+]% floating serial number
    Header in skeleton: None, every line is detail
                        $HEADER, $TRAILER, $DETAIL

Build-Batch/Command-files──────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                              Esc
                                                                                     ◄┘
```

## Command skeleton

A command skeleton is a text file with one or more commands, that are to be repeated for each occurrence within the file mask. The command skeleton will normally contains symbols (variables), which are replaced during file mask processing. Following symbols may be used:

| Symbol/Variable | Content |
|---|---|
| %F% | Full file description, i.e. C:\PATH\FILE.EXT |
| %P% | Path description only, i.e. C:\PATH |
| %N% | File name only, i.e. FILE |
| %E% | File extension only, i.e. EXT |
| %NE% | File name and extension only, i.e. FILE.EXT |
| %9% | Fixed length serial number, i.e. 00001 |
| %Z% | Floating length serial number, i.e. 1 |
| %9+% | Like %9% except serial number is incremented after use |
| %Z+% | Like %Z% except serial number is incremented after use |

Serial number is incremented for each file mask occurrence. A start value may be specified, otherwise the default serial number is one (1).

By default, all lines of the command skeleton are processed for each occurrence in the file mask, however, you may designate some lines as Header and/or Trailer lines. $HEADER, $TRAILER and $DETAIL (separate line entry) lines are used to separate lines of the command skeleton.

Header lines are generated once, unless the -Append argument is given.

Trailer lines are only generated when -Trailer argument is given.

### File mask

A standard OS file mask may be used to control generation of the command file, i.e. *.CBL or C:\SOURCE\A*.CBL.

If -File argument is given, the file mask specify a text file with one or more lines, where each line contains a file specification.

## G8WB Commands

Most of the facilities and functions within G8WB workbench are also available through commands, which can be invoked outside the interactive workbench. Commands are typically used to build command files, i.e. bulk compiles, batch test suites.

G8WB support two types of command files:

- OS Command File
- WB Command File

In general WB command file are best suited for activities like bulk compile, whereas OS command files are best suited for activities like test suites.

### OS Command File

An OS command file (BAT for DOS, CMD for OS/2) is a collection of commands, which can be executed as a batch. Command files may have conditional and/or cyclic constructs. Command file may also be executed implicitly from tools like MAKE etc.

A G8WB command may be invoked as follows:

**[XM] WB W8 command [argument(s)]**

*Note: XM is only required if you are running DOS.*

### WB Command File

A WB command file (CBT) is a collection of G8WB commands, which can be executed as a batch. In contrast to an OS command file, a WB command file does not allow conditional or cyclic constructs, and only G8WB commands are supported.

A WB command file may contain G8WB commands as follows:

**command [argument(s)]**

A WB command file may be launched directly form the interactive workbench (Batch File Facility), or through the G8WB batch command as follows:

**[XM] WB BATCH command-file**

*Note: XM is only required if you are running DOS.*

## Commands

Each command is defined by a unique command name and most commands require one or more arguments. Following describes the G8WB commands available.

## Checking

| Name | Argument(s) | Function |
|---|---|---|
| CHKCBL74 | source-name [directive(s)] | Check a COBOL-74 Source |
| CHKTPR74 | TPR-name [directive(s)] | Check a COBOL-74 TPR |
| CHKPRG74 | Prg-name [directive(s)] | Check a COBOL-74 Program |
| CHKCBL85 | source-name [directive(s)] | Check a COBOL-85 Source |
| CHKTPR85 | TPR-name [directive(s)] | Check a COBOL-85 TPR |
| CHKPRG85 | Prg-name [directive(s)] | Check a COBOL-85 Program |
| CHKCBLSQ | source-name [directive(s)] | Check a COBOL-85/SQL Source |
| CHKTPRSQ | TPR-name [directive(s)] | Check a COBOL-85/SQL TPR |
| CHKPRGSQ | Prg-name [directive(s)] | Check a COBOL-85/SQL Program |

Directives can be any valid Checker directive, please refer to MFWB documentation for details. All checking commands produce full listing by default (LIST""), the ERRLIST directive can be used to obtain a list of errors only.

| Name | Argument(s) | Function |
|---|---|---|
| COMPILE | source-name [directive(s)] | Compile to GNT |
| GENERATE | source-name [directive(s)] | Compile to GNT (like COMPILE) |

Directives can be any valid Generate directive, please refer to MFWB documentation for details. SEGSIZE(63536) and 64KSECT directives are default, the 64KPARA directive can be used if you experience problems with segmentation, i.e. code segment > 64KB.

| Name | Argument(s) | Function |
|---|---|---|
| BELOG | [path] | Browse Compiler Error Log |
| DELOG | [path] | Delete Compiler Error & Info Log's |
| UELOG | command-line | Update Compile Error Log |

```
W8UELOG - Update Compile Error-log (1.2.00/1.2.00)
Defaults: Error only, No listing
[Programname|Source-filename] [-Info-also] [-List-include]
```

## TP

| Name | Argument(s) | Function |
|---|---|---|
| ANIMTP | [directives] END | Animate DMIV-TP or TP8 |
| ZOOMTP | [directives] END | Animate DMIV-TP or TP8 with immediate zoom |
| RUNTP | | Run DMIV-TP or TP8 |

Directives can be any valid Animator directive, please refer to MFWB documentation for details.

## Batch

| Name | Argument(s) | Function |
|---|---|---|
| ANIMPRG | program [directives(s)] | Animate batch program |
| ZOOMPRG | program [directives(s)] | Animate batch program with immediate zoom |
| RUNPRG | program | Run batch program |

Directives can be any valid Animator directive, please refer to MFWB documentation for details.

| Name | Argument(s) | Function |
|---|---|---|
| G8WFL | command-line | Convert GCOS-8 files |

```
W8G8WFL - GCOS-8 Workbench File Loader (1.2.00/1.2.00)
Defaults: Profile Ext=G8P, Input Ext=G8F, Output Ext=DAT, Output=Input
Profile [Input-file [Output-file]] [-Stats]
```

| Name | Argument(s) | Function |
|---|---|---|
| PRCEXP | command-line | Expand multiple report print files |

```
W8PRCEXP - Print Report Code Expansion Utility (1.2.00/1.2.00)
Defaults: ext=DAT, generated ext=Drc
Print-filename[.ext] [Output-filename]
```

| Name | Argument(s) | Function |
|---|---|---|
| PSWT | command-line | Test PSW setting |

```
W8PSWT - Program Switch Word Test (1.2.00/1.2.00)
Switch (18-35)
```

Error-level = 1 if switch is set.

| Name | Argument(s) | Function |
|---|---|---|
| PSWS | command-line | Set/Reset PSW switch |

```
W8PSWS - Program Switch Word Setting (1.2.00/1.2.00)
Switches: 18-35 or 99(all), +=Set, -=Reset
+|-nn [...]
```

## IDS-II Database

| Name | Argument(s) | Function |
|---|---|---|
| TSCH | command-line | Translate Schema DDL and DMCL |

```
W8TSCH - IDS-II Schema Translator Utility (1.2.00/1.2.00)
Defaults: Ext=.SD, DMCL Ext=.SDM, List Ext=.SL
Schema-filename[.ext]
```

| Name | Argument(s) | Function |
|---|---|---|
| VSCH | command-line | Validate Schema |

```
W8VSCH - IDS-II Schema Validator Utility (1.2.00/1.2.00)
Schema-name
```

| Name | Argument(s) | Function |
|---|---|---|
| TSSCH | command-line | Translate Subschema DDL |

```
W8TSSCH - IDS-II Subschema Translator Utility (1.2.00/1.2.00)
Defaults: Ext=.SSD, List Ext=.SSL
Subschema-filename[.ext]
```

| Name | Argument(s) | Function |
|---|---|---|
| VSSCH | command-line | Validate Subschema |

```
W8VSSCH - IDS-II Subschema Validator Utility (1.2.00/1.2.00)
Subschema-name
```

| Name | Argument(s) | Function |
|---|---|---|
| IIDS | command-line | Run Interactive IDS-II Utility |

```
W8IIDS Interactive IDS-II Utility (1.2.00/1.2.00)
Subschema [FILE filename [Zoom]]
```

| Name | Argument(s) | Function |
|---|---|---|
| Q2UT | command-line | Run IDS-II Database Utility |

```
W8Q2UT IDS-II Database Utility (1.2.00/1.2.00)
Schema [Dump|INITIAL -Area name|ALL]
```

| Name | Argument(s) | Function |
|---|---|---|
| DBBCK | command-line | Run Backup Database Utility |

```
W8DBBCK - Database Backup Utility (1.2.00/1.2.00)
Schema Backup-set -Delete
```

| Name | Argument(s) | Function |
|---|---|---|
| DBRES | command-line | Run Restore Database Utility |

```
W8DBRES - Database Restore Utility (1.2.00/1.2.00)
Schema Backup-Set
```

| Name | Argument(s) | Function |
|---|---|---|
| DBUNLOAD | command-line | Run Database Unload Utility |

```
W8USCH - Database Unload Utility (1.2.00/1.2.00)
Schema-name unload-file  Backup-Set
Schema-name unload-file [-Only Record-name|Area-name|RECORDS|SETS]
```

| Name | Argument(s) | Function |
|---|---|---|
| DBLOAD | command-line | Run Database Load Utility |

```
W8LSCH - Database Load Utility (1.2.00/1.2.00)
Schema-name load-file [-Update [Only]]
```

| Name | Argument(s) | Function |
|---|---|---|
| SDEL | command-line | Run Delete Schema Utility |

```
W8SDEL - IDS-II Schema Delete Utility (1.2.00/1.2.00)
Schema-name
```

| Name | Argument(s) | Function |
|---|---|---|
| SSDEL | command-line | Run Delete Subschema Utility |

```
W8SSDEL - IDS-II Subschema Delete Utility (1.2.00/1.2.00)
Subschema-name
```

| Name | Argument(s) | Function |
|---|---|---|
| SMOVE | command-line | Run Move Schema Utility |

```
W8MVSCH - IDS-II Schema Move Utility (1.2.00/1.2.00)
Schema-name Target-directory
```

| Name | Argument(s) | Function |
|---|---|---|
| SSMOVE | command-line | Run Move Subschema Utility |

```
W8MVSSCH - IDS-II Subschema Move Utility (1.2.00/1.2.00)
Subschema-name Target-directory
```

## INTEREL Database

| Name | Argument(s) | Function |
|---|---|---|
| XDBBCK | command-line | Run INTEREL Backup Database Utility |

```
W8XDBBCK - XDB Database Backup Utility (1.2.00/1.2.00)
Location Backup-set -Delete
```

| Name | Argument(s) | Function |
|---|---|---|
| XDBRES | command-line | Run INTEREL Restore Database Utility |

```
W8XDBRES - Database Restore Utility (1.2.00/1.2.00)
Location Backup-Set
```

| Name | Argument(s) | Function |
|---|---|---|
| XDBIED | command-line | Run INTEREL Import/Export Database Utility |

```
W8XDIED - Database Import/Export Utility (1.2.00/1.2.00)
Location Setup|Import|Export [-Table name] [-Path path]
```

| Name | Argument(s) | Function |
|---|---|---|
| XDBMCF | command-line | Run XDB/INTEREL Command File Migrate Utility |

```
W8XDMCF - Migrate Command File Utility (1.2.00/1.2.00)
Defaults: Ext=IQL, Output ext=SQL, -INTEREL
          Ext=SQL, Output ext=IQL, -XDB
input-file.ext [output-file[.ext] [-XDB|-INTEREL]
```

## Forms

| Name | Argument(s) | Function |
|---|---|---|
| FORMS |  | Run Workbench Screen Painter |
| FDIR | command-line | Run Forms Directory Utility |

```
W8FDIR - Forms Directory Utility (1.2.00/1.2.00)
Mask=?* (name,language-code,version,type-forms)
Defaults: type=configured, output=CON:
Form-name [-Language code] [-Vers T] [-Type forms] [-Output dev/file] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FDELE | command-line | Run Forms Delete Utility |

```
W8FDELE - Delete Form Utility (1.2.00/1.2.00)
Mask=?* (name,language-code,version,type-forms)
Defaults: type=configured
Form-name [-Language code] [-Version T] [-Type forms] [-Force no confirmation]
```

| Name | Argument(s) | Function |
|---|---|---|
| FCOPY | command-line | Run Forms Copy Utility |

```
W8FCOPY - Forms Copy Utility (1.2.00/1.2.00)
Defaults: type=configured
Form-name [-Language code] [-Vers T] [-Type forms]
-FROM form-name [-FLanguage code] [-FVers T]
```

## TPFF

| Name | Argument(s) | Function |
|---|---|---|
| FFIMP | command-line | Run TPFF Forms Import Utility |

```
W8FFIMP - TPFF Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.S output=CON:
Source-filename[.ext] [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FFEXP | command-line | Run TPFF Forms Export Utility |

```
W8FFEXP - TPFF Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=formname, ext=.S, output=CON:
Formname [Source-filename[.ext]] [-Output device/file] [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FFGEN | command-line | Run TPFF Forms Generate Utility |

```
W8FFGEN - TPFF Forms Generate Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: output=CON:
Formname [-Output device/file] [-Source-list] [-Pause]
```

## FORMAT

| Name | Argument(s) | Function |
|---|---|---|
| FAIMP | command-line | Run FORMAT Forms Import Utility |

```
W8FAIMP - FORMAT Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.S output=CON:
Source-filename[.ext] [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FAEXP | command-line | Run FORMAT Forms Export Utility |

```
W8FAEXP - FORMAT Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=formname, ext=.S, output=CON:
Formname [Source-filename[.ext]] [-Output device/file] [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FAGEN | command-line | Run FORMAT Forms Generate Utility |

```
W8FAGEN - FORMAT Forms Generate Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: output=CON:
Formname [-Output device/file] [-Source-list] [-Pause]
```

## Pilspråk

| Name | Argument(s) | Function |
|---|---|---|
| FPIMP | command-line | Run Pilspråk Forms Import Utility |

```
W8FPIMP - PILSPråk Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.S output=CON:
Source-filename[.ext] [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FPEXP | command-line | Run Pilspråk Forms Export Utility |

```
W8FPEXP - PILSPråk Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=formname, ext=.S, output=CON:
Formname [Source-filename[.ext]] [-Output device/file]
         [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FPGEN | command-line | Run Pilspråk Forms Generate Utility |

```
W8FPGEN - PILSPråk Forms Generate Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: output=CON:
Formname [-Output device/file] [-Source-list] [-Pause]
```

## DIMS8

| Name | Argument(s) | Function |
|---|---|---|
| FDIMP | command-line | Run DIMS8 Forms Import Utility |

```
W8FDIMP - DIMS8 Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.DD output=CON:
Source-filename[.ext] [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FDEXP | command-line | Run DIMS8 Forms Export Utility |

```
W8FDEXP - DIMS8 Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=formname, ext=.DD, output=CON:
Formname [Source-filename[.ext]] [-Output device/file]
         [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FSDIMP | command-line | Run DIMS8 Storage Import Utility |

```
W8FSDIMP - DIMS8 Storage Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.SD output=CON:
Source-filename[.ext] [-Output device/file]
               [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FSDEXP | command-line | Run DIMS8 Storage Export Utility |

```
W8FSDEXP - DIMS8 Storage Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=storagename, ext=.SD, output=CON:
Storagename [Source-filename[.ext]] [-Output device/file]
            [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FTDIMP | command-line | Run DIMS8 Transaction Import Utility |

```
W8FTDIMP - DIMS8 Transaction Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.TD, output=CON:
Source-filename[.ext] [-Output device/file]
                     [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FTDEXP | command-line | Run DIMS8 Transaction Export Utility |

```
W8FTDEXP - DIMS8 Transaction Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=transactionname, ext=.TD, output=CON:
Transactionname [Source-filename[.ext]] [-Output device/file]
                [-Source-list] [-Pause]
```

## SOLD

| Name | Argument(s) | Function |
|---|---|---|
| FSIMP | command-line | Run SOLD Forms Import Utility |

```
W8FSMP - SOLD Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.FRM output=CON:
Source-filename[.ext] [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FSEXP | command-line | Run SOLD Forms Export Utility |

```
W8FSEXP - SOLD Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=formname, ext=.FRM, output=CON:
Formname [Source-filename[.ext]] [-Output device/file] [-Source-list] [-Pause]
```

## TSM8

| Name | Argument(s) | Function |
|---|---|---|
| FTIMP | command-line | Run TSM8 Forms Import Utility |

```
W8FTIMP - TSM8 System/Forms Import Utility (1.2.00/1.2.00)
Defaults: filename-ext=.ALL  output=CON:
                    [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| FTEXP | command-line | Run TSM8 Forms Export Utility |

```
W8FTEXP - TSM8 System/Forms Export Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: source-filename=name, ext=.ALL/.SYS/.FRM/.MSG, output=CON:
Directive name [-L language-code] [Source-filename[.ext]] [-Output device/file]
         [-Source-list] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| FTGEN | command-line | Run TSM8 Forms Generate Utility |

```
W8FTGEN - TSM Forms Generate Utility (1.2.00/1.2.00)
Mask=?* (name)
Defaults: output=CON:
Formname [-L language] [-Output device/file] [-Source-list] [-Pause]
```

## Source Handling

| Name | Argument(s) | Function |
|---|---|---|
| SIMP | command-line | Run Source Import Utility |

```
W8SIMP - Source Import Utility (1.2.00/1.2.00)
Defaults: ext=XFR, type=source, output=CON:
Source-filename[.ext] [-Type-select] [-Classify T[pr]|P[rogram]]
                      [-Output device/file] [-Source-list] [-Pause-screen]
```

| Name | Argument(s) | Function |
|---|---|---|
| SEXP | command-line | Run Source Export Utility |

```
W8SEXP - Source Export Utility (1.2.00/1.2.00)
Defaults: ext=XFR, output=CON:
Extract-filename[.ext]  [-Confirm] [-Mask|-Force *? (name)]
                        [-Expand] [-Append]
                        [-Source-list] [-Pause-screen] [-Output device/file]
```

| Name | Argument(s) | Function |
|---|---|---|
| SCRE | command-line | Run Source Create Utility |

```
W8SCRE - Source Register/Create Utility (1.2.00/1.2.00)
Defaults: ext=CBL
Source-filename[.ext] Host-Cat-file
```

| Name | Argument(s) | Function |
|---|---|---|
| SCOUT | command-line | Run Source Checkout Utility |

```
W8SCOUT - Check-out Source Utility (1.2.00/1.2.00)
Reference-directory Source-name [-UD userdir]
```

| Name | Argument(s) | Function |
|---|---|---|
| SCIN | command-line | Run Source Checkin Utility |

```
W8SCIN - Check-in Source Utility (1.2.00/1.2.00)
Compile types: CBL74,CBL85,CBLSQL,PRG74,PRG85,PRGSQL,TPR74,TPR85,TPRSQL
Reference-directory Source-name [-Compile type] [-UD userdir]
```

## Copy Handling

| Name | Argument(s) | Function |
|---|---|---|
| CIMP | command-line | Run Copy Import Utility |

```
W8CIMP - Copy Import Utility (1.2.00/1.2.00)
Defaults: ext=XFR, output=CON:, Global Map
Directory: dir or dir1+dir2 etc. (last is base)
Source-filename[.ext] Library-tag [-Map Local|Project|Global] [Directories]
                                  [-Source-list] [-Output device/file] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| CEXP | command-line | Run Copy Export Utility |

```
W8CEXP - Copy Export Utility (1.2.00/1.2.00)
Defaults: ext=XFR, output=CON:
Source-filename[.ext] Library-tag [-Confirm] [-Mask|-Force *? (name)]
                        [-Append]
                        [-Source-list] [-Pause-screen] [-Output device/file]
```

| Name | Argument(s) | Function |
|---|---|---|
| CCRE | command-line | Run Copy Create Utility |

```
W8CCRE - Copy Create Utility (1.2.00/1.2.00)
For TP specify: TP-DMIV, TP-C74 or TP-C85
Member-name Library-tag
```

| Name | Argument(s) | Function |
|---|---|---|
| CEDT | command-line | Run Copy Edit Utility |

```
W8CEDT - Copy Edit Utility (Find->Editor) (1.2.00/1.2.00)
Mask=?* (name), Tag=.L
For TP specify: TP-DMIV, TP-C74 or TP-C85
Member-name [Library-tag]
```

| Name | Argument(s) | Function |
|---|---|---|
| CDIR | command-line | Run Copy Directory Utility |

```
W8CDIR - Copy Directory Utility (1.2.00/1.2.00)
Mask=?* (name)
For TP specify: TP-DMIV, TP-C74 or TP-C85
Defaults: output=CON:
[Copy-member-mask] Library-tag [-Output dev/file] [-Pause]
```

| Name | Argument(s) | Function |
|---|---|---|
| CCOUT | command-line | Run Copy Checkout Utility |

```
W8CCOUT - Check-out Copy-member Utility (1.2.00/1.2.00)
Library-tag Member-name [-UD userdir]
```

| Name | Argument(s) | Function |
|---|---|---|
| CCIN | command-line | Run Copy Checkin Utility |

```
W8CCIN - Check-in Copy-member Utility (1.2.00/1.2.00)
Library-tag Member-name [-UD userdir]
```

## Dispatch8

| Name | Argument(s) | Function |
|---|---|---|
| D8E | command-line | Run Dispatch8 Print Executive |

```
W8D8E - DISPATCH8 Executive Utility (1.2.00/1.2.00)
List   [logical-printer-no] [form-no] [-Waiting|-Completed]
Print  logical-printer-no form-no [-First|-All|-Specific ddd [hhmmsshh]]
Delete logical-printer-no [form-no] [-Completed|-All|-Specific ddd [hhmmsshh]]
```

## Miscellaneous

| Name | Argument(s) | Function |
|---|---|---|
| BATCH | file-name | Run G8WB command file |
| LIBRARY | command-line | Run MF Library Utility |
| DFED | command-line | Run MF Data File Editor |
| DIFF | command-line | Run MF Source Comparison Utility |
| WFL | command-line | Run MF Workbench File Loader |
| HEXEDIT | command-file | Run MF Hex Editor |
| EDIT | command-file | Run MF Editor |
| COLOR | command-file | Run MF Color Utility |
| MFWB | | Start Micro Focus Workbench (PC development) |
| XDOS | command-line | Execute OS command with $symbol replace |
| XBAT | command-line | Execute OS command file with $symbol  replace |
| BCF | command-file | Build OS/WB command file utility |

```
W8BCF - Build Command File Utility (1.2.00/1.2.00)
Defaults: Ext(I)=SCB, Ext(O)=CBT
Symbols: %F% %P% %N% %E% %NE% %9[+]% %Z[+]%
File-file-mask Command-skeleton[.ext] [Command-file.[ext]]
               [-File] [-Append] [-Serial nnnn] [-Trailer]
```

| Name | Argument(s) | Function |
|---|---|---|
| STRIP | command-file | Strip COBOL line numbers |

```
W8STRIP - Strip Line numbers Utility (1.2.00/1.2.00)
Defaults: ext=.CBL, Original file->.BAK
Source-filename[.ext] [-Cobol|-Makecobol]
```

| Name | Argument(s) | Function |
|---|---|---|
| DOCSA | command-file | Adjust for national character set |

```
W8DOCSA - Apply Character-Set-Adjust to Text-file (1.2.00/1.2.00)
Text-filename -Import|-Export
```

| Name | Argument(s) | Function |
|---|---|---|
| FCRLF | command-file | Fix “funny” CR-LF in text file |

```
W8CRLF - Fix CrLf for Line Sequential (1.2.00/1.2.00)
Input-filename Output-filename
```

## Test suites

Test suites are like GCOS-8 Batch jobs, where multiple programs are tested in a pre-defined sequence. Test suites are possible either with OS or WB command files, however, conditional constructs are only available with OS command files.

Following illustrate an  OS command file:

```
rem My example test suite
rem
rem Restore TESTSCM database
XM WB W8 DBRES testscm samples-reference
rem
rem Execute BTEST2
XM WB W8 RUNPRG btest2
rem
rem Reset PSW’s
XM WB W8 PSWS -99
rem
rem Execute BTEST3
XM WB W8 RUNPRG btest3
rem
rem Test Swicth 21
XM WB W8 PSWT 21
IF ERRORLEVEL 1 GOTO label2
rem
rem Switch 21 is OFF
:label1
rem
rem Backup TESTSCM database
XM WB W8 DBBCK testscm workpac
rem
rem Switch 21 is ON
:label2
```

## Bulk compiles

Bulk compiles are possible either with OS or WB command files. Following example illustrate a WB command file:

```
CHKCBL74 program1
CHKCBL74 c:\source\p1\program2
CHKCBL74 $p1\program3
```

Checker will look for program.CBL (CBL is default) in the current directory, unless you explicitly give the full file specification. If you specify only program, then the WB command file can only be launched from the directory where the source is located.

Specifying the full file specification is more flexible, as the WB command file can be lanched from any directory, however in order not to have explicit drive and directory information in WB command file, you should the $symbol to assign a logical full file specification. If you define SET P1=c:\source\p1, $p1 will thus dynamically be expanded, and this setting can be changes later without impacting the WB command file.

Checker will place related files, i.e. INT, IDY, LST etc., in the same directory where the program.CBL file is found.

Although above WB command file makes it quite easy to bulk compile programs, it gives you no flexible way of checking where any or all compiles went ok.

## Error Log

G8WB provides the Compile Error Log feature to ease verification of bulk compiles. To build the error log, each check command must be followed by the UELOG command, which will read the results from checking and update the error log accordingly.

The error log is always named CERROR.LOG, and it is placed in the same directory where the program.CBL file is found. The WB command file in above example would thus need to be amended with UELOG commands as follows:

```
CHKCBL74 program1
UELOG program1
CHKCBL74 c:\source\p1\program2
UELOG c:\source\p1\program2
CHKCBL74 $p1\program3
UELOG $p1\program3
```

You would then be able to verify the results from the bulk compile, by viewing the CERROR.LOG file. Following illustrate its format:

```
program2   yy.mm.dd hh.mm.ss
COBOL/2    Compiled with errors
           COPY “c:\source\p1\program2.LST”.
```

In above example, program2 did not compile clean and was thus logged into the CERROR.LOG file. The COPY line provides a pointer to the compile listing, and you may open this file using ALT-F2, while you have CERROR.LOG loaded in the Editor. You may then browse the listing and the error messages to determine the cause of compile failure. Browsing function is available from the Compile Log Facility menu.

While browsing within the LST file, you may open another window with the CBL file and make correction as you go along. Once you Escape out of the LST file, you return to the CERROR.LOG file and may thus continue with the next entry in the log.

You may automate the processing of CERROR.LOG, through the Process function of the Compile Facility menu. Instead of bringing CERROR.LOG into the Editor, you may instead process each entry in the log one by one. For each entry, the Editor will be invoked just like you were returning from interactive checking. The CBL file or whatever CPY file containing the first error, will be opened with positioning at the first error encountered and the error message at the bottom of the screen. You can continue browsing through error using F2-COBOL and F7/F8/F9 keys to navigate between errors. You may make required modications to the source as you go along. Once you Escape out of the Editor, you will move on to the next entry in the log.

CERROR.LOG must be explicitly deleted, as UELOG command will always append to the file, except when the file is initially created. The G8WB command DELOG may be used to delete the log. Typically you would place the DELOG command in front of your WB command file, so you always start with a fresh CERROR.LOG.

```
DELOG $p1
CHKCBL74 $p1\program1
UELOG $p1\program1
CHKCBL74 $p1\program2
UELOG $p1\program2
CHKCBL74 $p1\program3
UELOG $p1\program3
```

## Building command files

G8WB provides a simple Build Command File Utility, which ease the task of building WB command files as described in above examples. The G8WB command BCF may be used to invoke the Build Command File Utility.

BCF work on the basis of a list and a command skeleton. Using above example, we could create a command skeleton named CHECK.SCB with the following content:

```
$HEADER
DELOG %P%
$DETAIL
CHKCBL74 %F%
UELOG %F%
```

We would the be able to create the WB command file as follows:

**XM WB W8 BCF *.cbl check**

The resulting WB command file named CHECK.CBT would look as follows:

```
DELOG c:\source\p1
CHKCBL74 c:\source\p1\PROGRAM1.CBL
UELOG c:\source\p1\PROGRAM1.CBL
CHKCBL74 c:\source\p1\PROGRAM2.CBL
UELOG c:\source\p1\PROGRAM2.CBL
CHKCBL74 c:\source\p1\PROGRAM3.CBL
UELOG c:\source\p1\PROGRAM3.CBL
```

# Chapter 4 FORMS DEVELOPMENT MENU

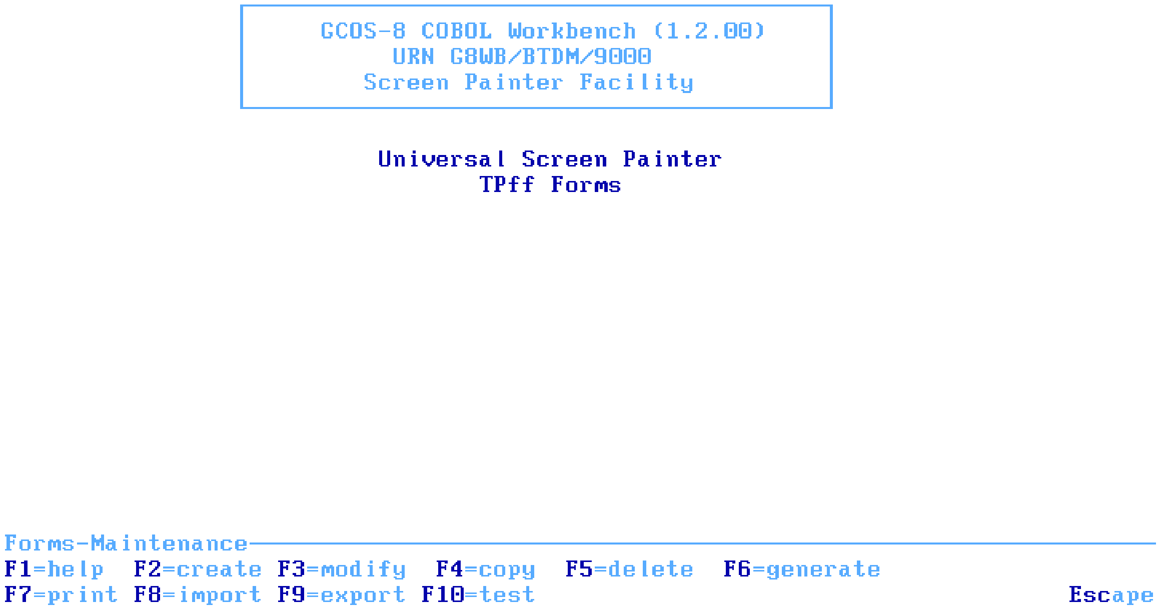
The Forms development menu is your place of work for painting, editing, importing, exporting and generating application forms. G8WB currently support TPFF, FORMAT, DIMS8, Pilspråk, SOLD and TSM8 forms software.

The Workbench Universal Screen Painter (WUSP), which may be used for all supported forms software, provides a user-friendly, comprehensive and flexible tool for forms development and maintenance. Customized panels are used to support specific forms software features and attributes. Existing GCOS-8 forms may be imported and once development or maintenance activities are complted, forms may be exported back to GCOS-8. WUSP make forms development and maintenance substantially more productive than working with tools in the GCOS-8 environment.

## Generic

The basic principles of forms maintenance are the same and WUSP work functionally equivalent for all supported forms software. Some features and attributes may be specific to some forms software, and WUSP will dynamically adjust function menu's to reflect this, and provide you with Forms specific panels, once you get to field attribute level. You need to configure the Forms software you use, ref. Configuration chapter for details, before you start working with WUSP.

Forms Development menu is available through F4-Forms from the main menu. In the following, TPFF mode has been used to illustrate the generic features of WUSP.



```
                 GCOS-8 COBOL Workbench (1.2.00)
                      URN G8WB/BTDM/9000
                    Screen Painter Facility


                      Universal Screen Painter
                             TPff Forms










Forms-Maintenance────────────────────────────────────────────────────────────────
F1=help  F2=create F3=modify  F4=copy  F5=delete  F6=generate
F7=print F8=import F9=export F10=test                                      Escape
```

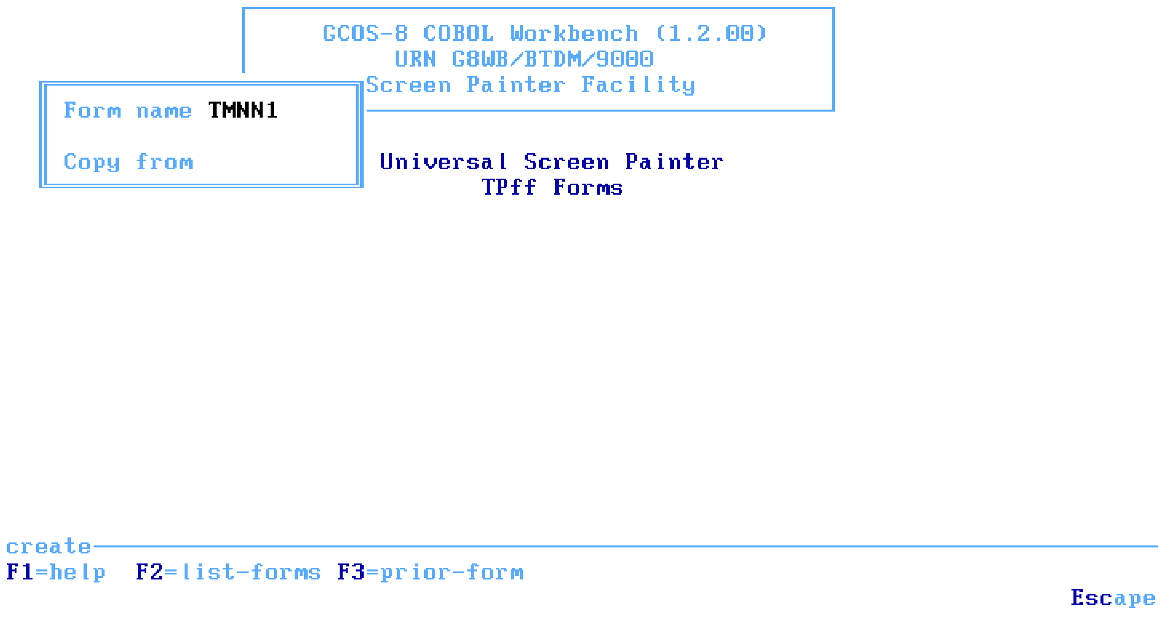
F1-Help key may be used request help information in most of the WUSP menu's.

### Painting & Editing

When you develop a new Form you have three ways of doing this:

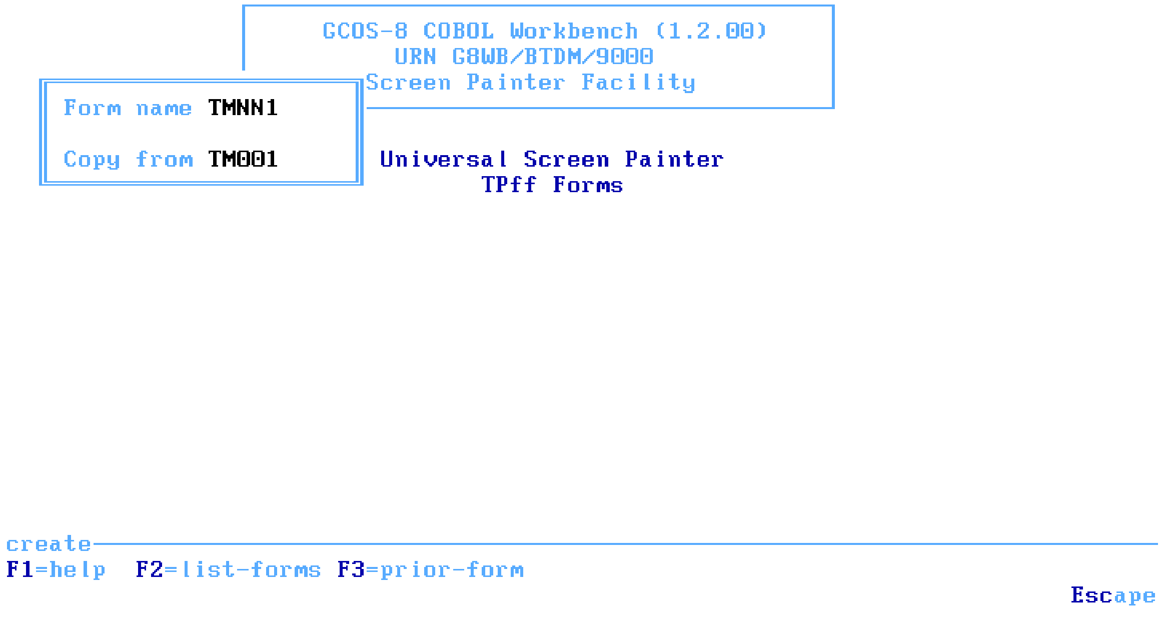
- Painting from scratch
- Copy an existing Form and re-paint it
- Copy an existing Form

F2-Create may be used to create a new form. After you specify the form name, i.e 1 to 8 characters, you are able to paint a new form from scratch.



```
                      GCOS-8 COBOL Workbench (1.2.00)
                           URN G8WB/BTDM/9000
 Form name TMNN1      Screen Painter Facility

 Copy from             Universal Screen Painter
                               TPff Forms

create
F1=help  F2=list-forms F3=prior-form
                                                               Escape
```

F2-Create may also be used to create a new form based on an existing form. You then specify both the new form name and the form name you wish use as the basis for a new form. Instead of starting from scratch, the painting panel will be pre-filled with all the fields from the basis form.

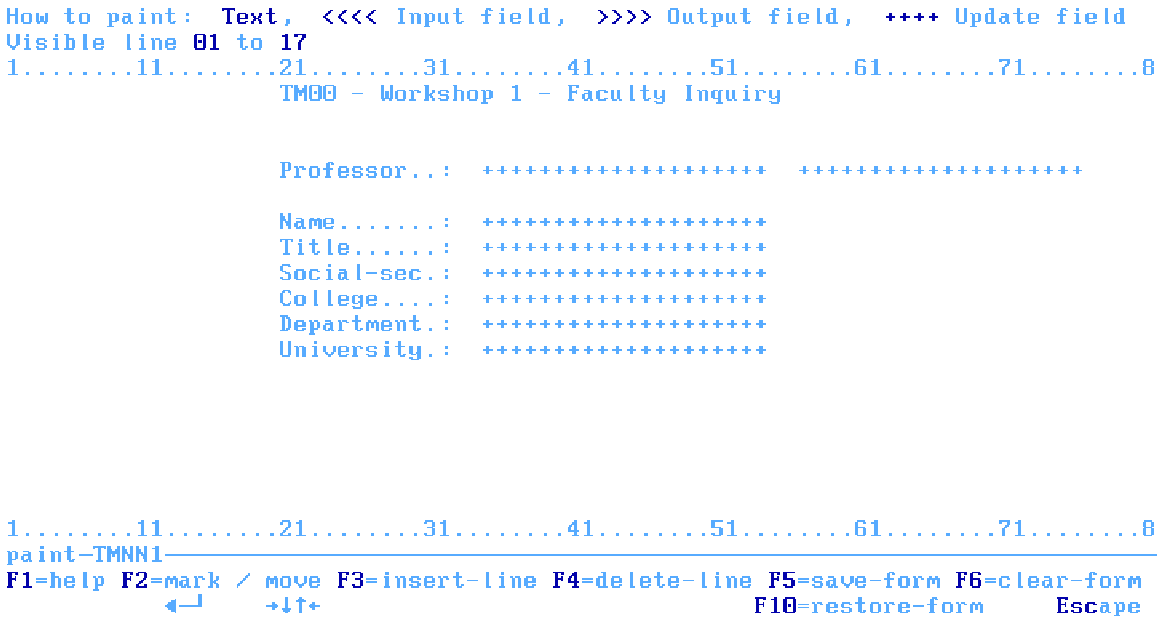


```
                      GCOS-8 COBOL Workbench (1.2.00)
                           URN G8WB/BTDM/9000
 Form name TMNN1      Screen Painter Facility

 Copy from TM001       Universal Screen Painter
                               TPff Forms

create
F1=help  F2=list-forms F3=prior-form
                                                               Escape
```

The basic painting rules are documented in the painting panel. WUSP support following basic field types:

- Text, entered asis
- <...<, specify Input field and implied length
- >...>, specify Output field and implied length
- +...+, specify Update (Input and Output) field and implied length

Although some forms software may not support all basic field types directly, i.e TPFF only supported Text and Named fields, the classification as Input or Output fields is used by WUSP to assign different default attributes, i.e. Input fields have UNPROTECTED attribute whereas Output fields have PROTECTED attribute as default. For some forms software there is no difference between Input and Update fields.



```
How to paint:  Text,  <<<< Input field,  >>>> Output field,  ++++ Update field
Visible line 01 to 17
1.......11.......21.......31.......41.......51.......61.......71.......8
                 TM00 - Workshop 1 - Faculty Inquiry


                 Professor..:  ++++++++++++++++++++  ++++++++++++++++++++

                 Name.......:  ++++++++++++++++++++
                 Title......:  ++++++++++++++++++++
                 Social-sec.:  ++++++++++++++++++++
                 College....:  ++++++++++++++++++++
                 Department.:  ++++++++++++++++++++
                 University.:  ++++++++++++++++++++





1.......11.......21.......31.......41.......51.......61.......71.......8
paint-TMNN1────────────────────────────────────────────────────────────────
F1=help F2=mark / move F3=insert-line F4=delete-line F5=save-form F6=clear-form
           ◄┘      →↓↑←                               F10=restore-form    Escape
```

You should always try to get initial painting as close to the end result as possible. Although WUSP provides features to align and move fields around during modification, this is done field by field, and you cannot go back to re-paint once you have created a form.

During initial painting you have following editing possibilities:

F3-Insert-line

F4-Delete-line

INS, set/reset insert mode for typing

DEL, delete character

F6-Clear-form, restart from scratch

The painting panel is scrollable both for lines and columns, and you may exceed the normal limits of 24 lines and 80 columns during painting. You should, however, make sure that your form is within these limits before you finish painting.

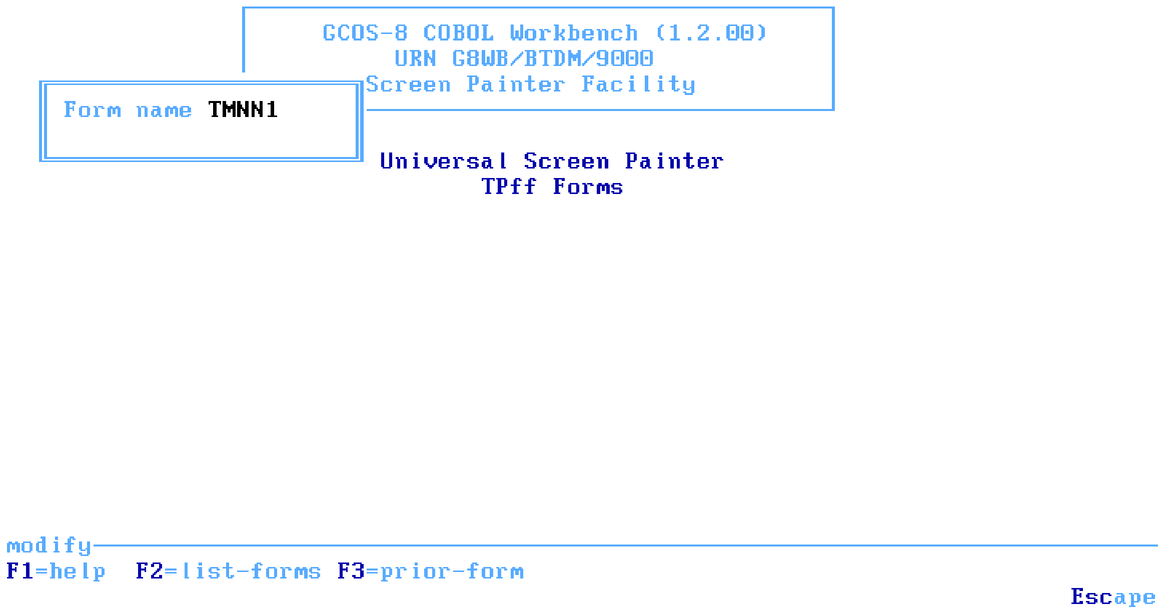
For repeated fields, i.e. columns oriented (rows) or line oriented (array), you should only paint one occurrence (the first). Once you enter form modify function, you will be able to insert repeat structures around fields.

You need to use F5-Save-form before you exit painting through Escape. You may perform intermediate saves during painting, i.e. save what you have painted until now, and you may revert to a previous save through F10-Restore-form (one level only).

Once you have finished painting, Escape will take you into the Form modify function, just like you had just requested modify function for an existing form.

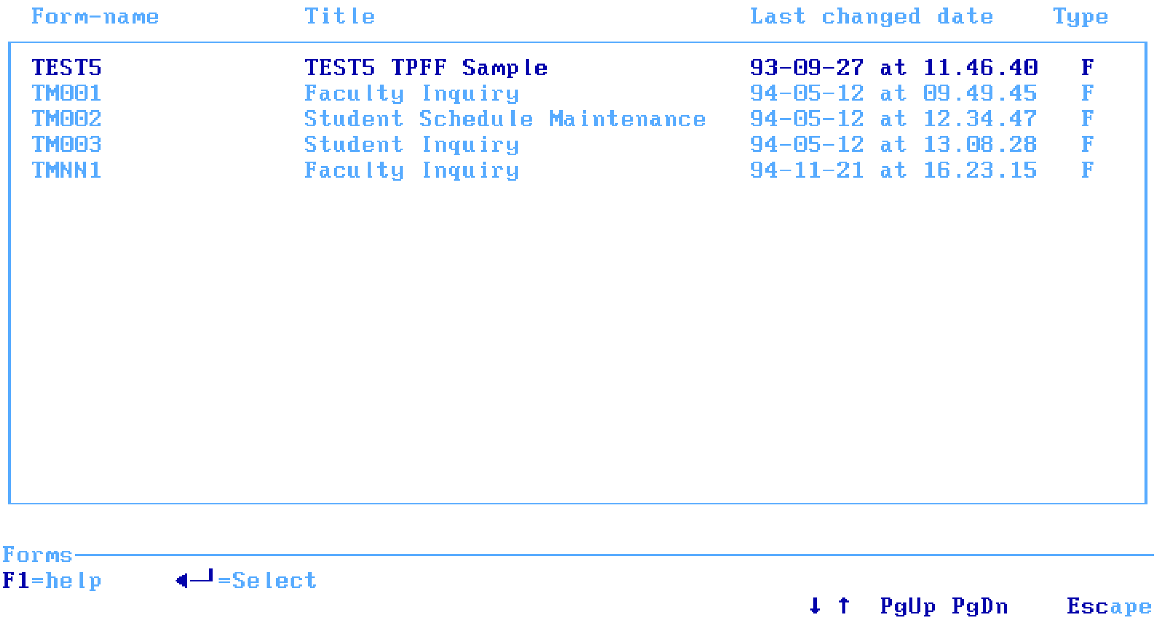
## Modifying

F3-Modify may be used to modify an existing new form.



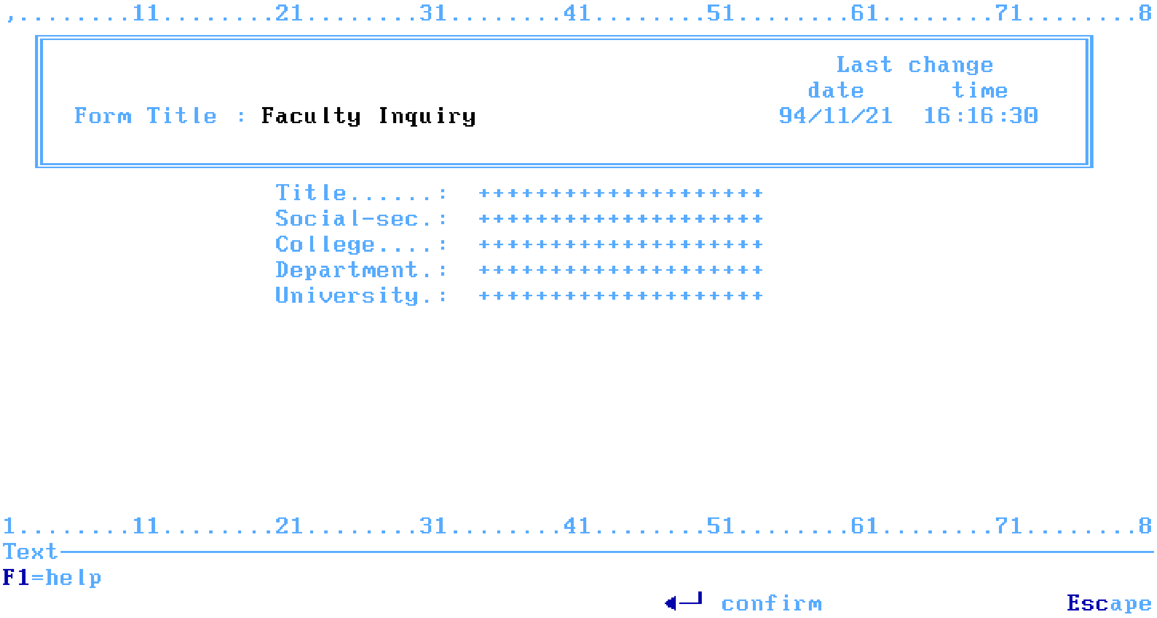
```
                       GCOS-8 COBOL Workbench (1.2.00)
                            URN G8WB/BTDM/9000
 Form name  TMNN1          Screen Painter Facility

                            Universal Screen Painter
                                   TPff Forms

modify
F1=help  F2=list-forms F3=prior-form
                                                               Escape
```

F2-List-forms may be used to obtain an list of existing forms and to select a form from the list. You may specify a form name mask, i.e. * or ?, to limit the list.



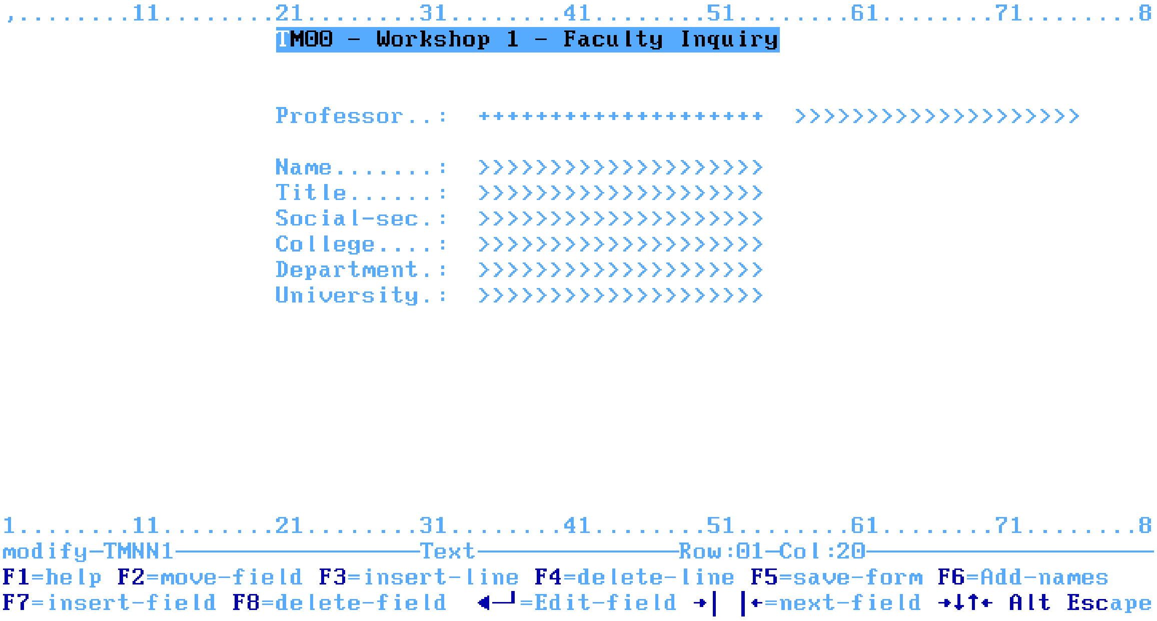
```
  Form-name           Title                         Last changed date     Type

  TEST5               TEST5 TPFF Sample             93-09-27 at 11.46.40    F
  TM001               Faculty Inquiry               94-05-12 at 09.49.45    F
  TM002               Student Schedule Maintenance  94-05-12 at 12.34.47    F
  TM003               Student Inquiry               94-05-12 at 13.08.28    F
  TMNN1               Faculty Inquiry               94-11-21 at 16.23.15    F

Forms
F1=help     ◄┘=Select
                                                ↓ ↑  PgUp PgDn     Escape
```

Various general information and attributes may apply to a form. When you enter modify function, the Title panel may be used to define or modify such general information and attributes. The Title panel is specific to the forms software currently configured, i.e. TPFF only has a form title attribute.



```
,.......11.......21.......31.......41.......51.......61.......71.......8
                                                   Last change
                                                 date      time
    Form Title : Faculty Inquiry               94/11/21  16:16:30

                 Title......:  ++++++++++++++++++++
                 Social-sec.:  ++++++++++++++++++++
                 College....:  ++++++++++++++++++++
                 Department.:  ++++++++++++++++++++
                 University.:  ++++++++++++++++++++

1.......11.......21.......31.......41.......51.......61.......71.......8
Text────────────────────────────────────────────────────────────────────
F1=help
                                          ◄┘ confirm                Escape
```

Once into form modify, a simulated view of the form is presented in a scrollable panel. A cursor is used to indicated you current position within the form. A visual cursor marker is shown in the view panel and information on line and column is shown in the menu panel. If you are currently positioned in a field, i.e. default positioning is first field, the field is highlighted and the field type in indicated in the menu panel.



```
,.......11.......21.......31.......41.......51.......61.......71.......8
                 TM00 - Workshop 1 - Faculty Inquiry

                 Professor..:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

                 Name.......:  >>>>>>>>>>>>>>>>>>>>
                 Title......:  >>>>>>>>>>>>>>>>>>>>
                 Social-sec.:  >>>>>>>>>>>>>>>>>>>>
                 College....:  >>>>>>>>>>>>>>>>>>>>
                 Department.:  >>>>>>>>>>>>>>>>>>>>
                 University.:  >>>>>>>>>>>>>>>>>>>>

1.......11.......21.......31.......41.......51.......61.......71.......8
modify-TMNN1────────────────Text─────────────Row:01-Col:20──────────────
F1=help F2=move-field F3=insert-line F4=delete-line F5=save-form F6=Add-names
F7=insert-field F8=delete-field  ◄┘=Edit-field →| |←=next-field →↓↑← Alt Escape
```

You are NOT able to type in the view panel, i.e. like paint panel, however you have the following basic editing possibilities:

F3-Insert-line

F4-Delete-line, empty line or line including fields

INS, set/reset insert mode for SPACEBAR

DEL, delete character

Besides Arrow Up/Down and Page Up/Down, you may use TAB (next field) and Shift-TAB (prior field) to move around in the form.

The view panel is scrollable both for lines and columns, and you may exceed the normal limits of 24 lines and 80 columns during editing. You should, however, make sure that your form is within these limits before you finish editing.

You have following other editing possibilities:

Return/Enter, edit current field

F7-Insert-field

F8-Delete-field

F2-Move-field

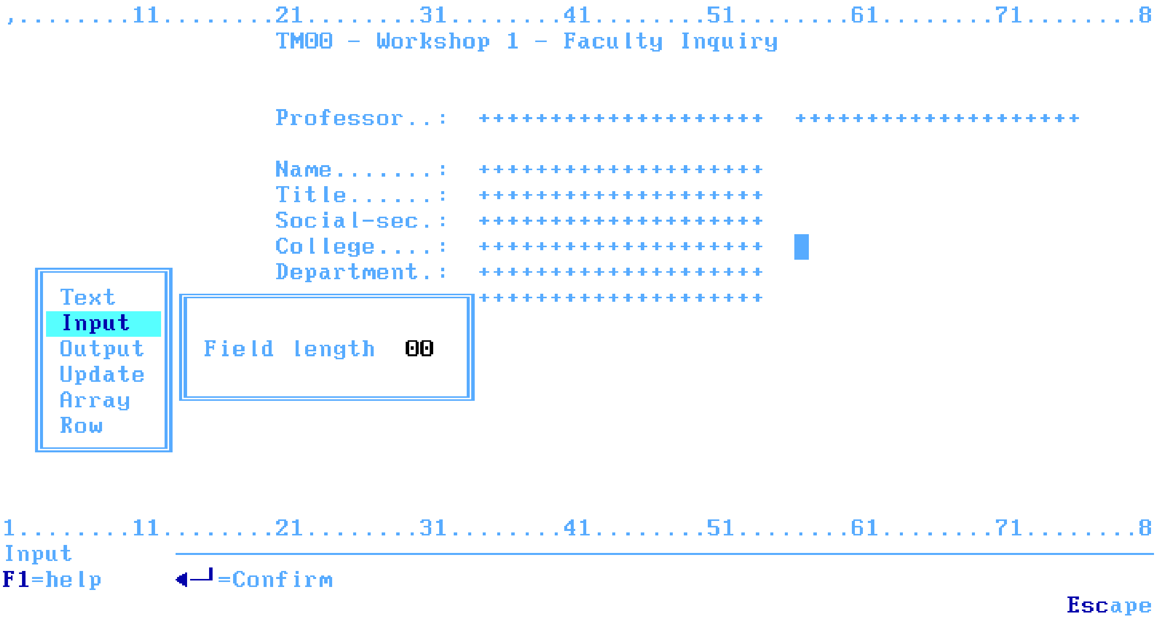
F6-Add-names, edit all fields one by one

Alt-F2-Reenter-title, go back to Title panel

Once you have completed editing, you need to make changes permanent through F5-Save-form. You may perform intermediate saves as required during editing. If you attempt to exit without save, you have the option to save or discard the changes you have made.

## Insert Field

May be used to insert a new field at the current position.



```
,........11........21........31........41........51........61........71........8
                   TM00 - Workshop 1 - Faculty Inquiry

                   Professor..:  ++++++++++++++++++++  ++++++++++++++++++++

                   Name.......:  ++++++++++++++++++++
                   Title......:  ++++++++++++++++++++
                   Social-sec.:  ++++++++++++++++++++
                   College....:  ++++++++++++++++++++  █
   Text            Department.:  ++++++++++++++++++++
   Input                         ++++++++++++++++++++
   Output     Field length  00
   Update
   Array
   Row

1........11........21........31........41........51........61........71........8
Input
F1=help     ◄┘=Confirm
                                                                          Escape
```

For Text fields, you need to type the text asis in the pop-up panel.

For Input, Output and Update fields, you need to specify the field length in the pop-up panel. To assign other attributes, you need to use Edit field after insert, i.e. Enter.

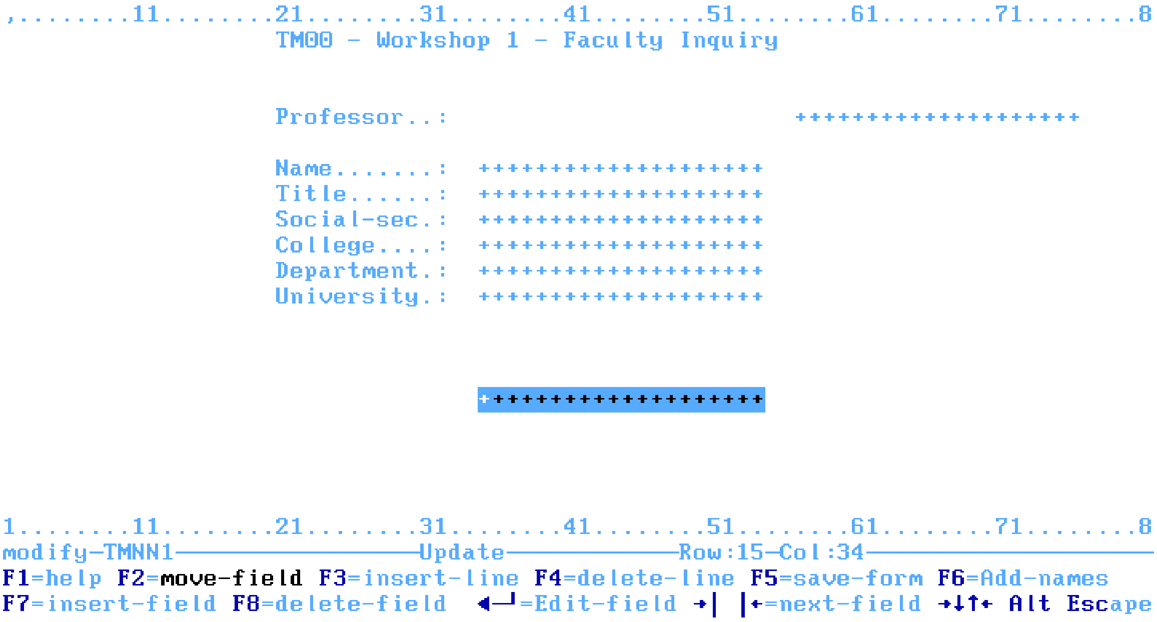
For Array fields, you need to specify the number of lines to repeat and the number of repeat occurences.

For Row fields, you need first to indicate the number of columns, i.e. move cursor to end of repeat, and then the number of occurences.

Array and Row setting may later be changed through Edit field, i.e. Enter, when you are positioned in the Array or Row field. Deleting an Array or Row field will not affect fields inside repeat, except that they are no longer repeated.

## Move Field

May be used to move any field, except Array and Row, to a new location. F2-Move-field is used both to activate and de-activate move operation. When you activate, the current field may be moved with cursor keys Up/Down/Right/Left. Once you are at the new location, you de-activate move operation again.

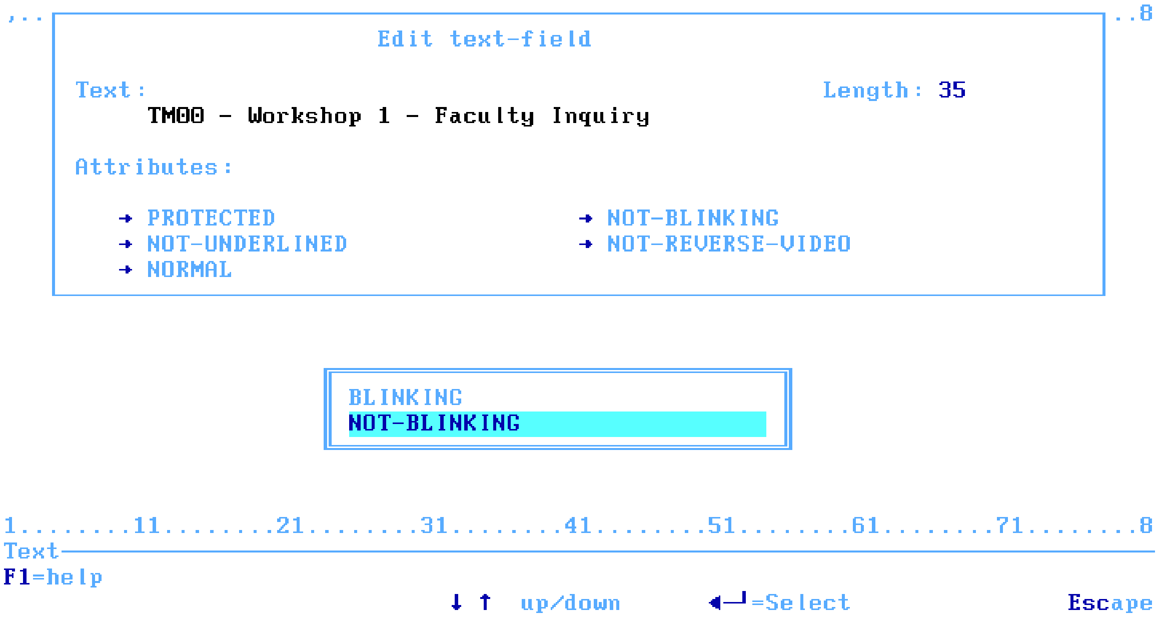


```
,........11........21........31........41........51........61........71........8
                   TM00 - Workshop 1 - Faculty Inquiry


                   Professor..:                        ++++++++++++++++++++

                   Name.......:  ++++++++++++++++++++
                   Title......:  ++++++++++++++++++++
                   Social-sec.:  ++++++++++++++++++++
                   College....:  ++++++++++++++++++++
                   Department.:  ++++++++++++++++++++
                   University.:  ++++++++++++++++++++




                                 ++++++++++++++++++++




1........11........21........31........41........51........61........71........8
modify-TMNN1─────────────────Update────────────Row:15─Col:34────────────────────
F1=help F2=move-field F3=insert-line F4=delete-line F5=save-form F6=Add-names
F7=insert-field F8=delete-field  ◄─┘=Edit-field →| |←=next-field →↓↑← Alt Escape
```

“Move-field” is highlighted in the menu panel, when active.

## Text Field

Edit function may be used to change the actual text and the attributes assigned to a Text field. The maximum text length is 40 characters, and painted text fields > 40 characters will automatically be split in two fields. You may use TAB, Shift-TAB and cursor keys to move around in the edit panel. Attribute assignment may be changed through a pop-up list, which is activated using Enter key.

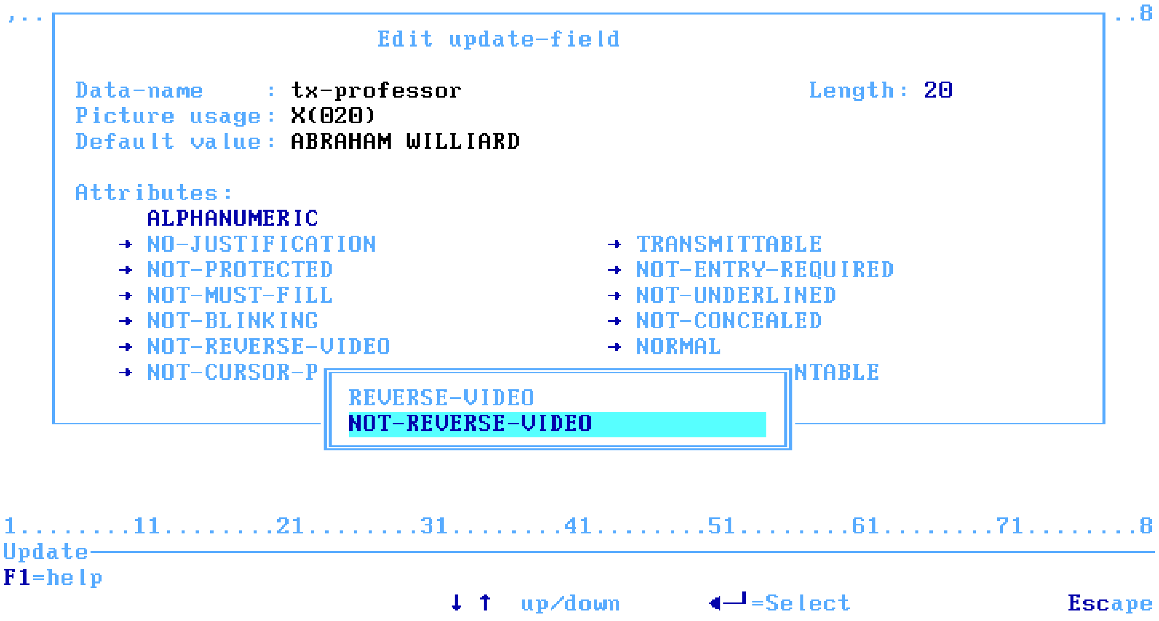


```
,..                                                                                    ..8
                        Edit text-field

   Text:                                                 Length: 35
        TM00 - Workshop 1 - Faculty Inquiry

   Attributes:

      → PROTECTED                       → NOT-BLINKING
      → NOT-UNDERLINED                  → NOT-REVERSE-VIDEO
      → NORMAL


                     BLINKING
                     NOT-BLINKING


1........11........21........31........41........51........61........71........8
Text
F1=help
                                 ↓ ↑  up/down      ↵=Select            Escape
```

Once you have completed editing, Escape will take you back to the main editing menu.

## Input Field

Edit function may be used to assign or change attributes for an Input field. You may use TAB, Shift-TAB and cursor keys to move around in the edit panel. Attribute assignment may be changed through a pop-up list, which is activated using Enter key.



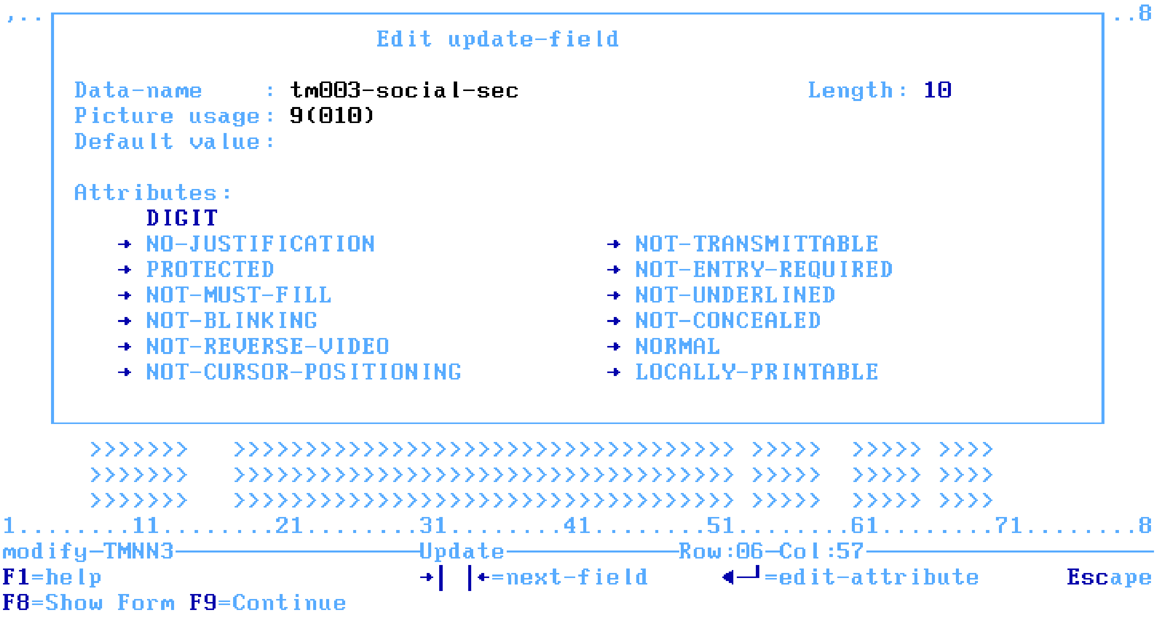
```
,..                                                                                    ..8
                        Edit update-field

   Data-name     : tx-professor                        Length: 20
   Picture usage : X(020)
   Default value : ABRAHAM WILLIARD

   Attributes:
        ALPHANUMERIC
      → NO-JUSTIFICATION                  → TRANSMITTABLE
      → NOT-PROTECTED                     → NOT-ENTRY-REQUIRED
      → NOT-MUST-FILL                     → NOT-UNDERLINED
      → NOT-BLINKING                      → NOT-CONCEALED
      → NOT-REVERSE-VIDEO                 → NORMAL
      → NOT-CURSOR-P                     NTABLE
                     REVERSE-VIDEO
                     NOT-REVERSE-VIDEO


1........11........21........31........41........51........61........71........8
Update
F1=help
                                 ↓ ↑  up/down      ↵=Select            Escape
```

Once you have completed editing, Escape will take you back to the main editing menu.

*Note: TPFF treat Input fields like Update fields.*

## Output Field

Edit function may be used to assign or change attributes for an Output field. You may use TAB, Shift-TAB and cursor keys to move around in the edit panel. Attribute assignment may be changed through a pop-up list, which is activated using Enter key.



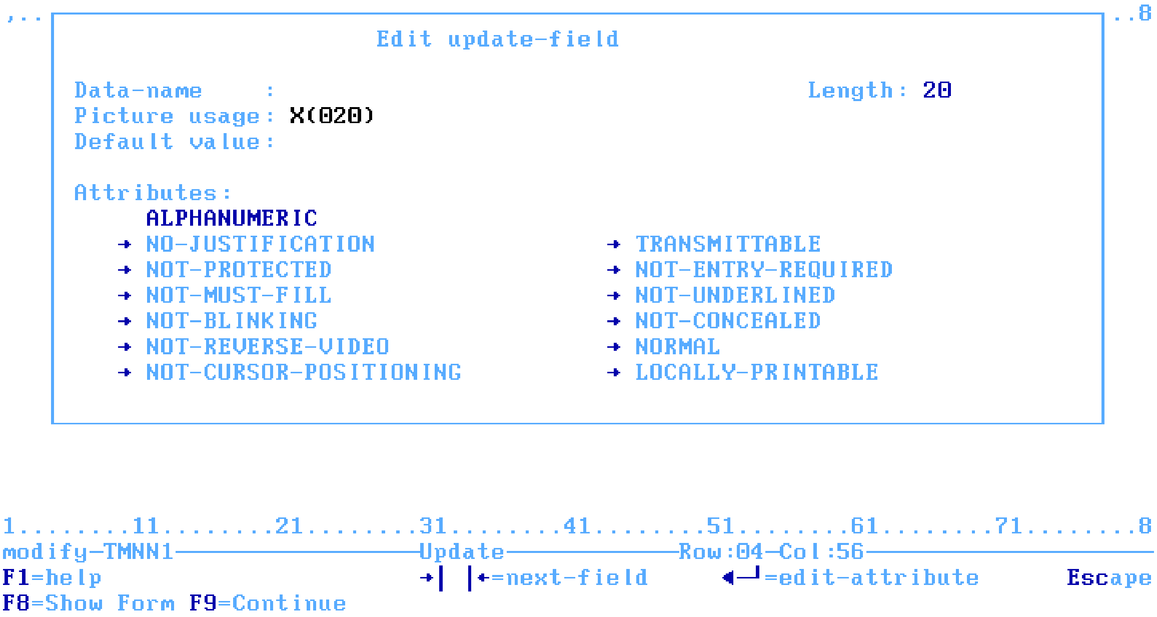
```
,..                                                                  ..8
            Edit update-field

   Data-name     : tm003-social-sec                    Length: 10
   Picture usage: 9(010)
   Default value:

   Attributes:
        DIGIT
      → NO-JUSTIFICATION                  → NOT-TRANSMITTABLE
      → PROTECTED                         → NOT-ENTRY-REQUIRED
      → NOT-MUST-FILL                     → NOT-UNDERLINED
      → NOT-BLINKING                      → NOT-CONCEALED
      → NOT-REVERSE-VIDEO                 → NORMAL
      → NOT-CURSOR-POSITIONING            → LOCALLY-PRINTABLE

     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN3──────────────────Update───────────Row:06─Col:57────────────────────
F1=help                        →| |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

Once you have completed editing, Escape will take you back to the main editing menu.

*Note: TPFF treat Output fields like Update fields, except for PROTECTED and NON-TRANSMITTABLE attribute setting.*

## Update Field

Edit function may be used to assign or change attributes for an Update field. You may use TAB, Shift-TAB and cursor keys to move around in the edit panel. Attribute assignment may be changed through a pop-up list, which is activated using Enter key.



```
,..                                                                  ..8
            Edit update-field

   Data-name     :                                     Length: 20
   Picture usage: X(020)
   Default value:

   Attributes:
        ALPHANUMERIC
      → NO-JUSTIFICATION                  → TRANSMITTABLE
      → NOT-PROTECTED                     → NOT-ENTRY-REQUIRED
      → NOT-MUST-FILL                     → NOT-UNDERLINED
      → NOT-BLINKING                      → NOT-CONCEALED
      → NOT-REVERSE-VIDEO                 → NORMAL
      → NOT-CURSOR-POSITIONING            → LOCALLY-PRINTABLE



1........11........21........31........41........51........61........71........8
modify-TMNN1──────────────────Update───────────Row:04─Col:56────────────────────
F1=help                        →| |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

Once you have completed editing, Escape will take you back to the main editing menu.

## Repeat Field

Repeat fields may be used to define Row (column oriented) or Array (line oriented) repeat structures. When using repeat fields, only one (the first) occurrence of  a field or fields, are painted or inserted. You next insert a repeat field in front of the field(s) to be repeated and specify the length of a repeat occurrence. WUSP will actually insert two repeat fields:

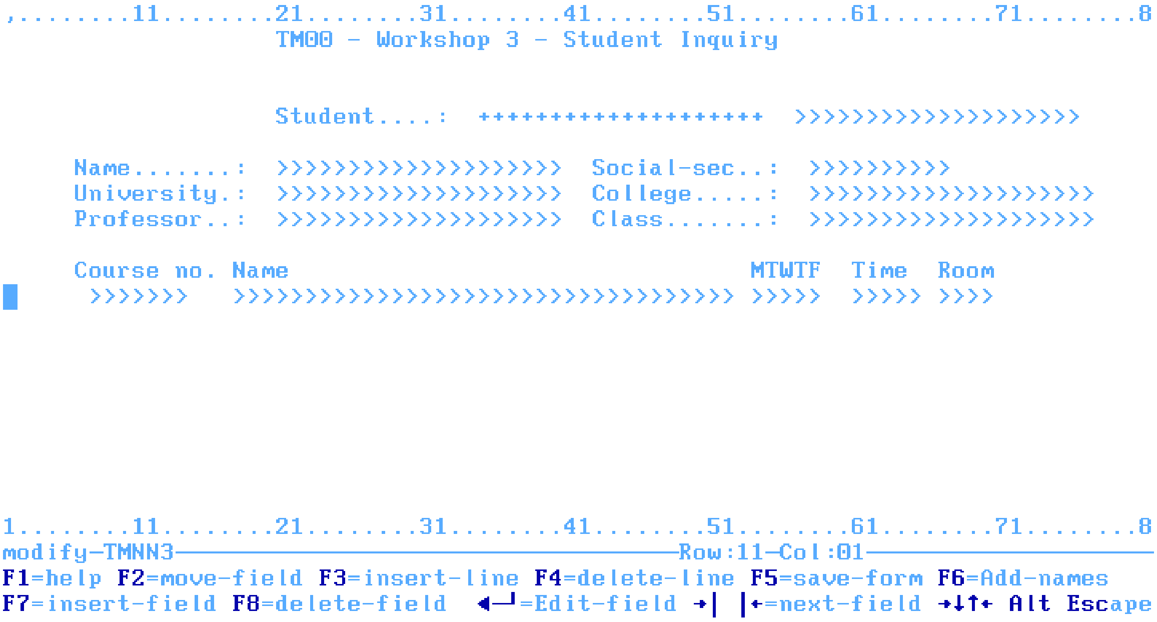
Repeat-start, signal start of a repeat structure at a specific position

Repeat-end, signal end of a repeat structure at a specific position

Between Repeat-start and Repeat-end you will have one or more Text, Input, Output or Update fields. As you move between fields, repeat-start end repeat-end are treated as separate fields. To Edit a repeat, either repeat-start or repeat-end can be used, i.e. Enter.

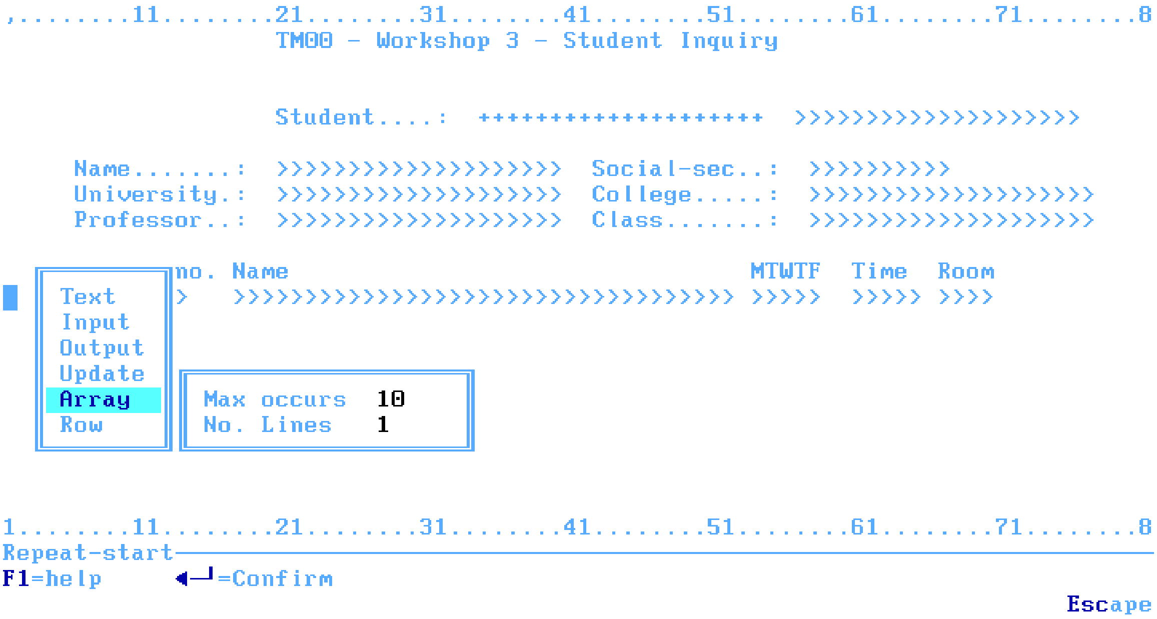
You may insert a repeat inside another repeat to create a two dimensional repeat structure, i.e. line and column oriented. Repeat structure posibilities may be limited by the current Forms software.

Following example illustrate a form with a line of fields that needs to be repeated.



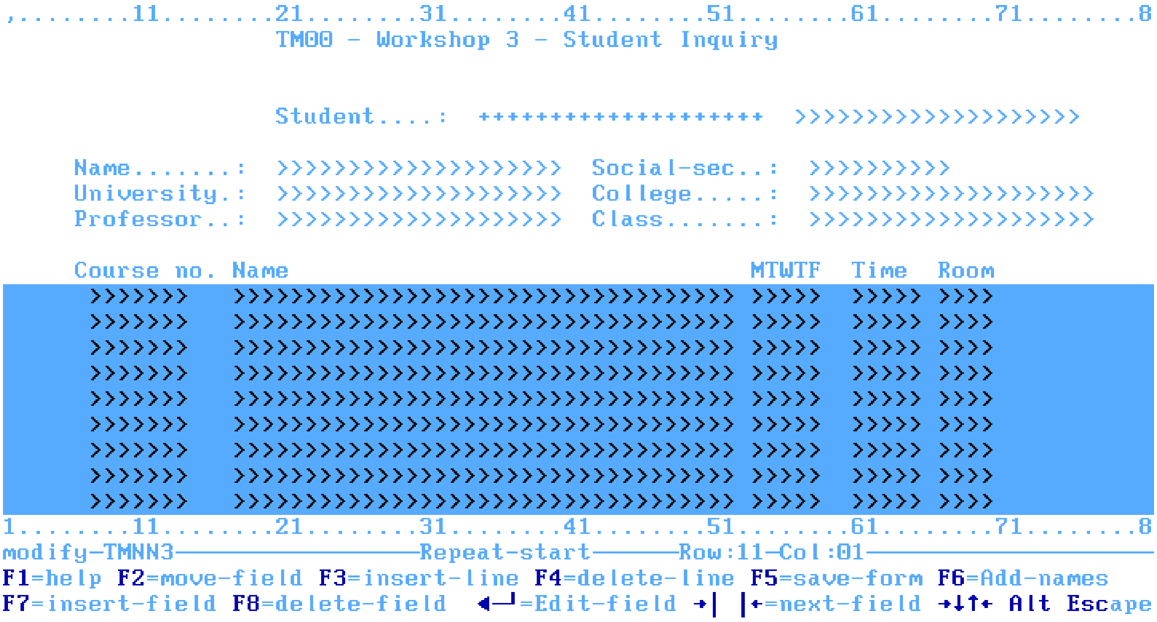
```
,........11........21........31........41........51........61........71........8
                   TM00 - Workshop 3 - Student Inquiry


                   Student....:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

     Name.......:  >>>>>>>>>>>>>>>>>>>>  Social-sec..:  >>>>>>>>>>
     University.:  >>>>>>>>>>>>>>>>>>>>  College.....:  >>>>>>>>>>>>>>>>>>>>
     Professor..:  >>>>>>>>>>>>>>>>>>>>  Class.......:  >>>>>>>>>>>>>>>>>>>>

     Course no. Name                                  MTWTF  Time  Room
█     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>





1........11........21........31........41........51........61........71........8
modify-TMNN3───────────────────────────────────Row:11-Col:01────────────────────
F1=help F2=move-field F3=insert-line F4=delete-line F5=save-form F6=Add-names
F7=insert-field F8=delete-field  ◄┘=Edit-field →| |←=next-field →↓↑← Alt Escape
```

To create the line oriented repeat, we need to position at the specific line in column 1, and then insert the repeat field.



```
,........11........21........31........41........51........61........71........8
                TM00 - Workshop 3 - Student Inquiry


                Student....:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

    Name.......:  >>>>>>>>>>>>>>>>>>>>  Social-sec..:  >>>>>>>>>>
    University.:  >>>>>>>>>>>>>>>>>>>>  College.....:  >>>>>>>>>>>>>>>>>>>>
    Professor..:  >>>>>>>>>>>>>>>>>>>>  Class.......:  >>>>>>>>>>>>>>>>>>>>

  ╔═══════╗no. Name                                MTWTF  Time  Room
█ ║ Text  ║>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
  ║ Input ║
  ║ Output║
  ║ Update║╔════════════════════╗
  ║ Array ║║ Max occurs   10    ║
  ║ Row   ║║ No. Lines    1     ║
  ╚═══════╝╚════════════════════╝


1........11........21........31........41........51........61........71........8
Repeat-start──────────────────────────────────────────────────────────────────────
F1=help     ◄┘=Confirm
                                                                            Escape
```

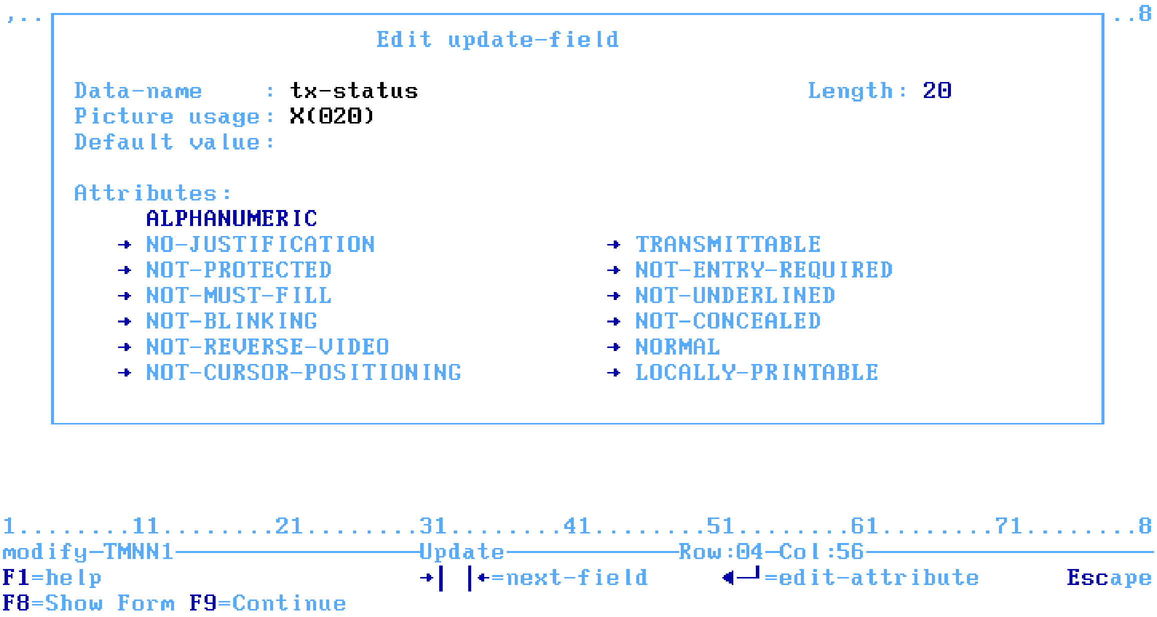
Array type must be used for a line-oriented repeat. Max occurs specify have many occurences and No. lines specify how many lines to repeat, i.e. each occurrence. The repeat is next inserted and Repeat-start becomes the current field. The entire repeat structure is thus highlighted.



```
,........11........21........31........41........51........61........71........8
                TM00 - Workshop 3 - Student Inquiry


                Student....:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

    Name.......:  >>>>>>>>>>>>>>>>>>>>  Social-sec..:  >>>>>>>>>>
    University.:  >>>>>>>>>>>>>>>>>>>>  College.....:  >>>>>>>>>>>>>>>>>>>>
    Professor..:  >>>>>>>>>>>>>>>>>>>>  Class.......:  >>>>>>>>>>>>>>>>>>>>

    Course no. Name                                MTWTF  Time  Room
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN3─────────────────Repeat-start─────Row:11-Col:01───────────────────────
F1=help F2=move-field F3=insert-line F4=delete-line F5=save-form F6=Add-names
F7=insert-field F8=delete-field  ◄┘=Edit-field →| |←=next-field →↓↑← Alt Escape
```

## Add names

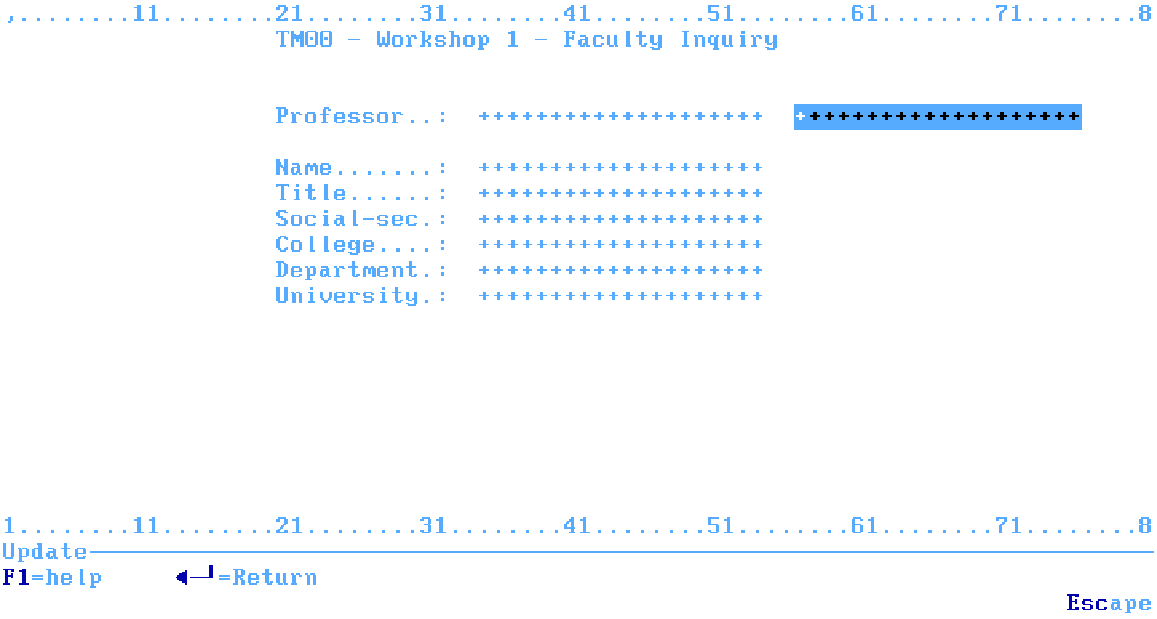
Add names may be used to assign/change attributes for all fields. Instead of stepping from field to field and then request Edit, Add names will automatically invoke Edit for each field one by one.



```
,..                                                                      ..8
    ┌──────────────────────────────────────────────────────────────────┐
    │                   Edit update-field                              │
    │                                                                  │
    │ Data-name     : tx-status                        Length: 20      │
    │ Picture usage: X(020)                                            │
    │ Default value:                                                   │
    │                                                                  │
    │ Attributes:                                                      │
    │      ALPHANUMERIC                                                │
    │    → NO-JUSTIFICATION                → TRANSMITTABLE             │
    │    → NOT-PROTECTED                   → NOT-ENTRY-REQUIRED        │
    │    → NOT-MUST-FILL                   → NOT-UNDERLINED            │
    │    → NOT-BLINKING                    → NOT-CONCEALED             │
    │    → NOT-REVERSE-VIDEO               → NORMAL                    │
    │    → NOT-CURSOR-POSITIONING          → LOCALLY-PRINTABLE         │
    │                                                                  │
    └──────────────────────────────────────────────────────────────────┘


1.......11.......21.......31.......41.......51.......61.......71.......8
modify-TMNN1────────────────Update───────────Row:04─Col:56──────────────
F1=help                      →| |←=next-field     ◄┘=edit-attribute     Escape
F8=Show Form F9=Continue
```

F9-Continue may be used to step to next field, as opposed to Escape, which will stop Add names function and take you back to the main editing menu.

If you loose track of where you are, i.e. Edit panel may overlap current position, you may use F8-Show form to get a view of where you are.



```
,.......11.......21.......31.......41.......51.......61.......71.......8
                   TM00 - Workshop 1 - Faculty Inquiry


                   Professor..:  ++++++++++++++++++++  ++++++++++++++++++++

                   Name.......:  ++++++++++++++++++++
                   Title......:  ++++++++++++++++++++
                   Social-sec.:  ++++++++++++++++++++
                   College....:  ++++++++++++++++++++
                   Department.:  ++++++++++++++++++++
                   University.:  ++++++++++++++++++++






1.......11.......21.......31.......41.......51.......61.......71.......8
Update──────────────────────────────────────────────────────────────────
F1=help     ◄┘=Return
                                                                   Escape
```

Return (Enter) will take you back to the Edit panel.

## Copying

F4-Copy may be used to copy an existing form. Copy will create a new form with all attributes preserved, whereas Create-from will not preserve attributes.

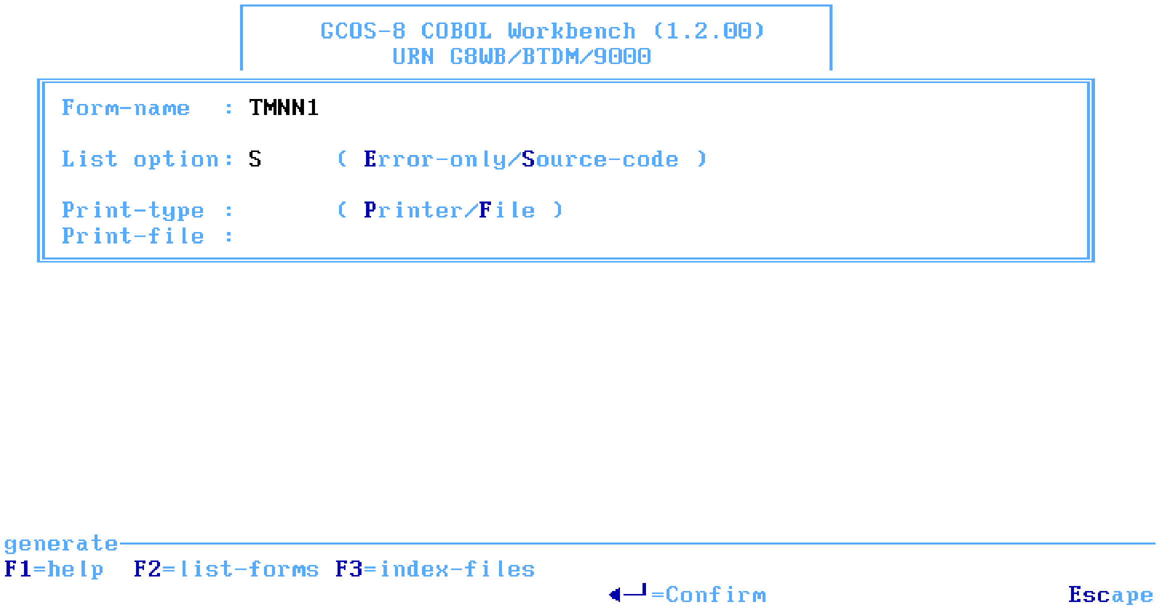
## Deleting

F5-Delete may be used to delete one or more forms. You may specify a form name mask, i.e. * and ?, to delete multiple forms.

## Generating

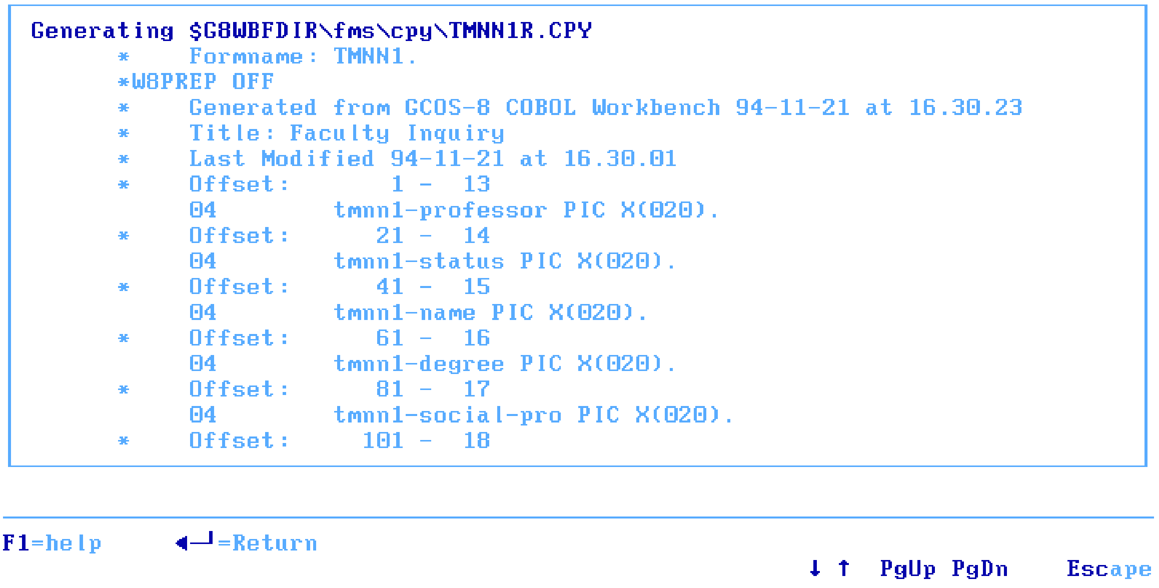
Most Forms software require a form to be generated, before you are able to use it in your application program. Generation will build one or more COBOL COPY's with a description of the form.

F6-Generate may be used to generate one or more forms.



```
                    GCOS-8 COBOL Workbench (1.2.00)
                        URN G8WB/BTDM/9000

   Form-name   : TMNN1

   List option: S       ( Error-only/Source-code )

   Print-type  :        ( Printer/File )
   Print-file  :


generate
F1=help  F2=list-forms F3=index-files
                                          ◄┘=Confirm                Escape
```

You may specify a form name mask, i.e. * and ?, to generate multiple forms. List option will, by default, present any errors detected during generation, in a browse panel. You may use S in List option to obtain a complete listing of the generated form, in a browse panel. Browse panel will use a temporary file, which is deleted when you exit generation. You may save the output from generation, through Print-type/File setting, i.e. direct output to a file or printer device. Following illustrate a browse panel:



```
Generating $G8WBFDIR\fms\cpy\TMNN1R.CPY
      *    Formname: TMNN1.
      *W8PREP OFF
      *    Generated from GCOS-8 COBOL Workbench 94-11-21 at 16.30.23
      *    Title: Faculty Inquiry
      *    Last Modified 94-11-21 at 16.30.01
      *    Offset:        1 -  13
           04        tmnn1-professor PIC X(020).
      *    Offset:       21 -  14
           04        tmnn1-status PIC X(020).
      *    Offset:       41 -  15
           04        tmnn1-name PIC X(020).
      *    Offset:       61 -  16
           04        tmnn1-degree PIC X(020).
      *    Offset:       81 -  17
           04        tmnn1-social-pro PIC X(020).
      *    Offset:      101 -  18
```

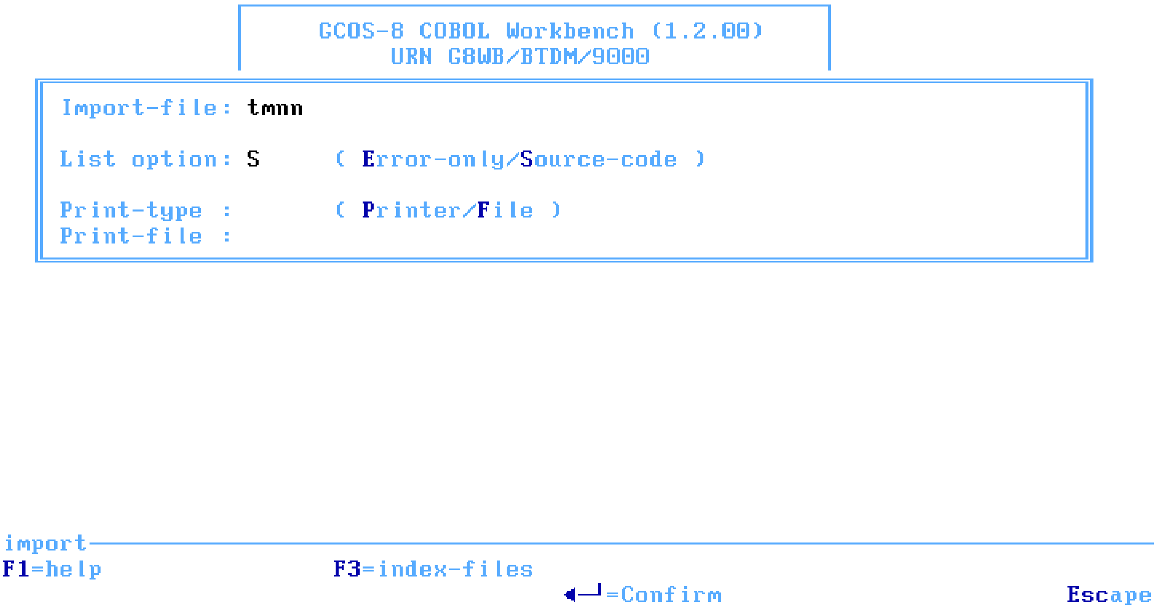
Generated COPY's are saved under $G8WBFDIR, which is automatically searched during preprocessing of your application program. Any specific Library-Tag used, i.e. COPY <name> of FL, need NOT be defined in the Copy Mapping configuration.

## Printing

This feature is currently not implemented.

## Importing

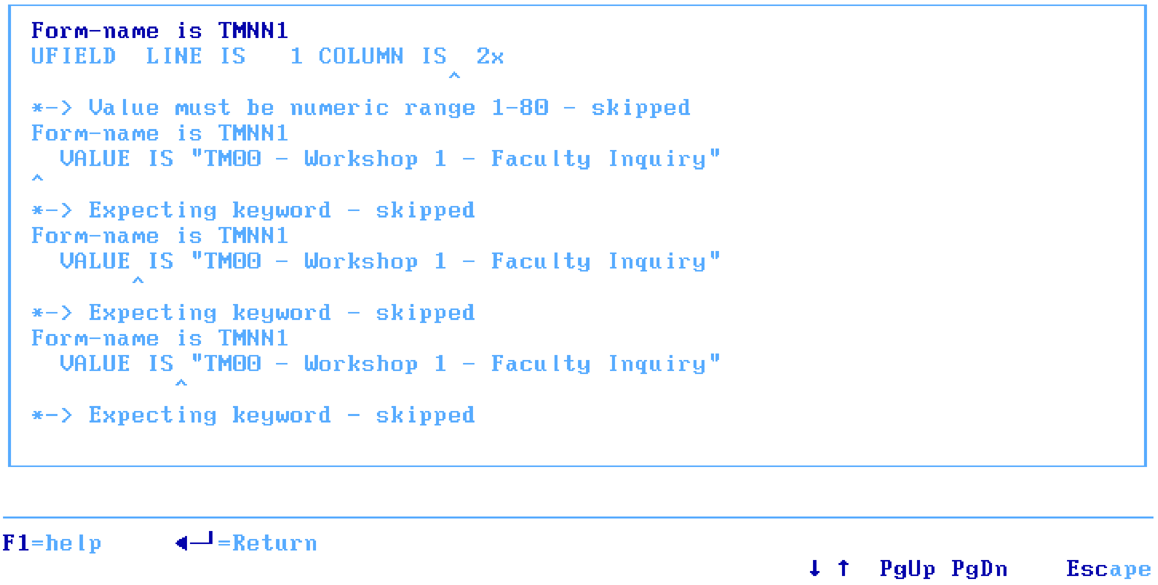
F8-Import may be used to import form(s), either from GCOS-8 or another G8WB environment. The import file is a text file describing one or more forms through a Forms software specific syntax.



```
GCOS-8 COBOL Workbench (1.2.00)
     URN G8WB/BTDM/9000

Import-file: tmnn

List option: S      ( Error-only/Source-code )

Print-type  :       ( Printer/File )
Print-file  :
```

The default extension for the import file is .S, i.e. TMNN.S.

List option will, by default, present any errors detected during import, in a browse panel. You may use S in List option to obtain a complete listing of the imported form(s), in a browse panel. Browse panel will use a temporary file, which is deleted when you exit import. You may save the output from import, through Print-type/File setting, i.e. direct output to a file or printer device. Following illustrate a browse panel with errors only:



```
Form-name is TMNN1
UFIELD  LINE IS   1 COLUMN IS  2x
                             ^
*-> Value must be numeric range 1-80 - skipped
Form-name is TMNN1
  VALUE IS "TM00 - Workshop 1 - Faculty Inquiry"
^
*-> Expecting keyword - skipped
Form-name is TMNN1
  VALUE IS "TM00 - Workshop 1 - Faculty Inquiry"
       ^
*-> Expecting keyword - skipped
Form-name is TMNN1
  VALUE IS "TM00 - Workshop 1 - Faculty Inquiry"
          ^
*-> Expecting keyword - skipped
```

## CrossePAC

With the CrossePAC option you are able to compress files before file transfer. This will reduce the file size by 70-90%. CrossePAC will create a text file archive, which you are able to transfer as a non-binary file.

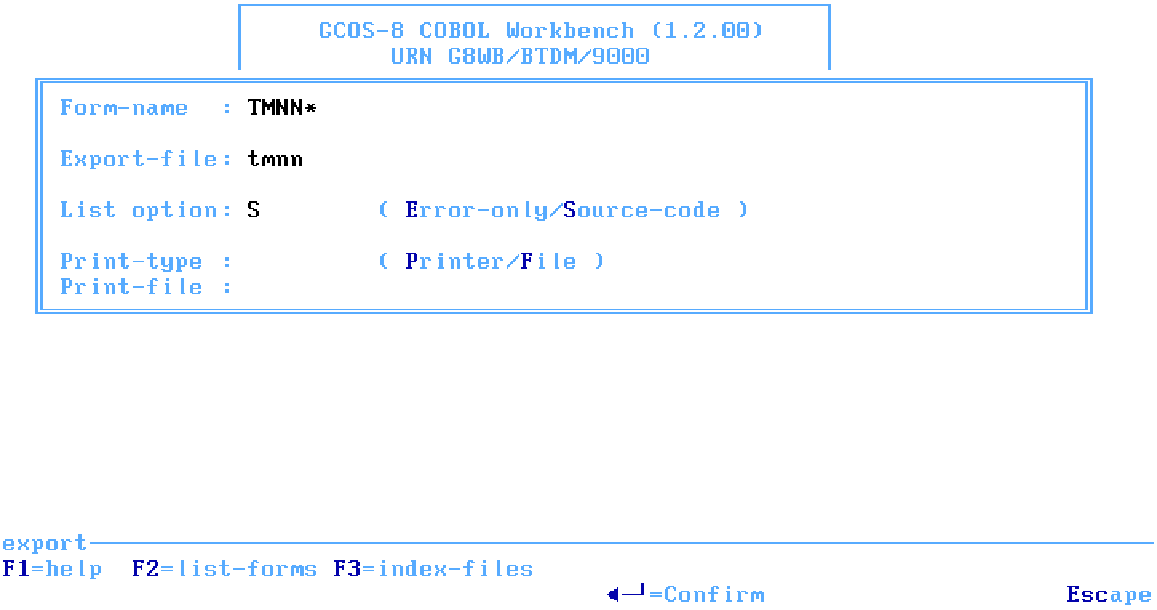
WUSP Import function will automatically recognize a CrossePAC archive file and decompress the file before importing it. It does, however, assume that the embedded file in the CrossePAC archive file is named CPACWORK.

If, as an example, you were to build a file named FORM.S on GCOS-8 for transfer to G8WB, you would instead build a file named CPACWORK and create a CrossePAC archive file named FORM.S. The CrossePAC command, TSS or Batch, would look like:

```
CPAC A  SP=N  /FORM.S  /CPACWORK
```

## Exporting

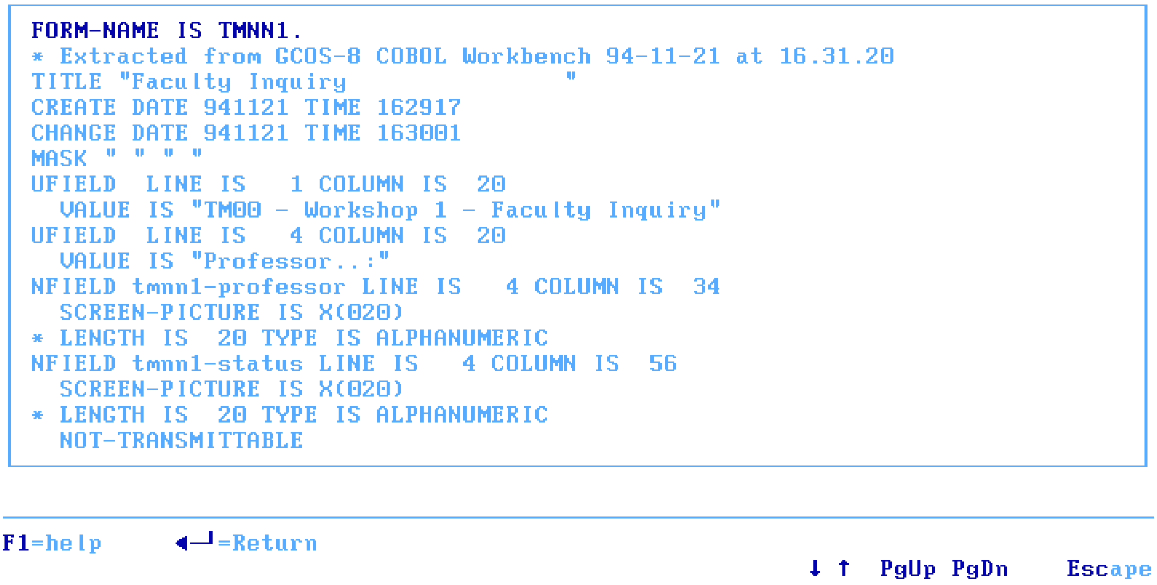
F9-Export may be used to export form(s), either to GCOS-8 or another G8WB environment. The export file is a text file describing one or more forms through a Forms software specific syntax.



```
             GCOS-8 COBOL Workbench (1.2.00)
                  URN G8WB/BTDM/9000

 Form-name  : TMNN*

 Export-file: tmnn

 List option: S        ( Error-only/Source-code )

 Print-type :          ( Printer/File )
 Print-file :


export------------------------------------------------------------------
F1=help  F2=list-forms F3=index-files
                                     ◄┘=Confirm                   Escape
```

You may specify a form name mask, i.e. * and ?, to export multiple forms. The export file is, by default, given the form name. When using a mask, no default apply to export file. The default extension for the export file is .S, i.e. TMNN.S.

List option will, by default, present any errors detected during export, in a browse panel. You may use S in List option to obtain a complete listing of the exported form(s), in a browse panel. Browse panel will use a temporary file, which is deleted when you exit export. You may save the output from export, through Print-type/File setting, i.e. direct output to a file or printer device. Following illustrate a browse panel:



```
 FORM-NAME IS TMNN1.
 * Extracted from GCOS-8 COBOL Workbench 94-11-21 at 16.31.20
 TITLE "Faculty Inquiry                    "
 CREATE DATE 941121 TIME 162917
 CHANGE DATE 941121 TIME 163001
 MASK " " " "
 VFIELD  LINE IS   1 COLUMN IS  20
   VALUE IS "TM00 - Workshop 1 - Faculty Inquiry"
 VFIELD  LINE IS   4 COLUMN IS  20
   VALUE IS "Professor..:"
 NFIELD tmnn1-professor LINE IS   4 COLUMN IS  34
   SCREEN-PICTURE IS X(020)
 * LENGTH IS  20 TYPE IS ALPHANUMERIC
 NFIELD tmnn1-status LINE IS   4 COLUMN IS  56
   SCREEN-PICTURE IS X(020)
 * LENGTH IS  20 TYPE IS ALPHANUMERIC
   NOT-TRANSMITTABLE

------------------------------------------------------------------------
F1=help     ◄┘=Return
                                            ↓ ↑  PgUp PgDn     Escape
```

## CrossePAC

If you have CrossePAC enabled, WUSP export function will automatically compress the export file and you need to decompress the file before you import the file on GCOS-8. WUSP will always name the embedded file CPACWORK, within the CrossePAC archive file.

If, as an example, you have used WUSP to build an export file named FORM.S, then you would need to extract CPACWORK from FORM.S and use CPACWORK as input to the GCOS-8 import program. The CrossePAC command, TSS or Batch, would look like:

```
CPAC X  /FORM.S
```

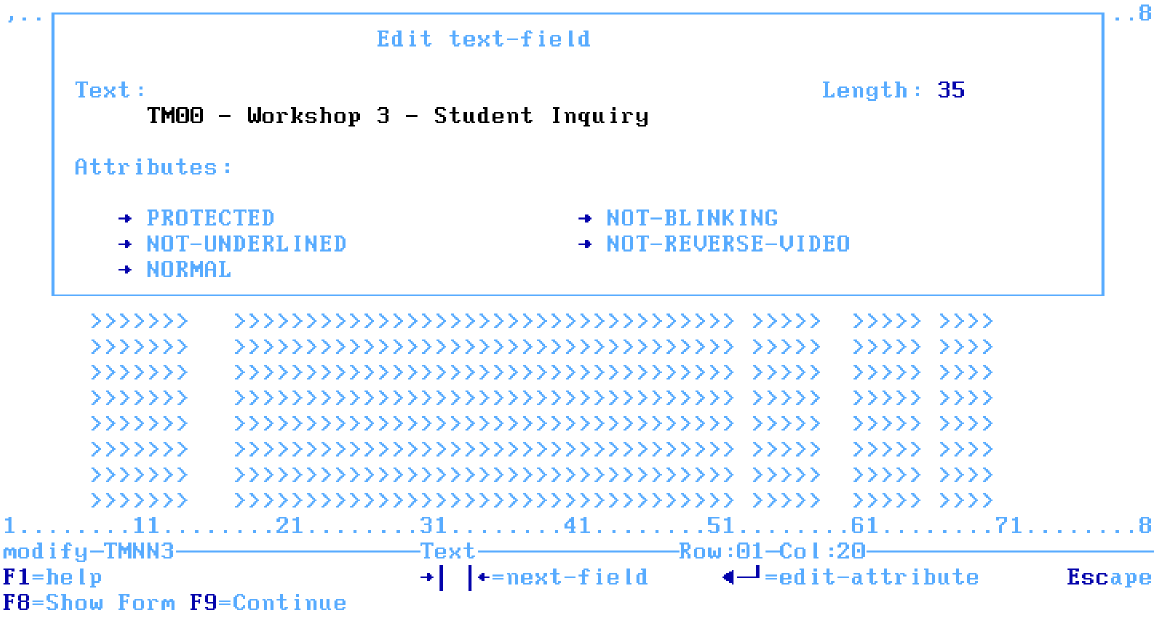
## Testing

This feature is currently not implemented.

# TPFF

Following illustrate the TPFF specific features, panels and restrictions in relation to WUSP. Please refer to BULL TPFF documentation for details on attribute settings etc.

## Text field



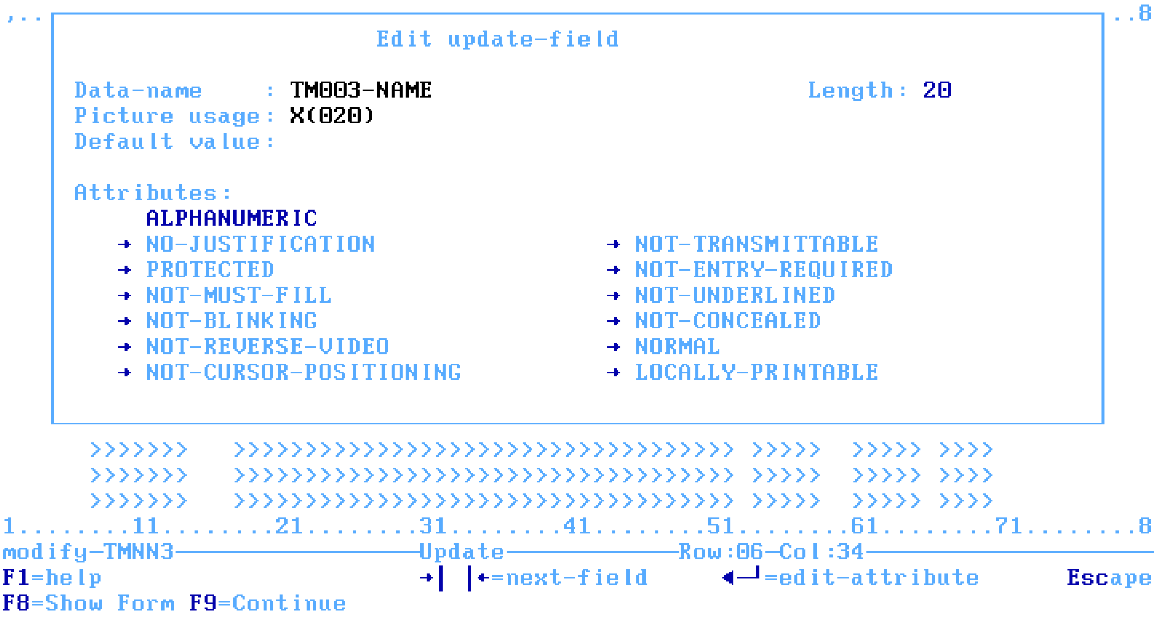
```
,...                                                                          ..8
                          Edit text-field

  Text:                                               Length: 35
       TM00 - Workshop 3 - Student Inquiry

  Attributes:

     → PROTECTED                       → NOT-BLINKING
     → NOT-UNDERLINED                  → NOT-REVERSE-VIDEO
     → NORMAL
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1.........11........21........31........41........51........61........71.......8
modify-TMNN3─────────────────Text──────────────Row:01─Col:20────────────────────
F1=help                      →| |←=next-field    ◄┘=edit-attribute        Escape
F8=Show Form F9=Continue
```

## Input field

Input fields are treated like Update fields. If you paint or insert Input fields, these are automatically changed to Update fields.

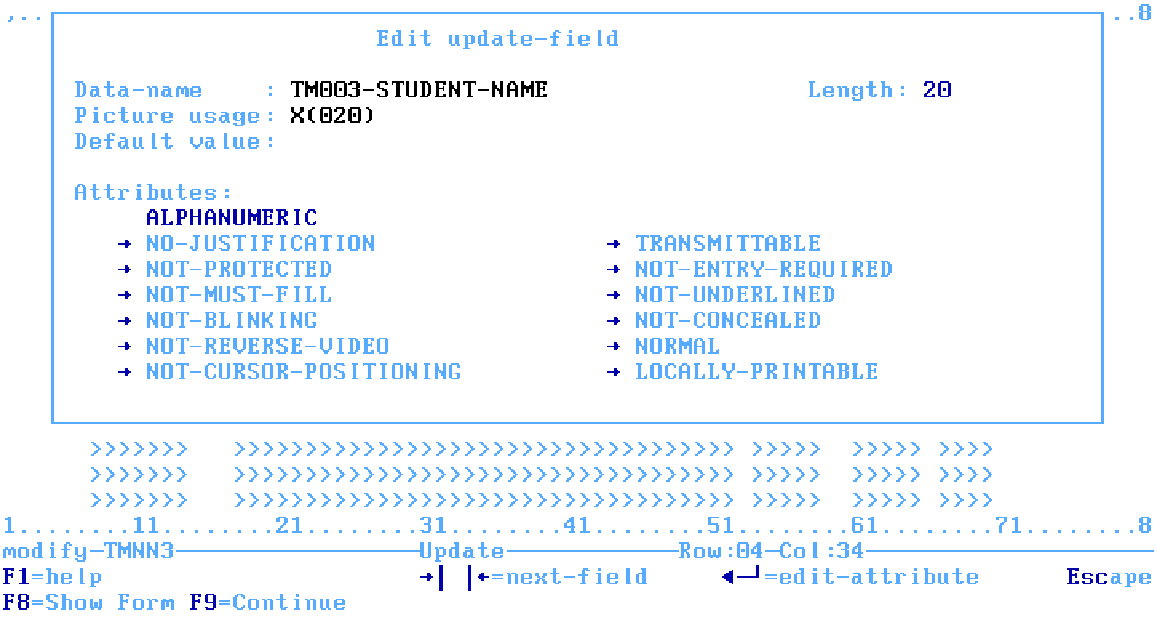
## Output field

Output fields are treated like Update fields, however if you paint or insert Output fields, these will have PROTECT and NOT-TRANSMITTABLE attributes set by default.



```
,..                                                                          ..8
                           Edit update-field

   Data-name     : TM003-NAME                          Length: 20
   Picture usage: X(020)
   Default value:

   Attributes:
        ALPHANUMERIC
      → NO-JUSTIFICATION                  → NOT-TRANSMITTABLE
      → PROTECTED                         → NOT-ENTRY-REQUIRED
      → NOT-MUST-FILL                     → NOT-UNDERLINED
      → NOT-BLINKING                      → NOT-CONCEALED
      → NOT-REVERSE-VIDEO                 → NORMAL
      → NOT-CURSOR-POSITIONING            → LOCALLY-PRINTABLE

      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN3─────────────────Update────────────Row:06─Col:34──────────────────────
F1=help                                  →| |←=next-field    ◄┘=edit-attribute     Escape
F8=Show Form F9=Continue
```

## Update field



```
,..                                                                          ..8
                           Edit update-field

   Data-name     : TM003-STUDENT-NAME                  Length: 20
   Picture usage: X(020)
   Default value:

   Attributes:
        ALPHANUMERIC
      → NO-JUSTIFICATION                  → TRANSMITTABLE
      → NOT-PROTECTED                     → NOT-ENTRY-REQUIRED
      → NOT-MUST-FILL                     → NOT-UNDERLINED
      → NOT-BLINKING                      → NOT-CONCEALED
      → NOT-REVERSE-VIDEO                 → NORMAL
      → NOT-CURSOR-POSITIONING            → LOCALLY-PRINTABLE

      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN3─────────────────Update────────────Row:04─Col:34──────────────────────
F1=help                                  →| |←=next-field    ◄┘=edit-attribute     Escape
F8=Show Form F9=Continue
```

## Repeat field

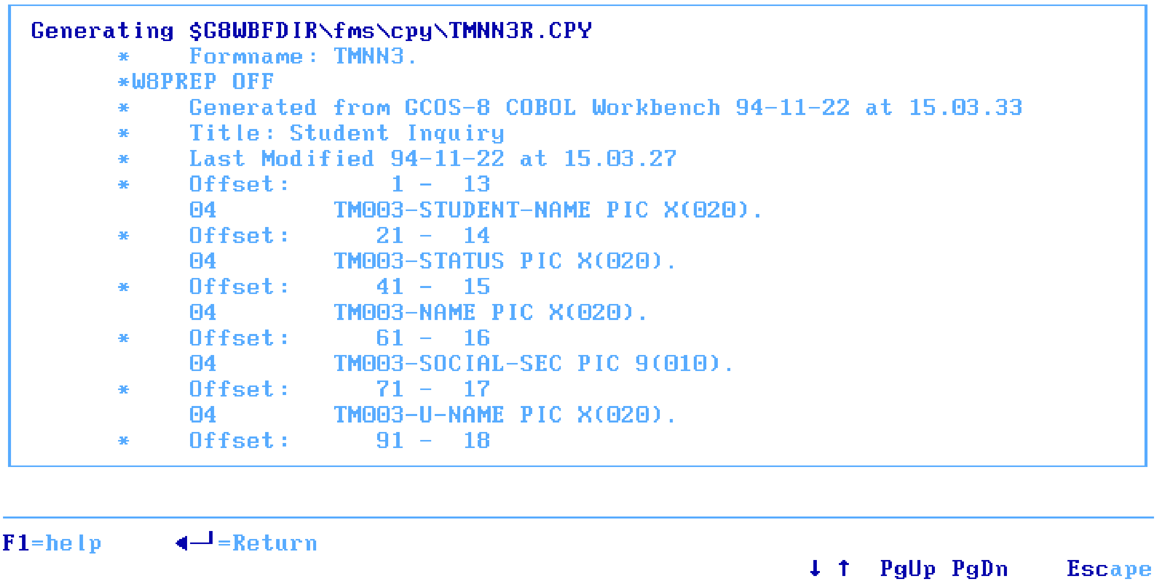
TPFF support both Array and Row repeat. An array is limited to one line (occurrence). For two dimentional repeats, only Row within Array is allowed.

## Generation

Form generation will generate the three standard TPFF COPY’s:

- Form Image, named FormnameI
- Form record, named FormnameR
- Form Vector, named FormnameV

Optionally, Form generation may generate a single COPY (named Formname) containing Image, Record and Vector, or both the combined and separate COPY's. Ref. Configuration chapter for details.



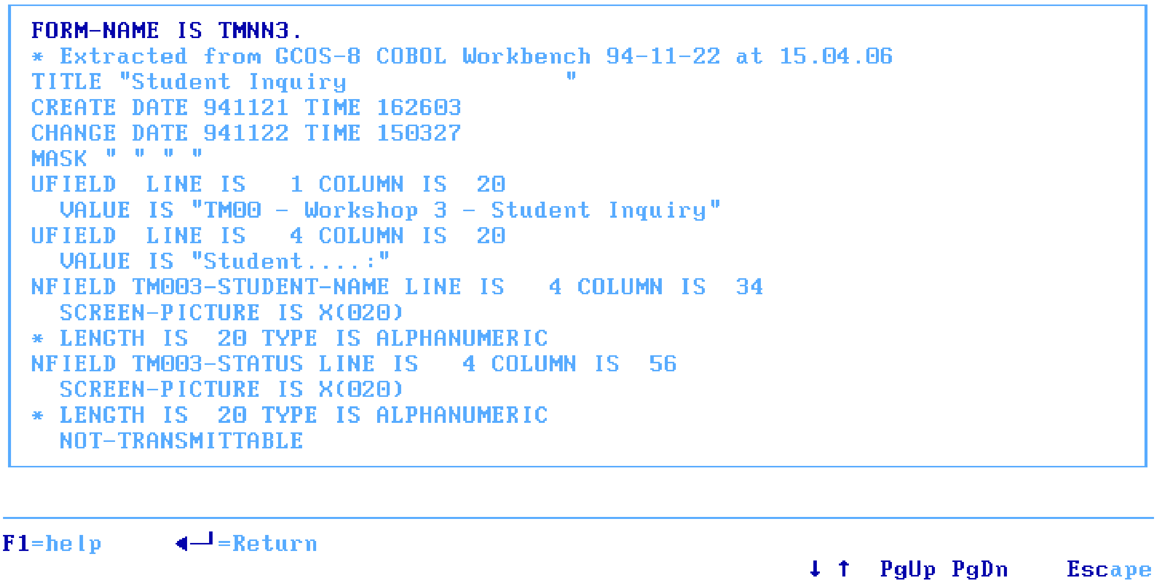
```
Generating $G8WBFDIR\fms\cpy\TMNN3R.CPY
      *    Formname: TMNN3.
      *W8PREP OFF
      *    Generated from GCOS-8 COBOL Workbench 94-11-22 at 15.03.33
      *    Title: Student Inquiry
      *    Last Modified 94-11-22 at 15.03.27
      *    Offset:        1 -  13
           04        TM003-STUDENT-NAME PIC X(020).
      *    Offset:       21 -  14
           04        TM003-STATUS PIC X(020).
      *    Offset:       41 -  15
           04        TM003-NAME PIC X(020).
      *    Offset:       61 -  16
           04        TM003-SOCIAL-SEC PIC 9(010).
      *    Offset:       71 -  17
           04        TM003-U-NAME PIC X(020).
      *    Offset:       91 -  18
```

```
F1=help     ◄┘=Return
                                              ↓ ↑  PgUp PgDn    Escape
```

*Note: The COPY's are not compatible with GCOS-8.*

## Import/Export

GCOS-8 TPFF provides very limited import/export facilities and you need to use G8WB Host import/export utilities instead, ref. User Guide on how to install these utilities on GCOS-8. The syntax used for forms exchange, is however similar to the import syntax described in TPFF documentation.

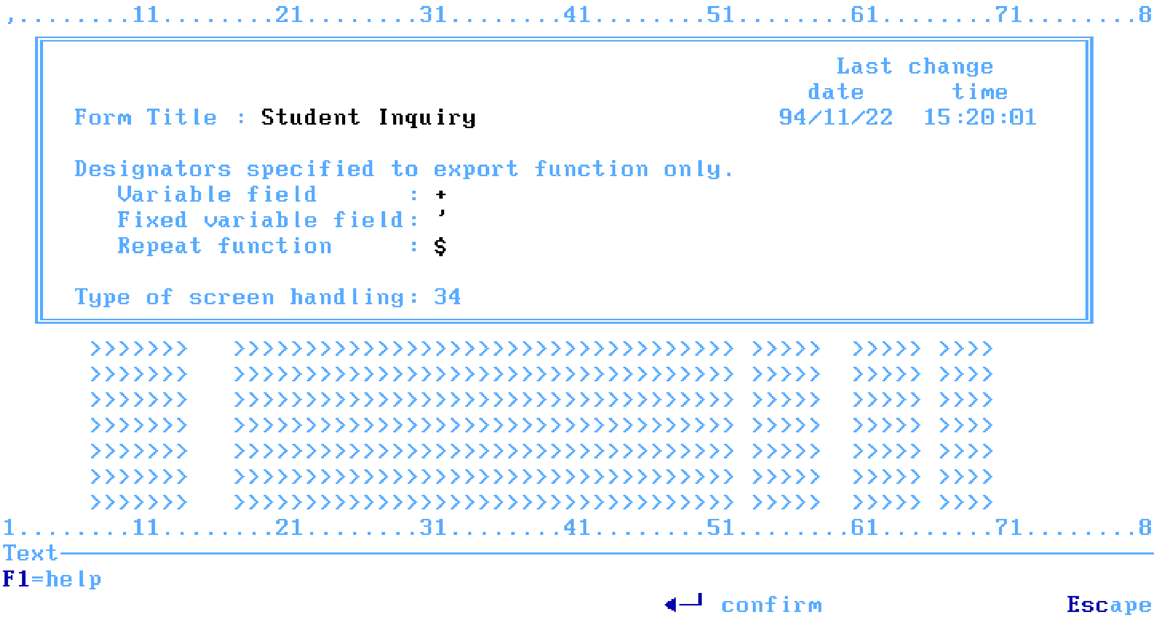


```
FORM-NAME IS TMNN3.
* Extracted from GCOS-8 COBOL Workbench 94-11-22 at 15.04.06
TITLE "Student Inquiry                "
CREATE DATE 941121 TIME 162603
CHANGE DATE 941122 TIME 150327
MASK " " " "
UFIELD  LINE IS   1 COLUMN IS  20
  VALUE IS "TM00 - Workshop 3 - Student Inquiry"
UFIELD  LINE IS   4 COLUMN IS  20
  VALUE IS "Student....:"
NFIELD TM003-STUDENT-NAME LINE IS   4 COLUMN IS  34
  SCREEN-PICTURE IS X(020)
* LENGTH IS  20 TYPE IS ALPHANUMERIC
NFIELD TM003-STATUS LINE IS   4 COLUMN IS  56
  SCREEN-PICTURE IS X(020)
* LENGTH IS  20 TYPE IS ALPHANUMERIC
  NOT-TRANSMITTABLE
```

# FORMAT

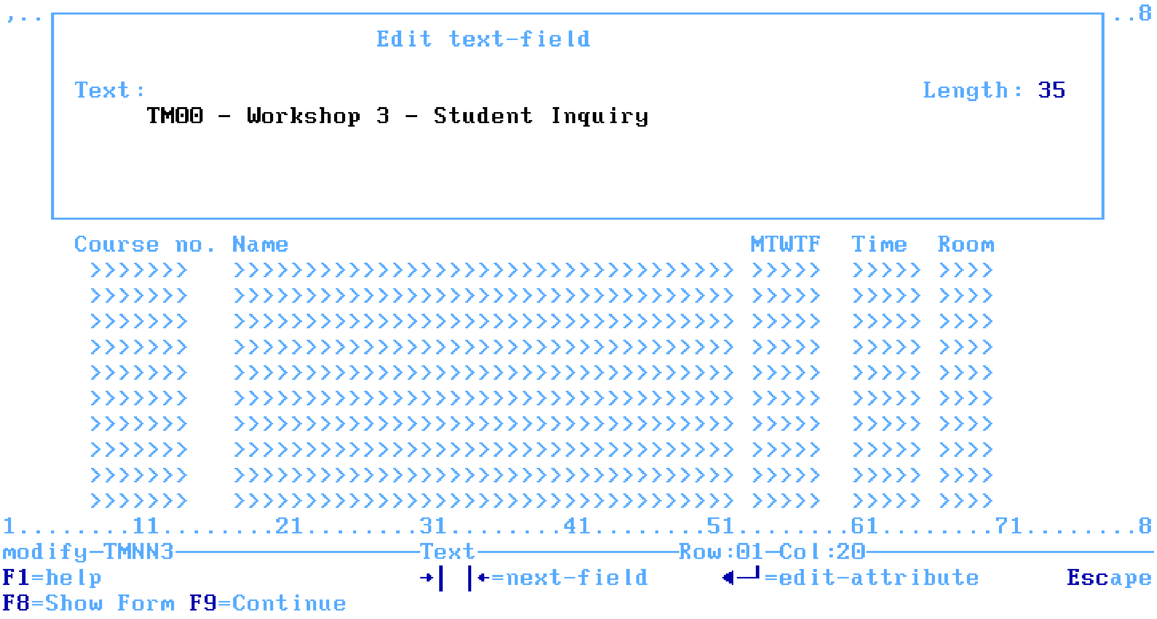
Following illustrate the FORMAT specific features, panels and restrictions in relation to WUSP. Please refer to BULL FORMAT documentation for details on attribute settings etc.

## Form title



```
,........11........21........31........41........51........61........71.......8
                                                         Last change
                                                       date      time
    Form Title : Student Inquiry                      94/11/22  15:20:01

    Designators specified to export function only.
       Variable field      : +
       Fixed variable field: '
       Repeat function     : $

    Type of screen handling: 34
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71.......8
Text──────────────────────────────────────────────────────────────────────────
F1=help
                                              ◄┘ confirm                Escape
```

## Text field

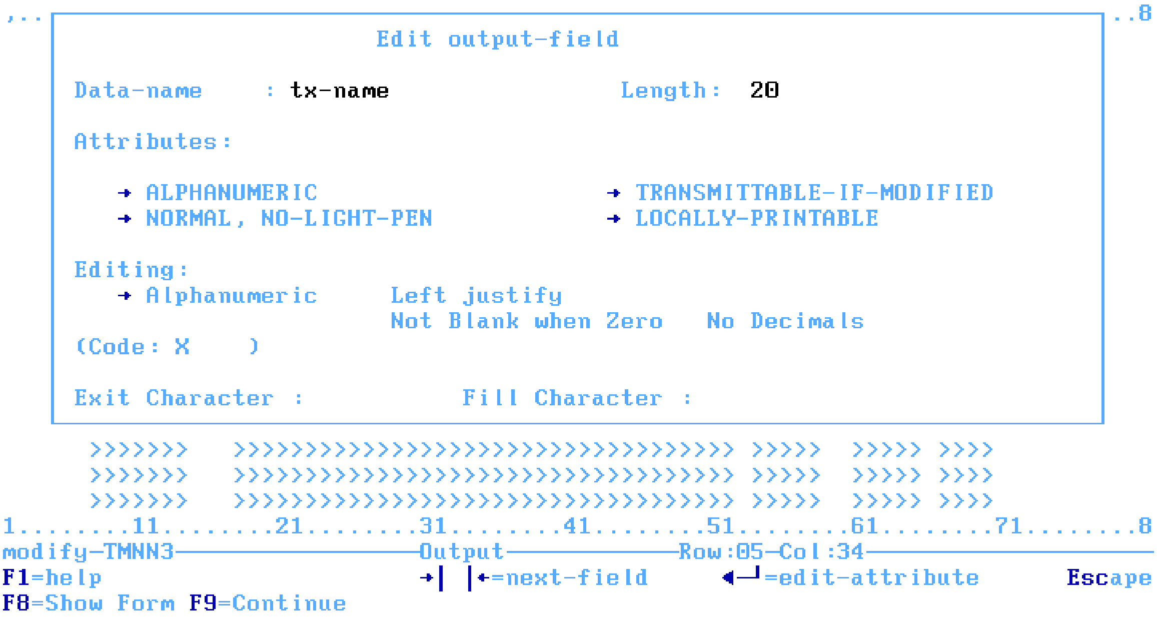


```
,..                                                                          ..8
                          Edit text-field

    Text:                                                         Length: 35
         TM00 - Workshop 3 - Student Inquiry



    Course no. Name                                MTWTF  Time  Room
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
      >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1........11........21........31........41........51........61........71.......8
modify-TMNN3────────────────Text─────────────Row:01-Col:20────────────────────
F1=help                         →| |←=next-field    ◄┘=edit-attribute     Escape
F8=Show Form F9=Continue
```

## Input field

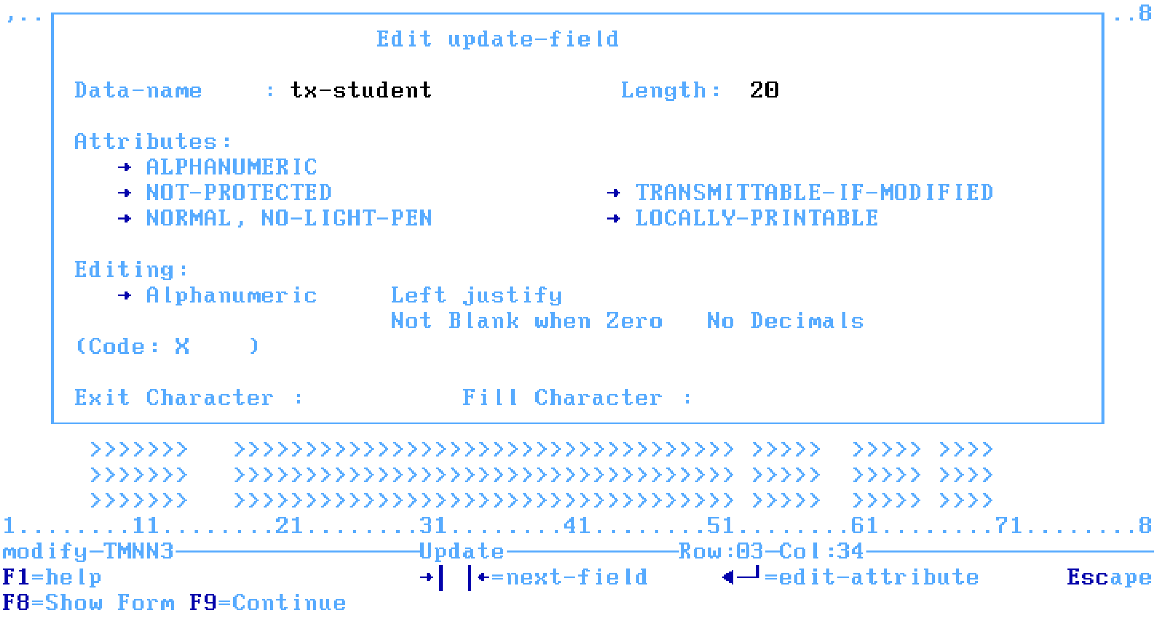
Input fields are treated like Update fields. If you paint or insert Input fields, these are automatically changed to Update fields.

## Output field



```
,..                                                                     ..8
                      Edit output-field

   Data-name    : tx-name                Length:  20

   Attributes:

      → ALPHANUMERIC                     → TRANSMITTABLE-IF-MODIFIED
      → NORMAL, NO-LIGHT-PEN             → LOCALLY-PRINTABLE

   Editing:
      → Alphanumeric      Left justify
                          Not Blank when Zero   No Decimals
   (Code: X    )

   Exit Character  :            Fill Character  :

     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1.........11.........21.........31.........41.........51.........61.........71.........8
modify-TMNN3─────────────────Output────────────Row:05─Col:34──────────────────
F1=help                        →|  |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Update field



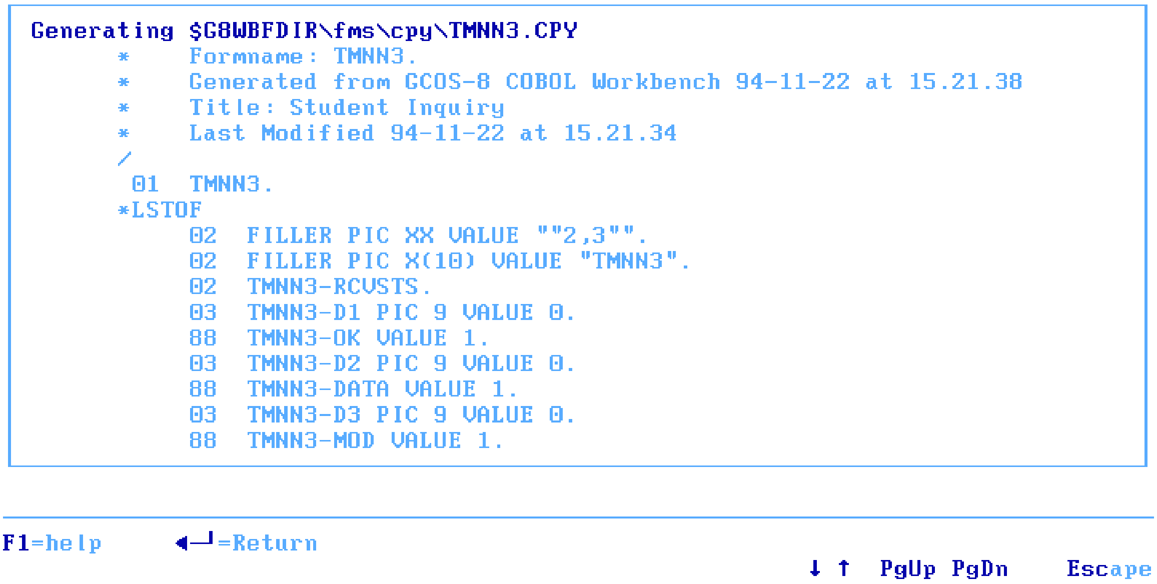
```
,..                                                                     ..8
                      Edit update-field

   Data-name    : tx-student             Length:  20

   Attributes:
      → ALPHANUMERIC
      → NOT-PROTECTED                    → TRANSMITTABLE-IF-MODIFIED
      → NORMAL, NO-LIGHT-PEN             → LOCALLY-PRINTABLE

   Editing:
      → Alphanumeric      Left justify
                          Not Blank when Zero   No Decimals
   (Code: X    )

   Exit Character  :            Fill Character  :

     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
     >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>
1.........11.........21.........31.........41.........51.........61.........71.........8
modify-TMNN3─────────────────Update────────────Row:03─Col:34──────────────────
F1=help                        →|  |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Repeat field

FORMAT support both Array and Row repeat. An array must start in column 1. Array within Array is not allowed.

## Generation

Form generation will generate the standard FORMAT COPY and give it the same name as the form.



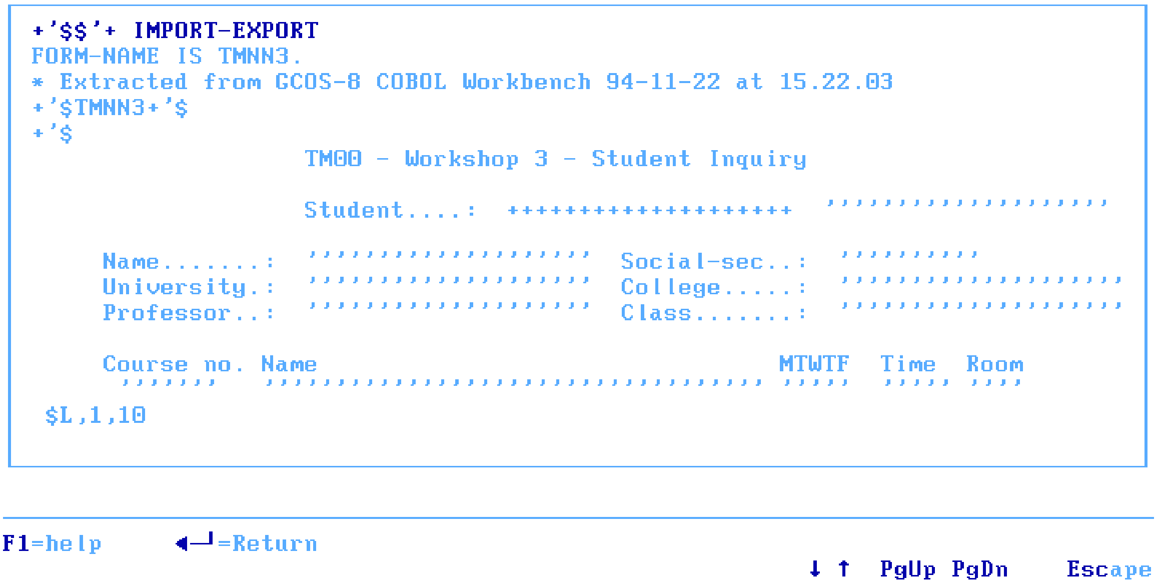
```
Generating $G8WBFDIR\fms\cpy\TMNN3.CPY
      *    Formname: TMNN3.
      *    Generated from GCOS-8 COBOL Workbench 94-11-22 at 15.21.38
      *    Title: Student Inquiry
      *    Last Modified 94-11-22 at 15.21.34
      /
       01  TMNN3.
      *LSTOF
           02  FILLER PIC XX VALUE ""2,3"".
           02  FILLER PIC X(10) VALUE "TMNN3".
           02  TMNN3-RCVSTS.
           03  TMNN3-D1 PIC 9 VALUE 0.
           88  TMNN3-OK VALUE 1.
           03  TMNN3-D2 PIC 9 VALUE 0.
           88  TMNN3-DATA VALUE 1.
           03  TMNN3-D3 PIC 9 VALUE 0.
           88  TMNN3-MOD VALUE 1.

F1=help     ◄┘=Return
                                            ↓ ↑  PgUp PgDn    Escape
```

*Note: The COPY is compatible with GCOS-8.*

If you use $$SELECT to include FORMAT forms, you need to map the GCOS-8 Catalog-string(s) used in $$SELECT to $G8WBFDIR\FMS\CPY, ref. Configuration chapter for details.

## Import/Export

GCOS-8 FORMAT provides very limited import/export facilities and you need to update FORMAT with G8WB Host import/export feature add-on, ref. User Guide on how to install this add-on on GCOS-8. The syntax used for forms exchange, is the same as the FORMAT library format, as described in FORMAT documentation.

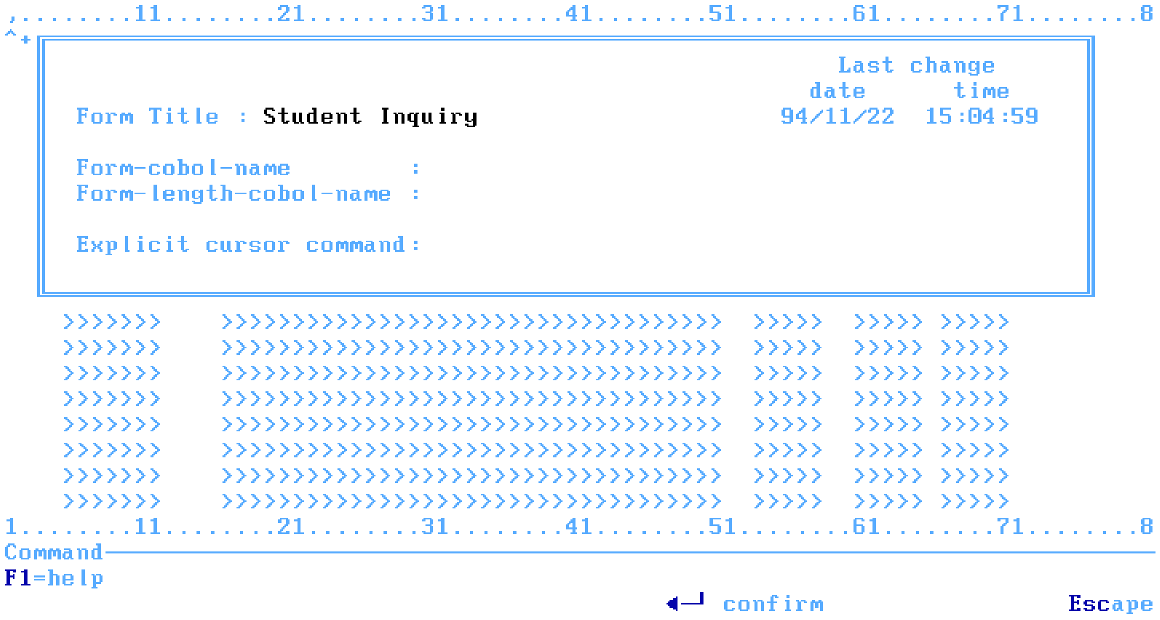


```
+'$$'+ IMPORT-EXPORT
FORM-NAME IS TMNN3.
* Extracted from GCOS-8 COBOL Workbench 94-11-22 at 15.22.03
+'$TMNN3+'$
+'$
                   TM00 - Workshop 3 - Student Inquiry

                   Student....:  ++++++++++++++++++++  ,,,,,,,,,,,,,,,,,,,,

     Name.......:  ,,,,,,,,,,,,,,,,,,,,  Social-sec..:  ,,,,,,,,,,
     University.:  ,,,,,,,,,,,,,,,,,,,,  College.....:  ,,,,,,,,,,,,,,,,,,,,
     Professor..:  ,,,,,,,,,,,,,,,,,,,,  Class.......:  ,,,,,,,,,,,,,,,,,,,,

     Course no. Name                                MTWTF  Time  Room
      ,,,,,,,   ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, ,,,,,  ,,,,, ,,,,
 $L,1,10

F1=help     ◄┘=Return
                                            ↓ ↑  PgUp PgDn    Escape
```

# Pilspråk

Following illustrate the Pilspråk specific features, panels and restrictions in relation to WUSP. Please refer to RFV Pilspråk documentation for details on attribute settings etc.

## Form title



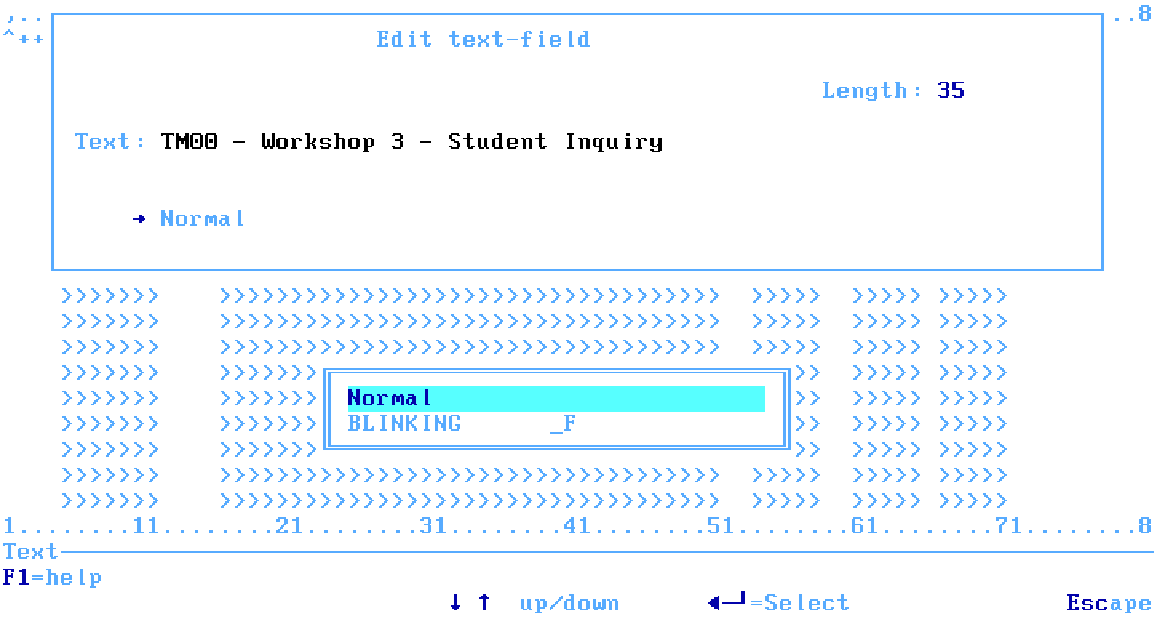
```
,........11........21........31........41........51........61........71........8
^+
                                                              Last change
                                                            date       time
    Form Title  :  Student Inquiry                        94/11/22   15:04:59

    Form-cobol-name          :
    Form-length-cobol-name   :

    Explicit cursor command:

   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   >>>>>>>     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71........8
Command────────────────────────────────────────────────────────────────────────
F1=help
                                                     ◄┘ confirm             Escape
```

Form-cobol-name may be used to include 01 <name> for the form in the generated form syntax. Default is 01 <formname>.

Form-length-cobol-name may be used to include 01 <name> for the form length in the generated form syntax. Default is 01 <formname>-LENGTH.

Explicit-cursor-command may be used to include a Pilspråk command at the very end of the generated form. This may be used as an alternative to CURSOR attribute on Update field.

## Text field

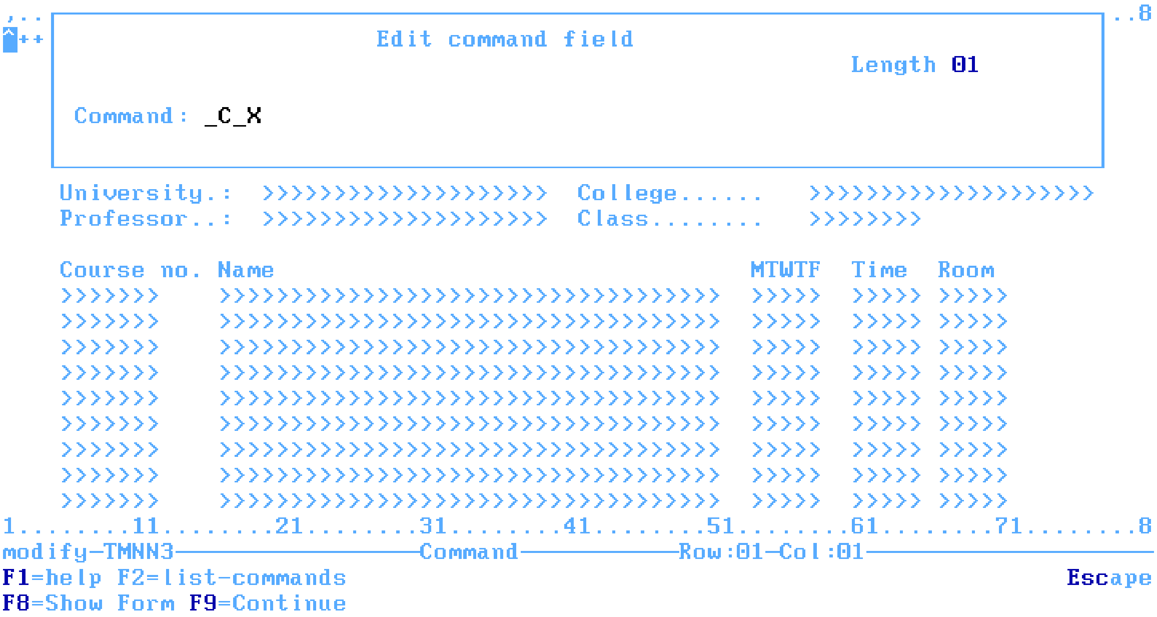


```
,..                                                                    ..8
^++                   Edit text-field

                                                        Length: 35

   Text: TM00 - Workshop 3 - Student Inquiry


       → Normal

    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>┌───────────────────────────────┐>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>│ Normal                        │>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>│ BLINKING      _F              │>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>└───────────────────────────────┘>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71........8
Text────────────────────────────────────────────────────────────────────────────
F1=help
                                ↓ ↑  up/down      ↵=Select                Escape
```

## Command field

Command fields may be used to insert explicit Pilspråk commands in the generated form. The position of the command field define where the commands are inserted.

For new forms, WUSP will insert a standard command field in Line1 Column 1.

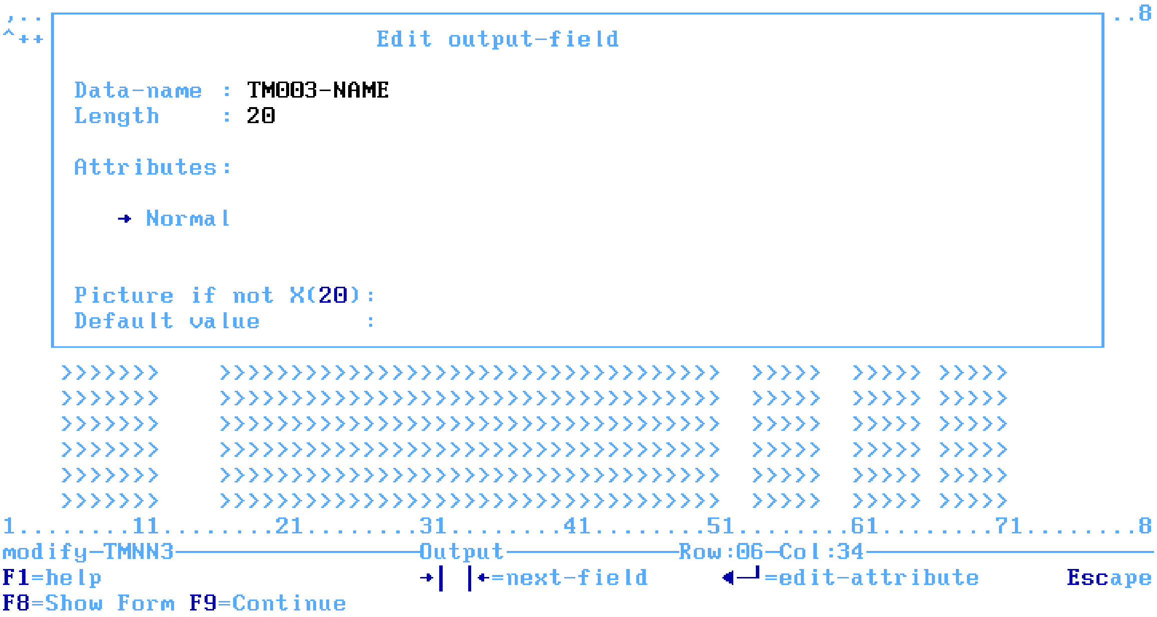


```
,..                                                                    ..8
^++                   Edit command field
                                                          Length 01

   Command: _C_X


    University.:  >>>>>>>>>>>>>>>>>>>>  College......   >>>>>>>>>>>>>>>>>>>>
    Professor..:  >>>>>>>>>>>>>>>>>>>>  Class........   >>>>>>>>

    Course no. Name                                 MTWTF  Time   Room
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71........8
modify-TMNN3──────────────────Command──────────Row:01-Col:01────────────────────
F1=help F2=list-commands                                                  Escape
F8=Show Form F9=Continue
```

## Input field

Input fields are treated like Update fields. If you paint or insert Input fields, these are automatically changed to Update fields.

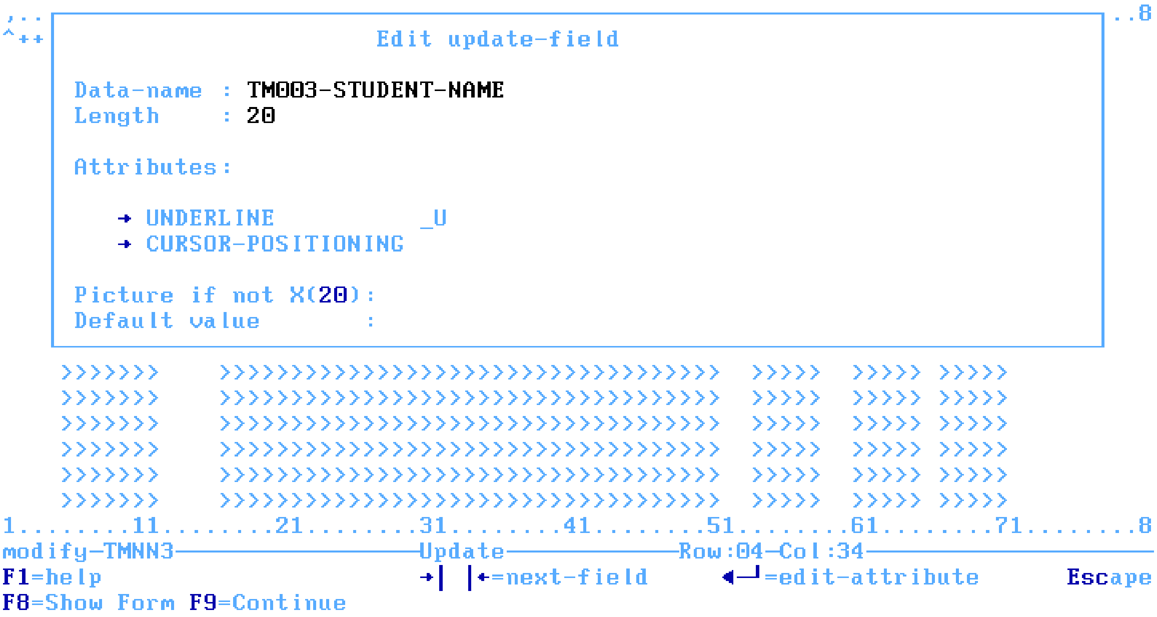
## Output field



```
;..                                                                          ..8
^++                      Edit output-field

     Data-name  : TM003-NAME
     Length     : 20

     Attributes:

        → Normal


     Picture if not X(20):
     Default value       :

    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71.......8
modify-TMNN3────────────────Output────────────Row:06-Col:34──────────────────────
F1=help                      →| |←=next-field     ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

*Note: The field position is maintained regardless of attributes assigned, i.e. cursor position is automatically adjusted in the generated form.*

## Update field

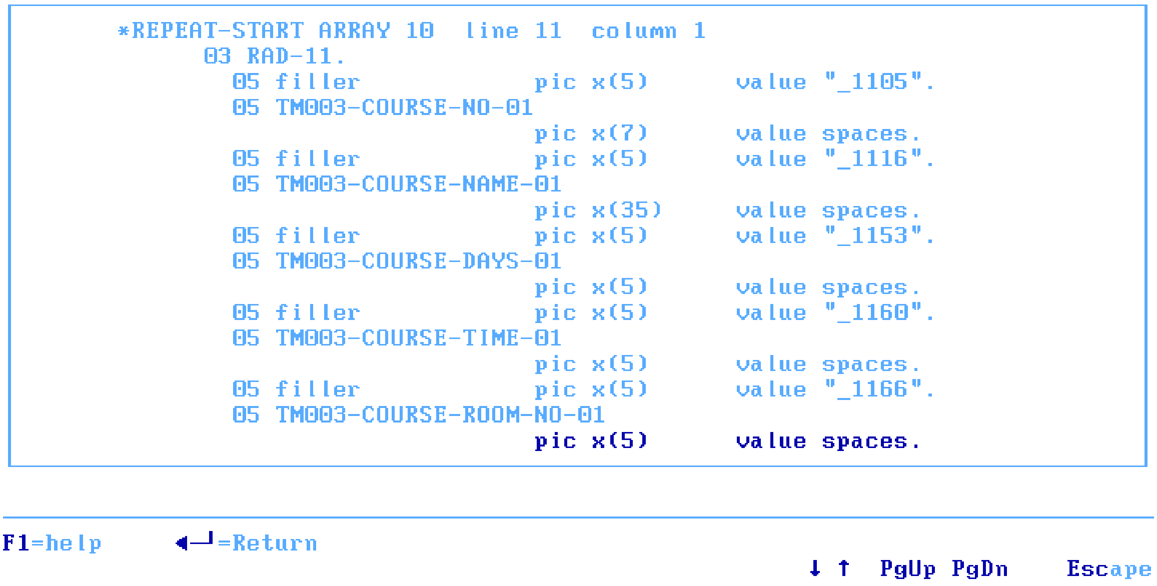


```
;..                                                                          ..8
^++                      Edit update-field

     Data-name  : TM003-STUDENT-NAME
     Length     : 20

     Attributes:

        → UNDERLINE         _U
        → CURSOR-POSITIONING

     Picture if not X(20):
     Default value       :

    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71.......8
modify-TMNN3────────────────Update────────────Row:04-Col:34──────────────────────
F1=help                      →| |←=next-field     ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

*Note: The field position is maintained regardless of attributes assigned, i.e. cursor position is automatically adjusted in the generated form.*

## Repeat field

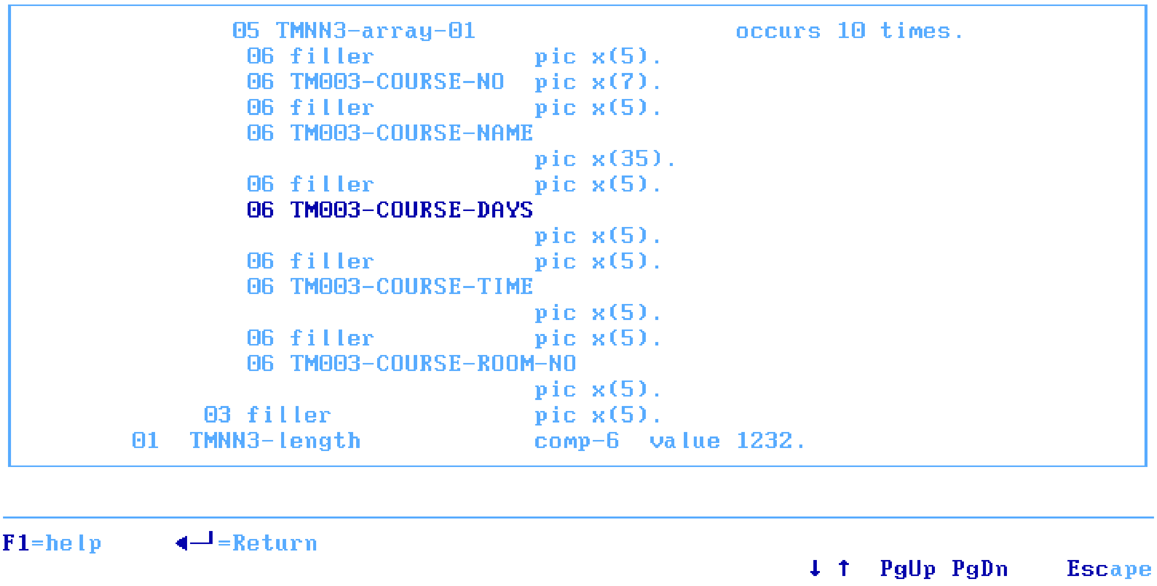
Pilspråk does directly support repeat structures, however, you may use repeat structures when you create a new or modify and existing form. Repeat structures are expanded during form generation. The fields within a repeat structure will include the occurrence number as suffix, i.e. <Field-name>-<suffix>.



```
       *REPEAT-START ARRAY 10  line 11  column 1
             03 RAD-11.
               05 filler            pic x(5)      value "_1105".
               05 TM003-COURSE-NO-01
                                    pic x(7)      value spaces.
               05 filler            pic x(5)      value "_1116".
               05 TM003-COURSE-NAME-01
                                    pic x(35)     value spaces.
               05 filler            pic x(5)      value "_1153".
               05 TM003-COURSE-DAYS-01
                                    pic x(5)      value spaces.
               05 filler            pic x(5)      value "_1160".
               05 TM003-COURSE-TIME-01
                                    pic x(5)      value spaces.
               05 filler            pic x(5)      value "_1166".
               05 TM003-COURSE-ROOM-NO-01
                                    pic x(5)      value spaces.
```

```
F1=help     ◄┘=Return
                                                   ↓ ↑  PgUp PgDn    Escape
```

If you have Extended COPY enabled, ref. Configuration chapter for details, repeat fields may also be referenced with a subscript, i.e. <Field-name> (subscript).



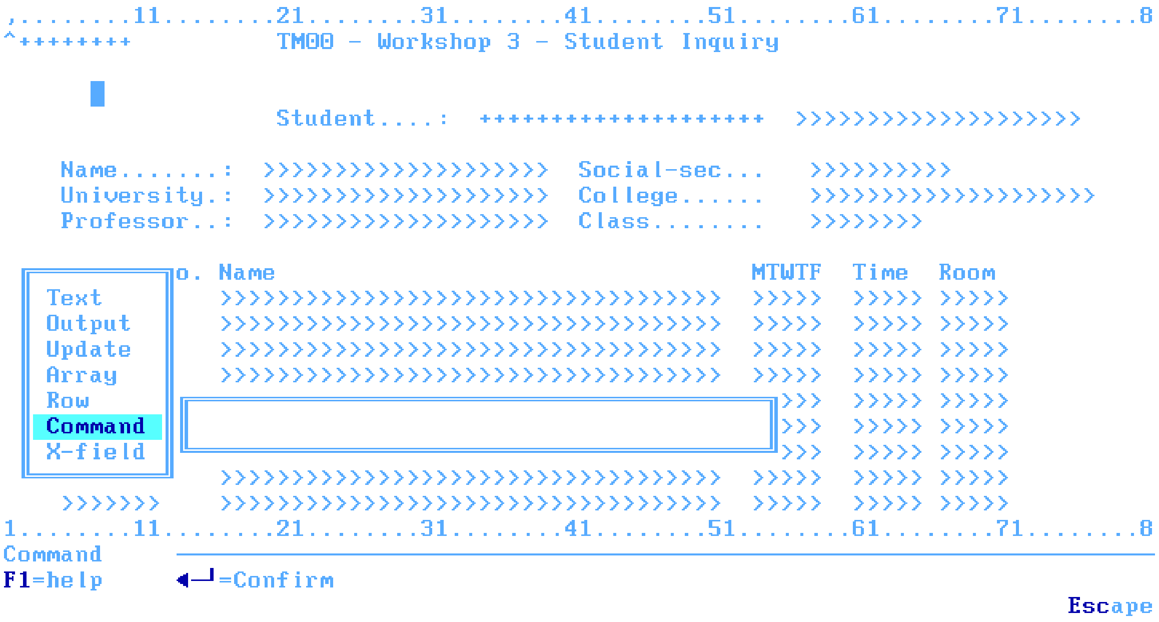
```
               05 TMNN3-array-01                      occurs 10 times.
                06 filler           pic x(5).
                06 TM003-COURSE-NO  pic x(7).
                06 filler           pic x(5).
                06 TM003-COURSE-NAME
                                    pic x(35).
                06 filler           pic x(5).
                06 TM003-COURSE-DAYS
                                    pic x(5).
                06 filler           pic x(5).
                06 TM003-COURSE-TIME
                                    pic x(5).
                06 filler           pic x(5).
                06 TM003-COURSE-ROOM-NO
                                    pic x(5).
             03 filler              pic x(5).
        01  TMNN3-length            comp-6  value 1232.
```

```
F1=help     ◄┘=Return
                                                   ↓ ↑  PgUp PgDn    Escape
```

## Insert field

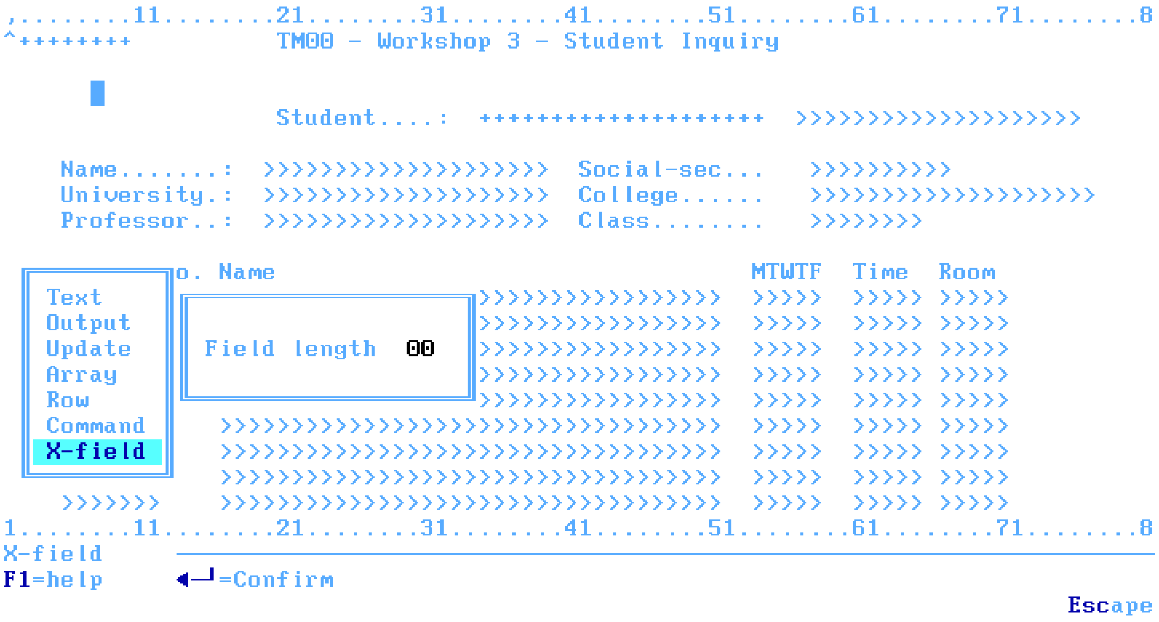
Insert field will allow you to insert Command fields and X-Field, in addition to the standard WUSP field types.

Command fields may be used to insert explicit Pilspråk commands in the generated form. The position of the command field define where the commands are inserted.



```
,........11........21........31........41........51........61........71........8
^++++++++          TM00 - Workshop 3 - Student Inquiry


                   Student....:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

    Name.......:  >>>>>>>>>>>>>>>>>>>>  Social-sec...   >>>>>>>>>>
    University.:  >>>>>>>>>>>>>>>>>>>>  College......   >>>>>>>>>>>>>>>>>>>>
    Professor..:  >>>>>>>>>>>>>>>>>>>>  Class........   >>>>>>>>

           o. Name                                 MTWTF  Time  Room
   Text       >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Output     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Update     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Array      >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Row                                               >>>  >>>>> >>>>>
   Command                                           >>>  >>>>> >>>>>
   X-field                                           >>>  >>>>> >>>>>
              >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71........8
Command
F1=help     ↵=Confirm
                                                                          Escape
```

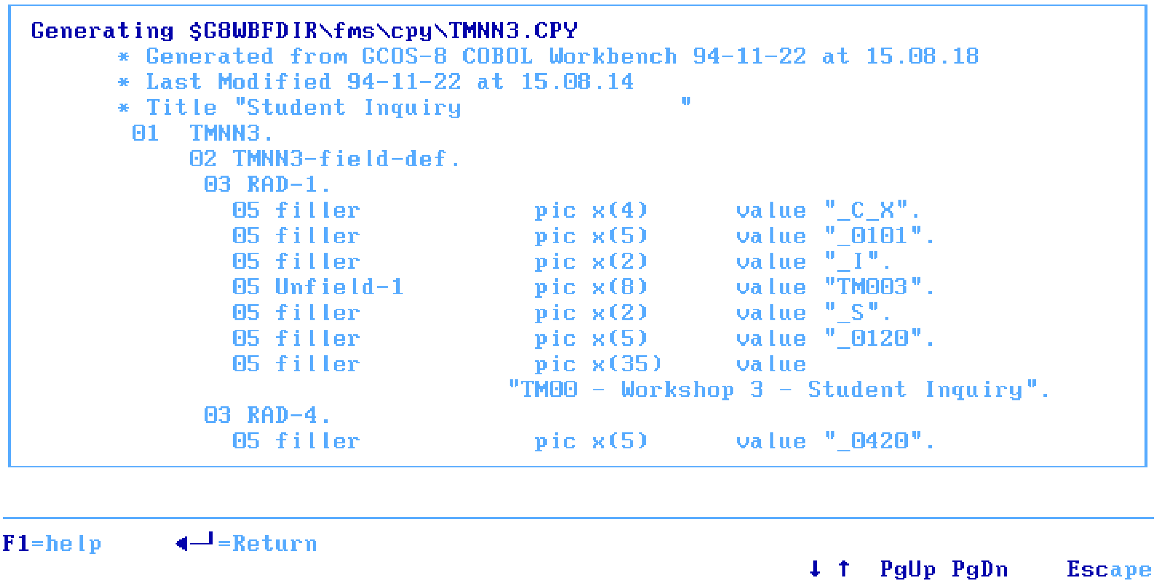
X-fields may be used to insert an X-field definition in the generated form. The position of the X-field define where the sequence is inserted.



```
,........11........21........31........41........51........61........71........8
^++++++++          TM00 - Workshop 3 - Student Inquiry


                   Student....:  ++++++++++++++++++++  >>>>>>>>>>>>>>>>>>>>

    Name.......:  >>>>>>>>>>>>>>>>>>>>  Social-sec...   >>>>>>>>>>
    University.:  >>>>>>>>>>>>>>>>>>>>  College......   >>>>>>>>>>>>>>>>>>>>
    Professor..:  >>>>>>>>>>>>>>>>>>>>  Class........   >>>>>>>>

           o. Name                                 MTWTF  Time  Room
   Text                          >>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Output                        >>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Update      Field length   00 >>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Array                         >>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Row                           >>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   Command    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
   X-field    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
              >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
    >>>>>>>   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  >>>>>  >>>>> >>>>>
1........11........21........31........41........51........61........71........8
X-field
F1=help     ↵=Confirm
                                                                          Escape
```

## Generation

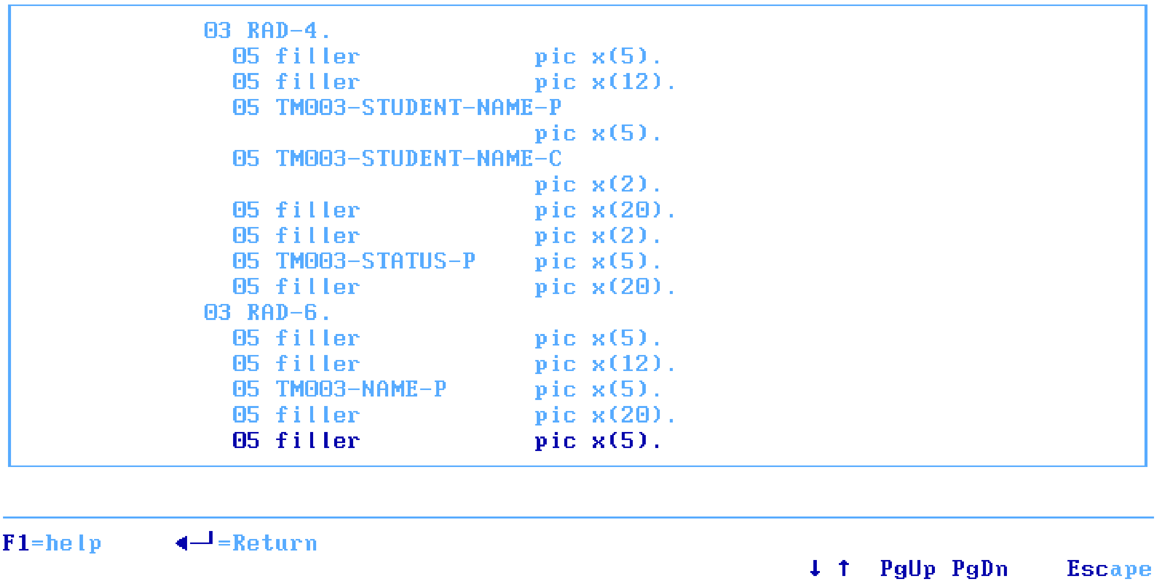
Form generation will generate a COBOL description of the Pilspråk form and give it the same name as the form. Generated forms may be included in your apllication program through COPY <formname>.



```
Generating $G8WBFDIR\fms\cpy\TMNN3.CPY
      * Generated from GCOS-8 COBOL Workbench 94-11-22 at 15.08.18
      * Last Modified 94-11-22 at 15.08.14
      * Title "Student Inquiry               "
       01  TMNN3.
           02 TMNN3-field-def.
            03 RAD-1.
              05 filler             pic x(4)      value "_C_X".
              05 filler             pic x(5)      value "_0101".
              05 filler             pic x(2)      value "_I".
              05 Unfield-1          pic x(8)      value "TM003".
              05 filler             pic x(2)      value "_S".
              05 filler             pic x(5)      value "_0120".
              05 filler             pic x(35)     value
                                  "TM00 - Workshop 3 - Student Inquiry".
            03 RAD-4.
              05 filler             pic x(5)      value "_0420".

F1=help     ◄┘=Return
                                                      ↓ ↑  PgUp PgDn    Escape
```

*Note: The COPY is compatible with GCOS-8.*

If you have Extended COPY enabled, ref. Configuration chapter for details, the Position and attribute fields in the generate form will be labeled, i.e. <Field-name>-P for position and <Field-name>-C for attribute.



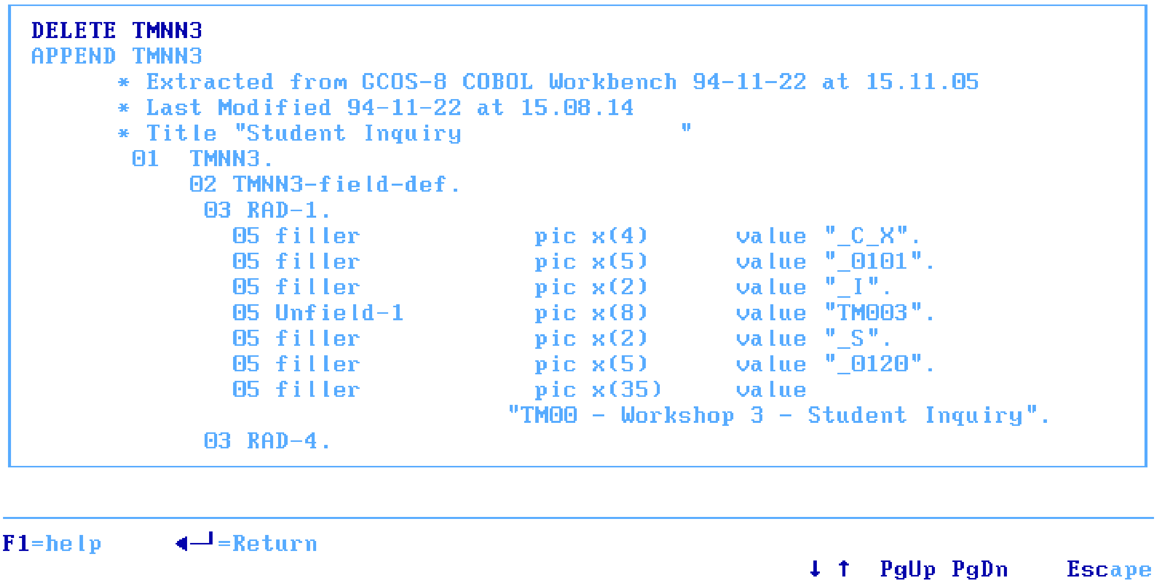
```
            03 RAD-4.
              05 filler             pic x(5).
              05 filler             pic x(12).
              05 TM003-STUDENT-NAME-P
                                    pic x(5).
              05 TM003-STUDENT-NAME-C
                                    pic x(2).
              05 filler             pic x(20).
              05 filler             pic x(2).
              05 TM003-STATUS-P     pic x(5).
              05 filler             pic x(20).
            03 RAD-6.
              05 filler             pic x(5).
              05 filler             pic x(12).
              05 TM003-NAME-P       pic x(5).
              05 filler             pic x(20).
              05 filler             pic x(5).

F1=help     ◄┘=Return
                                                      ↓ ↑  PgUp PgDn    Escape
```

You may use the labeled Position and/or Attribute fields to dynamically change attribute setting in the Pilspråk form or build dynamic update strings to update a Pilspråk form, without sending the entire form.

## Import/Export

Pilspråk forms are plain COBOL structures, which may be exchanged with GCOS-8. WUSP will assume Pilspråk forms are maintained in GCOS-8 COPY libraries. Import files are thus assumed to be in SRCLIB or SSTLIB extract file format. WUSP will build export files, which may be used to update a COPY library on GCOS-8 either through SRCLIB or SSTLIB.

During initial import of existing Pilspråk forms, WUSP will analyze the current COBOL description and break it down into Text, Output and Update fields. Any unrecognized sequence will be treated as a Command field. For Text, Output and Update fields the position is adjusted in according with the attribute assignment. When you subsequently export a form, the generated COBOL description will look different, but work functionally equivalent.



```
DELETE TMNN3
APPEND TMNN3
      * Extracted from GCOS-8 COBOL Workbench 94-11-22 at 15.11.05
      * Last Modified 94-11-22 at 15.08.14
      * Title "Student Inquiry               "
       01  TMNN3.
           02 TMNN3-field-def.
            03 RAD-1.
              05 filler            pic x(4)      value "_C_X".
              05 filler            pic x(5)      value "_0101".
              05 filler            pic x(2)      value "_I".
              05 Unfield-1         pic x(8)      value "TM003".
              05 filler            pic x(2)      value "_S".
              05 filler            pic x(5)      value "_0120".
              05 filler            pic x(35)     value
                                 "TM00 - Workshop 3 - Student Inquiry".
            03 RAD-4.
```

```
F1=help      ◄┘=Return
                                                  ↓ ↑  PgUp PgDn    Escape
```

# DIMS8

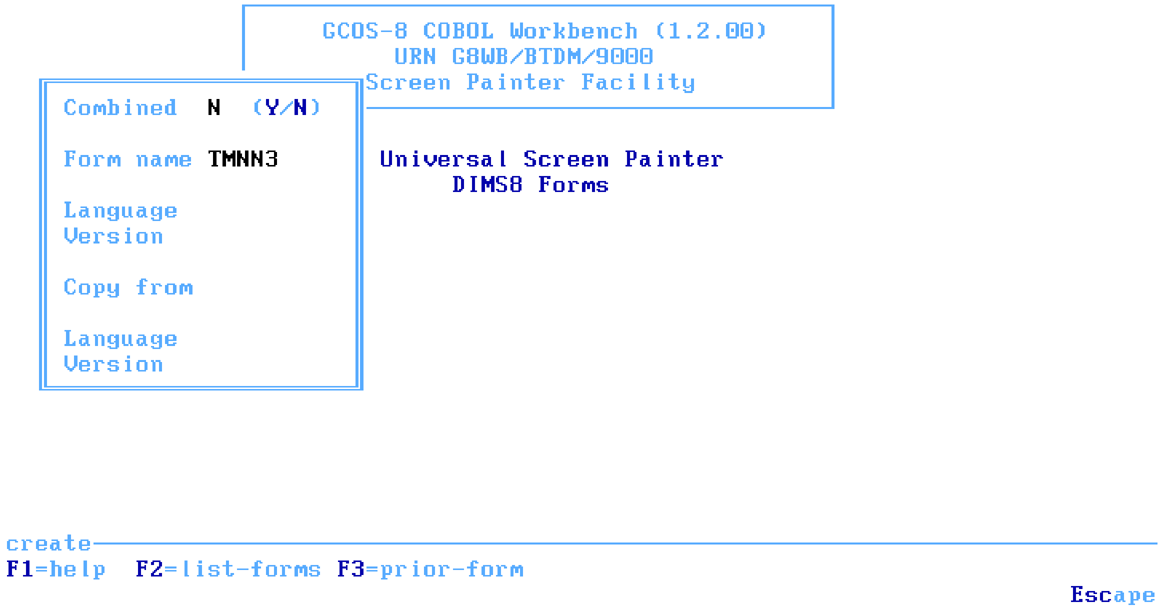
Following illustrate the DIMS8 specific features, panels and restrictions in relation to WUSP. Please refer to B & C Solution DIMS8 documentation for details on attribute settings etc.

WUSP support following DIMS8 definitions:

- Display Definition (Form)
- Storage Definition
- Transaction Definition

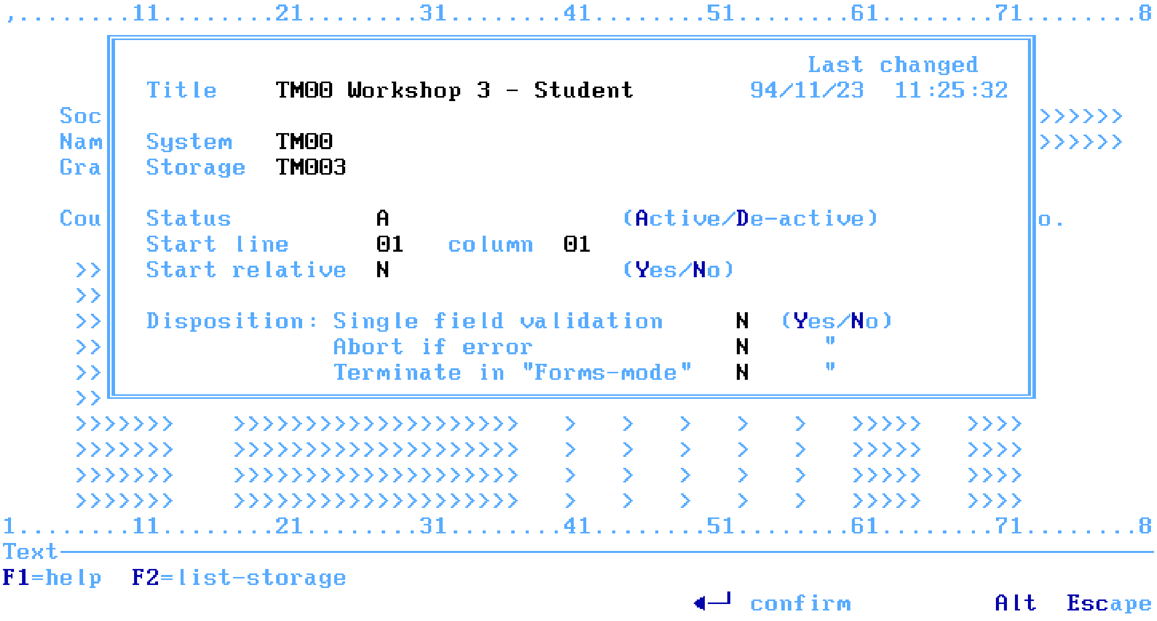
## Display Definition

Identification of a Display Definition may consist of a form name, a language code and a version code. When working with Form maintenance, the WUSP identification panels are extended to include this identification. When creating a form, you may also designate the form as either a Combined or Normal form.



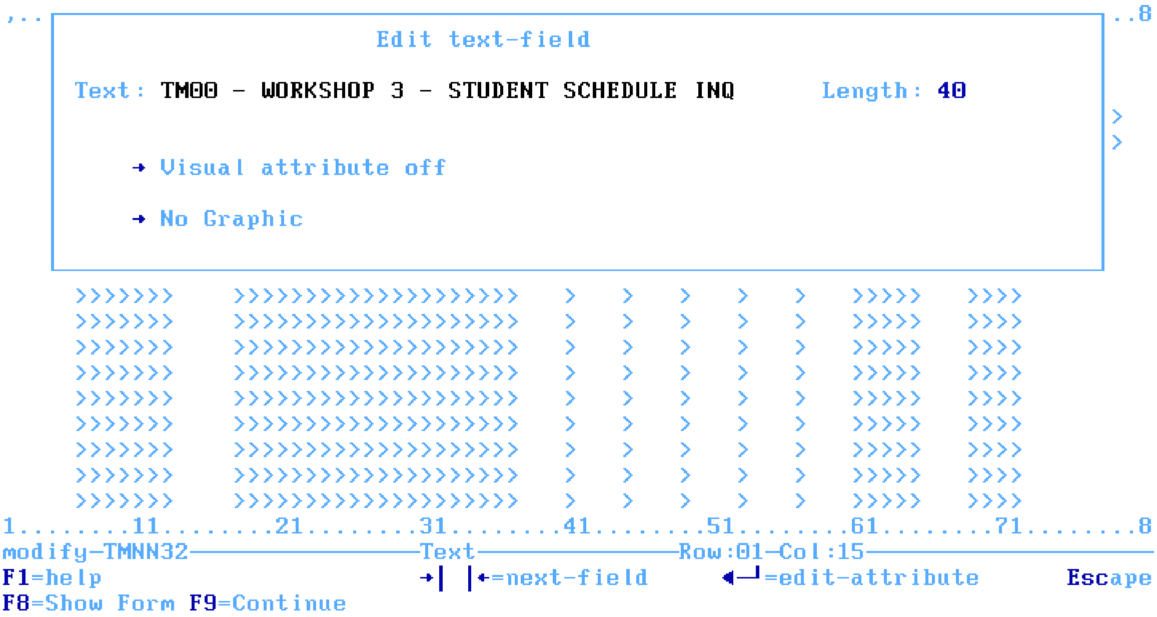
```
                     GCOS-8 COBOL Workbench (1.2.00)
                          URN G8WB/BTDM/9000
                    Screen Painter Facility
  Combined   N   (Y/N)

  Form name TMNN3          Universal Screen Painter
                                DIMS8 Forms
  Language
  Version

  Copy from

  Language
  Version




create
F1=help   F2=list-forms F3=prior-form
                                                                         Escape
```

## Form title



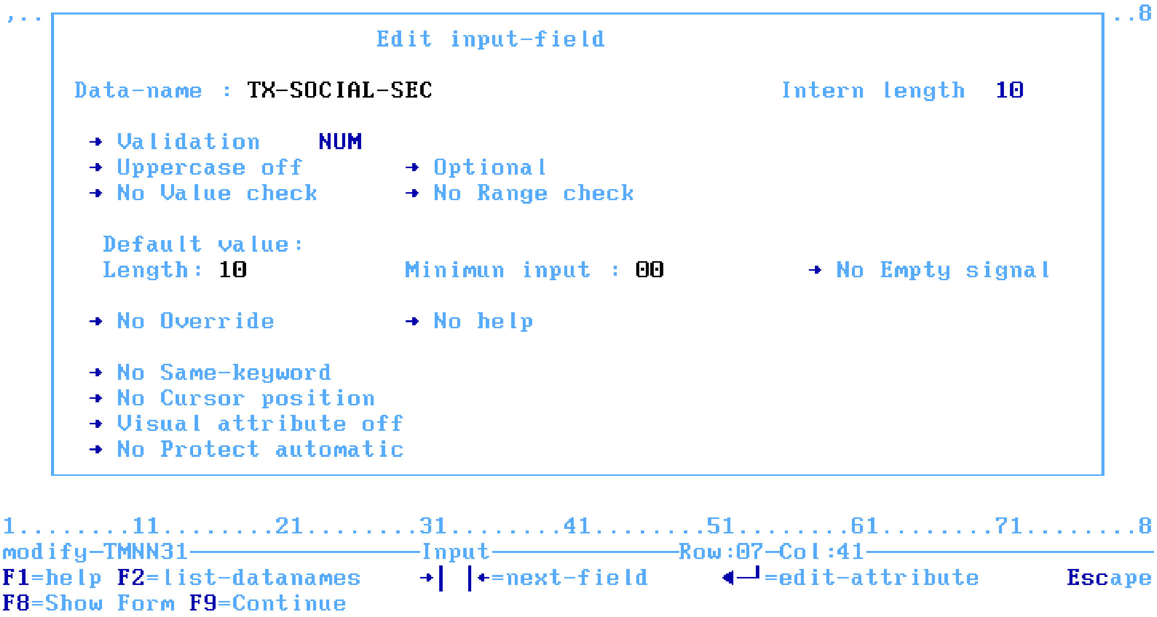
```
,........11........21........31........41........51........61........71........8
       ┌──────────────────────────────────────────────────────────────────┐
       │                                                    Last changed   │
       │  Title     TM00 Workshop 3 - Student            94/11/23  11:25:32│
   Soc │                                                                   │>>>>>>
   Nam │  System    TM00                                                   │>>>>>>
   Gra │  Storage   TM003                                                  │
       │                                                                   │
   Cou │  Status           A                 (Active/De-active)            │o.
       │  Start line       01   column  01                                 │
    >> │  Start relative   N                 (Yes/No)                      │
    >> │                                                                   │
    >> │  Disposition: Single field validation     N  (Yes/No)             │
    >> │               Abort if error              N      "                │
    >> │               Terminate in "Forms-mode"   N      "                │
    >> └──────────────────────────────────────────────────────────────────┘
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
1........11........21........31........41........51........61........71........8
Text───────────────────────────────────────────────────────────────────────────────
F1=help  F2=list-storage
                                                ◄┘ confirm          Alt  Escape
```

## Text field



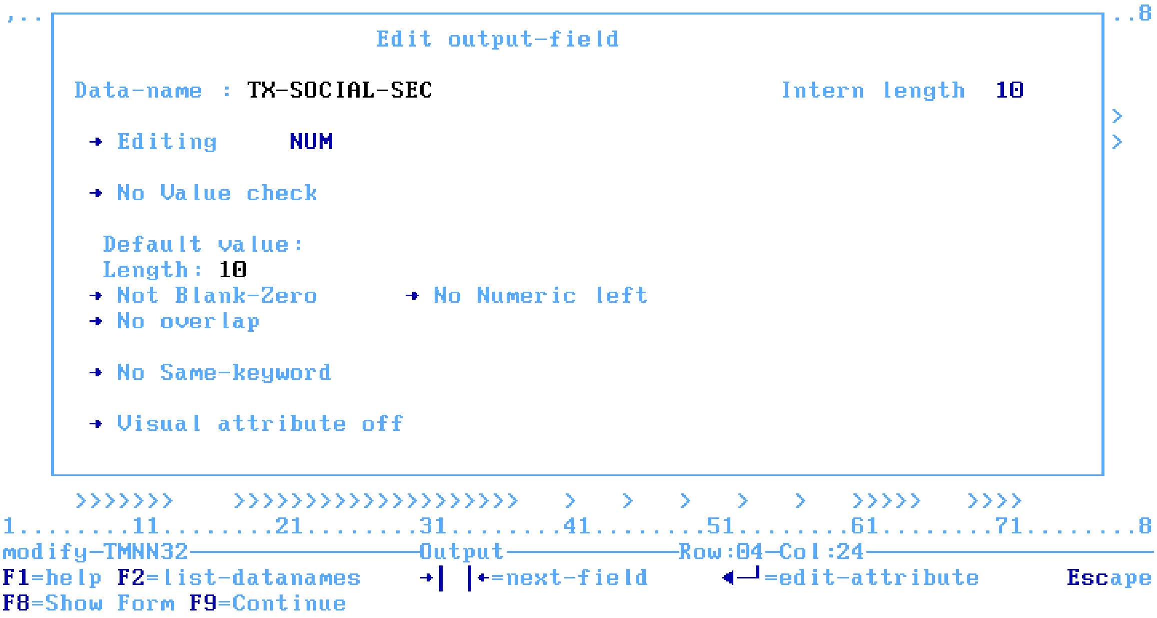
```
,..┌─────────────────────────────────────────────────────────────────────────┐..8
   │                      Edit text-field                                    │
   │                                                                         │
   │ Text: TM00 - WORKSHOP 3 - STUDENT SCHEDULE INQ       Length: 40         │
   │                                                                         │>
   │                                                                         │>
   │     → Visual attribute off                                              │
   │                                                                         │
   │     → No Graphic                                                        │
   │                                                                         │
   └─────────────────────────────────────────────────────────────────────────┘
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
    >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>   >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN32────────────────Text─────────────────Row:01-Col:15────────────────────
F1=help                      →| |←=next-field    ◄┘=edit-attribute       Escape
F8=Show Form F9=Continue
```

## Input field



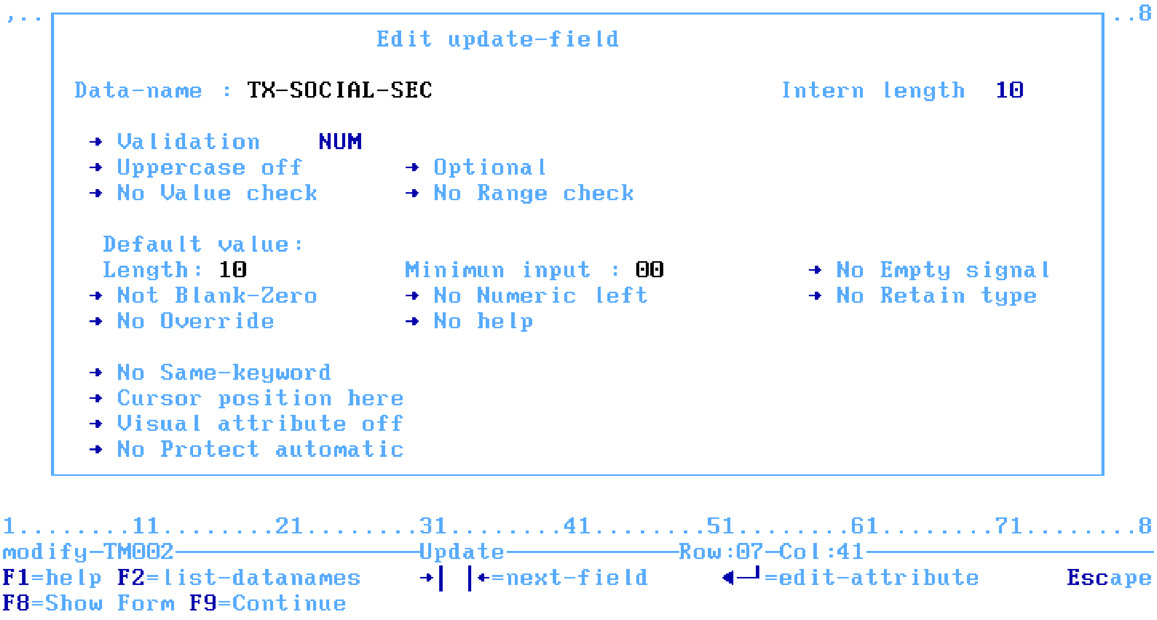
```
,..  ┌───────────────────────────────────────────────────────────────────────────┐ ..8
     │                      Edit input-field                                     │
     │                                                                           │
     │ Data-name : TX-SOCIAL-SEC                          Intern length  10      │
     │                                                                           │
     │  → Validation    NUM                                                      │
     │  → Uppercase off        → Optional                                        │
     │  → No Value check       → No Range check                                  │
     │                                                                           │
     │   Default value:                                                          │
     │   Length: 10            Minimun input : 00         → No Empty signal      │
     │                                                                           │
     │  → No Override          → No help                                         │
     │                                                                           │
     │  → No Same-keyword                                                        │
     │  → No Cursor position                                                     │
     │  → Visual attribute off                                                   │
     │  → No Protect automatic                                                   │
     └───────────────────────────────────────────────────────────────────────────┘

1........11........21........31........41........51........61........71........8
modify-TMNN31────────────────Input─────────────Row:07-Col:41─────────────────────
F1=help F2=list-datanames   →| |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Output field



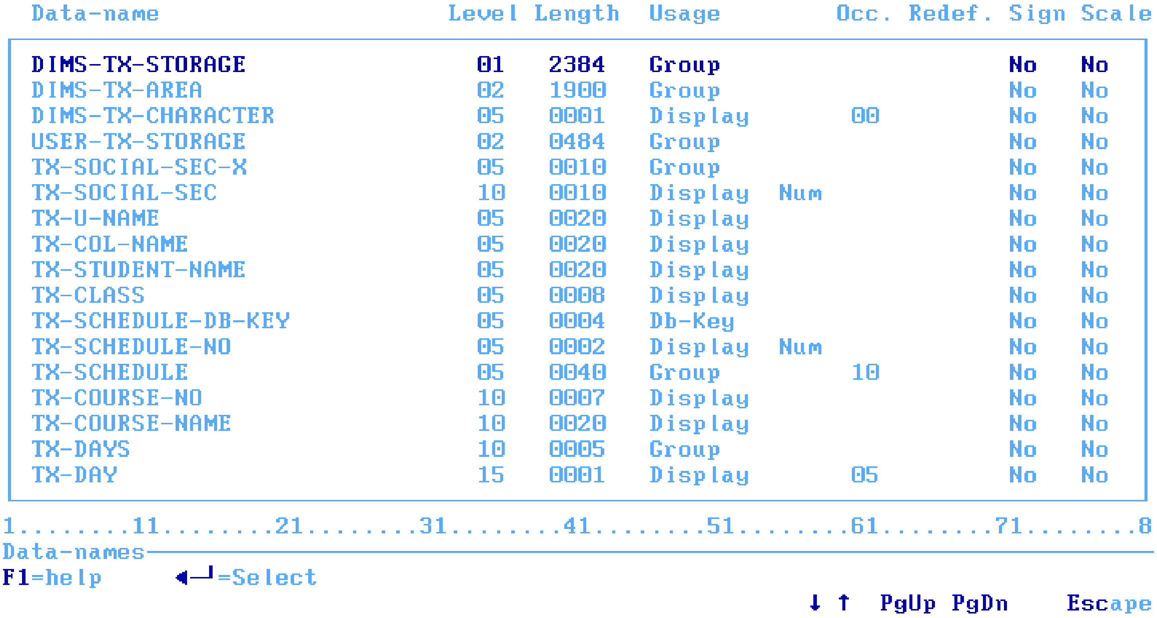
```
,..  ┌───────────────────────────────────────────────────────────────────────────┐ ..8
     │                      Edit output-field                                    │
     │                                                                           │
     │ Data-name : TX-SOCIAL-SEC                          Intern length  10      │ >
     │                                                                           │ >
     │  → Editing     NUM                                                        │
     │                                                                           │
     │  → No Value check                                                         │
     │                                                                           │
     │   Default value:                                                          │
     │   Length: 10                                                              │
     │  → Not Blank-Zero       → No Numeric left                                 │
     │  → No overlap                                                             │
     │                                                                           │
     │  → No Same-keyword                                                        │
     │                                                                           │
     │  → Visual attribute off                                                   │
     │                                                                           │
     └───────────────────────────────────────────────────────────────────────────┘
     >>>>>>>    >>>>>>>>>>>>>>>>>>>>   >   >   >   >   >   >>>>>    >>>>
1........11........21........31........41........51........61........71........8
modify-TMNN32────────────────Output────────────Row:04-Col:24─────────────────────
F1=help F2=list-datanames   →| |←=next-field    ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Update field



```
,..                                                                       ..8
                       Edit update-field

    Data-name : TX-SOCIAL-SEC                         Intern length   10

     → Validation     NUM
     → Uppercase off         → Optional
     → No Value check        → No Range check

      Default value:
      Length: 10             Minimun input : 00          → No Empty signal
     → Not Blank-Zero        → No Numeric left           → No Retain type
     → No Override           → No help

     → No Same-keyword
     → Cursor position here
     → Visual attribute off
     → No Protect automatic

1........11........21........31........41........51........61........71........8
modify-TM002─────────────────Update───────────Row:07─Col:41──────────────────────
F1=help F2=list-datanames    →| |←=next-field    ◄┘=edit-attribute        Escape
F8=Show Form F9=Continue
```

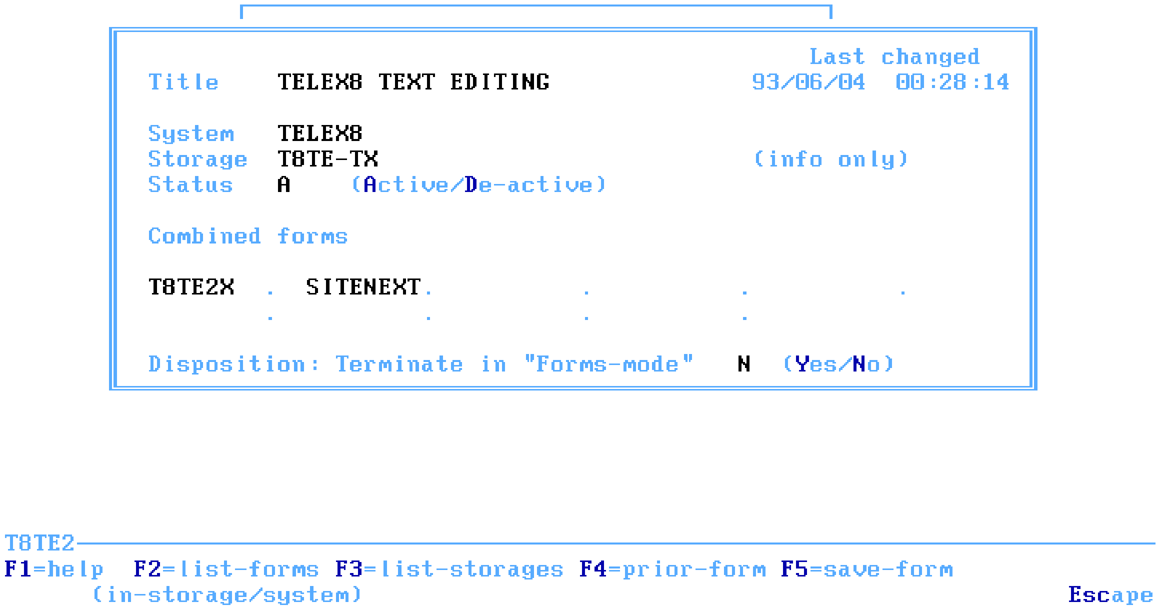
F2-List-datanames may be used to browse the Storage Definition and select an application field.



```
  Data-name                     Level Length  Usage        Occ. Redef. Sign Scale

  DIMS-TX-STORAGE                 01    2384   Group                     No   No
  DIMS-TX-AREA                    02    1900   Group                     No   No
  DIMS-TX-CHARACTER               05    0001   Display       00          No   No
  USER-TX-STORAGE                 02    0484   Group                     No   No
  TX-SOCIAL-SEC-X                 05    0010   Group                     No   No
  TX-SOCIAL-SEC                   10    0010   Display  Num              No   No
  TX-U-NAME                       05    0020   Display                   No   No
  TX-COL-NAME                     05    0020   Display                   No   No
  TX-STUDENT-NAME                 05    0020   Display                   No   No
  TX-CLASS                        05    0008   Display                   No   No
  TX-SCHEDULE-DB-KEY              05    0004   Db-Key                    No   No
  TX-SCHEDULE-NO                  05    0002   Display  Num              No   No
  TX-SCHEDULE                     05    0040   Group         10          No   No
  TX-COURSE-NO                    10    0007   Display                   No   No
  TX-COURSE-NAME                  10    0020   Display                   No   No
  TX-DAYS                         10    0005   Group                     No   No
  TX-DAY                          15    0001   Display       05          No   No

1........11........21........31........41........51........61........71........8
Data-names────────────────────────────────────────────────────────────────────────
F1=help     ◄┘=Select
                                                        ↓ ↑  PgUp PgDn     Escape
```

## Combined

Combined forms are maintained through a separate form panel.



```
                                                          Last changed
  Title     TELEX8 TEXT EDITING                       93/06/04  00:28:14

  System    TELEX8
  Storage   T8TE-TX                                   (info only)
  Status    A     (Active/De-active)

  Combined forms

  T8TE2X   .  SITENEXT.          .          .          .
           .          .          .          .

  Disposition: Terminate in "Forms-mode"   N  (Yes/No)




T8TE2
F1=help  F2=list-forms F3=list-storages F4=prior-form F5=save-form
      (in-storage/system)                                        Escape
```

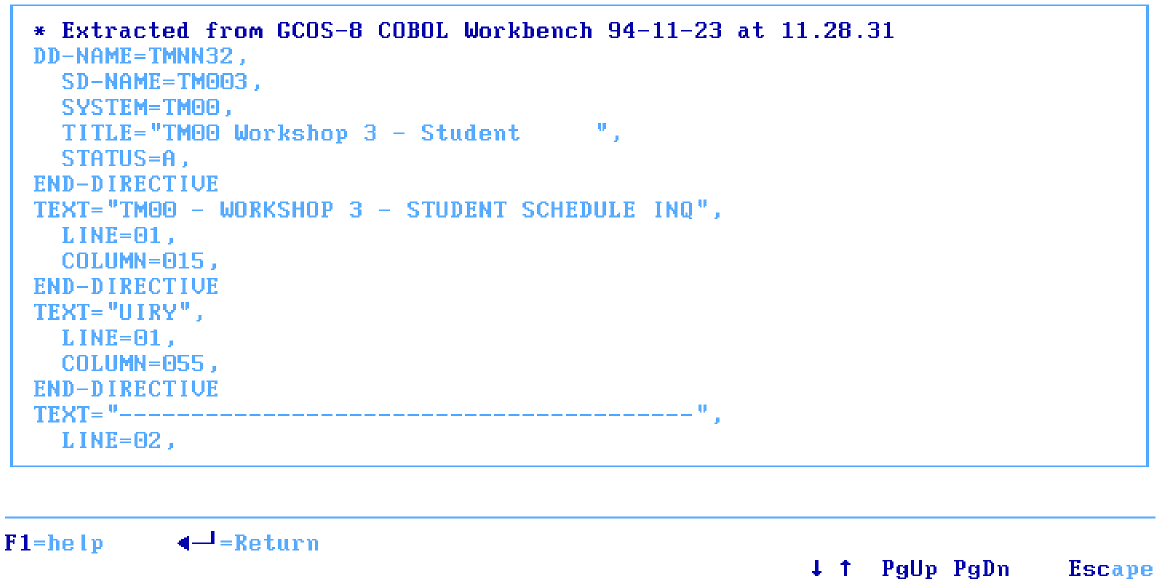
## Generation

Generation of a form is automatic and independent of the application program. Once a form is used, DIMS8 will build a run-time version of the form under $G8WBFDIR\FMS\RDD. The run-time version is automatically deleted if a form is changed, or if the Storage definition is changed.

## Import/Export

Export/import function may be used to exchange forms (Display Definitions) with GCOS-8 or another G8WB environment. The standard DIMS8 DCL syntax is used.

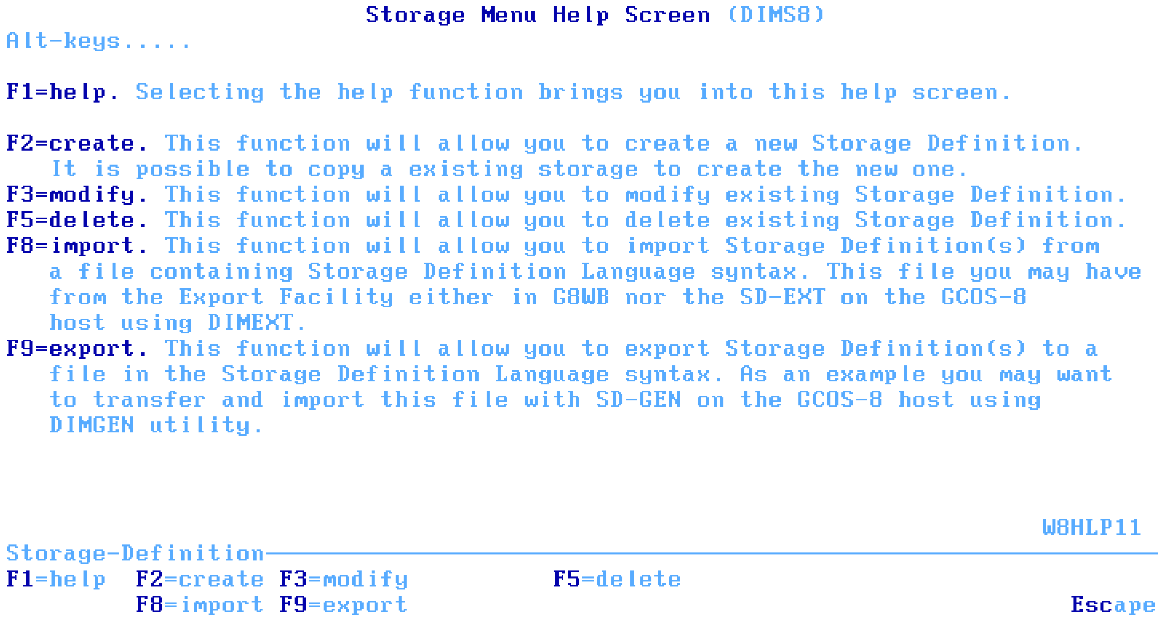
The export- and import-files have default extension of .DD, i.e. <form-name>.DD.



```
* Extracted from GCOS-8 COBOL Workbench 94-11-23 at 11.28.31
DD-NAME=TMNN32,
  SD-NAME=TM003,
  SYSTEM=TM00,
  TITLE="TM00 Workshop 3 - Student     ",
  STATUS=A,
END-DIRECTIVE
TEXT="TM00 - WORKSHOP 3 - STUDENT SCHEDULE INQ",
  LINE=01,
  COLUMN=015,
END-DIRECTIVE
TEXT="UIRY",
  LINE=01,
  COLUMN=055,
END-DIRECTIVE
TEXT="----------------------------------------",
  LINE=02,

F1=help     ◄┘=Return
                                              ↓ ↑  PgUp PgDn     Escape
```

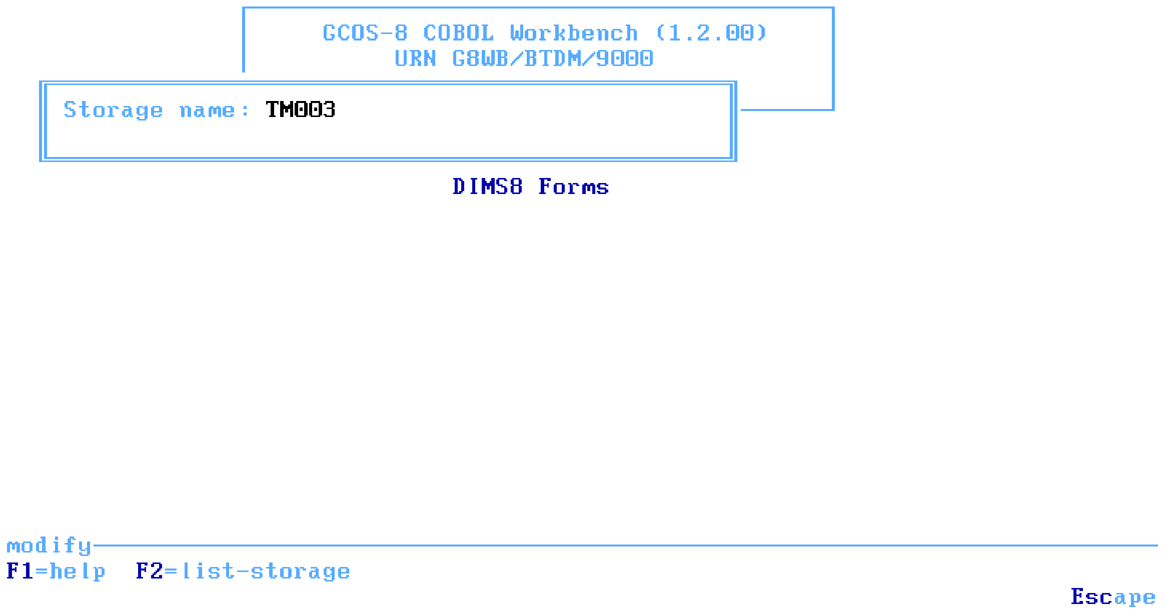
## Storage Definition

Storage definition maintenance is available through the ALT menu, which is enabled when you hold down ALT key while in the main forms menu.



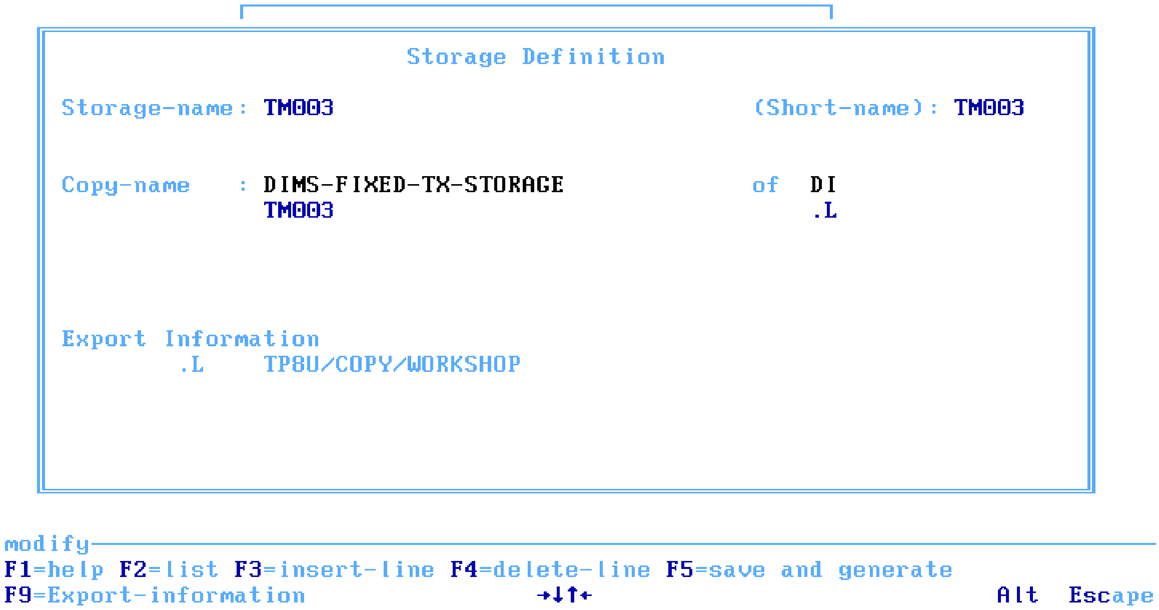
```
                              Storage Menu Help Screen (DIMS8)
Alt-keys.....

F1=help. Selecting the help function brings you into this help screen.

F2=create. This function will allow you to create a new Storage Definition.
   It is possible to copy a existing storage to create the new one.
F3=modify. This function will allow you to modify existing Storage Definition.
F5=delete. This function will allow you to delete existing Storage Definition.
F8=import. This function will allow you to import Storage Definition(s) from
   a file containing Storage Definition Language syntax. This file you may have
   from the Export Facility either in G8WB nor the SD-EXT on the GCOS-8
   host using DIMEXT.
F9=export. This function will allow you to export Storage Definition(s) to a
   file in the Storage Definition Language syntax. As an example you may want
   to transfer and import this file with SD-GEN on the GCOS-8 host using
   DIMGEN utility.



                                                                          W8HLP11
Storage-Definition────────────────────────────────────────────────────────────────
F1=help  F2=create F3=modify          F5=delete
         F8=import F9=export                                                Escape
```

Principles for Create, Modify, Delete, Import and Export functions for Storage Definition maintenance are equivalent to functions for Display Definition maintenance. Identification and detail panels are by nature different.



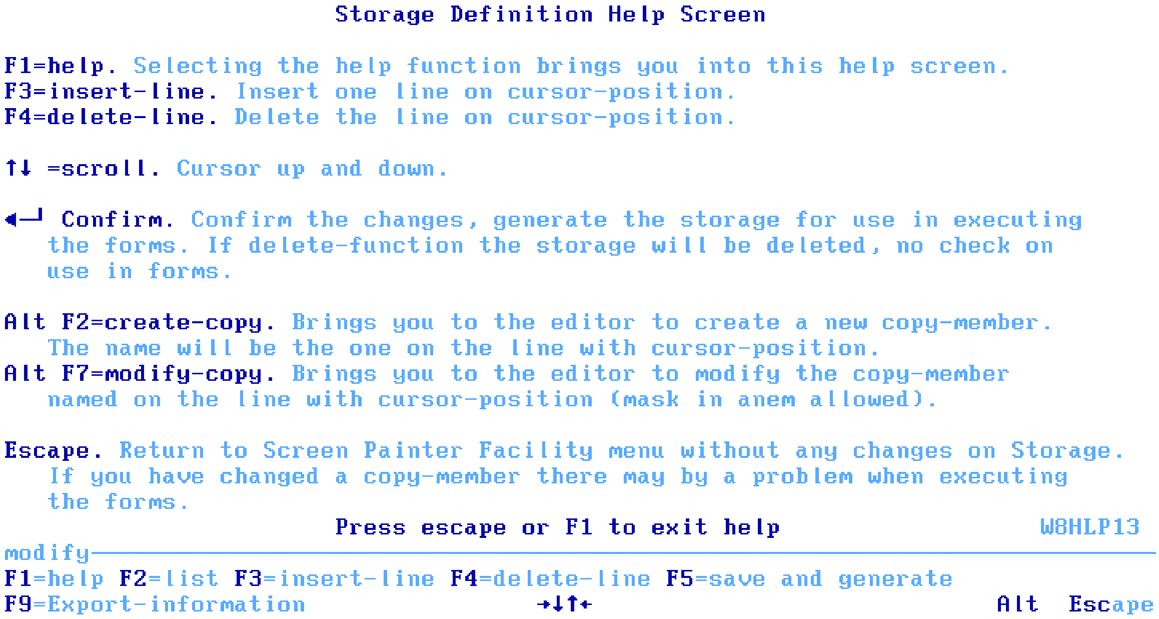
```
                    GCOS-8 COBOL Workbench (1.2.00)
                         URN G8WB/BTDM/9000
  Storage name: TM003


                               DIMS8 Forms


modify────────────────────────────────────────────────────────────────────────────
F1=help  F2=list-storage
                                                                            Escape
```

## Storage panel



```
                         Storage Definition

 Storage-name: TM003                            (Short-name): TM003


 Copy-name    : DIMS-FIXED-TX-STORAGE            of   DI
                TM003                                 .L




 Export Information
         .L     TP8U/COPY/WORKSHOP




modify──────────────────────────────────────────────────────────────────────────
F1=help F2=list F3=insert-line F4=delete-line F5=save and generate
F9=Export-information                →↓↑←                          Alt   Escape
```

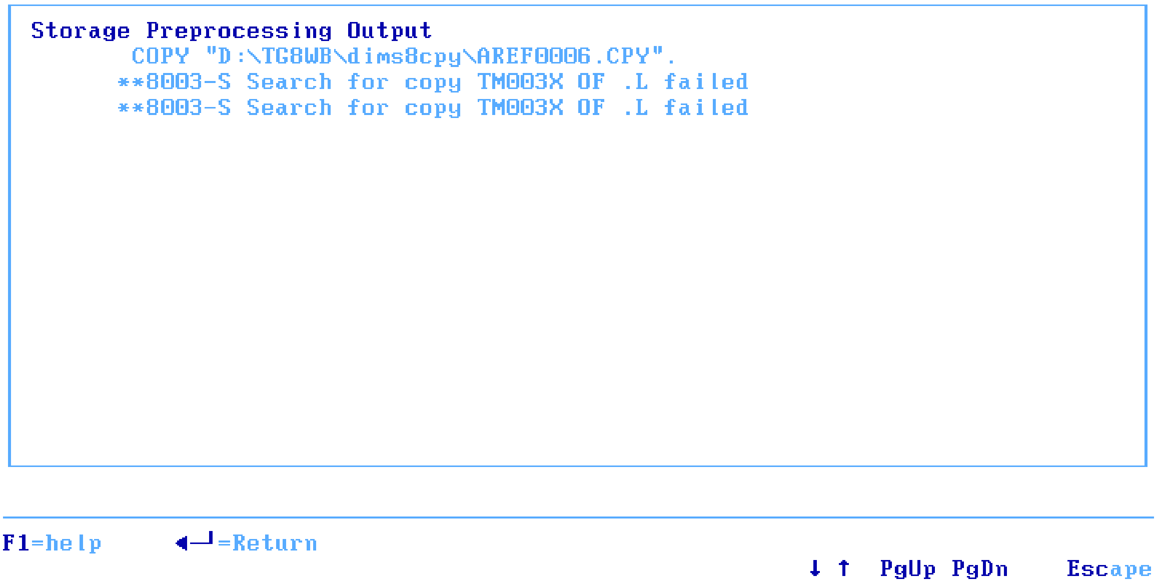
A short-name will be assigned if the storage-name is > 8 characters, i.e. SREF0001.



```
                      Storage Definition Help Screen

F1=help. Selecting the help function brings you into this help screen.
F3=insert-line. Insert one line on cursor-position.
F4=delete-line. Delete the line on cursor-position.

↑↓ =scroll. Cursor up and down.

◄┘ Confirm. Confirm the changes, generate the storage for use in executing
   the forms. If delete-function the storage will be deleted, no check on
   use in forms.

Alt F2=create-copy. Brings you to the editor to create a new copy-member.
   The name will be the one on the line with cursor-position.
Alt F7=modify-copy. Brings you to the editor to modify the copy-member
   named on the line with cursor-position (mask in anem allowed).

Escape. Return to Screen Painter Facility menu without any changes on Storage.
   If you have changed a copy-member there may by a problem when executing
   the forms.
                      Press escape or F1 to exit help                  W8HLP13
modify──────────────────────────────────────────────────────────────────────────
F1=help F2=list F3=insert-line F4=delete-line F5=save and generate
F9=Export-information                →↓↑←                          Alt   Escape
```

The Export information is used to created GCOS-8 compatible syntax during export, i.e. syntax must include Catalog-file-string(s) for your COPY library.

F5-Save-and-generate will both save the Storage Definition and “compile” the COBOL descriptions similar to a DIMS8 DIMGEN of the Storage Definition. If errors are detected during “compile”, these are presented in a browse panel.

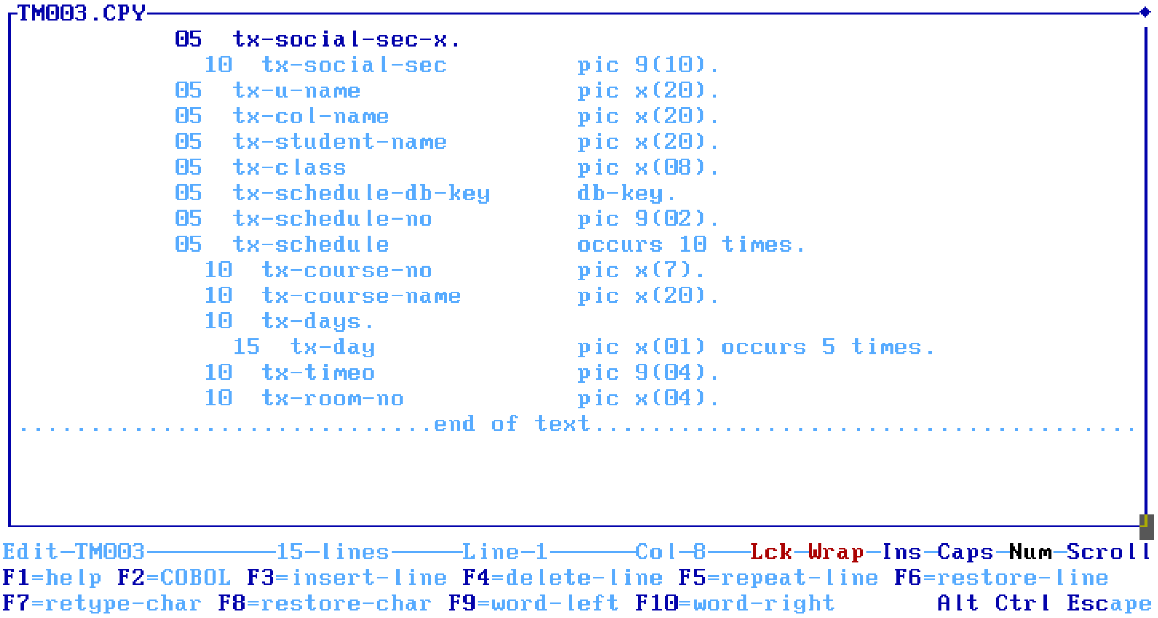


```
 Storage Preprocessing Output
        COPY "D:\TG8WB\dims8cpy\AREF0006.CPY".
       **8003-S Search for copy TM003X OF .L failed
       **8003-S Search for copy TM003X OF .L failed
```

```
F1=help     ◄┘=Return
                                                   ↓ ↑  PgUp PgDn    Escape
```

Alt-F7-modify-copy may be used to invoke Editor with the “current” COPY. You may make changes as required, resave, Escape back to Storage panel and retry Save-and-Generate.

Alt-F2-create-copy is similar to Alt-F7-modify-copy, except that “current” COPY is first created.



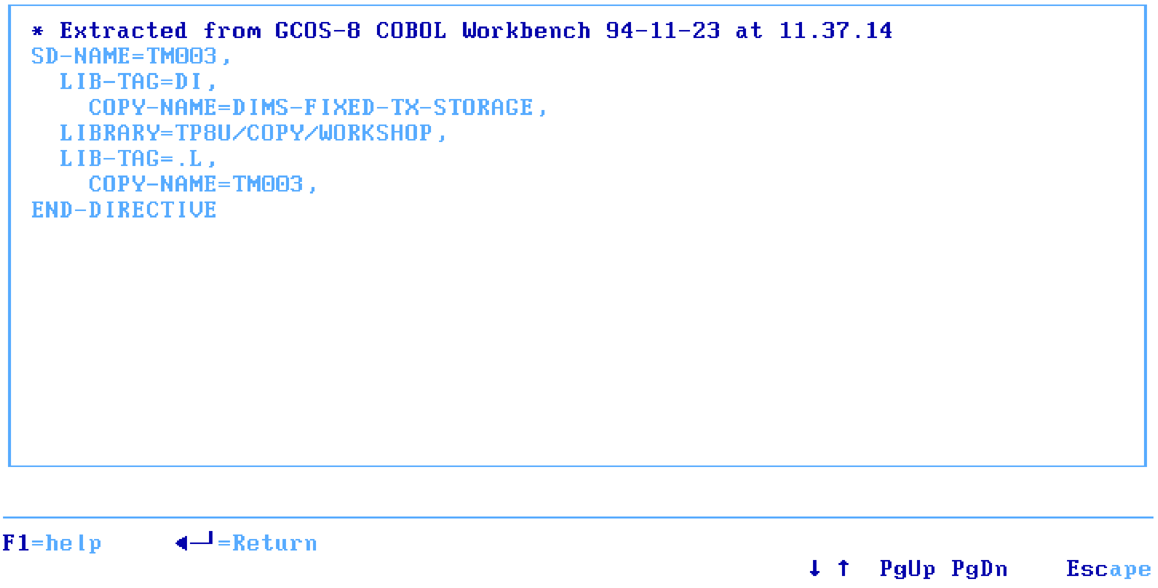
```
┌TM003.CPY──────────────────────────────────────────────────────────────────────
           05  tx-social-sec-x.
             10  tx-social-sec          pic 9(10).
           05  tx-u-name                pic x(20).
           05  tx-col-name              pic x(20).
           05  tx-student-name          pic x(20).
           05  tx-class                 pic x(08).
           05  tx-schedule-db-key       db-key.
           05  tx-schedule-no           pic 9(02).
           05  tx-schedule              occurs 10 times.
             10  tx-course-no           pic x(7).
             10  tx-course-name         pic x(20).
             10  tx-days.
               15  tx-day               pic x(01) occurs 5 times.
             10  tx-timeo               pic 9(04).
             10  tx-room-no             pic x(04).
.............................end of text.......................................
```

```
Edit─TM003─────────15─lines─────Line─1──────Col─8────Lck─Wrap─Ins─Caps─Num─Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

## Import/Export

Export/import function may be used to exchange Storage Definitions with GCOS-8 or another G8WB environment. The standard DIMS8 DCL syntax is used.

The export- and import-files have default extension of .SD, i.e. <storage-short-name>.SD.

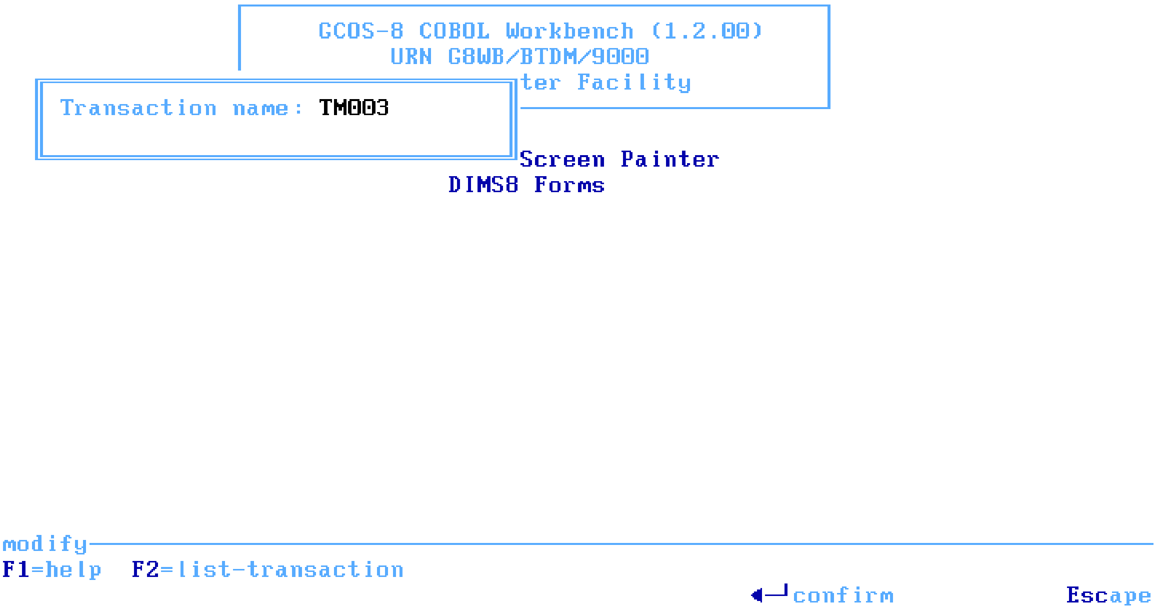


```
* Extracted from GCOS-8 COBOL Workbench 94-11-23 at 11.37.14
SD-NAME=TM003,
  LIB-TAG=DI,
    COPY-NAME=DIMS-FIXED-TX-STORAGE,
  LIBRARY=TP8U/COPY/WORKSHOP,
  LIB-TAG=.L,
    COPY-NAME=TM003,
END-DIRECTIVE
```

*Note: COPY’s are not included in the import or export.*

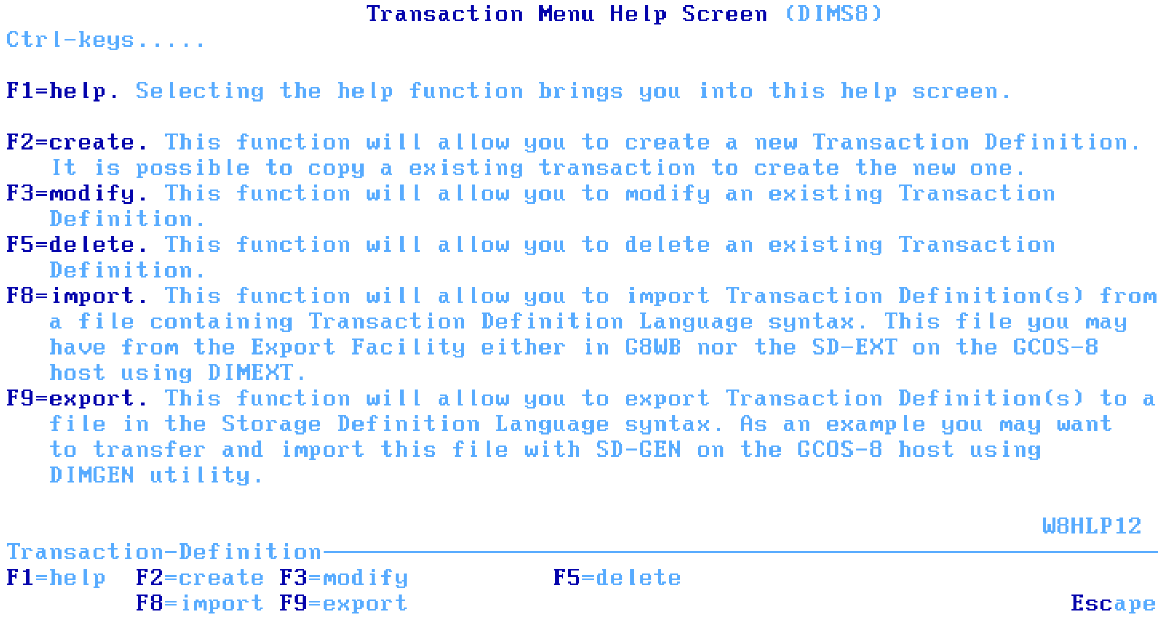
## Transaction Definition

Transaction definition maintenance is available through the CTRL menu, which is enabled when you hold down CTRL key while in the main forms menu.



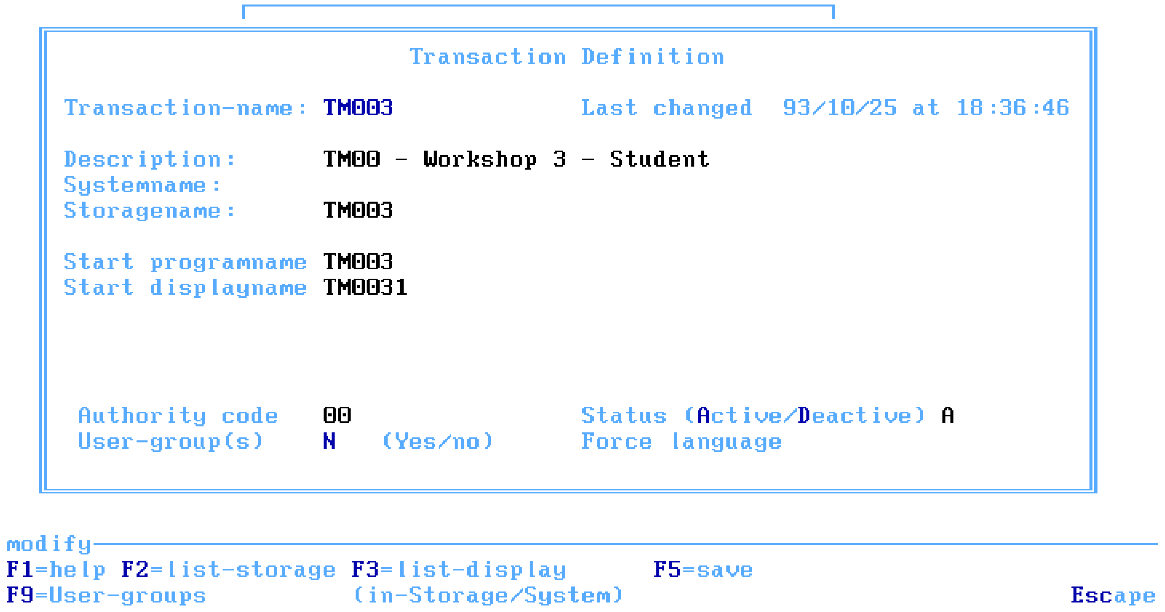
Transaction name: TM003

Principles for Create, Modify, Delete, Import and Export functions for Transaction Definition maintenance are equivalent to functions for Display Definition maintenance. Identification and detail panels are by nature different.



```
                         Transaction Menu Help Screen (DIMS8)
Ctrl-keys.....

F1=help. Selecting the help function brings you into this help screen.

F2=create. This function will allow you to create a new Transaction Definition.
   It is possible to copy a existing transaction to create the new one.
F3=modify. This function will allow you to modify an existing Transaction
   Definition.
F5=delete. This function will allow you to delete an existing Transaction
   Definition.
F8=import. This function will allow you to import Transaction Definition(s) from
   a file containing Transaction Definition Language syntax. This file you may
   have from the Export Facility either in G8WB nor the SD-EXT on the GCOS-8
   host using DIMEXT.
F9=export. This function will allow you to export Transaction Definition(s) to a
   file in the Storage Definition Language syntax. As an example you may want
   to transfer and import this file with SD-GEN on the GCOS-8 host using
   DIMGEN utility.

                                                                       W8HLP12
Transaction-Definition----------------------------------------------------------
F1=help  F2=create F3=modify          F5=delete
         F8=import F9=export                                             Escape
```

## Transaction panel

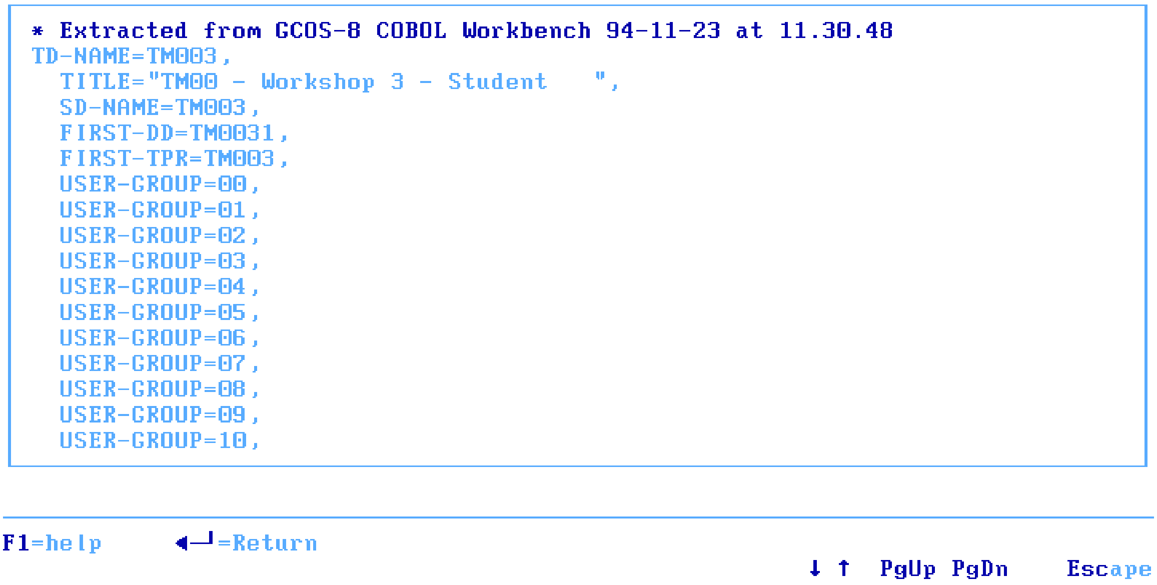


```
                          Transaction Definition

 Transaction-name: TM003            Last changed  93/10/25 at 18:36:46

 Description:      TM00 - Workshop 3 - Student
 Systemname:
 Storagename:      TM003

 Start programname TM003
 Start displayname TM0031



  Authority code   00               Status (Active/Deactive) A
  User-group(s)    N   (Yes/no)     Force language

modify--------------------------------------------------------------------------
F1=help F2=list-storage F3=list-display       F5=save
F9=User-groups          (in-Storage/System)                              Escape
```

## Import/Export

Export/import function may be used to exchange Transaction Definitions with GCOS-8 or another G8WB environment. The standard DIMS8 DCL syntax is used.

The export- and import-files have default extension of .TD, i.e. <transaction-name>.TD.

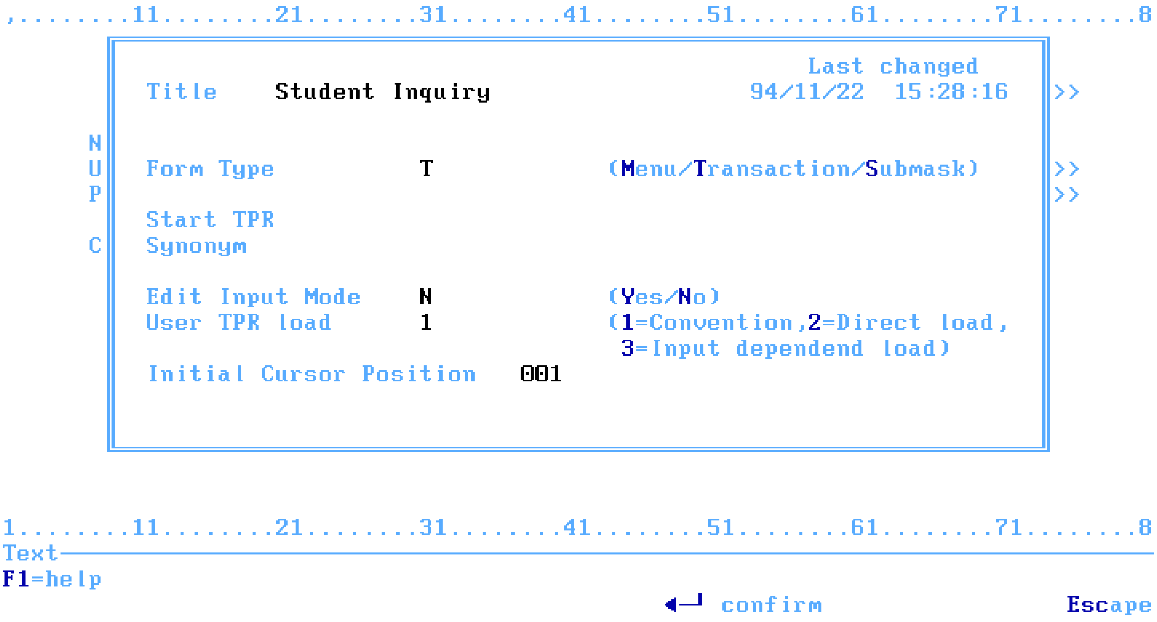


```
* Extracted from GCOS-8 COBOL Workbench 94-11-23 at 11.30.48
TD-NAME=TM003,
  TITLE="TM00 - Workshop 3 - Student    ",
  SD-NAME=TM003,
  FIRST-DD=TM0031,
  FIRST-TPR=TM003,
  USER-GROUP=00,
  USER-GROUP=01,
  USER-GROUP=02,
  USER-GROUP=03,
  USER-GROUP=04,
  USER-GROUP=05,
  USER-GROUP=06,
  USER-GROUP=07,
  USER-GROUP=08,
  USER-GROUP=09,
  USER-GROUP=10,
```

# SOLD

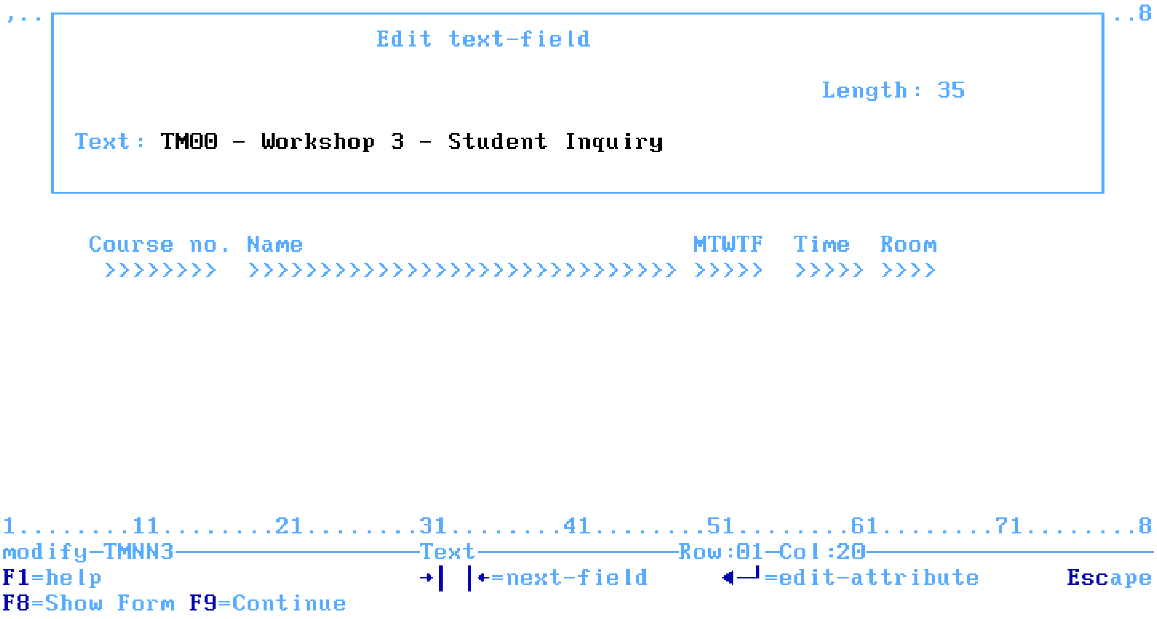
Following illustrate the SOLD specific features, panels and restrictions in relation to WUSP. Please refer to BVG SOLD documentation for details on attribute settings etc.

## Form title



```
,.......11.......21.......31.......41.......51.......61.......71.......8
   ┌────────────────────────────────────────────────────────────────┐
   │                                             Last changed       │
   │  Title    Student Inquiry                 94/11/22  15:28:16  │>>
   │                                                                │
  N│                                                                │
  U│  Form Type         T         (Menu/Transaction/Submask)        │>>
  P│                                                                │>>
   │  Start TPR                                                     │
  C│  Synonym                                                       │
   │                                                                │
   │  Edit Input Mode   N         (Yes/No)                          │
   │  User TPR load     1         (1=Convention,2=Direct load,      │
   │                               3=Input dependend load)          │
   │  Initial Cursor Position  001                                  │
   │                                                                │
   └────────────────────────────────────────────────────────────────┘

1.......11.......21.......31.......41.......51.......61.......71.......8
Text────────────────────────────────────────────────────────────────────
F1=help
                                            ◄┘ confirm           Escape
```

## Text field

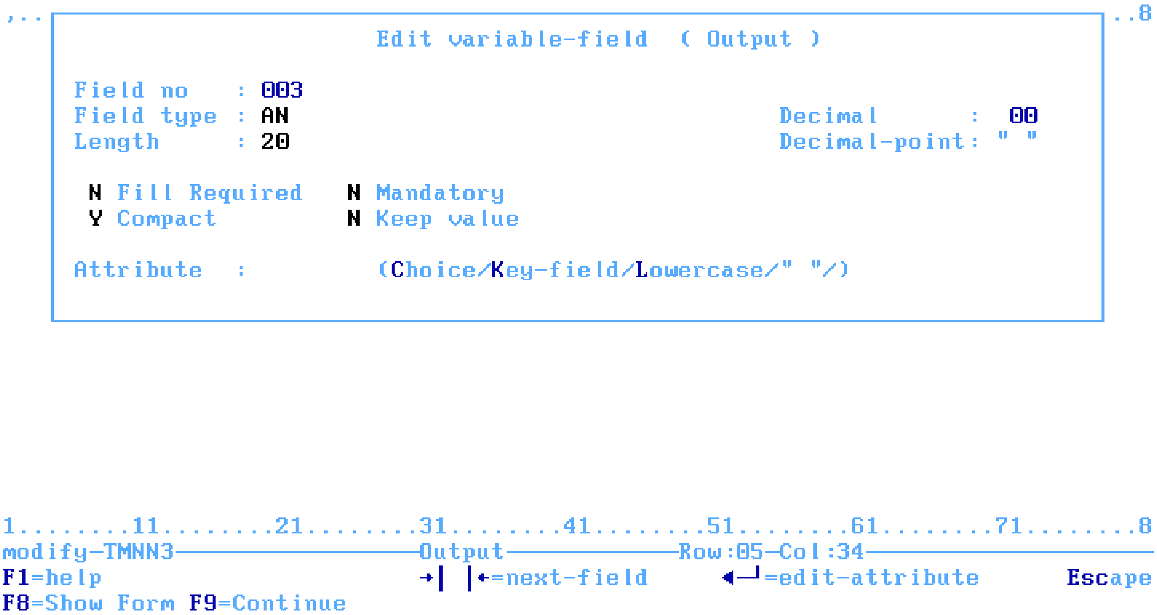


```
,..┌──────────────────────────────────────────────────────────────────┐..8
   │                     Edit text-field                              │
   │                                                                  │
   │                                              Length: 35          │
   │                                                                  │
   │ Text: TM00 - Workshop 3 - Student Inquiry                        │
   │                                                                  │
   └──────────────────────────────────────────────────────────────────┘

     Course no. Name                           MTWTF  Time  Room
      >>>>>>>>  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>> >>>>>  >>>>> >>>>









1.......11.......21.......31.......41.......51.......61.......71.......8
modify-TMNN3─────────────Text─────────────Row:01-Col:20─────────────────
F1=help                  →| |←=next-field    ◄┘=edit-attribute   Escape
F8=Show Form F9=Continue
```

## Input field

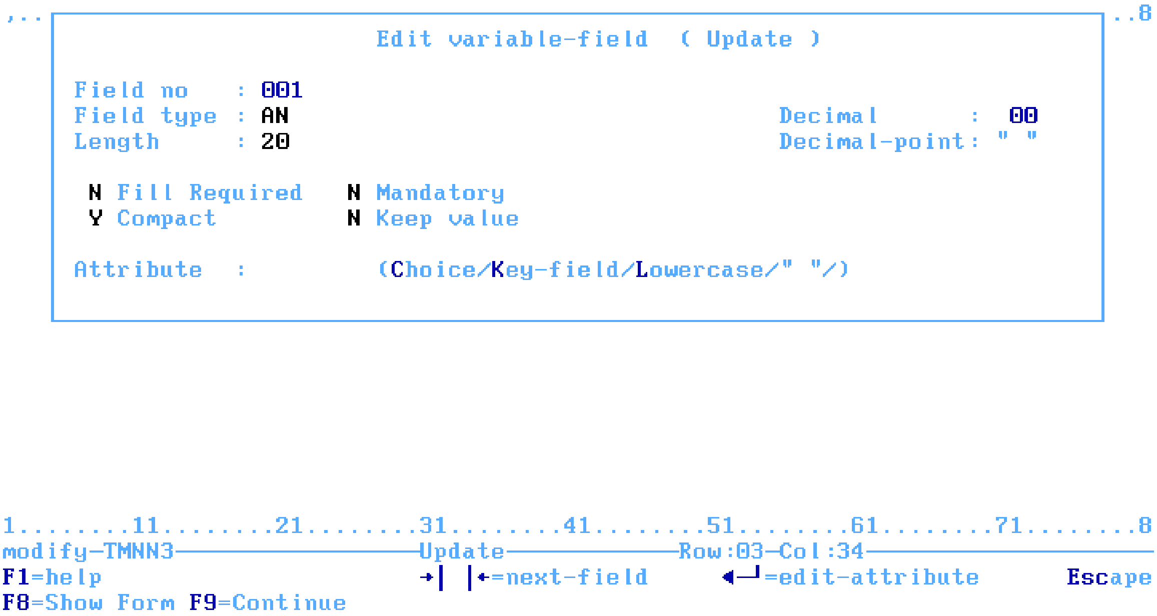
Input fields are treated like Update fields. If you paint or insert Input fields, these are automatically changed to Update fields.

## Output field



```
,..                                                                         ..8
      Edit variable-field  ( Output )

 Field no    : 003
 Field type  : AN                                  Decimal       :  00
 Length      : 20                                  Decimal-point: " "

  N Fill Required    N Mandatory
  Y Compact          N Keep value

 Attribute   :          (Choice/Key-field/Lowercase/" "/)


1........11........21........31........41........51........61........71........8
modify-TMNN3─────────────────Output────────────Row:05─Col:34─────────────────────
F1=help                       →| |←=next-field     ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Update field



```
,..                                                                         ..8
      Edit variable-field  ( Update )

 Field no    : 001
 Field type  : AN                                  Decimal       :  00
 Length      : 20                                  Decimal-point: " "

  N Fill Required    N Mandatory
  Y Compact          N Keep value

 Attribute   :          (Choice/Key-field/Lowercase/" "/)


1........11........21........31........41........51........61........71........8
modify-TMNN3─────────────────Update────────────Row:03─Col:34─────────────────────
F1=help                       →| |←=next-field     ◄┘=edit-attribute      Escape
F8=Show Form F9=Continue
```

## Repeat field

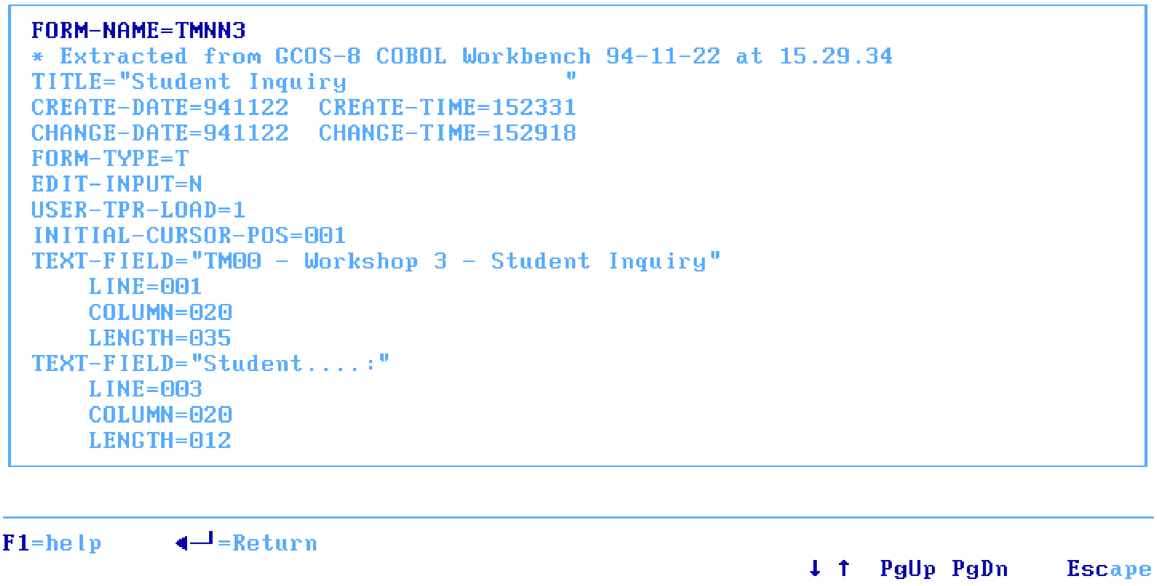
Repeat fields are not supported.

## Generation

Generate function is performed externally by SOLD software.

## Import/Export

Import/export may be used to exchange forms with SOLD software. The syntax used for exchange is specific for SOLD.



```
FORM-NAME=TMNN3
* Extracted from GCOS-8 COBOL Workbench 94-11-22 at 15.29.34
TITLE="Student Inquiry              "
CREATE-DATE=941122  CREATE-TIME=152331
CHANGE-DATE=941122  CHANGE-TIME=152918
FORM-TYPE=T
EDIT-INPUT=N
USER-TPR-LOAD=1
INITIAL-CURSOR-POS=001
TEXT-FIELD="TM00 - Workshop 3 - Student Inquiry"
    LINE=001
    COLUMN=020
    LENGTH=035
TEXT-FIELD="Student....:"
    LINE=003
    COLUMN=020
    LENGTH=012
```

## Syntax

| Syntax per Form | Default | Values |
|---|---|---|
| FORM-NAME=form-name | none | |
| TITLE="free text ... max 30 char. " | space | |
| CREATE-DATE=yymmdd CREATE-TIME=hhmmss | current | |
| CHANGE-DATE=yymmdd CHANGE-TIME=hhmmss | current | |
| FORM-TYPE=t | T | T(ransaction)<br>M(enu)<br>S(ubmask) |
| START-TPR=[tpr-name] | space | |
| SYNONYM=[name] | space | |
| EDIT-INPUT=yn | Y | Y(es)<br>N(o) |
| USER-TPR-LOAD=n | 1 | 1(=convention)<br>2(=direct-load)<br>3(=input depend) |
| INITIAL-CURSOR-POS=nnn | 000 (var field=0)<br>001 (var fields>0) | 000-999 |

| Syntax per Text field | Default | Values |
|---|---|---|
| TEXT-FIELD="text ..max 40 char." | none | |
| LINE=nnn | none | 1-999 |
| COLUMN=nnn | none | 1-999 |
| LENGTH=nnn | none | 1-999 |

| Syntax per Update field | Default | Values |
|---|---|---|
| UPDATE-FIELD=nnn | sequence starting from 1 | 1-999 |
| LINE=nnn | none | 1-999 |
| COLUMN=nnn | none | 1-999 |
| LENGTH=nnn | none | 1-999 |
| FIELD-TYPE=ft | AN | AN(=alphanumeric)<br>N(=numeric)<br>AL(=alphabetic)<br>AM(=amount)<br>D(=date) |
| DECIMALS=nn | 0 | 1-18 |
| DECIMAL-POINT="," | "," if numeric<br>space if not numeric | “,”<br>“.”<br>space |
| ATTRIBUTE=c | space | C(hoice)<br>K(ey-field)<br>L(owercase) |
| OUTPUT-ONLY | (no) | |
| FILL-REQUIRED-OFF | (yes) | |
| FILL-REQUIRED | (no) | |
| OPTIONAL | (yes) | |
| MANDATORY | (no) | |
| COMPACT-OFF | (no) | |
| COMPACT-ON | (yes) | |
| CLEAR-VALUE | (yes) | |
| KEEP-VALUE | (no) | |

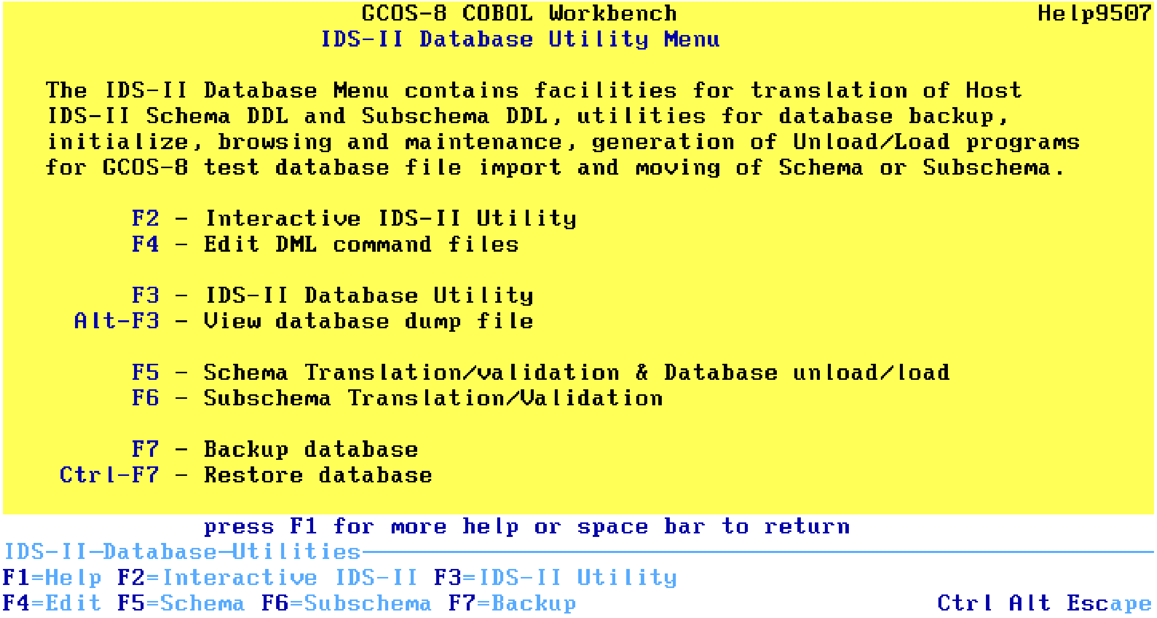
Comment lines are identified by “*” in column 1.

## TSM8

This feature is currently not documented.

# Chapter 5 IDS-II DATABASE MENU

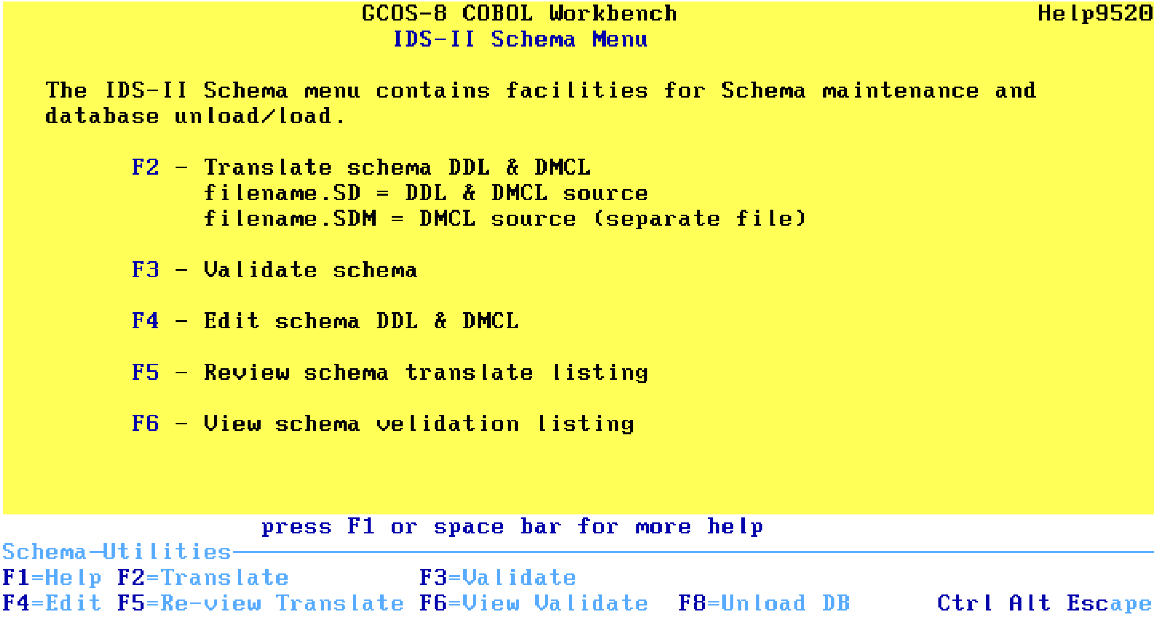
The IDS-II Database Menu is your place of work for all activities related to IDS-II database maintenance. G8WB provides a complete implementation of the GCOS-8 IDS-II Database system and includes utilities for maintenance of Schema/Subschema definition, database maintenance, database backup/restore and database unload/load.



```
                         GCOS-8 COBOL Workbench                         Help9507
                      IDS-II Database Utility Menu

  The IDS-II Database Menu contains facilities for translation of Host
  IDS-II Schema DDL and Subschema DDL, utilities for database backup,
  initialize, browsing and maintenance, generation of Unload/Load programs
  for GCOS-8 test database file import and moving of Schema or Subschema.

        F2 - Interactive IDS-II Utility
        F4 - Edit DML command files

        F3 - IDS-II Database Utility
    Alt-F3 - View database dump file

        F5 - Schema Translation/validation & Database unload/load
        F6 - Subschema Translation/Validation

        F7 - Backup database
   Ctrl-F7 - Restore database

              press F1 for more help or space bar to return
IDS-II-Database-Utilities------------------------------------------------------
F1=Help F2=Interactive IDS-II F3=IDS-II Utility
F4=Edit F5=Schema F6=Subschema F7=Backup                          Ctrl Alt Escape
```

## Schema

Schema's are supported through the standard IDS-II DDL and DMCL syntax and involves translation and validation activities similar to GCOS-8.



```
                         GCOS-8 COBOL Workbench                         Help9520
                           IDS-II Schema Menu

  The IDS-II Schema menu contains facilities for Schema maintenance and
  database unload/load.

        F2 - Translate schema DDL & DMCL
             filename.SD = DDL & DMCL source
             filename.SDM = DMCL source (separate file)

        F3 - Validate schema

        F4 - Edit schema DDL & DMCL

        F5 - Review schema translate listing

        F6 - View schema velidation listing


                  press F1 or space bar for more help
Schema-Utilities---------------------------------------------------------------
F1=Help F2=Translate          F3=Validate
F4=Edit F5=Re-view Translate F6=View Validate  F8=Unload DB       Ctrl Alt Escape
```

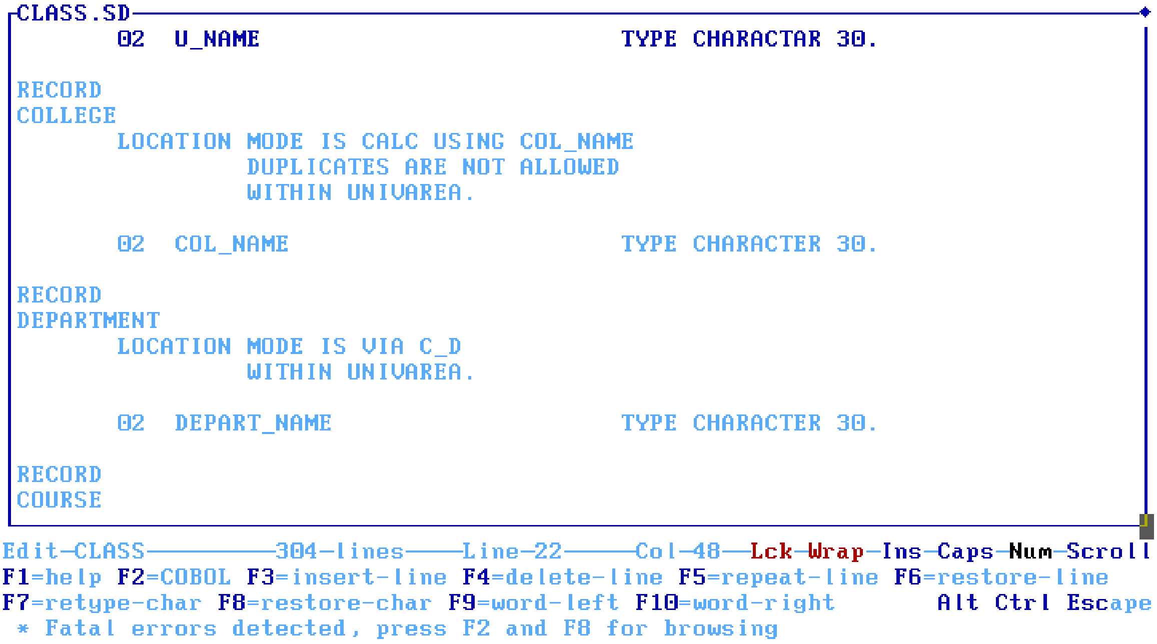
### Translation

You need to transfer your Schema DDL and DMCL syntax from GCOS-8 to get started. You may leave any DBACS directives (ignored), however any JCL must be removed. The DDL should be placed in a file named <schema>.SD. Use an abbreviated name if longer than 8 character, i.e. SCHEMA1.SD. If you keep DMCL in a separate file on GCOS-8, you should place this in a file named

<schema>.SDM. Schema Translator will look for this file if no DMCL is present in the <schema>.SD file.

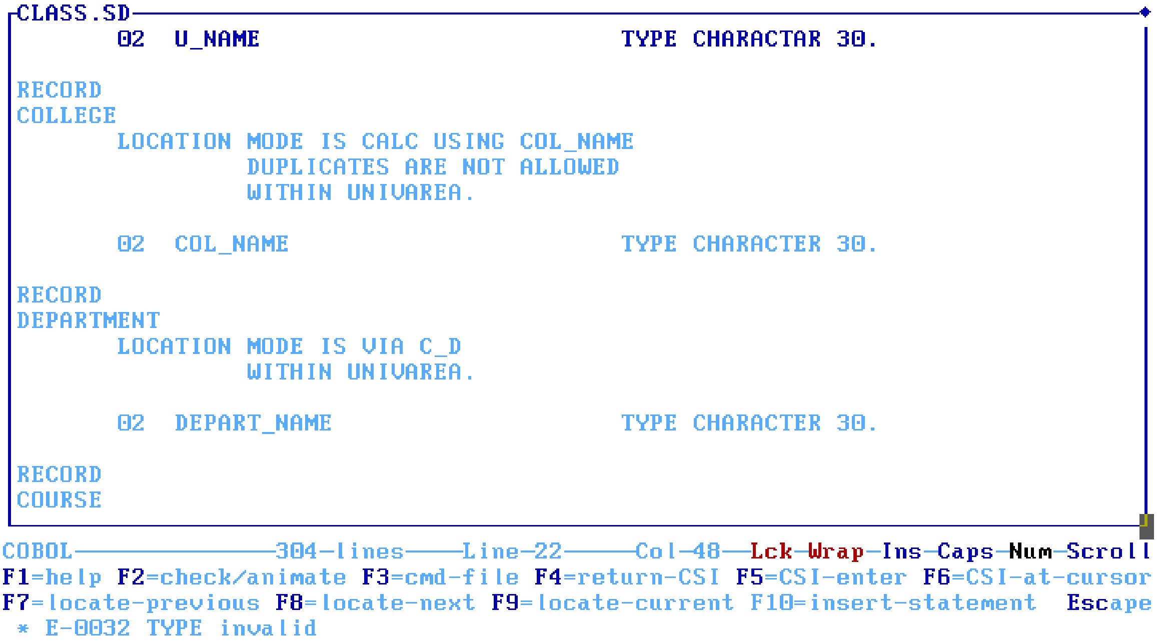
DMCL is optional, except for Database Procedure declaration syntax, however as G8WB will perform a complete syntax check of both DDL and DMCL, we recommeded you transfer both. This will allow you to implement changes and test these within G8WB, before transferring DDL and DMCL back to GCOS-8.

Schema Translator will use the MF Editor to present syntax errors, very much like checking a program. Following example illustrate a translate with syntax errors, which automatically enable the MF Editor.



```
CLASS.SD
       02  U_NAME                        TYPE CHARACTAR 30.

RECORD
COLLEGE
       LOCATION MODE IS CALC USING COL_NAME
                DUPLICATES ARE NOT ALLOWED
                WITHIN UNIVAREA.

       02  COL_NAME                      TYPE CHARACTER 30.

RECORD
DEPARTMENT
       LOCATION MODE IS VIA C_D
                WITHIN UNIVAREA.

       02  DEPART_NAME                   TYPE CHARACTER 30.

RECORD
COURSE
Edit-CLASS---------304-lines-----Line-22------Col-48--Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
 * Fatal errors detected, press F2 and F8 for browsing
```

When your select F2-COBOL and F8-Locate-next, Editor will show you the first syntax error and position to the corresponding location in the DDL source. You are able to make corrections as you go along, and the F7/F8/F9 function keys may be used to browse between the syntax errors found.



```
CLASS.SD
       02  U_NAME                        TYPE CHARACTAR 30.

RECORD
COLLEGE
       LOCATION MODE IS CALC USING COL_NAME
                DUPLICATES ARE NOT ALLOWED
                WITHIN UNIVAREA.

       02  COL_NAME                      TYPE CHARACTER 30.

RECORD
DEPARTMENT
       LOCATION MODE IS VIA C_D
                WITHIN UNIVAREA.

       02  DEPART_NAME                   TYPE CHARACTER 30.

RECORD
COURSE
COBOL--------------304-lines-----Line-22------Col-48--Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=check/animate F3=cmd-file F4=return-CSI F5=CSI-enter F6=CSI-at-cursor
F7=locate-previous F8=locate-next F9=locate-current F10=insert-statement  Escape
 * E-0032 TYPE invalid
```

Once you corrected all errors, you Escape from COBOL mode, and then resave the DDL syntax file. Another Escape will take you back to Schema menu, where you can re-invoke translation.

Translator will create a translate listing named <schema>.SL, which you may browse for syntax warnings or print a hardcopy. You will find a summary at the end of the listing. Use the prefix “* I-”, “* W-” or “* E-” to find information, warning and error message in the listing.

*Note: Schema translation does NOT disturb any current schema implementation. You may translate as many times as required and it is only when you validate the schema, that schema run-time files are changed.*

## Configuration

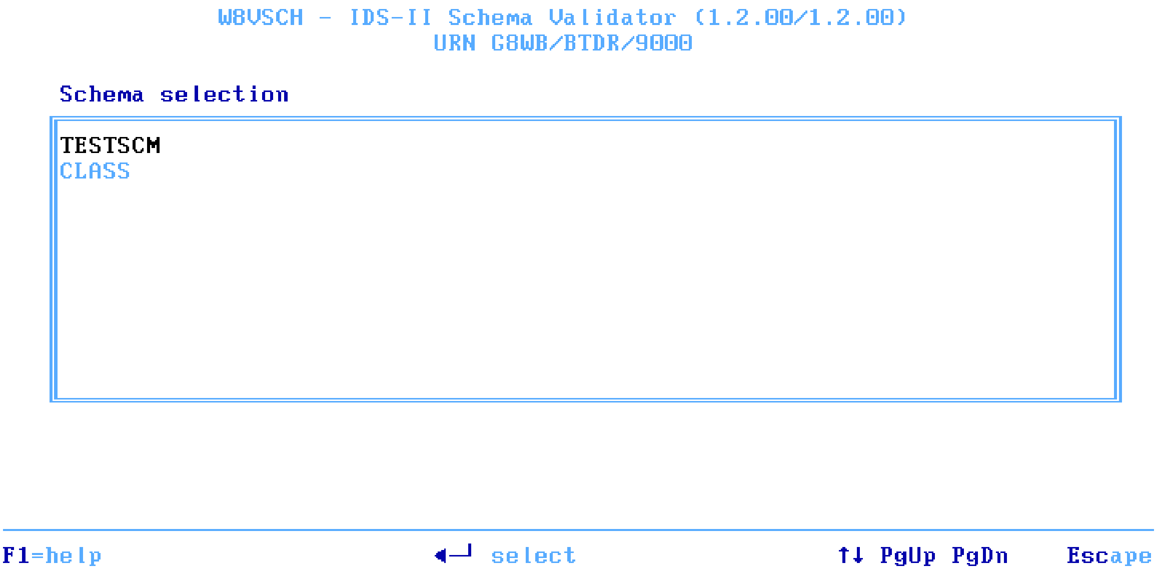
Schema Translator will perform an automatic configuration of the Schema when you perform the initial translation. The Schema will be assigned a unique internal number, which is used for identification of support files.

This configuration may also be done manually, before the initial translation. Ref. Configuration chapter for details.

*Note: Once a internal number is assigned, you may NOT change it in the configuration. Instead you have to delete the schema, which remove the configuration, then configure the schema manually with the number preferred, and then translate the schema again.*

## Validation

When you request schema validation, G8WB will present you with a list of configured schema’s. You use cursor keys to select the schema and return key to launch the validation. Escape will cancel the request and take you back to the Schema menu.



```
W8VSCH - IDS-II Schema Validator (1.2.00/1.2.00)
            URN G8WB/BTDR/9000

Schema selection

TESTSCM
CLASS

F1=help            ◄┘ select            ↑↓ PgUp PgDn    Escape
```

The validation activity is more of a generation activity, since translation will perform the complete syntax and logic checking of your schema. Validation will perform the following:

- Transfer the translated schema into the database directory. If the schema already exist, it is overridden and the schema version number is incremented. Initial schema version is numbered 1.
- Copy the DDL and DMCL syntax into the database directory, so you always have the source attached to your current schema version.

- Copy the translation list file into the database directory and amend the listing file during validation with information and statistics about the schema.
- Create the run-time support files, i.e. tables describing schema objects, which are loaded dynamically during use of a schema.
- Generate and compile a schema interface module, which is used to interface with database procedures, i.e. area selection, encode, decode, check.
- Generate a reference subschema, which is a complete image of the schema. The reference subschema is named SS-<schema-name>.
- Translate and validate the reference subschema.
- Generate Host Unload, Workbench Unload and Load programs, and compile the Workbench Unload and Load programs.

If you have a large schema, above activities may take quite some time. Information will be logged on the screen for each step of validation. Once complete you may view the validation listing, which contain both translation and validation listings, through the F6-View validation function.

Your schema is now ready for use and you need to initialize the database through the Database Utility. You may unload existing host test databases or you may build a new test database through various utilities. These activities are described further on in this chapter.

## Database Procedures

G8WB support database procedures written in either COBOL-74 or COBOL-85. Such procedures are called dynamically when applicable, while you are accessing the database through a subschema.

Otherwise G8WB treat database procedures like any other programs, and you thus need to import and check your database procedures, before you start to acces the database through a subschema. Ref. File-transfer and TP/Batch development chapters for details on how to import and check programs.

Since database procedures may be accessed either through application programs or G8WB database utilities, you must make sure these procedures are always accessible. You would thus normally choose a global (shared) directory for storage and include this directory as standard in your COBDIR setting.

As an example, database procedures could be placed under E:\DBMS\<schema>\PROC directory, and you would have a setting like:

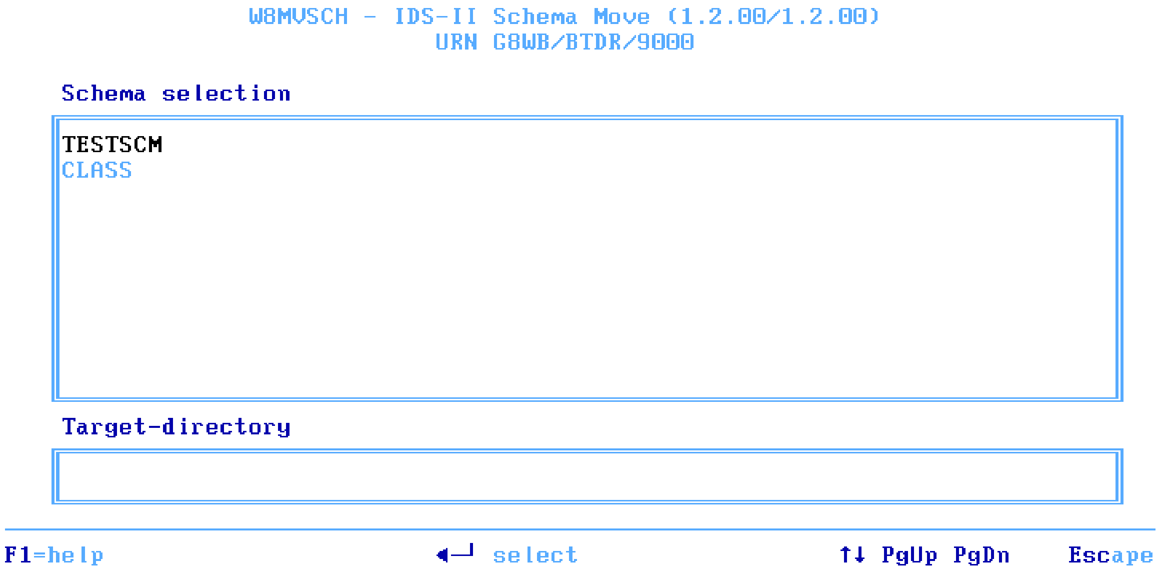
```
SET DBPROC1=E:\DBMS\<schema>\PROC
SET COBDIR=$DBPROC1;.....;C:\COBOL\LBR;C:\COBOL\EXEDLL
```

Yoy may keep database procedures in either INT (Checked) or GNT (Compiled) format, however, keep in mind that INT will make the database procedure visible during animation.

While you are testing database procedures, you probably want to use INT so you can animate the coding during test, while you would use GNT once you have completed test and move the procedure into the reference directory, i.e. $DBPROC1 in above example.

## Move

Schema configuration will alllow you to work with multiple versions, like on GCOS-8. Once you have established a new schema version, you may use the Schema Move Utility to move the schema and all related files into another environment.



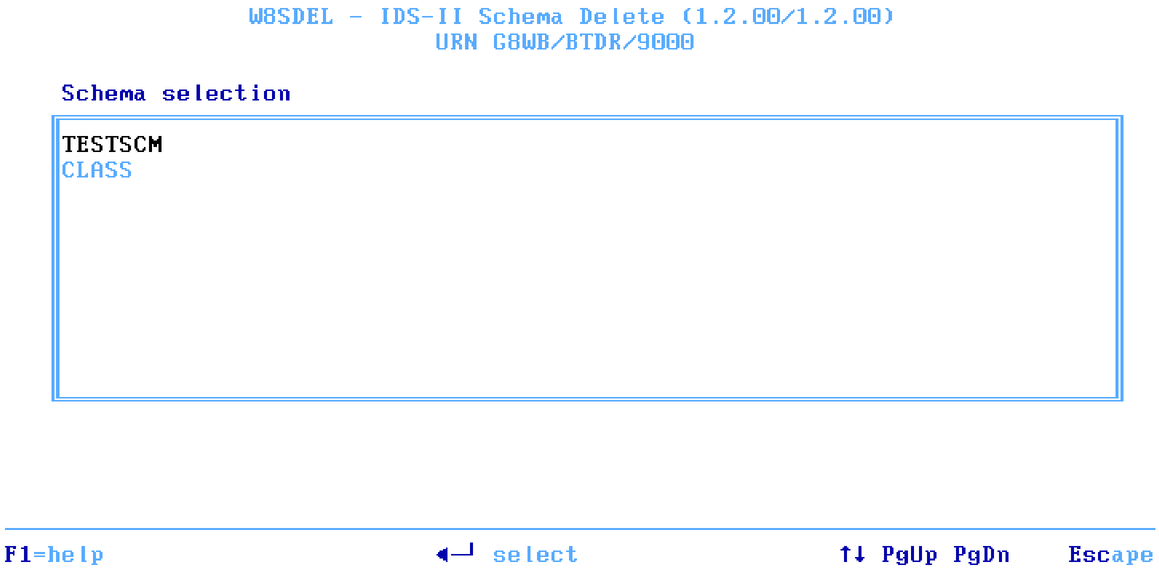
```
                W8MVSCH - IDS-II Schema Move (1.2.00/1.2.00)
                           URN G8WB/BTDR/9000

    Schema selection
    TESTSCM
    CLASS

    Target-directory

F1=help                      ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

Target-directory, define the directory where you want to move the schema to, i.e. some G8WBDDIR setting. The subdirectory SCHEMA will be appended, i.e. Target-directory\SCHEMA.

Ref. Configuration chapter for more details on IDS-II database configuration and environment organization.

## Delete

Schema Delete Utility may be used to remove a schema completely. This includes schema and all related files, except your original DDL and DMCL source, the generated reference subschema, the database files and any related backup-sets, and finally de-configuration.



```
                W8SDEL - IDS-II Schema Delete (1.2.00/1.2.00)
                           URN G8WB/BTDR/9000

    Schema selection
    TESTSCM
    CLASS

F1=help                      ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

## Field-Redefinition

As long as your schema view represent the data fields directly, i.e. each field is described in its basic format, schema/subschema transformation will adjust differences between the physical schema representation and the subschema COBOL representation. The data formats Character and Unspecified may, however, describe a group of fields, which are later redefined in the subschema view, or even in Working-Storage of the accessing application program.

The implementation of non DISPLAY format fields is different between GCOS-8 and G8WB, and the physical size may thus be different. When group fields are used, truncation may occur if the G8WB implementation is longer than the schema group field. Following example illustartion the potential truncation problem:

Schema view:

```
02 Field-a TYPE CHARACTER 2.
```

Subschema view:

```
02 Field-a.
   03 Field-a-1 PIC 9(3) COMP-4.
   03 Field-a-2 PIC 9(1) COMP-4.
```

G8WB implementation does not support half-bytes, thus the subschema view will have a physical length of three bytes. Truncation will occur during move to and from the schema, and as a result Field-a-2 is lost. During subschema validation a truncation warning will be given, and you should check for such warnings to ensure such truncations are identified.

You may solve this truncation problem by changing the schema DDL, i.e CHARACTER 3 instead of 2. Such change would, however, make the DDL incompatible with GCOS-8. Instead you may use the G8WB Increment directive, which is implemented as a comment to make it compatible with GCOS-8. The increment directive must be placed immediately before the field definition, where the dynamic increment is to be applied.

Schema view:

```
COMMENT”+1”
   02    Field-a TYPE CHARACTER 2.
```

This dynamic change from CHARACTER 2 to 3 will be documented with an information message during schema translation.

You need only fix truncation problems where actual data is lost, i.e. sum of actual fields in subschema view is smaller than the size allocation in the schema view. The subschema view can in some instances be shorter, or you may have a FILLER in the subschema view that accomodate the size difference (assuming FILLER is not used).

If the subschema view describe group fields, which are redefined in Working-Storage of the application program, you will not get any warning during subschema validation. Dynamic change of size needs to be done both for schema and subschema DDL, ref. subschema redefinition for details.

Subschema view:

```
02 Field-a PIC X(2).
```

Working-Storage view:

```
02 Field-a.
   03 Field-a-1 PIC 9(3) COMP-4.
   03 Field-a-2 PIC 9(1) COMP-4.
```

Following tables illustrate the size differences between GCOS-8 and G8WB implementation.

| Data type | PIC 9(n) | GCOS-8 #Bytes | G8WB #Bytes |
|---|---|---|---|
| COMP, COMP-3 & 5 | 1 | 1 | 1 |
| | 2 | 2 | 1 |
| | 3 | 2 | 2 |
| | 4 | 2 | 2 |
| | 5 | 3 | 3 |
| | 6 | 3 | 3 |
| | 7 | 4 | 3 |
| | 8 | 4 | 4 |
| | 9 | 5 | 4 |
| | 10 | 5 | 5 |
| | 11 | 6 | 5 |
| | 12 | 6 | 5 |
| | 13 | 7 | 6 |
| | 14 | 7 | 6 |
| | 15 | 8 | 7 |
| | 16 | 8 | 7 |
| | 17 | 9 | 8 |
| | 18 | 9 | 8 |
| COMP, COMP-3 & 5 | S1 | 1 | 1 |
| | S2 | 2 | 1 |
| | S3 | 2 | 2 |
| | S4 | 3 | 2 |
| | S5 | 3 | 3 |
| | S6 | 4 | 3 |
| | S7 | 4 | 4 |
| | S8 | 5 | 4 |
| | S9 | 5 | 4 |
| | S10 | 6 | 5 |
| | S11 | 6 | 5 |
| | S12 | 7 | 6 |
| | S13 | 7 | 6 |
| | S14 | 8 | 6 |
| | S15 | 8 | 7 |
| | S16 | 9 | 7 |
| | S17 | 9 | 8 |
| | S18 | 10 | 8 |

*Note: G8WB does not support PIC > S9(18).*

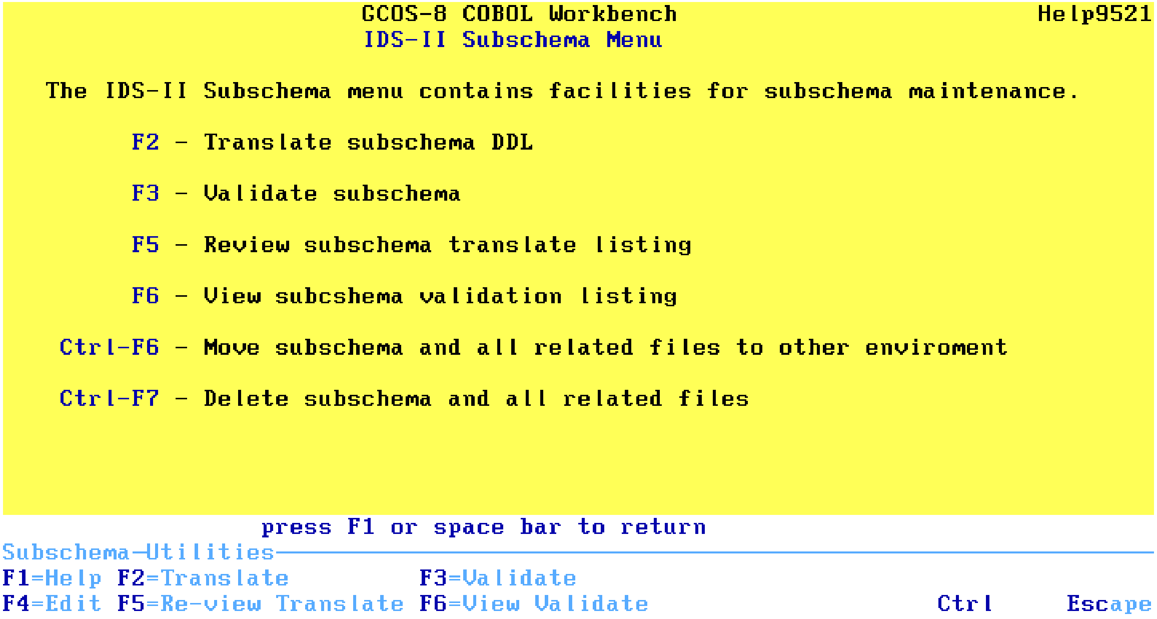
| Data type | PIC 9(n) | GCOS-8 #Bytes | G8WB #Bytes |
|---|---|---|---|
| COMP-4 | 1 | ½ | 1 |
| | 2 | 1 | 1 |
| | 3 | 1½ | 2 |
| | 4 | 2 | 2 |
| | 5 | 2½ | 3 |
| | 6 | 3 | 3 |
| | 7 | 3½ | 3 |
| | 8 | 4 | 4 |
| | 9 | 4½ | 4 |
| | 10 | 5 | 5 |
| | 11 | 5½ | 5 |
| | 12 | 6 | 5 |
| | 13 | 6½ | 6 |
| | 14 | 7 | 6 |
| | 15 | 7½ | 7 |
| | 16 | 8 | 7 |
| | 17 | 8½ | 8 |
| | 18 | 9 | 8 |
| COMP-4 | S1 | 1 | 1 |
| | S2 | 1½ | 1 |
| | S3 | 2 | 2 |
| | S4 | 2½ | 2 |
| | S5 | 3 | 3 |
| | S6 | 3½ | 3 |
| | S7 | 4 | 4 |
| | S8 | 4½ | 4 |
| | S9 | 5 | 4 |
| | S10 | 5½ | 5 |
| | S11 | 6 | 5 |
| | S12 | 6½ | 6 |
| | S13 | 7 | 6 |
| | S14 | 7½ | 6 |
| | S15 | 8 | 7 |
| | S16 | 8½ | 7 |
| | S17 | 9 | 8 |
| | S18 | 9½ | 8 |

| Data type | PIC 9(n) | GCOS-8 #Bytes | G8WB #Bytes |
|---|---|---|---|
| COMP-1 | n/a | 2 | 2 |
| COMP-2 | n/a | 4 | 4 |
| COMP-6 | n/a | 4 | 4 |
| COMP-6 36-bit | n/a | 4 | 5 |
| DB-KEY | n/a | 4 | 4 |
| DB-KEY 36-bit | n/a | 4 | 5 |

*Note: G8WB does not enforce word alignment for COMP-6.*

# Subschema

Subschema's are supported through the standard IDS-II DDL syntax and involves translation and validation activities similar to GCOS-8.

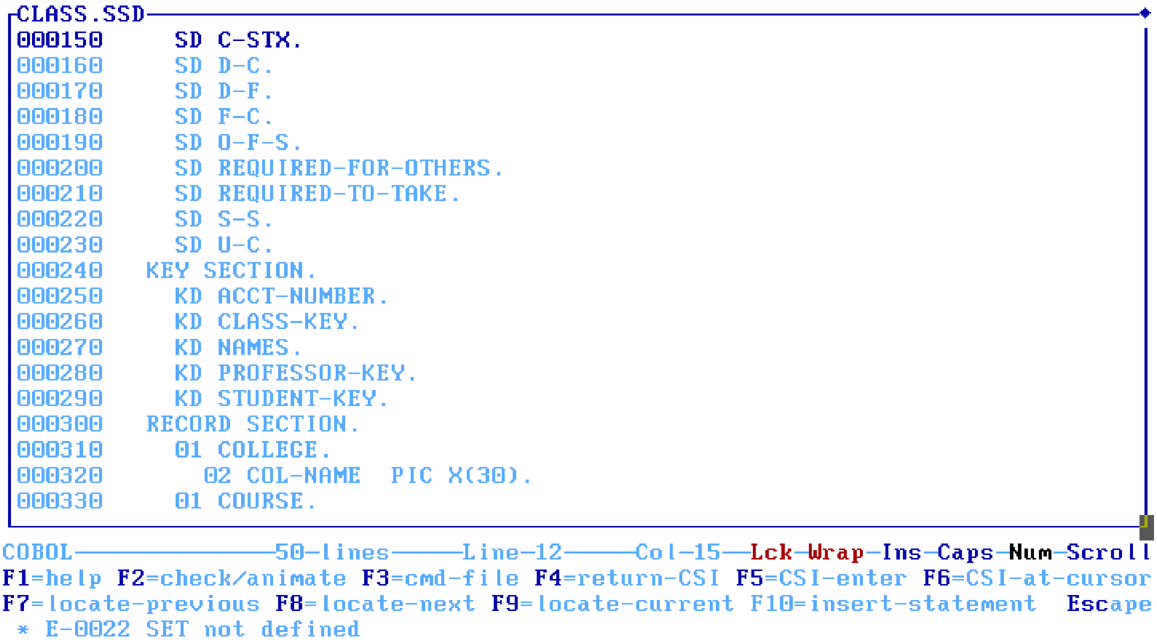


```
                              GCOS-8 COBOL Workbench                      Help9521
                              IDS-II Subschema Menu

   The IDS-II Subschema menu contains facilities for subschema maintenance.

         F2 - Translate subschema DDL

         F3 - Validate subschema

         F5 - Review subschema translate listing

         F6 - View subcshema validation listing

    Ctrl-F6 - Move subschema and all related files to other enviroment

    Ctrl-F7 - Delete subschema and all related files



                  press F1 or space bar to return
Subschema-Utilities-----------------------------------------------------------
F1=Help F2=Translate          F3=Validate
F4=Edit F5=Re-view Translate  F6=View Validate                   Ctrl     Escape
```

## Translation

You need to transfer your Subschema DDL syntax from GCOS-8 to get started. You may leave any DBACS directives (ignored), however any JCL must be removed. The DDL should be placed in a file named <subschema>.SSD. Use an abbreviated name if longer than 8 character, i.e. SUBSCH1.SSD.

Schema Translator will use the MF Editor the present syntax errors, similar to Schema Translator as described above. Following example illustrate a translate with syntax errors, which automatically enable the MF Editor.



```
CLASS.SSD---------------------------------------------------------------------
000150     SD C-STX.
000160     SD D-C.
000170     SD D-F.
000180     SD F-C.
000190     SD O-F-S.
000200     SD REQUIRED-FOR-OTHERS.
000210     SD REQUIRED-TO-TAKE.
000220     SD S-S.
000230     SD U-C.
000240   KEY SECTION.
000250     KD ACCT-NUMBER.
000260     KD CLASS-KEY.
000270     KD NAMES.
000280     KD PROFESSOR-KEY.
000290     KD STUDENT-KEY.
000300   RECORD SECTION.
000310     01 COLLEGE.
000320       02 COL-NAME  PIC X(30).
000330     01 COURSE.
COBOL-------------50-lines-----Line-12-----Col-15--Lck-Wrap-Ins-Caps-Num-Scroll
F1=help F2=check/animate F3=cmd-file F4=return-CSI F5=CSI-enter F6=CSI-at-cursor
F7=locate-previous F8=locate-next F9=locate-current F10=insert-statement  Escape
 * E-0022 SET not defined
```

Once you corrected all errors, you Escape from COBOL mode, and then resave the DDL syntax file. Another Escape will take you back to Subschema menu, where you can re-invoke translation.

Translator will create a translate listing named <subschema>.SSL, which you may browse for syntax warnings or print a hardcopy. You will find a summary at the end of the listing. Use the prefix “* I-”, “* W-” or “* E-” to find information, warning and error message in the listing.

Translator will generate the subschema view is a COPY, which is included when you check application programs.

## Configuration

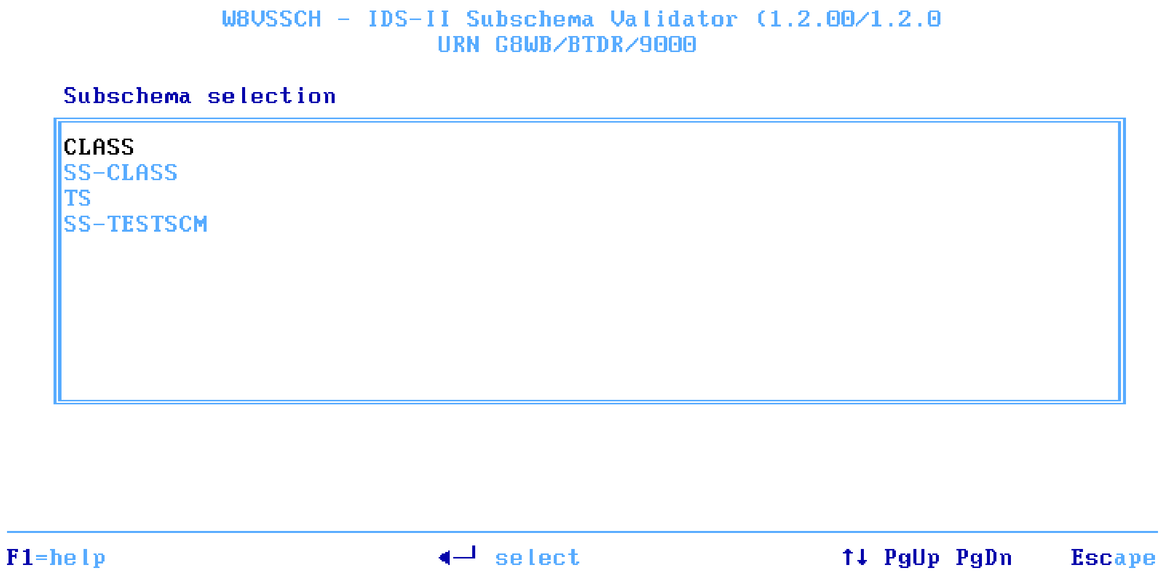
Subschema Translator will perform an automatic configuration of the Subschema when you perform the initial translation. The Subchema will be assigned a unique short name (6 character), which is used for identification of support files.

This configuration may also be done manually, before the initial translation. Automatic configuration will use the first 6 characters of the subschema name as short name. If this is not unique, translator will fail to configure the subschema and you need to do it manually, and re-translate. Ref. Configuration chapter for details on subschema configuration.

*Note: Once a short name is assigned, you may NOT change it in the configuration. Instead you have to delete the subschema, which remove the configuration, then configure the subschema manually with the number preferred, and then translate the subschema again.*

## Validation

When you request subschema validation, G8WB will present you with a list of configured subschema’s. You use cursor keys to select the subschema and return key to launch the validation. Escape will cancel the request and take you back to the Subschema menu.



```
W8VSSCH - IDS-II Subschema Validator (1.2.00/1.2.0
URN G8WB/BTDR/9000

Subschema selection

CLASS
SS-CLASS
TS
SS-TESTSCM
```

The validation activity is both a validation and a generation activity. You should consult the validation listing and verify for any warnings related to validation. You need to perform subschema validation either because the subschema has been translated, or because the schema has been re-translated and validated.

Validation after translation of subschema will perform the following:

- Transfer the translated subschema to the database directory. If the subschema already exist, it is overridden.

- Copy the DDL syntax into the database directory, so you always have the source attached to your current subschema version.

- Copy the translation list file into the database directory and amend the listing file during validation with information, warnings and statistics about the subschema.

- Warnings will be given for any Record restrictions, which prevent STORE or MODIFY of a record. These restrictions relate to incomplete inclusion of Realms, Sets, Keys or fields. Other types of restrictions are not documented, i.e. FIND WITHIN <set>, without <set> being included in subschema, however, all restrictions will be enforced by G8WB Preprocessor and Interactive IDS-II Utility.

Validation without translation of subschema will perform the following:

- Verify the existing subschema against the new schema version.

- Amend the listing file during validation with information, warnings and statistics about the subschema.

- Warnings will be given for entities, i.e. records, field etc., which no longer exist in the schema. Such entities are removed during re-validation.

- Warnings will be given for any Record restrictions, which prevent STORE or MODIFY of a record. These restrictions relate to incomplete inclusion of Realms, Sets, Keys or fields. Other types of restrictions are not documented, i.e. FIND WITHIN <set>, without <set> being included, however, all restrictions will be enforced by G8WB Preprocessor and Interactive IDS-II Utility.

In addition to either of above, validation will perform the following:

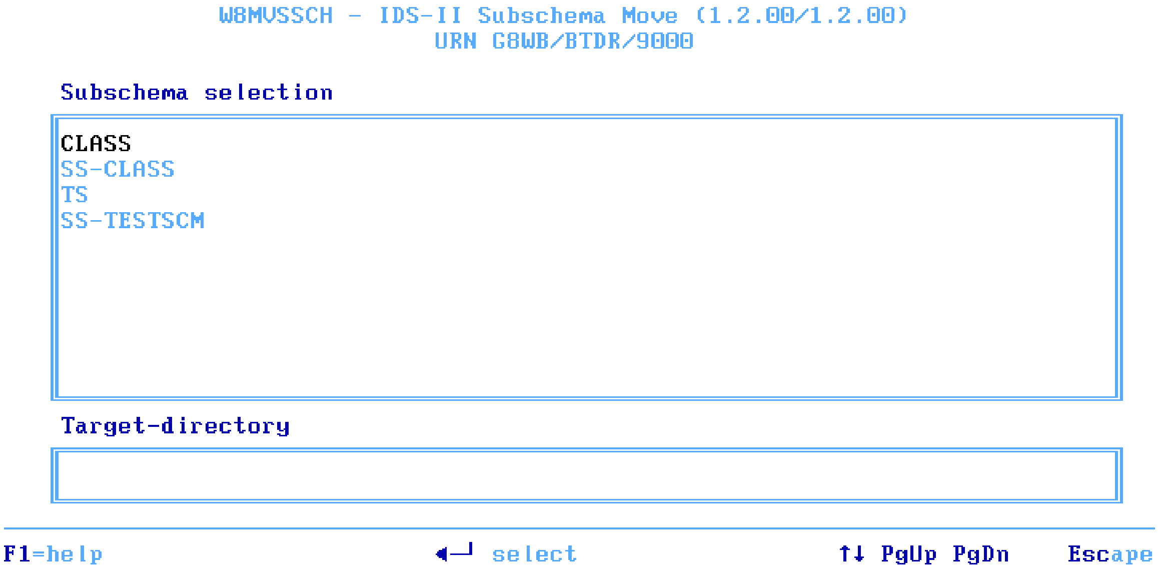
- Create the run-time support files, i.e. tables describing subschema objects, which are loaded dynamically during use of a subschema.

- Generate and compile a subschema domain module, which is used to provide a schema and subschema domain for the application, i.e. tables etc. are loaded only once into the shared domain.

Your subschema is now ready for use.

## Move

Subchema configuration will alllow you to work with multiple versions, like on GCOS-8. Once you have established a new subschema version, you may use the Subschema Move Utility to move the subschema and all related files into another environment.



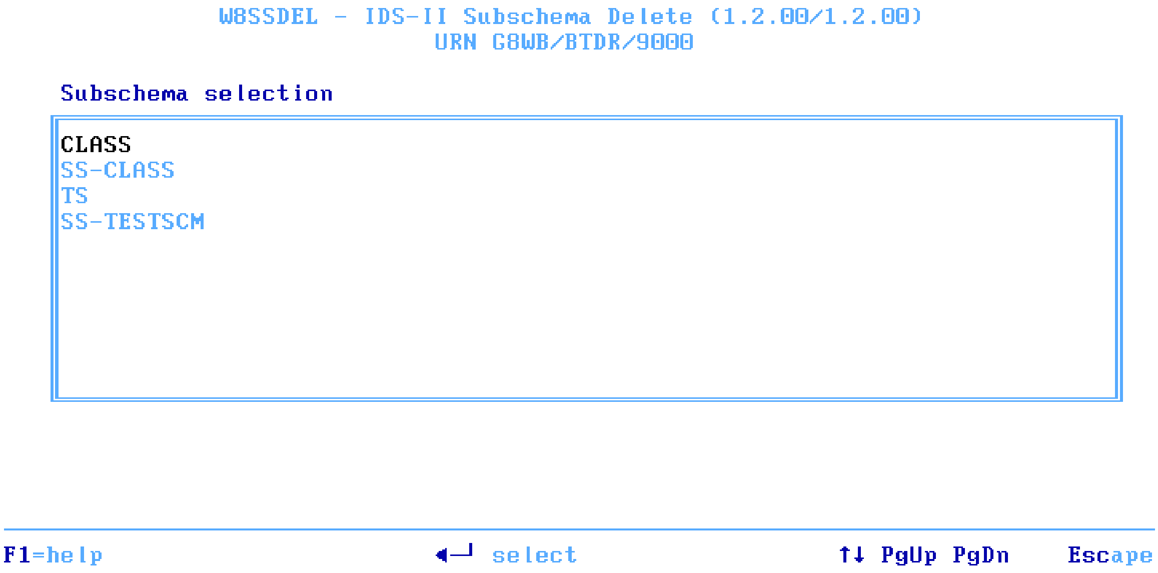
```
          W8MVSSCH - IDS-II Subschema Move (1.2.00/1.2.00)
                         URN G8WB/BTDR/9000

    Subschema selection
    CLASS
    SS-CLASS
    TS
    SS-TESTSCM

    Target-directory

F1=help                   ◄┘ select              ↑↓ PgUp PgDn    Escape
```

Target-directory, define the directory where you want to move the subschema to, i.e. some G8WBDDIR setting. The subdirectory SSCHEMA will be appended, i.e. Target-direct\SSCHEMA.

Ref. Configuration chapter for more details on IDS-II database configuration and environment organization.

## Delete

Subschema Delete Utility may be used to remove a subschema completely. This includes subschema and all related files, except your original DDL source, and de-configuration.



```
          W8SSDEL - IDS-II Subschema Delete (1.2.00/1.2.00)
                         URN G8WB/BTDR/9000

    Subschema selection
    CLASS
    SS-CLASS
    TS
    SS-TESTSCM

F1=help                   ◄┘ select              ↑↓ PgUp PgDn    Escape
```

## Field-Redefinition

Under schema redefinition described above, we showed an example where the schema view need dynamic size adjustment due to the difference in implementation of non DISPLAY format fields between GCOS-8 and G8WB. Similar adjustment may be required for a subschema, when group fields are used.

Schema view:

```
COMMENT"+1"
   02   Field-a TYPE CHARACTER 2.
```

Subschema view:

```
02 Field-a PIC X(2).
```

Working-Storage view:

```
02 Field-a.
   03 Field-a-1 PIC 9(3) COMP-4.
   03 Field-a-2 PIC 9(1) COMP-4.
```

You may solve this truncation problem by changing the subschema DDL, i.e PIC X(3) instead of PIC X(2). Such change would, however, make the DDL incompatible with GCOS-8. Instead you may use a G8WB preprocessor Increment directive, which is implemented as a comment to make it compatible with GCOS-8. The increment directive must be placed immediately before the field definition, where the dynamic increment is to be applied.

Subschema view:

```
*W8PREP G8WB+1
   02   Field-a PIC X(2).
```

This dynamic change from PIC X(2) to PIC X(3) will be documented with an information message during subschema translation.

You need only fix truncation problems where actual data is lost, i.e. sum of actual fields in Working-Storage view is smaller than the size allocation in the subschema view. The Working-Storage view can in some instances be shorter, or you may have a FILLER in the subschema view that accomodate the size difference (assuming FILLER is not used).

Subschema DDL is only translated during subschema translation, thus when you check a program which use a subschema, a translated (pre-processed) subschema view is included. Working-Storage views are however pre-processed during each program checking. Adjustment of subschema view must thus take place during subschema translation, which support some of the G8WB Preprocessor directives.

## COMP-6

COMP-36-BIT-SUPPORT setting, within Database configuration, defines how COMP-6 fields should be translated for a subschema. This setting apply to all subschema's, however you may also specify setting in the DDL source via SET directives.

Subschema translator, by default, treat COMP-6 fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate translation mode, i.e. *W8PREP ALT active in DDL source, COMP-6 fields are implemented as 32 bit unsigned binary fields.

ON setting may be used to change alternate translation, so COMP-6 fields are implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physical size of a COMP-6 field will however change from 4 to 5 bytes.

FORCE setting may be used to change default translation, so COMP-6 fields are implemented as 39 bit signed binary fields. In alternate translation, COMP-6 fields are implemented as 31 bit signed binary fields.

*Note: You should only enable 36-bit-support if you really need it, and you must pay attention to potential problems with redefines in Working storage of programs, due to the change in physical field size.*

## DB-KEY

DBK-36-BIT-SUPPORT setting, within Database configuration, defines how DB-KEY fields should be translated for a subschema. This setting apply to all subschema's, however you may also specify setting in the DDL source via SET directives.

Subschema translator, by default, treat DB-KEY fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate translation, i.e. *W8PREP ALT active in DDL source, DB-KEY fields are implemented likewise.

FORCE setting may be used to change translation, so DB-KEY fields are by default implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physizal size of a DB-KEY field will howver change from 4 to 5 bytes. In alternate translation, DB-KEY fields are implemented as 31 bit signed binary fields.

*Note: 36-bit-support is ONLY relevant, although this is not very likely, if you unload from host databases with VERY large DBK values, because original DBK's are re-used within G8WB. New DBK's are assigned as highest DBK + 1. A locally created database will thus start with DBK = 1.*

## COMP-3, -4 & -5

G8WB has a potential fundamental problem with COMP-3, -4 and COMP-5 (referred to as COMP-X), besides variances in physical size as described above.

GCOS-8 will generally allow redefinition of COMP-X fields. Redefinition of signed fields are seldom, because you need to consider GCOS-8' physical implementation of packed-decimal format, however redefinition of unsigned COMP-4 and even length COMP-3 and -5, are more often seen.

G8WB ONLY support redefinition of numeric DISPLAY fields. Although this limitation is quite fundamental, it is not nessecarily a problem for your application. G8WB provides various automated and manual solutions to overcome such redefinition problems, ref. COMP anlyzer in Configuration chapter for details.

In relation to subschema, you may need to change COMP-X in the subschema view due to redefinitions which occur in your application. This change may be implemented dynamically without impacting GCOS-8 compatibility for the subschema DDL.

Where applicable, you need to include G8WB Preprocessor directives that trigger subschema translator to work in Alternate or Normal translation mode. Following example illustrate a dynamic change to numeric DISPLAY format:

Subschema view:

```
*W8PREP ALT
 02  Field-a.
     03 Field-a-1 PIC 9(3) COMP-4.
     03 Field-a-2 PIC 9(1) COMP-4.
*W8PREP ON
 02  Field-b PIC 9(4) COMP-4.
```

Field-a-1 and Field-a-2 are dynamically changed to numeric DISPLAY, whereas Field-b is left in COMP-4 format.

You will notice that dynamic usage change may affect the size of group fields in the schema view. The schema size adjustment example used above, would thus change as follows:

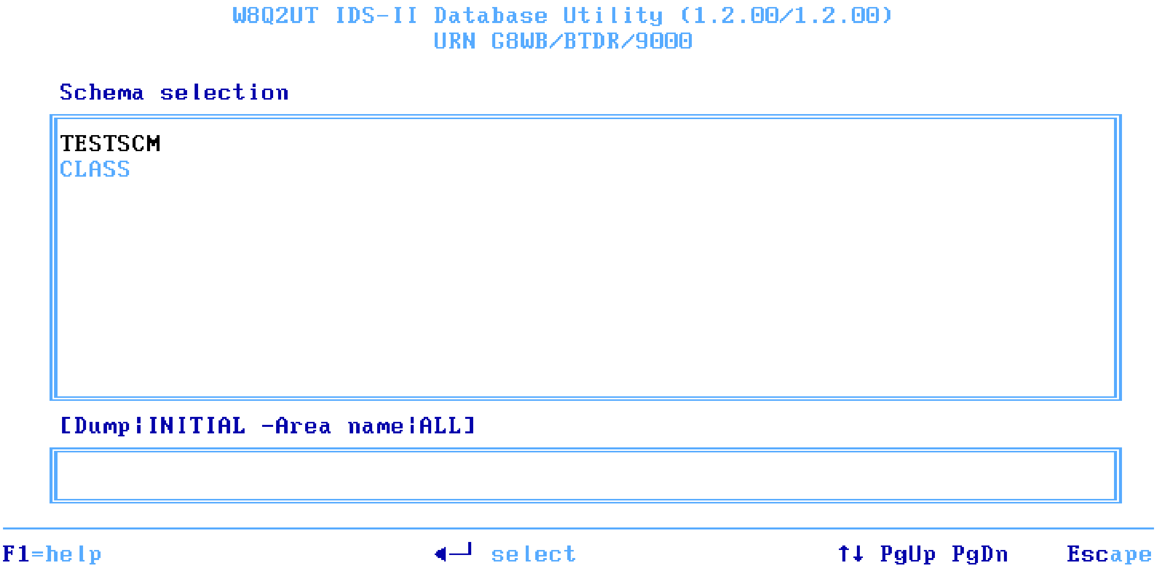
Schema view:

**COMMENT”+2”**
**02 Field-a TYPE CHARACTER 2.**

# IDS-II Database Utility

The IDS-II Database Utility (Q2UT) is quite similar to the Q2UT utility available on GCOS-8. Q2UT is primarily used to initialize the database, however it also provides features that allow you to analyze and repair the physical database. It should be noted that the G8WB database implementation is very much different from IDS-II on GCOS-8.

You select the Schema, and thereby database that you wish to work with, from the list of configured schema’s.



```
W8Q2UT IDS-II Database Utility (1.2.00/1.2.00)
              URN G8WB/BTDR/9000

Schema selection
TESTSCM
CLASS

[Dump|INITIAL -Area name|ALL]
```

The D(ump) and INITIAL command may be passed directly, otherwise Q2UT will enter a prompt mode until you exit.

## Commands

Following commands are available:

INITIAL

Initialize database Area(s)

P[RINT]

Display one or more Area's

D[UMP]

Dump one or more Area's to file

V[ERIFY]

Verify database integrity

R[EBUILD]

Rebuild indexes from data files (if corrupted)

PATCH

Not for general use

DONE (Quit)

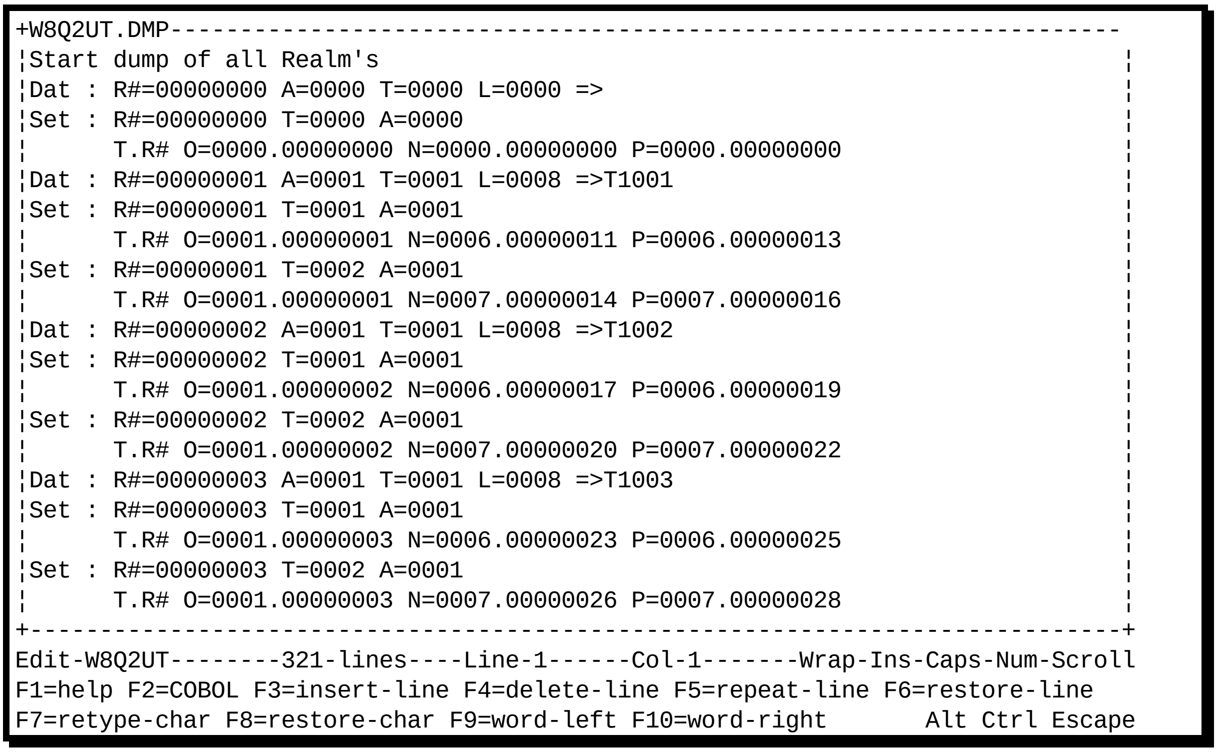
## Initial

May be used to Initialize a specific area or all areas of the database. A specific area is identified by the schema area name, i.e. SCHED_AREA.

*Note: Areas are implemented as a logical entities, as G8WB implement all areas into one physical file. A specific area initialize will thus remove all records pertaining to the logical area.*

## Dump

May be used to dump information ,for a specific area or all areas of the database, to a file. This file is named W8Q2UT.DMP and is placed in your current path. MF Editor is automatically invoked when dumping is complete, and you may thus browse through the dump file.



```
+W8Q2UT.DMP--------------------------------------------------------------------
¦Start dump of all Realm's                                                     ¦
¦Dat : R#=00000000 A=0000 T=0000 L=0000 =>                                     ¦
¦Set : R#=00000000 T=0000 A=0000                                               ¦
¦      T.R# O=0000.00000000 N=0000.00000000 P=0000.00000000                    ¦
¦Dat : R#=00000001 A=0001 T=0001 L=0008 =>T1001                                ¦
¦Set : R#=00000001 T=0001 A=0001                                               ¦
¦      T.R# O=0001.00000001 N=0006.00000011 P=0006.00000013                    ¦
¦Set : R#=00000001 T=0002 A=0001                                               ¦
¦      T.R# O=0001.00000001 N=0007.00000014 P=0007.00000016                    ¦
¦Dat : R#=00000002 A=0001 T=0001 L=0008 =>T1002                                ¦
¦Set : R#=00000002 T=0001 A=0001                                               ¦
¦      T.R# O=0001.00000002 N=0006.00000017 P=0006.00000019                    ¦
¦Set : R#=00000002 T=0002 A=0001                                               ¦
¦      T.R# O=0001.00000002 N=0007.00000020 P=0007.00000022                    ¦
¦Dat : R#=00000003 A=0001 T=0001 L=0008 =>T1003                                ¦
¦Set : R#=00000003 T=0001 A=0001                                               ¦
¦      T.R# O=0001.00000003 N=0006.00000023 P=0006.00000025                    ¦
¦Set : R#=00000003 T=0002 A=0001                                               ¦
¦      T.R# O=0001.00000003 N=0007.00000026 P=0007.00000028                    ¦
+------------------------------------------------------------------------------+
Edit-W8Q2UT--------321-lines----Line-1------Col-1-------Wrap-Ins-Caps-Num-Scroll
F1=help F2=COBOL F3=insert-line F4=delete-line F5=repeat-line F6=restore-line
F7=retype-char F8=restore-char F9=word-left F10=word-right       Alt Ctrl Escape
```

The DUMP information show the four records types used in the G8WB implementation:

- Data record
- Set record (occurence)
- Key record (index key)
- Calc record (calc key)

The data content of a record is dumped in raw format only.

R# (DBK), A#(Area type) and T#(Record/Set/Key/Calc type) may be found in the Schema validate listing.

For Set records, each O(wner), N(ext) and P(rior) pointer contains a T# and R#.

You may use the View Database Dump function to re-view the dump file at a later stage, and the Delete Database Dump to delete the temporary dump file, once you have completed browsing.

### Print

May be used to dump information to the screen, otherwise it's function is equivalent to the Dump function, described above.

### Rebuild

If the database index file(s) are somehow corrupted, DBMS will abort with a I/O status 9/041 with an indication of the file involved, i.e. Data, Set, Calc or Key. This may happen if the workstation is booted while the database is opened for update.

You may use the Rebuild function to rebuild the index for a specific file or all files, and this will bring your database back in shape. You may naturally also use the Database Restore Utility, to re-establish your database.

### Verify

If you suspect your database is corrupted or you are told so by DBMS, you may use the Verify function to pinpoint the current problem. Verify will cross-check the references (via DBK) between the Set, Calc, Key and Data file and document any discrepencies. Verify will automatically correct some of the discrepencies, while others are only documented. The results from verification is presented via MF Editor.

In general it is recommended that you restore the database if it becomes corrupted.

*Note: Verify does not do any set walking and will NOT detect set chaining errors.*

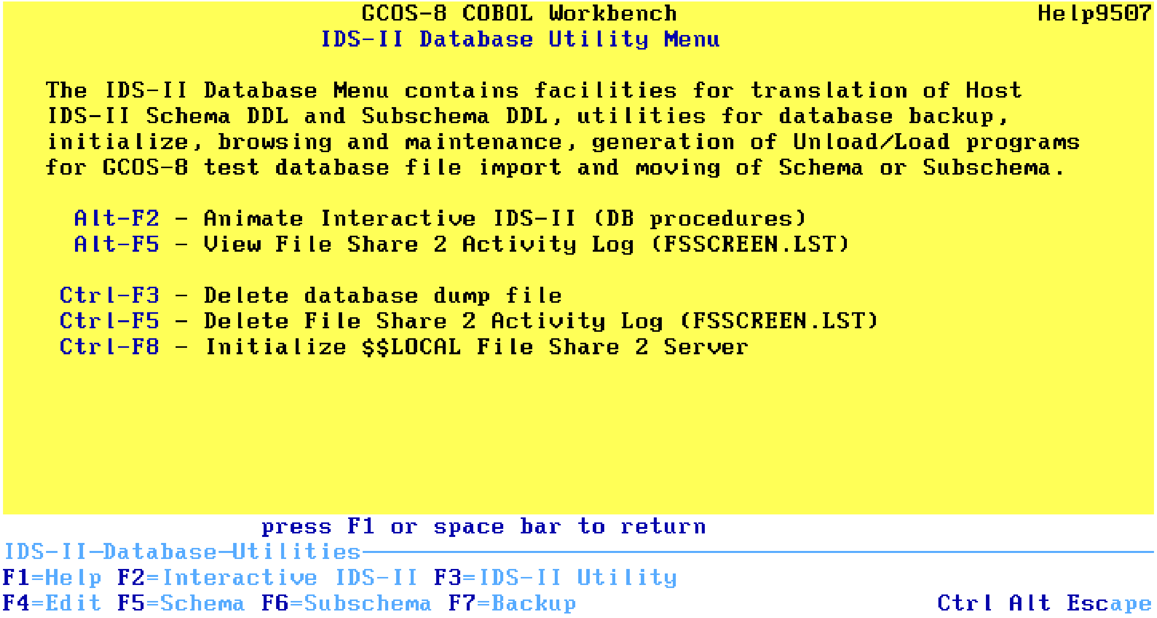
## File Share 2

You may enable Integrity Control for your database, to obtain a functionality similar to ABORT/ROLLBACK/ on GCOS-8. Ref. Configuration chapter for details.

When you have Integrity Control enabled, the database is protected through MF File Share 2. If you are unable to use your database, the cause of error could be related to File Share 2. File Share 2 will document error information on the FSSCREEN.LST file and you may view this file through the Alt-F5 function, to determine the cause of the error.

If the workstation crash or you are forced to boot during access to a protected database, File Share 2 may not be able to perform an automated recovery when you come back and start to use the database again. Typically your workbench will crash and refer to FHREDIR as the causing program. If this happens, you need to clear your local File Share 2 server. This may done through ALT-F8 function.

In general it is recommended that you perform this clear whenever you are forced to boot during use of a proctected database. The clear funtion is automatically perform after a database restore.

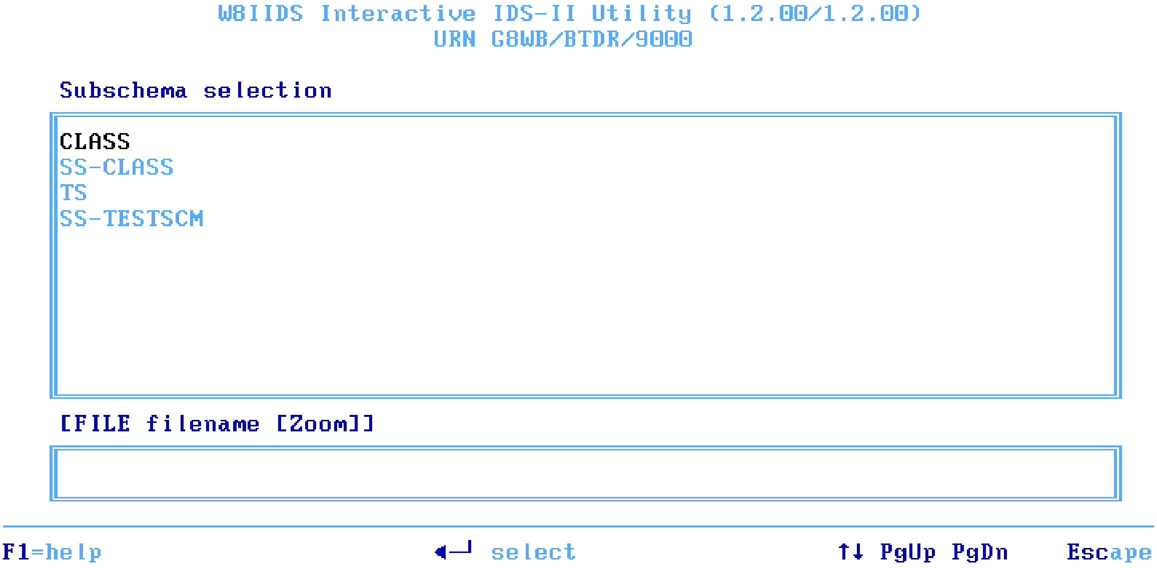


```
                    GCOS-8 COBOL Workbench                      Help9507
                  IDS-II Database Utility Menu

  The IDS-II Database Menu contains facilities for translation of Host
  IDS-II Schema DDL and Subschema DDL, utilities for database backup,
  initialize, browsing and maintenance, generation of Unload/Load programs
  for GCOS-8 test database file import and moving of Schema or Subschema.

    Alt-F2 - Animate Interactive IDS-II (DB procedures)
    Alt-F5 - View File Share 2 Activity Log (FSSCREEN.LST)

   Ctrl-F3 - Delete database dump file
   Ctrl-F5 - Delete File Share 2 Activity Log (FSSCREEN.LST)
   Ctrl-F8 - Initialize $$LOCAL File Share 2 Server

                 press F1 or space bar to return
IDS-II-Database-Utilities-----------------------------------------------
F1=Help F2=Interactive IDS-II F3=IDS-II Utility
F4=Edit F5=Schema F6=Subschema F7=Backup                 Ctrl Alt Escape
```

# Interactive IDS-II

The Interactive IDS-II Utility (IIDS) is quite similar to the IIDS utility available on GCOS-8. Although both work in a line oriented mode, G8WB provides a number of enhancements, which should make database file maintenance a little bit easier.

You select the Subschema, and thereby a schema and database that you wish to work with, from the list of configured subschema's.



```
              W8IIDS Interactive IDS-II Utility (1.2.00/1.2.00)
                           URN G8WB/BTDR/9000

   Subschema selection
   CLASS
   SS-CLASS
   TS
   SS-TESTSCM

   [FILE filename [Zoom]]

F1=help                      ◄┘ select              ↑↓ PgUp PgDn    Escape
```

The FILE command may be passed directly, otherwise IIDS will enter a prompt mode until you exit.

*Note: You are able to invoke IIDS in animated mode also. This will cause database procedures to be animated during IIDS execution.*

## Commands

The IIDS commands are similar to GCOS-8 and you should view BULL documentation for details on DML syntax. Following non-DML commands are available.

MOVE literal|value|ws-field TO ws-field|db-item

Move literal, numeric value or work-field to work-field or database-item (record, field, parameter).

LIST REGISTERS|STACK

List DB-Registers
List command stack

LIST CONTENT OF ws-field|db-record|db-item

List content of a record, field, parameter or work-field

LIST STATUS OF db-realm|db-record|db-set|db-key

List currencies for realm, record, set or key

LIST STATUS OF REALMS|RECORDS|SETS|KEYS

List currencies for all realms, records, sets or keys

LIST REALMS|SETS|RECORDS|PARAMETERS|FIELDS [OF db-record]

List defined realms
List defined sets
List defined records
List defined fields
List defined fields within record

REPEAT n-statements n-times|END [ALL]

Repeat command(s) from stack

FILE filename [Zoom]

Process command input file

RECORD OFF|filename

Record commands on output file

INIT db-record

Initialize all fields within a record

MASK db-record

Build a list of MOVE's (all fields) and invoke the Editor

EDIT filename

Edit any command file via Editor

DB-DEBUG ON|OFF

Display debug information (set status etc.) during processing

COMMIT

Make database commitment (only valid with Concurrency/Integrity enabled)

ROLLBACK

Rollback database changes (only valid with Integrity enabled)

DONE (quit)

The command prompt is a horizontally scrollable field with a maximum length of 250. You may use cursor, insert and delete keys while working in the field. A command or DML verb cannot be split on multiple lines.

The command prompt is also vertically scrollables within the 10 last entered lines. You may use cursor up-down keys to scroll, very much like the DOSKEY function. Scrolling does not affect the Stack, which is updated each time you execute a command or DML verb.

## Command files

Command files may be used to process or re-process a batch of DML verbs. The Zoom mode (process all directives without stopping) may be specified in the FILE command or after process of each verb(prompt). Command file processing terminates with an implicit LIST REGISTERS.

Command files may be captured through the RECORD command and edited through the EDIT command.

## DB Exceptions

Database exceptions are documented with db-status, a brief translation of the status and general status information (like LIST Registers).

```
get owner-rec1

*** Data base exception - GET
    Current record of run unit is not correct type

* DB-STATUS        = 0803300
* DB-REALM-NAME    =
* DB-SET-NAME      =
* DB-RECORD-NAME   = OWNER-REC1
* DB-KEY-NAME      =
* DB-DATA-NAME     =
* DIRECT-REFERENCE = +0000000079

=
```

## Limitations

- MOVE db-item TO ws-field or db-item is not supported.
- Number of work fields is limited to 10.
- Command stack is limited to 20 entries.

## Dynamic IDS-II Debugger

The Dynamic IDS-II Debugger is basically IIDS invoked dynamically during run-time, whether running batch or TP. Dynamic Debugger must be enabled via configuration and the exception filter configuration is used to select/deselect database exceptions, which should dynamically invoke the debugger, i.e. 0502100 may be consider normal, whereas 0503300 may not. Ref. Configuration chapter for details.

Once IIDS is invoked, the complete functionality is available to support debugging and corrective measures, i.e. establish a missing currency before return to program execution.

When IIDS is terminated (Done), you are given various options before returning to the origination application program:

- Clear (reset db-status, only appear if non zero)
- Return (return to execution statement following DML verb)
- Zoom (disable dynamic debugger and return)
- Break (force return to a break-point if animating)

*Note: If you use DB-EXCEPTION procedures, make sure you clear DB-STATUS before returning if you intend to restart the DML verb, otherwise your DB-EXCEPTION procedure may cause your program to abort or stop.*

*Note: Via Break you will return to the origination application program, where you may restart you program anywhere (including re-execution of failed DML verb), using the standard Restart feature of the ANIMATOR.*

During animation, you cannot execute a DML verb via the DO feature of Animator, because a DML verb is not a COBOL statement. You may, however, invoke Dynamic Debugger through the DO feature, and thus execute any DML verb once you are into the Dynamic Debugger. The DO statement to invoke Dynamic Debugger is:

**CALL “DB_<subschema-name>“**

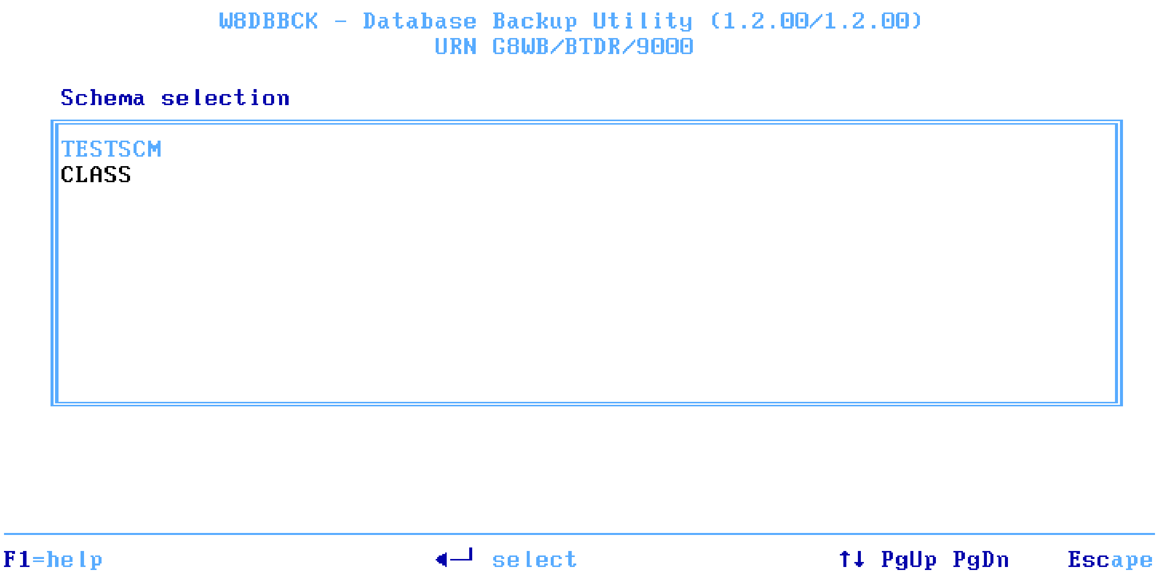
where <subschema-name> is the name of the subschema currently being used.

## Backup/Restore Database

The Backup/Restore Database Utilities may be used for flexible database version manipulation. Backup/Restore works on the basis of Backup-Sets, which are identified by a informative keyword. Ref. Configuration chapter for details on backup-set configuration.

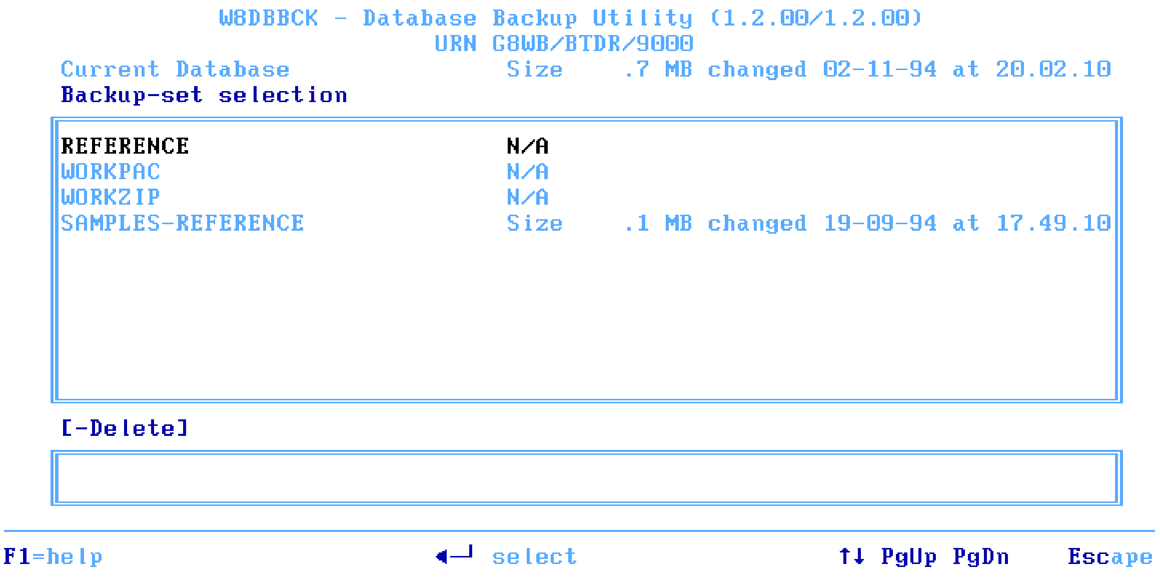
## Backup Database

You select the Schema, and thereby database that you wish to backup, from the list of configured schema's.



```
               W8DBBCK - Database Backup Utility (1.2.00/1.2.00)
                              URN G8WB/BTDR/9000

    Schema selection

    TESTSCM
    CLASS


F1=help                        ◄┘ select                 ↑↓ PgUp PgDn    Escape
```

Backup Utility will provide you with status information on your current database and give you a list of the configured backup-sets with status information. The time stamp information is taken from the physical database file(s), so the time stamp on the backup-set does NOT reflect the time of backup, but the time-stamp of the physical database file(s) at the time of backup.

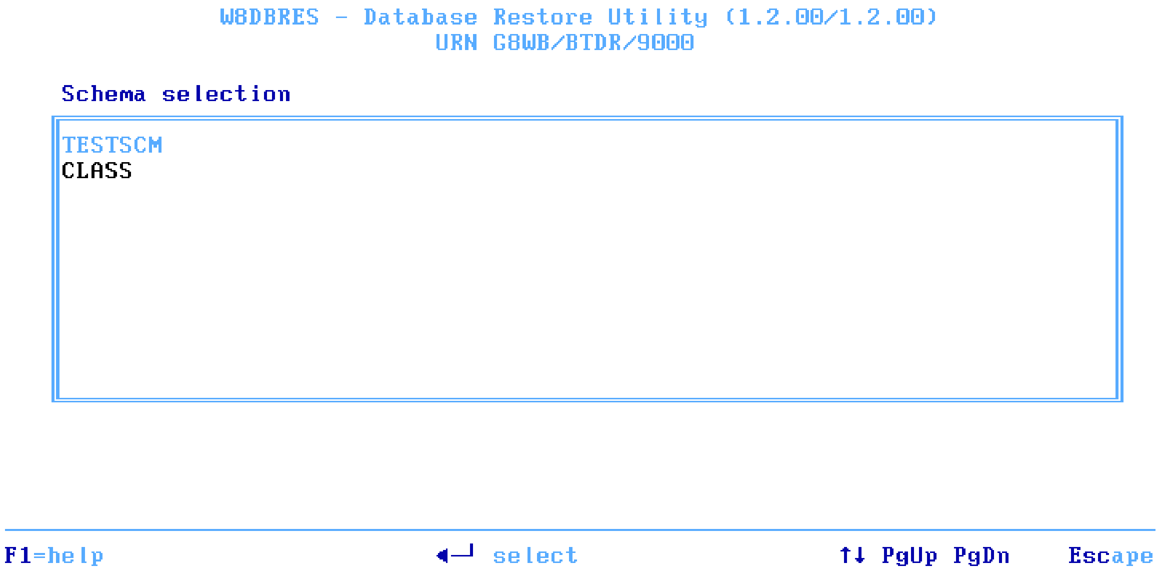


```
               W8DBBCK - Database Backup Utility (1.2.00/1.2.00)
                              URN G8WB/BTDR/9000
    Current Database                Size    .7 MB changed 02-11-94 at 20.02.10
    Backup-set selection

    REFERENCE                       N/A
    WORKPAC                         N/A
    WORKZIP                         N/A
    SAMPLES-REFERENCE               Size    .1 MB changed 19-09-94 at 17.49.10

    [-Delete]

F1=help                        ◄┘ select                 ↑↓ PgUp PgDn    Escape
```

-D(elete), may be used to delete a backup-set, instead of performing a backup.

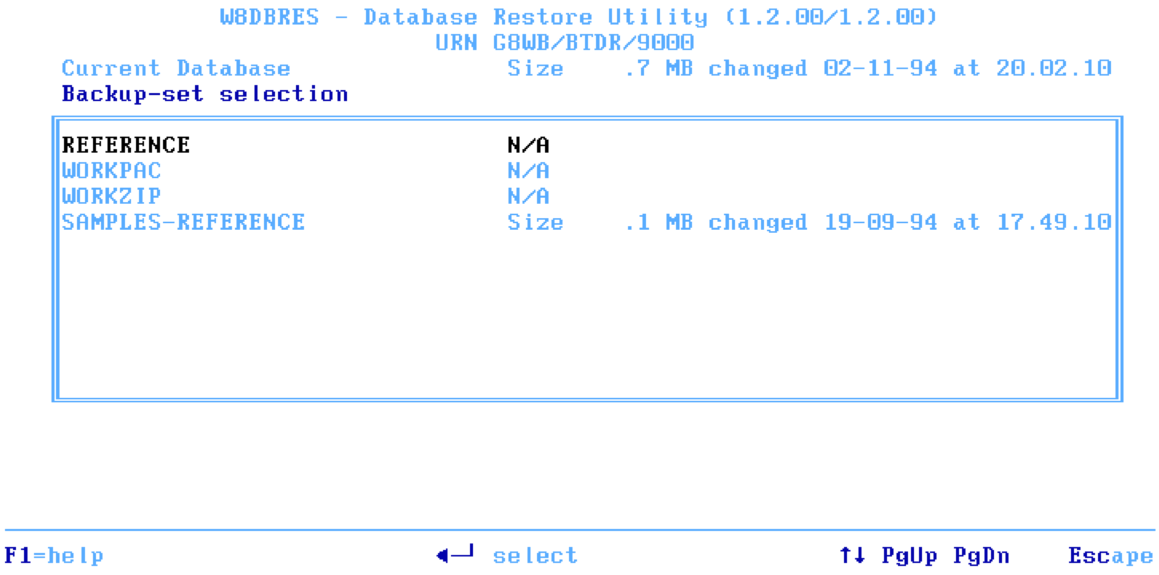
## Restore Database

You select the Schema, and thereby database that you wish to restore, from the list of configured schema's.



```
                 W8DBRES - Database Restore Utility (1.2.00/1.2.00)
                               URN G8WB/BTDR/9000

    Schema selection

    TESTSCM
    CLASS

F1=help                            ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

Restore Utility will provide you with status information on your current database and give you a list of the configured backup-sets with status information. The time stamp information is taken from the physical database file(s), so the time stamp on the backup-set does NOT reflect the time of backup, but the time-stamp of the physical database file(s) at the time of backup.



```
                 W8DBRES - Database Restore Utility (1.2.00/1.2.00)
                               URN G8WB/BTDR/9000
    Current Database               Size    .7 MB changed 02-11-94 at 20.02.10
    Backup-set selection

    REFERENCE                      N/A
    WORKPAC                        N/A
    WORKZIP                        N/A
    SAMPLES-REFERENCE              Size    .1 MB changed 19-09-94 at 17.49.10

F1=help                            ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

# Host Database Unload

Existing GCOS-8 test database files may be unloaded, transferred and loaded into G8WB environment, using any existing unload/load application programs. The DB Unload file must be converted through G8WFL, before being used by the application load program. If the DB unload file has non-DISPLAY type date, you need to use mask(s) to ensure all non-DISPLAY data fields are converted correctly. Ref. Files chapter for details on G8WFL.

G8WB also provides a generic Host Database Unload feature, which will allow you to unload existing test database either completely or selectively.

## Generic

Schema validation automatically generate all the various components need to perform unload of host test databases.

The generated reference subschema is, by default, the basis for database unload and load activities. The reference subschema is a complete view of the schema, and it is named SS-<schema-name>. Any underscores (_) in the schema name is replaced by dashes (-).

The generated Host unload program, is a COBOL-74 program containing all required logic to unload all or parts of the host database into a UFAS sequential file. Binary and packed fields are converted to DISPLAY form, thus the unload file will only contain DISPLAY format data, and does not required any field level conversion to be used by G8WB. The program is named <schema-number>HDBU.CBL and is placed under $G8WBDDIR\SCHEMA, i.e. $G8WBDDIR\SCHEMA\25HDBU.CBL.

The generated Workbench load program, is a COBOL/2 program containing required logic to read the schema specific unload file and load entities into the database. If the unload file represent a complete unload of the database, the load program will initialize the database before loading.

The generated Workbench unload program, is a COBOL/2 program containing logic to unload all or parts of the G8WB database. G8WB unload/load may be used to migrate existing database between different schema versions.

When you initially translate and validate a schema, G8WB will assign the initial version number one (1). When you later need to change your schema, then modification of DDL/DMCL and schema translation activity, will NOT affect your current schema version and you may repeat this cycle until your changes are ok. When you next validate the schema, then you override the current schema and the current version number for the schema is incremented, i.e. two (2) after second validate. Schema validator will create a new reference subschema, translate and validate it, and generate new unload programs also.

Both Host and Workbench unload programs will relate to the new schema version, whereas the Workbench load programs will relate both to the new schema version and any previous schema versions, by default. This will allow the load program to process unload file from both current and previous versions of the schema.

## Non-customized

Initially we will assume that you want to unload the entire host test database and that the schema view represent the data fields directly, i.e. each field is described in its basic format. We will use the CLASS schema, which is pre-configured with schema-number 26, as an example.

After completing validation, you need to transfer the DDL for the reference subschema and the host unload program to GCOS-8. These files are located under G8WBDDIR directory as follows:

**$G8WBDDIR\SSCHEMA\26SCHS.SSD**

**$G8WBDDIR\SCHEMA\26HDBU.CBL**

You will notice that the short name for a generated reference subschema is nnSCHS, where nn is the schema number.

Next you need to translate and validation the reference subschema against your host schema. DBACS directives for translate and validate is included in the DDL, however, JCL you need to build manually.

Next you need to compile the host unload program, using the reference subschema, or build a JCL job with a compile-link-go for the unload program. The unload file, which is a UFAS sequential file, must be allocated with file-code QQ. You should create the unload file with a minimal size and allow it to be grown during unload. This way the size of the file will reflect only the information being unloaded. Finally you should include one line of parameter input through the I* file as follows:

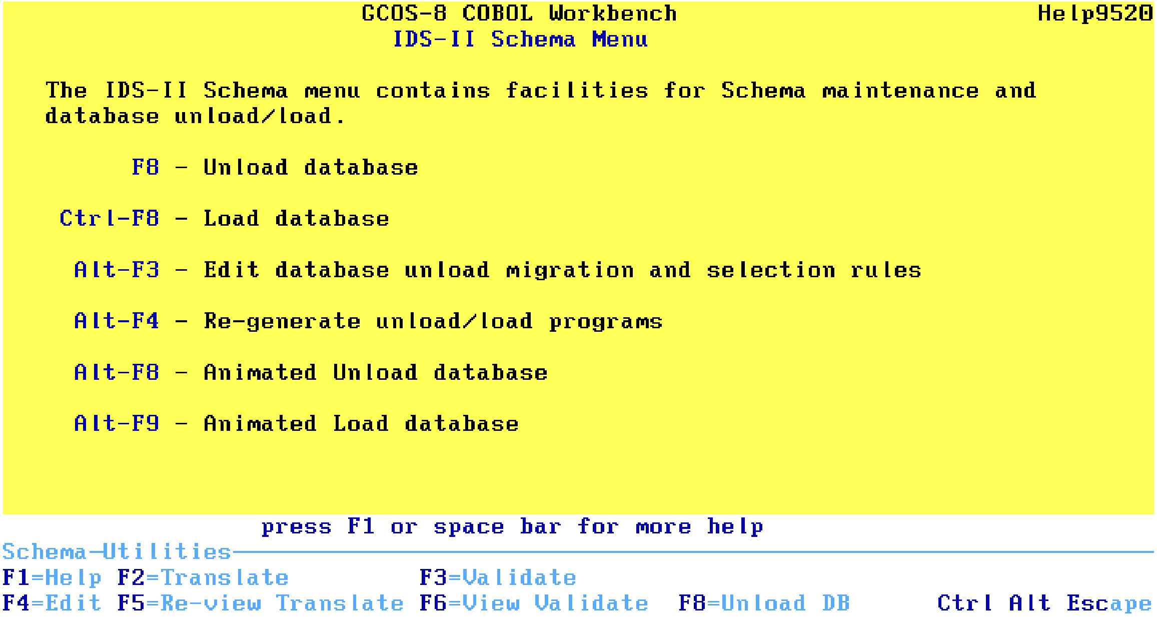
**UNLOAD=ALL**

This parameter will instruct the unload program to unload the entire database, using the fastest possible logic.

Next you need to transfer the unload file to your G8WB environment. Remember the file is UFAS (random) and you thus need to perform a BINARY file-transfer of the file.

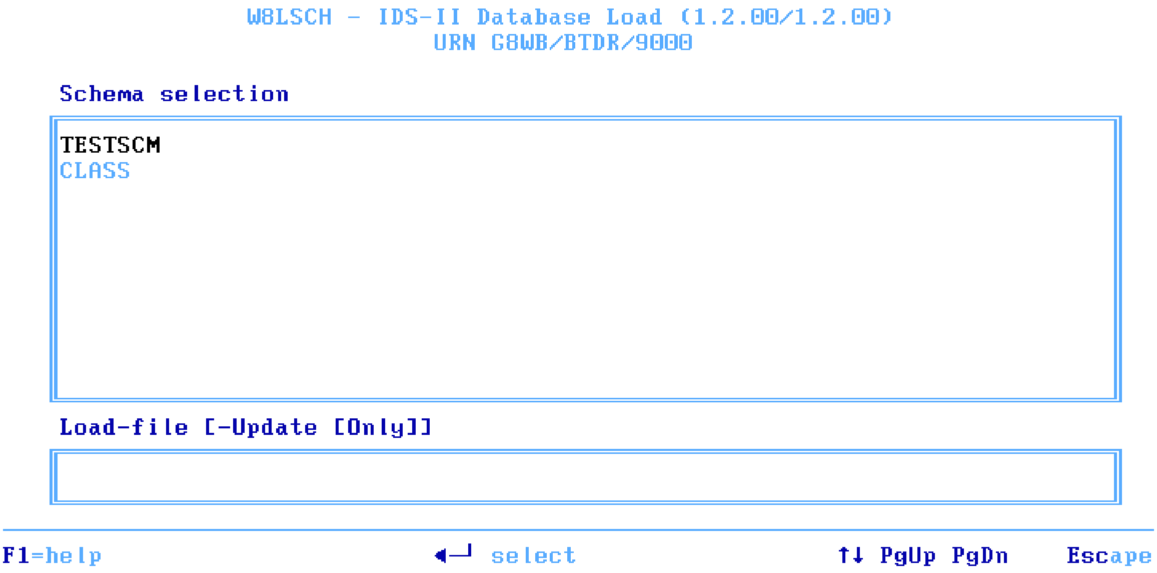
Next you need to convert the unload file from UFAS format to the equivalent G8WB format. You will use G8WFL, ref. Files chapter for details, which only need to be given the name of the input and output file. No mask is required, since the unload file contains DISPLAY data only.

You are now ready to load your host test database into your G8WB environment, and you select Ctrl-F8 from the schema menu to do this.



```
                          GCOS-8 COBOL Workbench                    Help9520
                            IDS-II Schema Menu

  The IDS-II Schema menu contains facilities for Schema maintenance and
  database unload/load.

        F8 - Unload database

   Ctrl-F8 - Load database

    Alt-F3 - Edit database unload migration and selection rules

    Alt-F4 - Re-generate unload/load programs

    Alt-F8 - Animated Unload database

    Alt-F9 - Animated Load database


                  press F1 or space bar for more help
Schema-Utilities────────────────────────────────────────────────────────────────
F1=Help F2=Translate         F3=Validate
F4=Edit F5=Re-view Translate F6=View Validate  F8=Unload DB      Ctrl Alt Escape
```

Select the schema using cursor and enter key, and specify the name of the unload file, i.e. output from G8WFL.

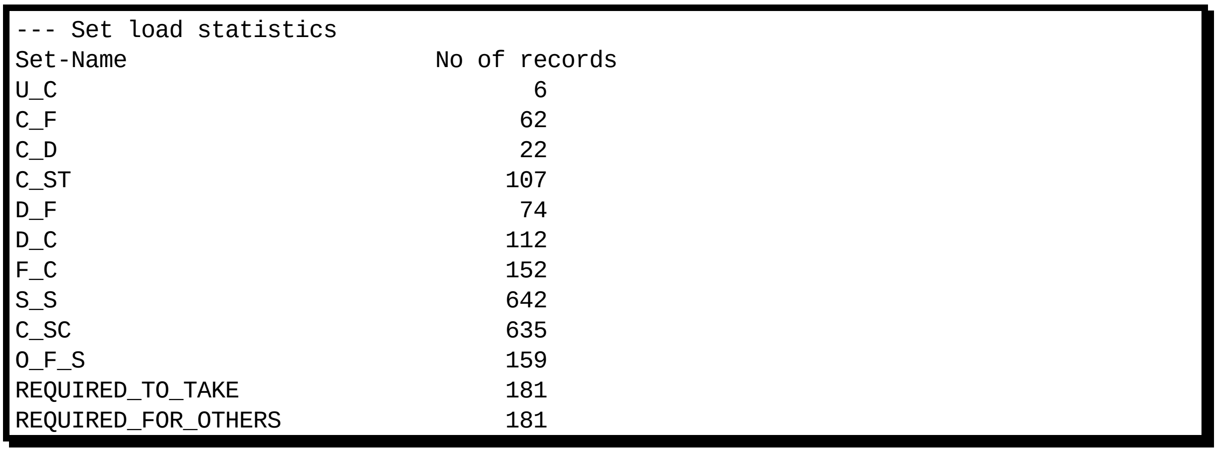


```
           W8LSCH - IDS-II Database Load (1.2.00/1.2.00)
                        URN G8WB/BTDR/9000

    Schema selection
    TESTSCM
    CLASS

    Load-file [-Update [Only]]
```

The database is first initialized and then loaded with the entities from your host test database. Upon completion, the load program with provide you load statistics on the screen. The same information is written to a file, which is named similar to the unload file, but with extension LLG.

Following example illustrate the content of the LLG file:

```
26WDBX Initialize load of database for CLASS
Input file is D:\G8WB.12\CLASS.DAT
Input file is unloaded with Schema version 001

--- Area load statistics
Area-Name                     No of records
UNIVAREA                           182
SCHEDAREA                          721
Total for all Area's               903

--- Record load statistics
Record-Name                   No of records
UNIVERSITY                           1
COLLEGE                              5
DEPARTMENT                          17
COURSE                              95
FACULTY                             57
STUDENT                            102
SCHEDULE                           540
PREREQUISITE                        86
```



```
--- Set load statistics
Set-Name                      No of records
U_C                                6
C_F                               62
C_D                               22
C_ST                             107
D_F                               74
D_C                              112
F_C                              152
S_S                              642
C_SC                             635
O_F_S                            159
REQUIRED_TO_TAKE                 181
REQUIRED_FOR_OTHERS              181
```

## CrossePAC

With the CrossePAC option enabled, G8WB will automatically compress the reference subschema DDL and the host unload program, and place the archive under your current path. The file is named <schema-number>HDBU.PAC, ex. 26HDBU.PAC. You need to transfer this file to GCOS-8 (non-BINARY) and extract the the two files. The CrossePAC command, TSS or batch, would look like:

**CPAC X /26HDBU**

With the CrossePAC option you are able to compress the database unload file (UFAS file) file before doing file transfer. This will reduce the file size by 70-90% and CrossePAC will create a text file archive, which you are able to transfer as a non-binary file.

The G8WFL Utility will automatically recognize a CrossePAC archive file and decompress the file before converting it. It does, however, assume that the embedded file in the CrossePAC archive file is named CPACWORK.

If, as an example, you were to build an unload file named UNLOAD.1 on GCOS-8 for transfer to G8WB, you would instead build a file named CPACWORK and create a CrossePAC archive file named UNLOAD.1. The CrossePAC command, TSS or Batch, would look like:

**CPAC A  SP=N  /UNLOAD.1 /CPACWORK**

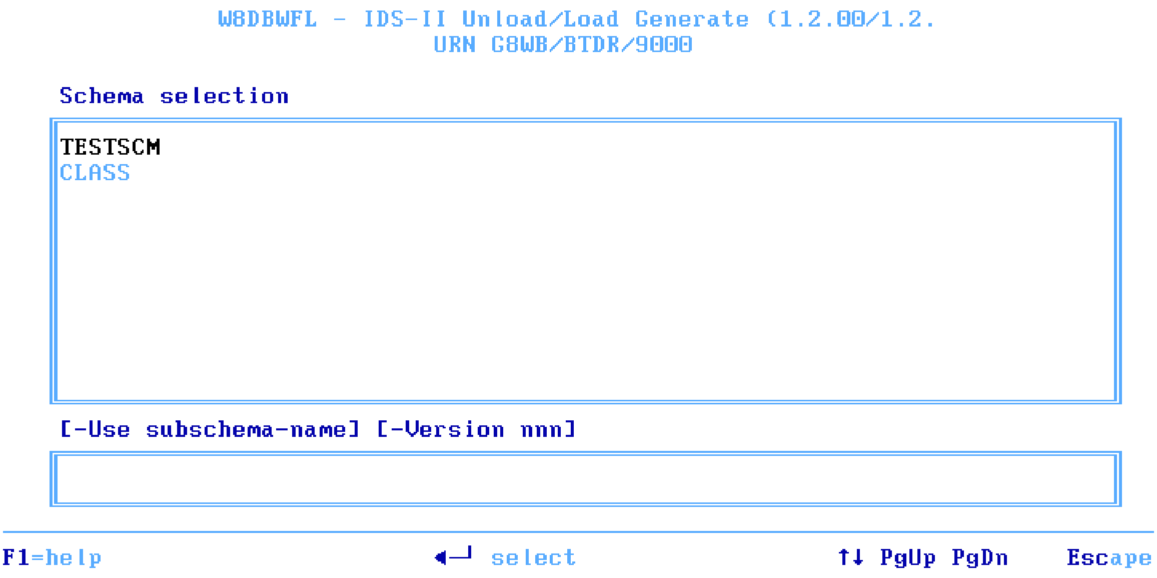
## Field-Migration

Unloading of a host test database is pretty straight forward, as long as the schema view represent the data fields directly, i.e. each field is described in its basic format. The data formats Character and Unspecified may, however, describe a group of fields, which are later redefined in the subschema view, or even in Working-Storage of the accessing application program.

The implementation of non DISPLAY format fields is different between GCOS-8 and G8WB, and database unload will NOT work correctly if fields are not described directly in the schema view. Generic database unload/load is by default based on the reference subschema, which is generated during schema validation. The reference subschema will have the same view as the schema, although this view is described using COBOL data types, i.e Character becomes PIC X, Decimal becomes PIC 9 COMP-4.

If you have a subschema, where data groups are expanded, so each field becomes directly described with appropriate data types, then you may use this subschema instead of the reference subschema, to generate unload and load programs. This would resolve the above described problem with data field migration.

Once you translated and validated your subschema, you may then re-generate the unload and load programs using Alt-F4 funtion from the schema menu.



```
W8DBWFL - IDS-II Unload/Load Generate (1.2.00/1.2.
URN G8WB/BTDR/9000

Schema selection

TESTSCM
CLASS

[-Use subschema-name] [-Version nnn]
```

Select the schema using cursor and enter key, and then specify -Use subschema-name to trigger generate of unload and load programs, based on this subschema. The subschema most represent a coherent view, otherwise the unload program may fail to access a component, i.e. you have described a SET and it’s member(s), but not it’s owner.

Otherwise database unload/load activities are as described above under Non-Customized.

If, however, such redefinition of schema fields cannot be solved by an existing subschema, either because it does not exist, the redefinition takes place in the application itself or the redefinition is dynamic, then you need to establish a set of field migration rules.

Such field migration rules, which are described in COBOL, may be merged into the logic of the generated unload and load program. The purpose being that all elementary data fields (non DISPLAY format) are properly converted into the same format, which is being used by your application while executing with G8WB environment.

A dummy rule configuration file will be generated during initial translation of your schema. This configuration may be amended as required, and merged into unload and load programs during re-generation as described above. The dummy configuration contain basic information on how to establish rules. You acces the rule configuration file through the Alt-F3 function from the schema menu.

```
; Unload Rules Configuration V1.2.00
; ==================================
;
; Unload Rules Configuration may be used to migrate field(s) during
; the unload process and/or perform a selective unload of a database.
;
; The Rules are merged into Unload and Load programs. Whenever you
; change this configuration file, you must then re-generate the
; Unload & Load programs via IDS-II Database Menu.
```

```
;
; -----------------------------
; Field Migration Configuration
; -----------------------------
;
; Field Migration is ONLY required when the subschema describe
; groups of fields, which are later redefined with other than
; display format. Such fields must be migrated, otherwise these
; are treated as ASCII data, and cannot be recognized correctly
; by G8WB when loaded, because non-display format data have different
; implementation on GCOS-8 and in G8WB.
;
; G8WB will, by default, generate a subschema from the schema to
; support the unload process both on GCOS-8 and in G8WB.
;
; Following simple example illustrate when field migration is required:
;
; A-FIELD type unspecified 100. (Schema)
;
; 02 A-FIELD pic x(100).        (Generated Subschema)
;
; 02 A-FIELD.                   (Program)
;  03 a-field-1 pic 9(4) comp-4.
;  03 filler    pic x(98).
;
; Continuing above example, this configuration file should be amended
; as follows:
;
;[OWN-WS] *> Define help fields for migration
       01  REDEFINE-1.
        02 REDEFINE-1-FIELD-1 pic 9(4) comp-4.
        02 REDEFINE-1-FIELD-2 pic x(98).
;[WS,A-FIELD,A-RECORD] *> Define replace fields in Unload/Load record
          02  UNIQUE-FIELD-1 pic 9(4).
      *       Note: All fields must be display format
          02  UNIQUE-FIELD-2 pic x(98).
;[PRU,A-FIELD,A-RECORD] *> Unload logic
          move A-FIELD in A-RECORD to REDEFINE-1
          move REDEFINE-1-FIELD-1 to
               UNIQUE-FIELD-1 of <unload-record>
          move REDEFINE-1-FIELD-2 to
               UNIQUE-FIELD-2 of <unload-record>.
;[PRL,A-FIELD,A-RECORD] *> Load logic
          move UNQIUE-FIELD-1 of <unload-record> to
               REDEFINE-1-FIELD-1
          move UNIQUE-FIELD-1 of <unload-record> to
               REDEFINE-1-FIELD-2
          move A-FIELD in A-RECORD to REDEFINE-1.
;
; Note: UNIQUE-FIELD-1 and -2 may be generated into several unload
;       record descriptions (one per schema version), and you thus
;       need to qualify when referencing these fields. Since the
;       qualification name (unload record description) is internally
;       generated (not known to you), you may instead specify a
;       generic qualification name of <unload-record>. This generic
;       name is then changed during generation of unload/load
;       programs.
;
; Note: When more complex logic is required for field migration, the
;       Unload/Load logic should perform SECTION's, so the fall-through
;       logic remains unchanged. Such SECTION's may be inserted via
;       OWN Procedure coding, ref. below.
```

```
;
; if, however, you have an existing subschema (must describe the
; whole schema), which describe the REDEFINE, the you may instead
; choose to generate the Unload & Load programs using this subschema.
;
; Following simple example illustrate when field migration is not required:
;
; A-FIELD type unspecified 100. (Schema)
;
; 02 A-FIELD.                   (Subschema)
;  03 a-field-1 pic 9(4) comp-4.
;  03 a-field-2 pic x(98).
;
; In above example, the Unload & Load programs will use the elementary
; fields, i.e. A-FIELD-1 and A-FIELD-2 instead of A-FIELD, and will
; automatically change these to display format as required.
;
; Following Rule Tag's may be used:
;
;[HDBU-INIT] Preprocessor directives for Host Unload Program
;
; Note: Used to preprocess OWN-WS(x) definitions.
;
;[OWN-WS] Own Unload & Load Working Storage Section definitions
;[OWN-WSU] Own Unload Working Storage Section definitions
;[OWN-WSL] Own Load Working Storage Section definitions
;
;[WS,field,record] Unload/Load Field definition
;   where field  = name of the subschema field
;         record = name of record for qualification (required)
;
;[PRU,field,record] Unload Field Move logic
;[PRL,field,record] Load Field Move logic
;
; Note: Last line must be terminated by (.).
;
;[OWN-PR] Own Unload & Load Procedure Division logic
;[OWN-PRU] Own Unload Procedure Division logic
;[OWN-PRL] Own Unload Procedure Division logic
;
; Note: Any lines following tag until next tag will be inserted.
;
; Note: OWN-xxx may be used to included further definitions and logic
;       to support PRU/PRL conversion logic.
;
; Note: A Tag [] must start in column 1 (above are comment lines
;       before the first real Tag.
```

In the following you find another example of field migration, which originate from the DIMS8 security database.

Within the record DIMS-OPERATOR-EXTENTION, the schema and subschema describe a group field, which is used as a kind of FILLER where field attributes can be added as required, without affecting the neither the schema nor the subschema view. The field DIMS-OPERATOR-OPTION, which is described as CHAR 100 and PIC X(100), is thus redefined in the application as follows:

```
         02        WS-OPERATOR-BLOCK-OPTION.
           05      OPERATOR-PHONE-EXT              PIC X(20).
           05      OPERATOR-PASSWORD-CHANGE        PIC 9(6) COMP-4.
           05      OPERATOR-PASSWORD-LOG.
             10    OPERATOR-PREVIOUS-PASSWORD      OCCURS 3 TIMES
                                                   PIC X(12).
           05      OPERATOR-DISTRIBUTION-GROUP     PIC 9(4).
           05      OPERATOR-SYSTEM-NAME            PIC X(8).
           05      OPERATOR-DEPT-NO-X.
             10    OPERATOR-DEPT-NO                PIC 9(8) COMP-4.
           05      OPERATOR-DISPLAY-MODES.
             10    OPERATOR-DISPLAY-MODE-X         OCCURS 4 TIMES
                                                       PIC 9(1) COMP-4.
           05      FILLER                          PIC X(33).
```

The field OPERATOR-PASSWORD-CHANGE and the other COMP-4 fields would be treated as DISPLAY data, and would NOT contain correct values after unload/load, because of the difference in implementation between GCOS-8 and G8WB. Using rules configuration, we thus need to make the fields visible and make the field conversion part of the unload/load processing. Following illustrate the rules required to do this:

```
[OWN-WS] Own Unload & Load Working Storage Section definitions
       01  WS-OPERATOR-BLOCK-OPTION.
           05      OPERATOR-PHONE-EXT              PIC X(20).
           05      OPERATOR-PASSWORD-CHANGE        PIC 9(6) COMP-4.
           05      OPERATOR-PASSWORD-LOG.
             10    OPERATOR-PREVIOUS-PASSWORD      OCCURS 3 TIMES
                                                   PIC X(12).
           05      OPERATOR-DISTRIBUTION-GROUP     PIC 9(4).
           05      OPERATOR-SYSTEM-NAME            PIC X(8).
           05      OPERATOR-DEPT-NO-X.
             10    OPERATOR-DEPT-NO                PIC 9(8) COMP-4.
           05      OPERATOR-DISPLAY-MODES.
             10    OPERATOR-DISPLAY-MODE-X         OCCURS 4 TIMES
                                                       PIC 9(1) COMP-4.
```

Above is the Working-Storage redefinition, which has been copied from the application program. Placement under OWN-WS tag will trigger merge into Working-Storage of the generated unload and load programs.

```
[WS,OPERATOR-BLOCK-OPTION,DIMS-OPERATOR-EXTENTION] Unload/Load Field definition
           02  operator-phone-ext            pic x(20).
           02  operator-password-change      pic 9(06).
           02  operator-password-log.
            03 operator-previous-password    occurs 3
                                             pic x(12).
           02  operator-distribution-group   pic 9(04).
           02  operator-system-name          pic x(08).
           02  operator-dept-no-x.
            03 operator-dept-no              pic 9(08).
           02  operator-display-modes.
            03 operator-display-mode-x       occurs 4
```

```
                                             pic 9(01).
```

Above is the request for redefinition of the OPERATOR-BLOCK-OPTION field, which will trigger merge of above field descriptions instead of the original PIC X(100) of the field. You will notice that COMP-4 field have been changed to DISPLAY format.

```
[PRU,OPERATOR-BLOCK-OPTION,DIMS-OPERATOR-EXTENTION] Unload Field Move logic
           perform unload-o-b-option.
[PRL,OPERATOR-BLOCK-OPTION,DIMS-OPERATOR-EXTENTION] Load Field Move logic
           perform load-o-b-option.
```

Above is the new unload and load logic, which will trigger merge of logic instead of the elementary MOVE normally inserted for each defined field. In above example the logic is a simple perform referencing a section. You may specify the logic directly, however no conditional constructs are allowed. If this is required, you need to perform a section and place this elsewhere.

```
[OWN-PRU] Own Unload Procedure Division logic
       unload-o-b-option section.
       section-entry.
           move operator-block-option to
                ws-operator-block-option
           move operator-phone-ext in
                ws-operator-block-option to
                operator-phone-ext in <unload-record>
           move operator-password-change in
                ws-operator-block-option to
                operator-password-change in <unload-record>
           move operator-password-log in
                ws-operator-block-option to
                operator-password-log in <unload-record>
           move operator-distribution-group in
                ws-operator-block-option to
                operator-distribution-group in <unload-record>
           move operator-system-name in
                ws-operator-block-option to
                operator-system-name in <unload-record>
           if operator-dept-no-x in ws-operator-block-option =
              spaces or low-values
              move spaces to operator-dept-no-x in
                             <unload-record>
           else
              move operator-dept-no in
                   ws-operator-block-option to
                operator-dept-no in <unload-record>.
           move operator-display-mode-x in
                ws-operator-block-option ( 1 ) to
                operator-display-mode-x in
                <unload-record> ( 1 )
           move operator-display-mode-x in
                ws-operator-block-option ( 2 ) to
                operator-display-mode-x in
                <unload-record> ( 2 )
           move operator-display-mode-x in
                ws-operator-block-option ( 3 ) to
                operator-display-mode-x in
                <unload-record> ( 3 )
           move operator-display-mode-x in
                ws-operator-block-option ( 4 ) to
                operator-display-mode-x in
                <unload-record> ( 4 ).
       section-exit. exit.
```

Above is the new unload logic, which will be merged into the generated unload programs. You will notice that all field references must be qualified. The name of the actual unload record, which is different for each schema version, will be dynamically inserted in place of <unload-record>.

```
[OWN-PRL] Own Unload Procedure Division logic
       load-o-b-option section.
       section-entry.
           move operator-phone-ext in <unload-record> to
                operator-phone-ext in
                ws-operator-block-option
           move operator-password-change in <unload-record> to
                operator-password-change in
                ws-operator-block-option
           move operator-password-log in <unload-record> to
                operator-password-log in
                ws-operator-block-option
           move operator-distribution-group in
                <unload-record> to
                operator-distribution-group in
                ws-operator-block-option
           move operator-system-name in <unload-record> to
                operator-system-name in
                ws-operator-block-option
           if operator-dept-no-x in <unload-record> =
              spaces or low-values
              move spaces to operator-dept-no-x in
                   ws-operator-block-option
           else
              move operator-dept-no in
                   <unload-record> to
                operator-dept-no in ws-operator-block-option.
           move operator-display-mode-x in
                <unload-record> ( 1 ) to
                operator-display-mode-x in
                ws-operator-block-option ( 1 )
           move operator-display-mode-x in
                <unload-record> ( 2 ) to
                operator-display-mode-x in
                ws-operator-block-option ( 2 )
           move operator-display-mode-x in
                <unload-record> ( 3 ) to
                operator-display-mode-x in
                ws-operator-block-option ( 3 )
           move operator-display-mode-x in
                <unload-record> ( 4 ) to
                operator-display-mode-x in
                ws-operator-block-option ( 4 ).
           move ws-operator-block-option to
                operator-block-option.
       section-exit. exit.
```

Above is the new load logic, which will be merged into the generated load program.

Once you have re-generated the new host unload program, you need to transfer this to GCOS-8 to perform the actual database unload. The host unload program is a COBOL-74 program, and you may naturally use G8WB to test the program before going to GCOS-8. You can either Check and animate the program directly in the G8BWDDIR\SCHEMA directory or copy it into your own directory structure.

Once you have unloaded the host test database, transferred and converted this through G8WFL, you are ready to load using the newly generated load program. This program is automatically compile into GNT format (non-animated), however if you wish to test the customized load logic, you may do so. First you need to load the load program into MF Editor. Then use F2-COBOL and F2-Check to check the program. You must set language to PC-ANS85 or ANS85, using F6-Lang. Finally use Alt-F9 from schema menu to perform an animated execution of the load program.

## Selective-Unload

Large volumes of data may slow down your otherwise responsive and flexible G8WB environment. Depending on your workstation and network environment, you may find that handling test databases of more than 2-10 MB (this figure depend on your specific environment is) make you impatient.

G8WB's IDS-II implementation  will generally ensure database files are as small as possible, since no space is pre-allocated. The database will only contain information for real records and sets. GCOS-8 is not as sensitive about file sizes and test databases are usually restored/copied through batch jobs. Although waiting time is also an issue, it is not as explicit. Results being, that very often you have quite large test databases.

As a general guide-line, you should try to keep test database which are frequently updated to a minimum size, so backup/restore operations can be performed as swiftly as possible.

If you already have a host test database with a substantial amount of data, where the extreme is a copy of a production database, the task of reducing it's size can be quite complex, unless you have application utilities that can perform this type of task.

G8WB's Selective-Unload feature is an attempt to assist you in getting the reduction task done, without having to write a lot of COBOL coding. When reducing the amount of data in the test database, this must be done logically, so the results are coherent, i.e. leaving out records must be reflected in sets etc.

There are two types of selection rules:

- Record/Realm exclusion

- Selective record exclusion

Record and/or Realm exclusion are defined at run-time through parameter input, ref. below for details. You may combine the two rule types as required.

Like field migration rules, you are able to establish a set of selection rules. Such rules, which are described in COBOL, may be merged into the logic of the generated unload program. Selection rules are described in the same configuration file as field migration rules, ref. above for details on how to change rules and re-generate.

```
;
; -----------------------
; Selection Configuration
; -----------------------
;
; Selection configuration may be used to define rules for a selective
; unload of your host test database, and thus create a smaller test
; database for use within G8WB.
;
; Selection logic may be combined with dynamic run-time options for the
; unload program. Via dynamic run-time option you may select/deselect
; specific records, and thus prevent unload of specific record types.
;
; Selection logic is specified at RECORD level, where you may apply
; specific test logic to select only certain occurences of the RECORD
; during unload. Whenever a RECORD is de-selected, any records subordinate
; to the RECORD is also de-selected. A subordinate record is ONLY selected
; when all it's owner(s) is selected also.
```

```
;
; Following simple example illustrate how selection can be configured:
;
; Record A, Owner of Set A-B
; Record B, Member of Set A-B, Owner of Set B-C
; Record C, Member of Set B-C
;
;[SEL,B] Host Unload Selection logic for Record
           if A-key-field > 100 and
                          < 200
              next sentence
           else
              perform de-select-record.
;
; Note: Default is Select.
;
; Selection Tag(s) is described as follows:
;
;[SEL,record] Host Unload Selection logic for Record
;   where record = subschema record-name
;
; Note: Any lines following tag until next tag will be inserted.
;
; Note: OWN-xxx may be used to included further definitions and logic
;       to support selection logic.
;
; Note: A Tag [] must start in column 1 (above are comment lines
;       before the first real Tag.
```

In the following you find another example of selection, which relate to the DIMS8 security database. The test database is actually a copy of the production database, so it describes all TP end-user's as well as developers and analysts. In a pure test environment, we actually only need information about developers, analyst and a few select end-user. Assuming the test database is of a substantial size, we could thus reduce it size by elimination information about most of the end-user's.

The DIMS-OPERATOR record, one for each system user, would thus be the entity we would use as the basis for reduction. This record is a member of three different sets and is the owner of another set. When doing the selection, we need to logically remove the record in the image of the unloaded database, i.e. no longer member of set, subordinate set and members not unloaded etc. We are able to recognize a developer or analyst from the DIMS-OPERATOR-USER-GROUP field and the select end-user we can find by name (initials) in the DIMS-OPERATOR-ID field.

Our aim is thus to perform selection as follows:

1) Authority must be 63

2) Ident must be either “USER1” or “USER2”

The selection rules we establish are as follows:

```
[SEL,DIMS-OPERATOR] Host Unload Selection logic for Record
           if DIMS-OPERATOR-AUTHORITY = 63 or
              DIMS-OPERATOR-ID = “USER1 or “USER2”
              next sentence
           else
              perform de-select-record.
```

The host unload program will take various parameter input from I*.

TYPE=<type>

UNLOAD=<unload>

SELECT=<select-deselect)

DESELECT=<select-delect>

## TYPE

ALL, which is default type, will result in unload of all records and sets.

RECORDS, will limit unload to records.

SETS, will limit unload to set information.

## UNLOAD

SELECTIVE, which is default, will result in a logical unload, where selection logic rules are applied. Selective unload will substantially increase the number of logical I/O and physcial I/O, because each record needs to be verified against all owner's (and their owner's), before selection can be determined. It is recommended to use a buffer pool to reduce the number of physical I/O's.

ALL, will result in unload of the entire database, i.e selection logic ignored.

## SELECT

RECORDS, will result in unload of all record types, except those record occurences which are specifically de-selected by selection rules.

<record-name>, will result in unconditional de-select of all other record types.

REALMS, will result in unload of all realms, except those record occurences which are specifically de-selected by selection rules.

<realm-name>, will result in unconditional de-select of all other realms types.

# Version-Migration

Version migration may be used to handle schema changes, while retaining your existing test database within G8WB environment. On GCOS-8 the Q2LOGR utility is used to migrate between schema versions.

The basis of schema version migration is the Workbench unload program and the ability of the Workbench Load program to handle unload files from different schema versions. Internally G8WB will assign unique numbers to all schema entities, i.e. Areas, Records, Sets, Keys and fields. This numbering is independent of any DMCL type assigment, and the entity numbers are both used in run-time tables and in the physical database. Once numbering has been established, which first takes place during initial schema validation, these CANNOT change.

When you subsequently change a schema, i.e. adding, deleting, changing entities, G8WB must ensure that previous assigned entity numbers remain unchanged. Schema validator will thus build a new version of the schema and re-assign all numbers from the previous version, unless entities have been deleted. New entities will be assigned numbers coherent with existing numbers. Result being, an assigned entity number will never change.

*Note: Database unloads from current version must take place before you validate the new schema version.*

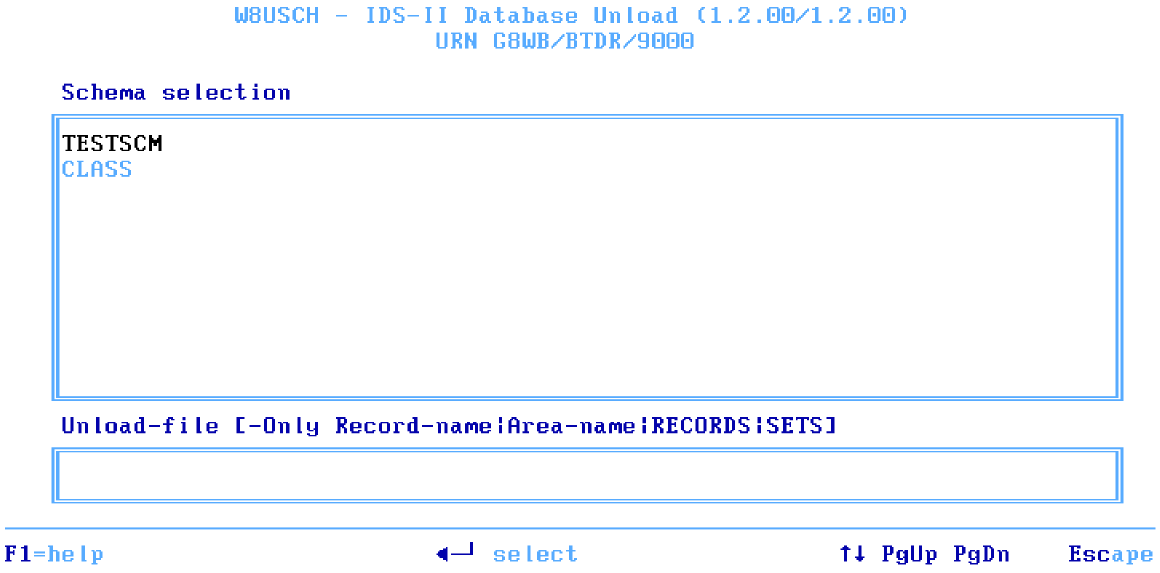
You select F8 from schema menu to invoke Workbench database unload. Select the schema using cursor and enter key, and specify the name of the unload file. Optionally you may limit the unload through -Only <option> as follows:

RECORDS, will unload all record, but no set information.

SETS, will unload all set information, but no records.

<Record-name>, will unload only the specified record, without any set information.

<Area-name>, will unload all records within specified area, without any set information.

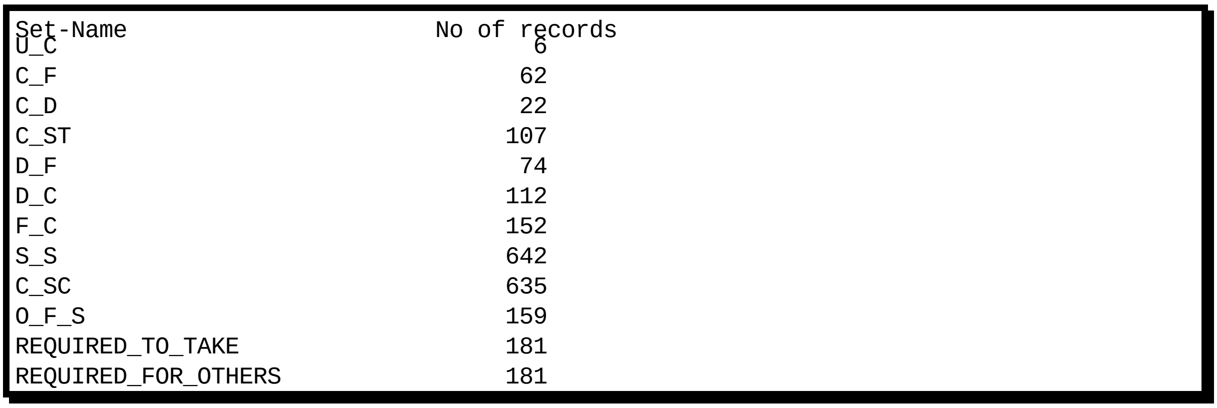


```
              W8USCH - IDS-II Database Unload (1.2.00/1.2.00)
                           URN G8WB/BTDR/9000

    Schema selection

    TESTSCM
    CLASS

    Unload-file [-Only Record-name|Area-name|RECORDS|SETS]

F1=help                    ↵ select                   ↑↓ PgUp PgDn    Escape
```

Upon completion, the unload program will provide you with unload statistics on the screen. The same information is written to a file, which is named similar to the unload file, but with extension ULG.

Following example illustrate the content of the ULG file:

```
26WDBX Complete Unload of database for CLASS
Output file is D:\G8WB.12\CLASS.DAT

--- Area unload statistics
Area-Name                     No of records
UNIVAREA                           182
SCHEDAREA                          721
Total for all Area's               903

--- Record unload statistics
Record-Name                   No of records
UNIVERSITY                           1
COLLEGE                              5
DEPARTMENT                          17
COURSE                              95
FACULTY                             57
STUDENT                            102
SCHEDULE                           540
PREREQUISITE                        86

--- Set unload statistics
```



```
Set-Name                      No of records
U_C                                6
C_F                               62
C_D                               22
C_ST                             107
D_F                               74
D_C                              112
F_C                              152
S_S                              642
C_SC                             635
O_F_S                            159
REQUIRED_TO_TAKE                 181
REQUIRED_FOR_OTHERS              181
```

Workbench load program will automatically recognize whether an unload file is partial or complete, thus when you later load the unload file, the database is only initialized if the unload file indicate complete.

## Areas

When you add new area(s), you need only migrate the database if any existing records will change area, i.e. change in area selection. In this case you need to unload all the areas involved, i.e one partial unload for each area. You then initialize each of the unloaded areas, and load the areas again from the unload files, once you have changed schema version and possibly area selection procedure.

A change in area selection logic, without change of schema version, involves similar unload, initialize and load activities as described above.

When you delete an area, you need only to initialize the specific area.

*Note: DMCL physical attributes are NOT used by G8WB, thus if you change such attributes you may use schema translator to verify the changes, but you need not validate nor migrate.*

## Records

When you add new record(s), no migration is required.

When you change a record’s location mode and/or duplicate clause, you need to unload the specific record, and load again after change of schema version. Duplicates check is performed during load, and the load program will fail if duplicates are found in the unload file. You may use MF Data File Editor to remove such duplicate records and then rerun the load program.

When you delete a record, no migration is required.

## Keys

When you add a new key to an existing record, you need to unload the specific record, and load again after change of schema version.

When you change a key, you need to unload the specific related record, and load again after change of schema version.

When you delete a key, no migration is required.

## Sets

When you add a new set that involved existing records (either owner or member), you need to unload the specific related record(s), and load again after change of schema version. No set linkage will be made, however dummy set membership records will be created.

When you delete a set, no migration is required.

## Fields

When you add a new field to an existing record, you need to unload the specific record, and load again after change of schema version. New fields are initialized during load, however you may also apply other value assignment through migration rules.

When you delete a field from an existing record, you migrate like for a new field.

When you modify a field in an existing record, you migrate like for a new field. You may need to include specific migration rules, if the field format change or otherwise require special logic.

## Version-handling

For each schema version, the schema validator will create following three copy files under G8WBDDIR\SCHEMA:

**<schema-number>RECORD.<version-number>**

**<schema-number>MOVEUL.<version-number>**

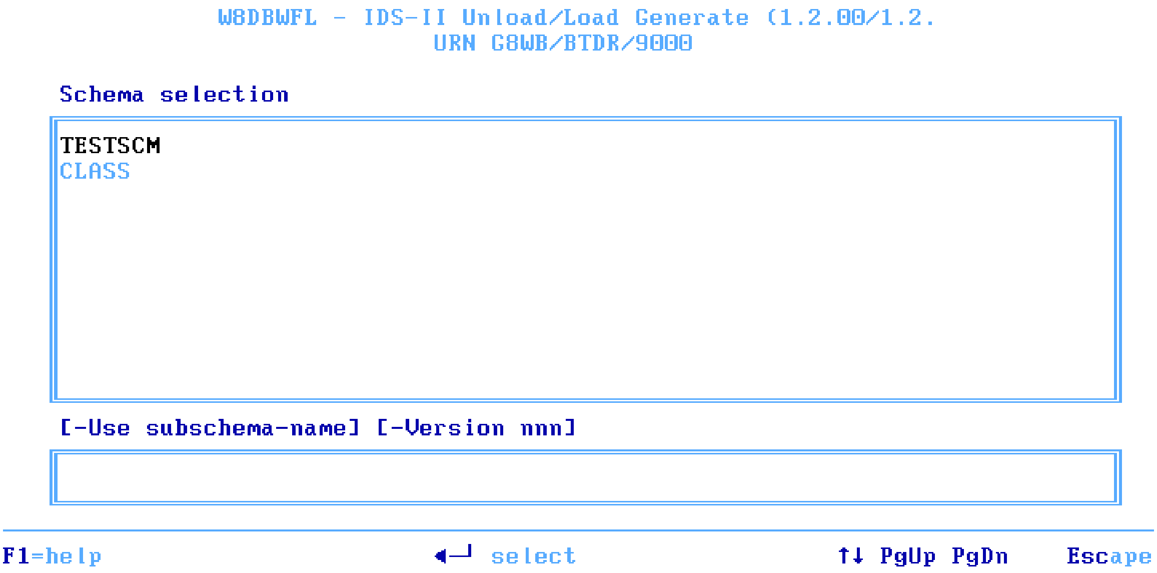
**<schema-number>MOVELD.<version-number>**

The RECORD copy contains a description of all records in the unload-file format. The current version is used by Host and Workbench unload programs, whereas all versions are used by the Workbench load program.

The MOVEUL copy contains unload logic for each record and each field within a record, merged with any migration rules. Only current version is used by Host and Workbench unload programs.

The MOVELD copy contains load logic for each record and each field within a record, merged with any migration rules. All versions are used by the Workbench load program.

When you re-generate the unload/load programs, you may optionally limit inclusion of versions through -Version. All versions, starting from specified until current, are used when generation Wokrbench load program, all others are ignored. This does not affect unload programs, which always use currenct version only.



```
            W8DBWFL - IDS-II Unload/Load Generate (1.2.00/1.2.
                             URN G8WB/BTDR/9000

    Schema selection

    TESTSCM
    CLASS

    [-Use subschema-name] [-Version nnn]
```

*Note: You may want to delete obsolete schema versions, especially for large schema's where the Workbench load program may become very large. Such deletion must be done through OS DELETE command.*

Since Workbench load program is generated with MOVELD copies for all versions, unless you limit this through -Version or delete obsolete versions, you need to take special consideration when you delete record and field entities in the schema or change migration rules.

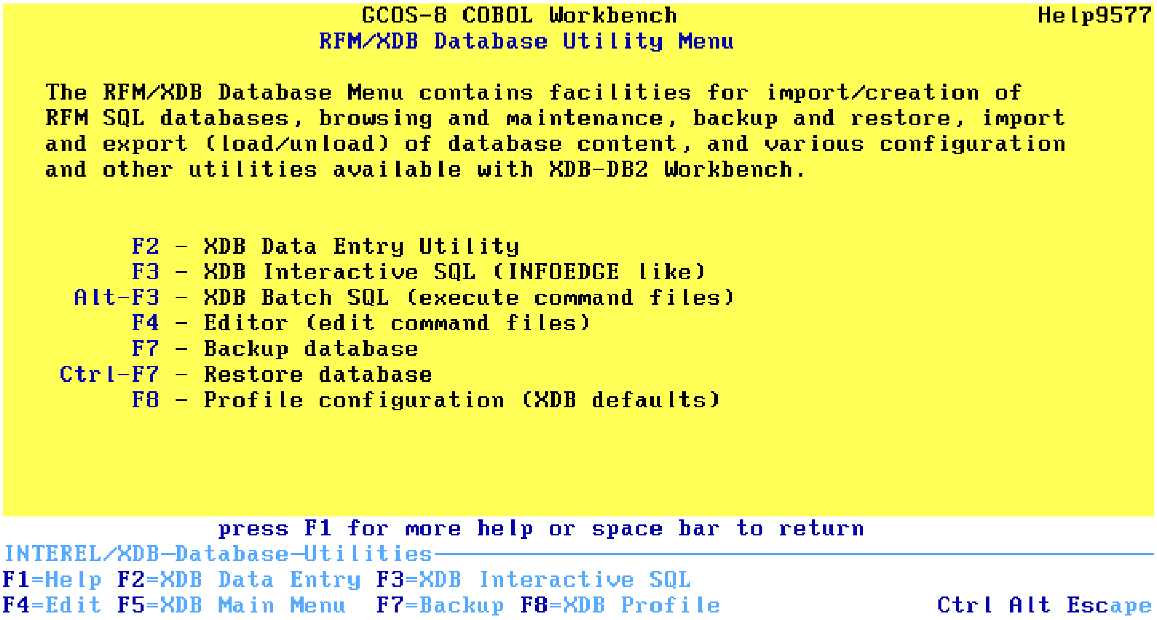
In general, deleted record and field entities must be moved to OWN-WS in the rules configuration file, otherwise you get compilation error due to references move in MOVELD of the previous version. Usually you cannot suppress MOVELD for the previous version, because you need to be able to load from an unload file built with the previous version. Such OWN-WS description are not used, except to satisfy references in load logic from the previous version.

Migration rules are merged into the MOVELD for each version. When you later change such rules, you cannot remove or change OWN-WS or OWN-PR related to previous versions, unless you exclude these from re-generation of the load program.

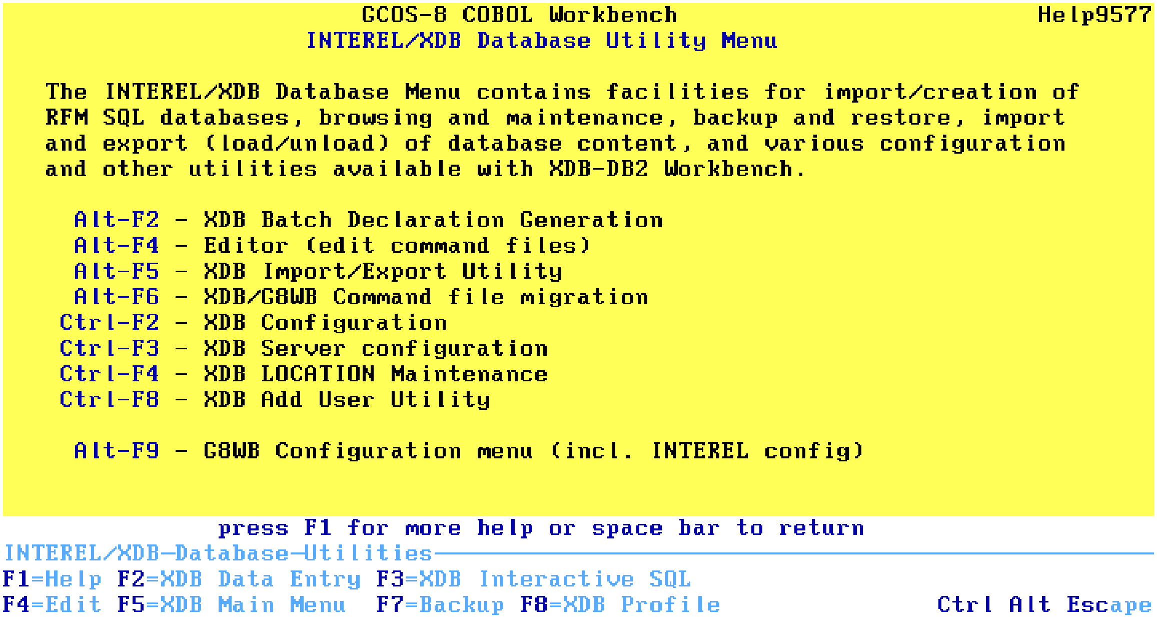
# Chapter 6 INTEREL DATABASE MENU

G8WB's support of the GCOS-8 INTEREL Database system is based on the XDB DB2-Workbench product, which provides an IBM DB2 SQL compatible environment. Since INTEREL is not completely compatible with DB2, various G8WB components will complement XDB and provide an almost completely INTEREL compatible environment. A range of a database maintenance utilities are available both with XDB and G8WB.

The Database Menu contains functions for translation of SQL DDL, database maintenance, configuration, backup/restore and unload/load utilities.



```
                         GCOS-8 COBOL Workbench                         Help9577
                      RFM/XDB Database Utility Menu

  The RFM/XDB Database Menu contains facilities for import/creation of
  RFM SQL databases, browsing and maintenance, backup and restore, import
  and export (load/unload) of database content, and various configuration
  and other utilities available with XDB-DB2 Workbench.


        F2 - XDB Data Entry Utility
        F3 - XDB Interactive SQL (INFOEDGE like)
    Alt-F3 - XDB Batch SQL (execute command files)
        F4 - Editor (edit command files)
        F7 - Backup database
   Ctrl-F7 - Restore database
        F8 - Profile configuration (XDB defaults)




               press F1 for more help or space bar to return
INTEREL/XDB-Database-Utilities------------------------------------------------
F1=Help F2=XDB Data Entry F3=XDB Interactive SQL
F4=Edit F5=XDB Main Menu  F7=Backup F8=XDB Profile                Ctrl Alt Escape
```



```
                         GCOS-8 COBOL Workbench                         Help9577
                     INTEREL/XDB Database Utility Menu

  The INTEREL/XDB Database Menu contains facilities for import/creation of
  RFM SQL databases, browsing and maintenance, backup and restore, import
  and export (load/unload) of database content, and various configuration
  and other utilities available with XDB-DB2 Workbench.

    Alt-F2 - XDB Batch Declaration Generation
    Alt-F4 - Editor (edit command files)
    Alt-F5 - XDB Import/Export Utility
    Alt-F6 - XDB/G8WB Command file migration
   Ctrl-F2 - XDB Configuration
   Ctrl-F3 - XDB Server configuration
   Ctrl-F4 - XDB LOCATION Maintenance
   Ctrl-F8 - XDB Add User Utility

    Alt-F9 - G8WB Configuration menu (incl. INTEREL config)


               press F1 for more help or space bar to return
INTEREL/XDB-Database-Utilities------------------------------------------------
F1=Help F2=XDB Data Entry F3=XDB Interactive SQL
F4=Edit F5=XDB Main Menu  F7=Backup F8=XDB Profile                Ctrl Alt Escape
```

## SQL DDL

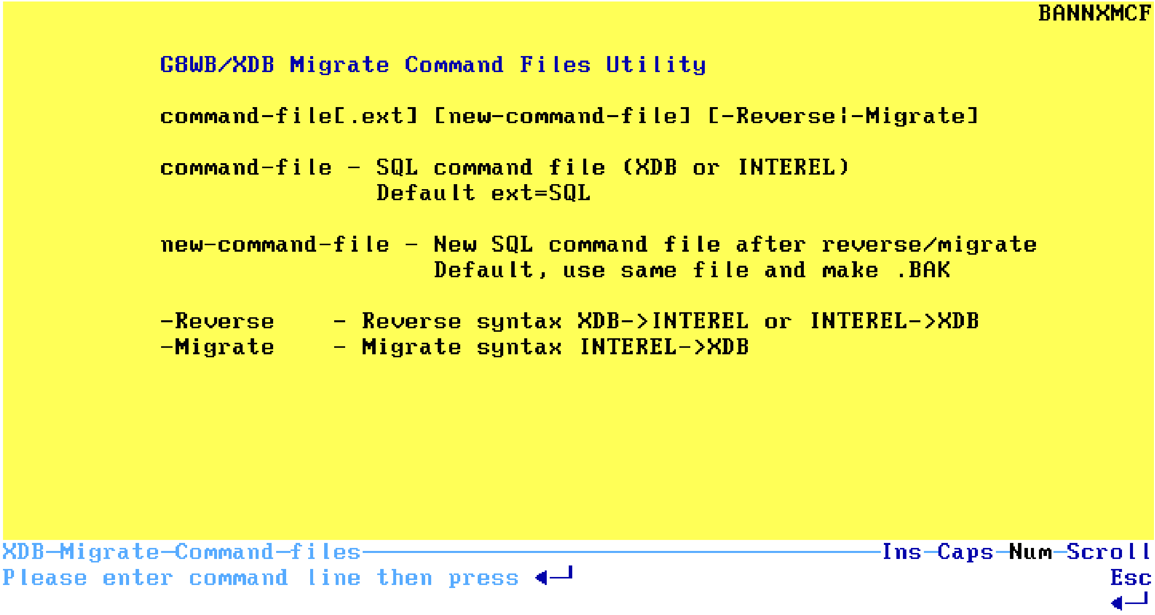
Before working with an INTEREL database, you first need to create a database Location through XDB Location Maintenance Utility. This activity is similar to CREATE MODEL on GCOS-8, however, XDB use another definition of Model. When working with G8WB INTEREL, keep in mind - an XDB Location is equivalent to an INTEREL Model.

For an existing INTEREL database, you need to transfer the SQL DDL syntax from GCOS-8 and place the file in your application directory with the extension .SQL. You could use the Model for naming, i.e. <model-name>.SQL. The application directory, which could be your current path, must be defined via XDB Profile Utility, otherwise you need to give a full file specification when referencing your SQL command file.

For a new INTEREL database, you can start building your database through XDB Interactive SQL or use Editor to build and SQL DDL syntax file.

## Migrate SQL

INTEREL SQL is not completely compatible with XDB, and SQL command files from GCOS-8 require migration before they can be used succesfully by XDB. G8WB provides the Migrate Utility to assist you with this.



```
                                                                      BANNXMCF

          G8WB/XDB Migrate Command Files Utility

          command-file[.ext] [new-command-file] [-Reverse|-Migrate]

          command-file - SQL command file (XDB or INTEREL)
                         Default ext=SQL

          new-command-file - New SQL command file after reverse/migrate
                             Default, use same file and make .BAK

          -Reverse    - Reverse syntax XDB->INTEREL or INTEREL->XDB
          -Migrate    - Migrate syntax INTEREL->XDB

XDB-Migrate-Command-files-----------------------------------Ins-Caps-Num-Scroll
Please enter command line then press ◄┘                                     Esc
                                                                             ◄┘
```

-Migration funtion will remove or adapt INITEREL SQL verbs and commands. The original syntax will be preserved, by turning it into a comment. Once migrated, the SQL command file may be input to XDB either via Interactive SQL or Batch SQL utility. Following illustrate this migration technique:

Original INTEREL SQL command file:

```
create table felanm
   (felnr          dec(5)        not null,
    reserv          char(8)       not null)
   volume (2000);

   grant select, insert, update, delete on felanm for tp8sys;

   done;
```

Migrated XDB SQL command file:

```
create table felanm
   (felnr          dec(5)        not null,
    reserv          char(8)       not null)
   -- INTEREL volume (2000);
   ; -- XDB

   -- INTEREL grant select, insert, update, delete on felanm for tp8sys;

   -- INTEREL done;
```

-Reverse function may be used to re-activate such comments again, so the SQL command file can be transferred to GCOS-8. Another reverse will make the SQL command file XDB compatible again.

If you manually edit SQL command files, you should use the same comment technique, so you are able to use Reverse function, whenever you require to use the command file either on GCOS-8 or within G8WB. The comment rules are:

- -- INTEREL before any other syntax indicate the syntax line is original INTEREL syntax, which is currently deactive
- -- XDB after other syntax indicate the syntax is original XDB syntax, which is currently active
- -- INTEREL after any other syntax indicate the syntax line is original INTEREL syntax, which is currently active
- -- XDB before any other syntax indicate the syntax is original XDB syntax, which is currently deactive

Please refer to Guide Lines chapter for details on compatibility.

# XDB Utilities

Please refer to XDB documentation for details.

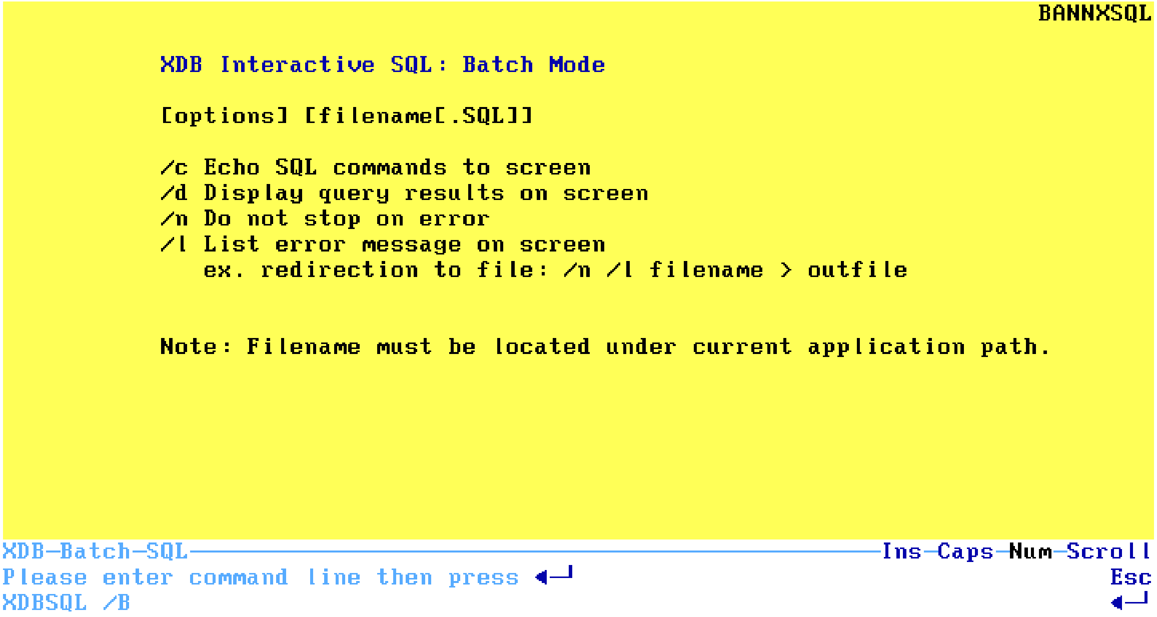
## XDB Interactive SQL

XDB Interactive SQL, available through F3, provides similar functionality to INTEREL INFOEDGE, however, the user interface is quite different.

Unlike INFOEDGE, you need not or cannot, identify the Model/Location when you start any of the XDB utilities. Instead you need to make your Model/Location CURRENT, before invoking XDB utilities. This is done through XDB Profile utility.

## XDB SQL Batch

XDB SQL Batch, available through Alt-F3, allows you to execute SQL command files.



```
                                                                        BANNXSQL

          XDB Interactive SQL: Batch Mode

          [options] [filename[.SQL]]

          /c Echo SQL commands to screen
          /d Display query results on screen
          /n Do not stop on error
          /l List error message on screen
             ex. redirection to file: /n /l filename > outfile


          Note: Filename must be located under current application path.






XDB-Batch-SQL─────────────────────────────────────────────────────Ins-Caps-Num-Scroll
Please enter command line then press ◄┘                                           Esc
XDBSQL /B                                                                          ◄┘
```

In

## XDB Data Entry

XDB Data Entry, available through F2, is a powerful database browsing and maintenance utility.

## XDB Profile

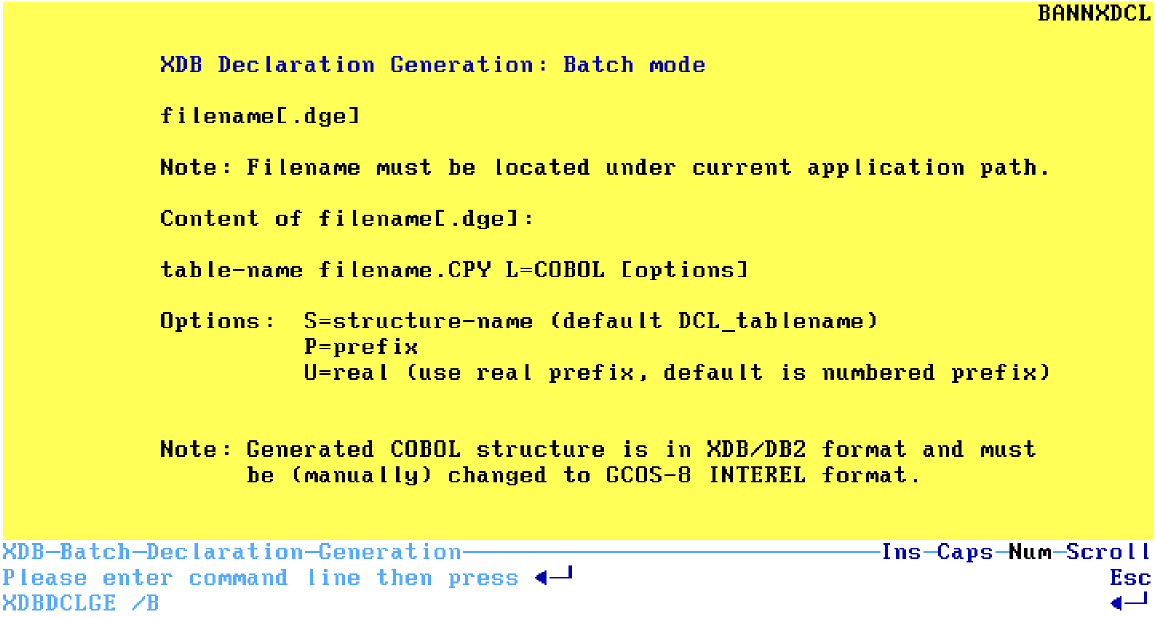
XDB Profile, available through F8, may be used to change/define your current Location and application directory.

## XDB Location Maintenance

XDB Location Maintenance, available through Ctrl-F4, may used to create/delete XDB Locations, i.e INTEREL Models.

## XDB Declaration Generation

XDB Declaration Generation, available through Alt-F2, allows you to generate COBOL declaration syntax files. These files are not INTEREL compatible, however they may be used for dynamic reference via the preprocessor COPY Map file feature, ref. Configuration chapter for details.



```
                                                                  BANNXDCL

         XDB Declaration Generation: Batch mode

         filename[.dge]

         Note: Filename must be located under current application path.

         Content of filename[.dge]:

         table-name filename.CPY L=COBOL [options]

         Options:  S=structure-name (default DCL_tablename)
                   P=prefix
                   U=real (use real prefix, default is numbered prefix)


         Note: Generated COBOL structure is in XDB/DB2 format and must
               be (manually) changed to GCOS-8 INTEREL format.

XDB-Batch-Declaration-Generation-----------------------------Ins-Caps-Num-Scroll
Please enter command line then press ◄┘                                      Esc
XDBDCLGE /B                                                                   ◄┘
```

## Backup/Restore Database

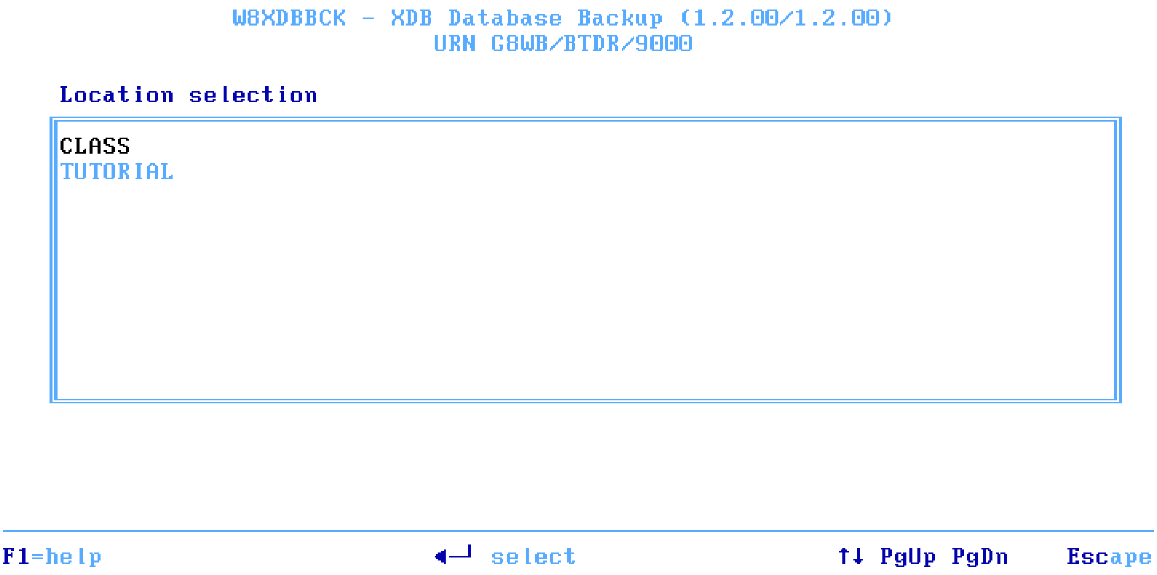
The G8WB Backup/Restore Database Utilities may be used for flexible database version manipulation. Backup/Restore works on the basis of Backup-Sets, which are identified by a informative keyword. Ref. Configuration chapter for details on backup-set configuration.

Backup/Restore of an INTEREL database, is a complete backup/restore of the XDB Location structure with database model definition, i.e. tables, columns, indexes, and the database content. Backup/Restore is performed either with CrossePAC or PKZIP compression software.

You may prepare an INTEREL database and then use Backup/Restore with shared backup-sets for distribution, however, although the backup-set will contain the entire XDB location structure, you may ONLY perform restore into your XDB environment, if the location is already known to XDB, i.e Create Location.

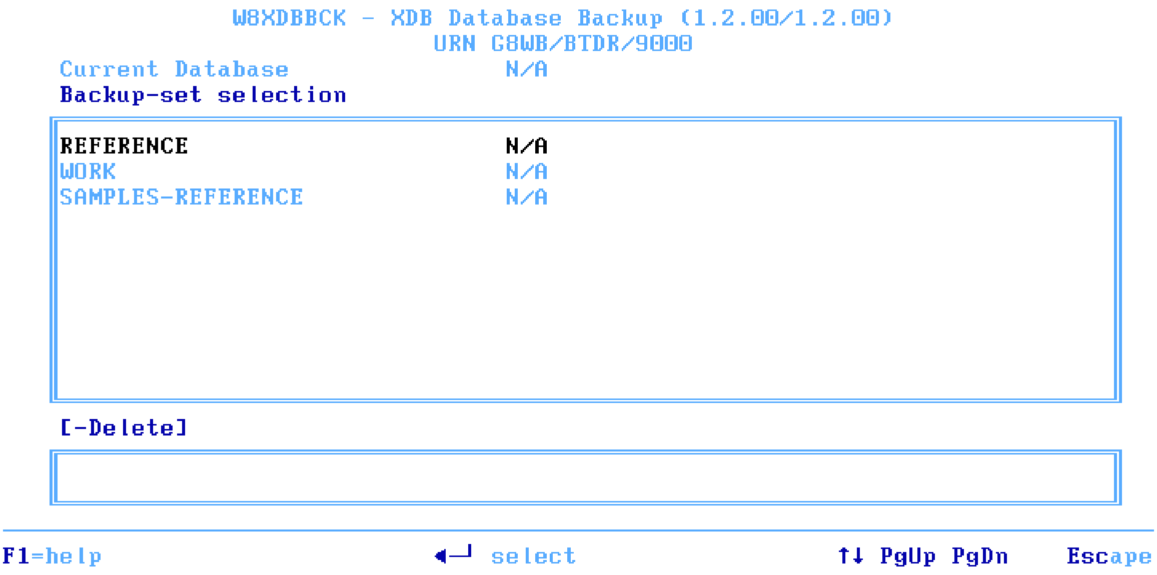
## Backup Database

You select the Location, and thereby database that you wish to backup, from the list of configured locations.



```
           W8XDBBCK - XDB Database Backup (1.2.00/1.2.00)
                         URN G8WB/BTDR/9000

    Location selection
    CLASS
    TUTORIAL

F1=help                ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

Backup Utility will provide you with status information on your current database and give you a list of the configured backup-sets with status information. The time stamp information is taken from the physical database file(s), so the time stamp on the backup-set does NOT reflect the time of backup, but the time-stamp of the physical database file(s) at the time of backup.

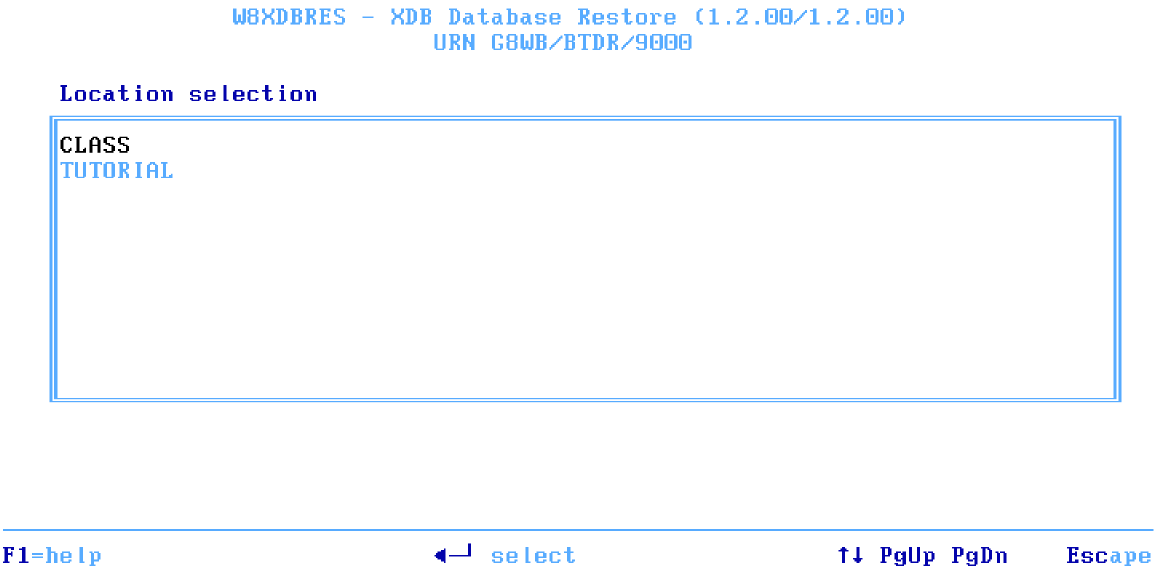


```
           W8XDBBCK - XDB Database Backup (1.2.00/1.2.00)
                         URN G8WB/BTDR/9000
    Current Database                N/A
    Backup-set selection
    REFERENCE                       N/A
    WORK                            N/A
    SAMPLES-REFERENCE               N/A

    [-Delete]

F1=help                ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

-D(elete), may be used to delete a backup-set, instead of performing a backup.

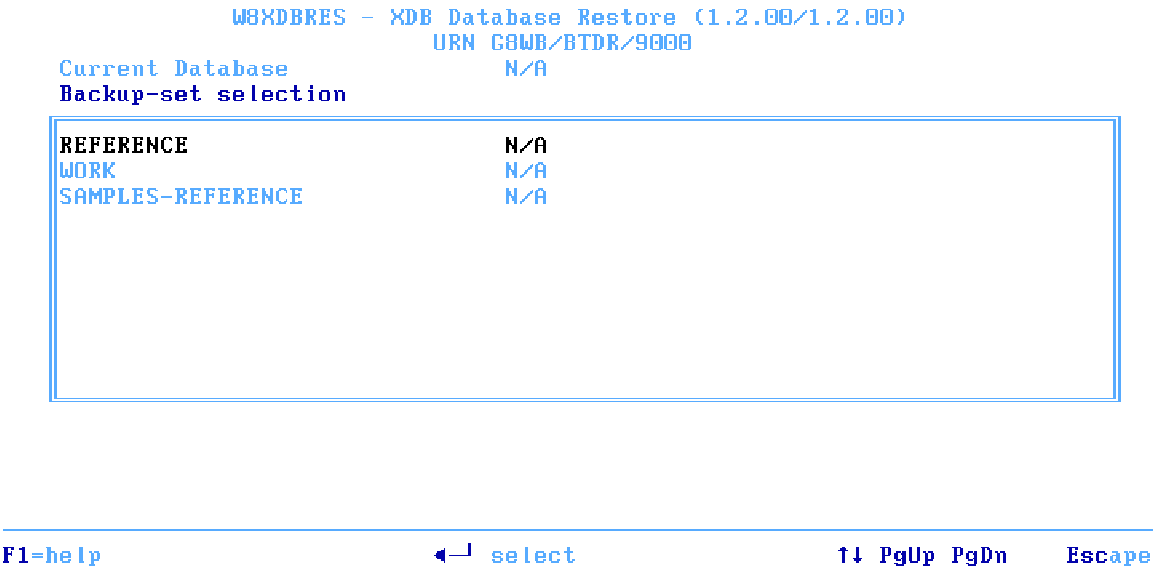
## Restore Database

You select the Location, and thereby database that you wish to restore, from the list of configured locations.



```
              W8XDBRES - XDB Database Restore (1.2.00/1.2.00)
                           URN C8WB/BTDR/9000

    Location selection

    CLASS
    TUTORIAL



F1=help                     ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

Restore Utility will provide you with status information on your current database and give you a list of the configured backup-sets with status information. The time stamp information is taken from the physical database file(s), so the time stamp on the backup-set does NOT reflect the time of backup, but the time-stamp of the physical database file(s) at the time of backup.



```
              W8XDBRES - XDB Database Restore (1.2.00/1.2.00)
                           URN C8WB/BTDR/9000
    Current Database                N/A
    Backup-set selection

    REFERENCE                       N/A
    WORK                            N/A
    SAMPLES-REFERENCE               N/A



F1=help                     ◄┘ select                  ↑↓ PgUp PgDn    Escape
```

# Host Database Unload

Existing GCOS-8 INTEREL test databases may be unloaded, transferred and loaded into G8WB environment. Likewise G8WB test databases may be unloaded, transferred and loaded into GCOS-8 INTEREL environment.

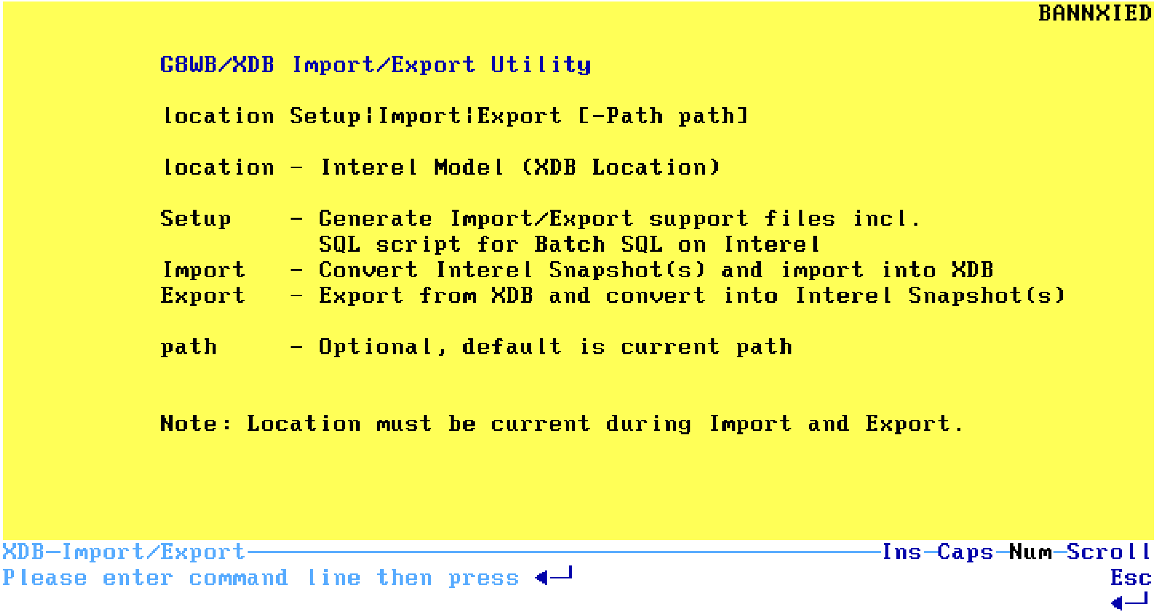
G8WB Import/Export utility work on the basis of the XDB systems tables, so you first need to create your database Location (Model) and define all database entities, i.e. loading SQL command file from GCOS-8.

Unload of GCOS-8 INTEREL databases is done in two steps:

- SETUP, generate command files for host unload
- IMPORT, convert and load database

Upload of GCOS-8 INTEREL databases is done in two steps:

- EXPORT, unload and convert database, and generate command files for host load
- IMPORT, load database on host



```
                                                                    BANNXIED

          G8WB/XDB Import/Export Utility

          location Setup|Import|Export [-Path path]

          location - Interel Model (XDB Location)

          Setup    - Generate Import/Export support files incl.
                     SQL script for Batch SQL on Interel
          Import   - Convert Interel Snapshot(s) and import into XDB
          Export   - Export from XDB and convert into Interel Snapshot(s)

          path     - Optional, default is current path


          Note: Location must be current during Import and Export.



XDB-Import/Export──────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                    Esc
                                                                            ◄┘
```

## Setup

Setup function will read XDB system tables for the specified Location (Model) and generate the following files:

**<model>.IQL**

An SQL command file containing Select/Save for all tables within the INTEREL database Model. This file may be used via INFOEDGE or Batch SQL to unload the INTEREL database. To ensure save files are unique and avoid any naming problems, each Save file is named TBL<nnnn>.IDA, where <nnnn> is the sequence number of the table within the model (sorted on table name). By default Model logon and Save commands are unqualified, i.e. only name is referenced. You may configure both Model and Save catalog-string through INTEREL.CFG, so Model logon and Save commands use fully qualified GCOS-8 catalog-file-strings. Refer to Configuration chapter for details.

**<model>.CPL**

If you have CrossePAC for GCOS-8 (G8WB Add-on option), you may use this CrossePAC Index file to combine and compress all Save file into a CrossePAC archive. The archive is a text file, which you are able to transfer (non-binary) to the G8WB environment. CrossePAC will reduce the file size by 70-90%. Using CrossePAC you need to do the following:

```
CPAC A /<model>.IPA X=/<model>.CPL
```

The presence of a CrossePAC archive is automatically detected by the Import function, and Save files are automatically extracted before conversion and import.

If you do not have CrossePAC for GCOS8, you need to transfer (non-binary) each of the Save files to the G8WB environment.

## Import

Import function will read XDB system tables for the specified Location (Model) to determine which tables and save files are involved. If any save files are missing, these will be ignored during conversion and XDB import.

You may optionally limit the import to one table only, using -Table <table-name>.

Import will convert each Save file into XDB Free Format ASCII files. The converted files are named TBL<nnnn>.XDA. During conversion any Character-Set-Adjust setting will be applied for “CHAR” data type fields. Import will next build an XDB import command file, named <model>.IMP, and then invoke XDB Batch Import, which will load each table from the corresponding INTEREL Save file.

*Note: Import will REPLACE ALL EXISTING data in a table.*

## Export

Export function will read XDB system tables for the specified Location (Model) to determine which tables and save files are involved.

You may optionally limit the export to one table only, using -Table <table-name>.

Export will build an XDB export command file, named <model>.EXP, and then invoke XDB Batch Export, which will unload each table into a separate Free Format ASCII file. The unload files are named TBL<nnnn>.XDA.

Export will next convert each XDB unload file into the corresponding INTEREL Save file. The converted files are named TBL<nnnn>.IDA. During conversion any Character-Set-Adjust setting will be applied for “CHAR” data type fields.

Finally Export will generate the following file:

**<model>.XQL**

An SQL command file containing Delete/Input for all tables (or a specific table) within the INTEREL database Model. This file may be used via INFOEDGE or Batch SQL to load the INTEREL database. To ensure save files are unique and avoid any naming problems, each Save file is named TBL<nnnn>.IDA, where <nnnn> is the sequence number of the table within the model (sorted on table name). By default Model logon and Input commands are unqualified, i.e. only name is referenced. You may configure both Model and Save catalog-string through INTEREL.CFG, so Model logon and Input commands use fully qualified GCOS-8 catalog-file-strings. Refer to Configuration chapter for details.

With the CrossePAC option enabled, Export will automatically compress all Save files and the <model>.XQL file, and place the archive under your current path. The file is named <model>.XPA. You need to transfer this file to GCOS-8 (non-BINARY) and extract all files. The CrossePAC command, TSS or batch, would look like:

```
CPAC X /<model>.XPA
```

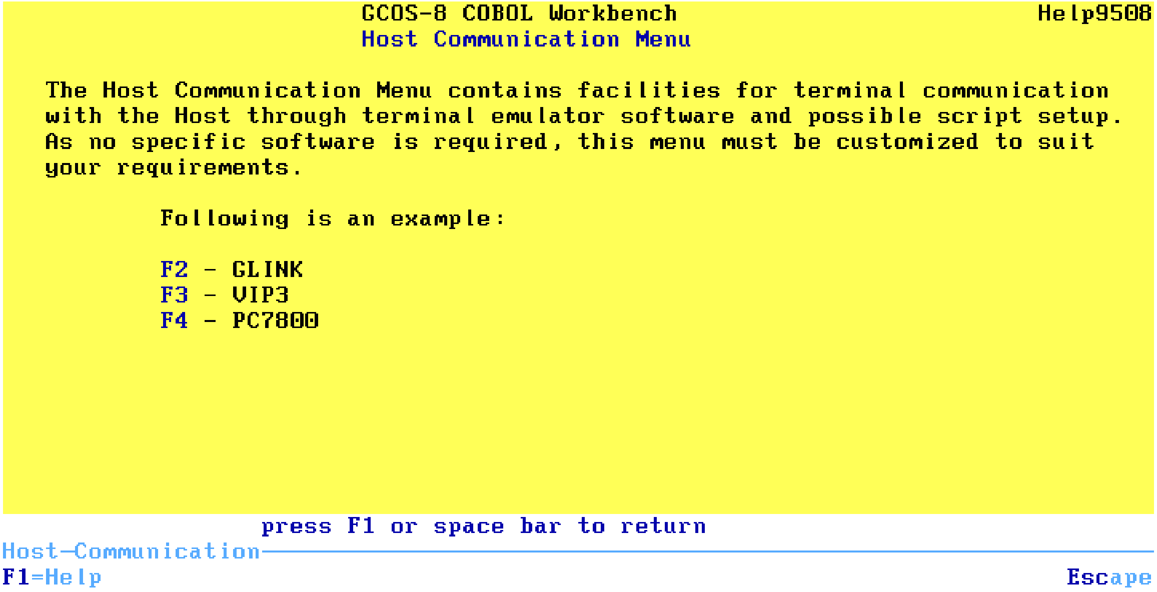
With the CrossePAC option disabled, you need to transfer (non-binary) each of the Save files, as well as the <model>.XQL file to GCOS-8 environemnt.

The <model>.XQL command file may then be used to load each table from the corresponding Save files.

*Note: Load will REPLACE ALL EXISTING data in a table.*

# Chapter 7 HOST MENU

The Host Menu is included in the workbench for practical reasons, as access to the GCOS-8 host may be required occasionally while working within G8WB. No special host communication software is provided with G8WB and you should amend the Host Menu to include invocation of your existing communication software. Please refer to the Configuration chapter for details.



```
                              GCOS-8 COBOL Workbench                          Help9508
                              Host Communication Menu

   The Host Communication Menu contains facilities for terminal communication
   with the Host through terminal emulator software and possible script setup.
   As no specific software is required, this menu must be customized to suit
   your requirements.

           Following is an example:

           F2 - GLINK
           F3 - VIP3
           F4 - PC7800







                      press F1 or space bar to return
Host-Communication──────────────────────────────────────────────────────────────────
F1=Help                                                                      Escape
```

If you are using Windows or OS/2, such functions could also be available elsewhere, i.e. not directly integrated in workbench menu’s.

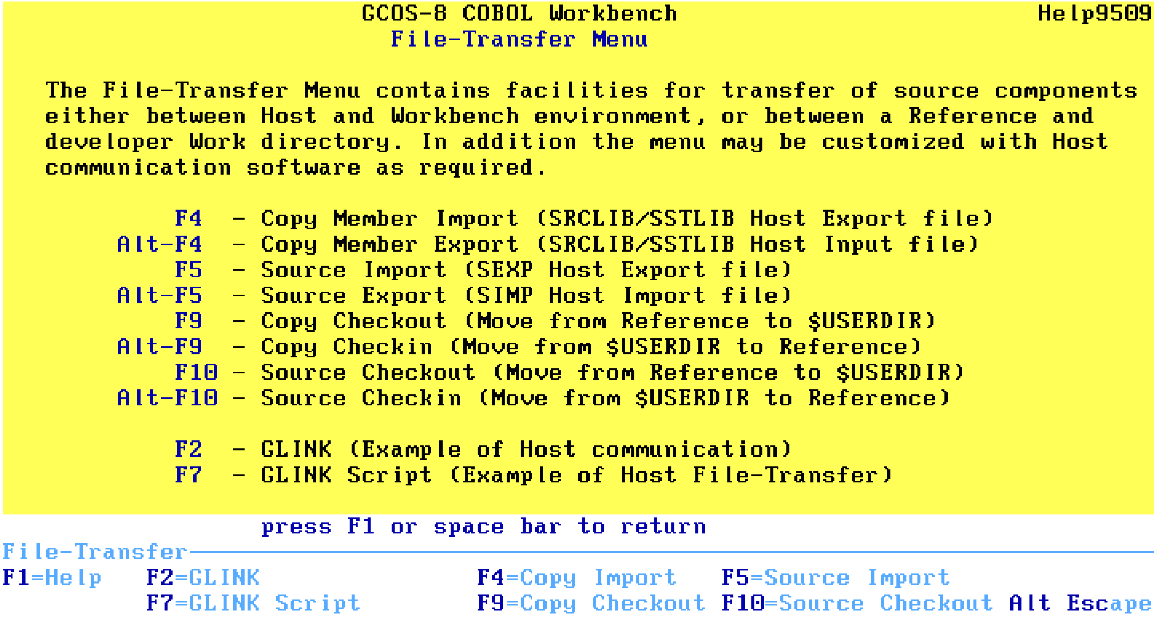
# Chapter 8 FILE TRANSFER MENU

The File Transfer Menu is your place of work when you import or export application components from the GCOS-8 Host environment. As with the Host Menu, no special host communication software is provided with G8WB and you should amend the File Transfer Menu to include invocation of your existing communication software. Please refer to Configuration chapter for details.

G8WB includes utilities to assist you with import and export of application system components to/from the workbench. Following components are involved:

- Copy source
- Program source
- Forms (definition language), refer to Forms chapter for details.

G8WB also includes utilities that will allow you to establish a file maintenance environment, where application components can be checked out for maintaince and later checked in again, once maintenance is completed.



```
                         GCOS-8 COBOL Workbench                        Help9509
                           File-Transfer Menu

   The File-Transfer Menu contains facilities for transfer of source components
   either between Host and Workbench environment, or between a Reference and
   developer Work directory. In addition the menu may be customized with Host
   communication software as required.

            F4   - Copy Member Import (SRCLIB/SSTLIB Host Export file)
        Alt-F4   - Copy Member Export (SRCLIB/SSTLIB Host Input file)
            F5   - Source Import (SEXP Host Export file)
        Alt-F5   - Source Export (SIMP Host Import file)
            F9   - Copy Checkout (Move from Reference to $USERDIR)
        Alt-F9   - Copy Checkin (Move from $USERDIR to Reference)
            F10 - Source Checkout (Move from Reference to $USERDIR)
        Alt-F10 - Source Checkin (Move from $USERDIR to Reference)

            F2   - GLINK (Example of Host communication)
            F7   - GLINK Script (Example of Host File-Transfer)

                  press F1 or space bar to return
File-Transfer─────────────────────────────────────────────────────────────────
F1=Help    F2=GLINK              F4=Copy Import    F5=Source Import
           F7=GLINK Script       F9=Copy Checkout  F10=Source Checkout Alt Escape
```

## File Transfer

When you customize the File Transfer Menu you may define one or more functions for invocation of file transfers. Examples of functions could be:

Simple - Used to transfer any single file

Script - Used to transfer multiple files

If you are using Windows or OS/2, such functions could also be available elsewhere, i.e. not directly integrated in workbench menu’s.

# Import

The File Import Utilities has been designed to assist you and make import of application system components as easy and secure as possible.

We do recommend you batch things and import a complete system or subsystem in one operation. This way you are sure all application components are available within the workbench. The Export Utilities will assist you in determining what to transfer back, once you have completed your development or maintenance work.

## Host Source

Host program sources are normally maintained in individual files. To ease import of multiple program sources, G8WB supports a file with the following format:

**APPEND name catalog-file-string**
**...**
***END***

You may manually generate such a file or use the provided Host Source Export Utility (TEX procedure).

```
TEX CALL /HOST/G8SEXP.TEX
G8SEXP - Host Source Export Utility
Enter cat-file for Program-List
/PROJ.LST
Enter cat-file for Export-File
/PROJ.XFR
...
Export-File /PROJ.XFR ready for transfer
```

The Program-List may be build from a CATA or otherwise edited into following format:

**catalog-file-string [name]**
**...**

Catalog-file-string must be complete and the program source name may optionally be specified, otherwise the file name from catalog-file-string becomes default.

## CrossePAC

With the CrossePAC option you are able to compress files before file transfer. This will reduce the file size by 70-90%. CrossePAC will create a text file archive, which you are able to transfer as a non-binary file.

You can enable CrossePAC use through a switch setting in the G8SEXP TEX procedure.

If you not do not use G8SEXP, you may enable CrossePAC manually or through other automated procedures. The G8WB Source Import Utility will automatically recognize a CrossePAC archive file and decompress the file before importing it. It does, however, assume that the embedded file in the CrossePAC archive file is named CPACWORK.

If, as an example, you were to build a file named SOURCEA.XFR on GCOS-8 for transfer to G8WB, you would instead build a file named CPACWORK and create a CrossePAC archive file named SOURCEA.XFR. The CrossePAC command, TSS or Batch, would look like:

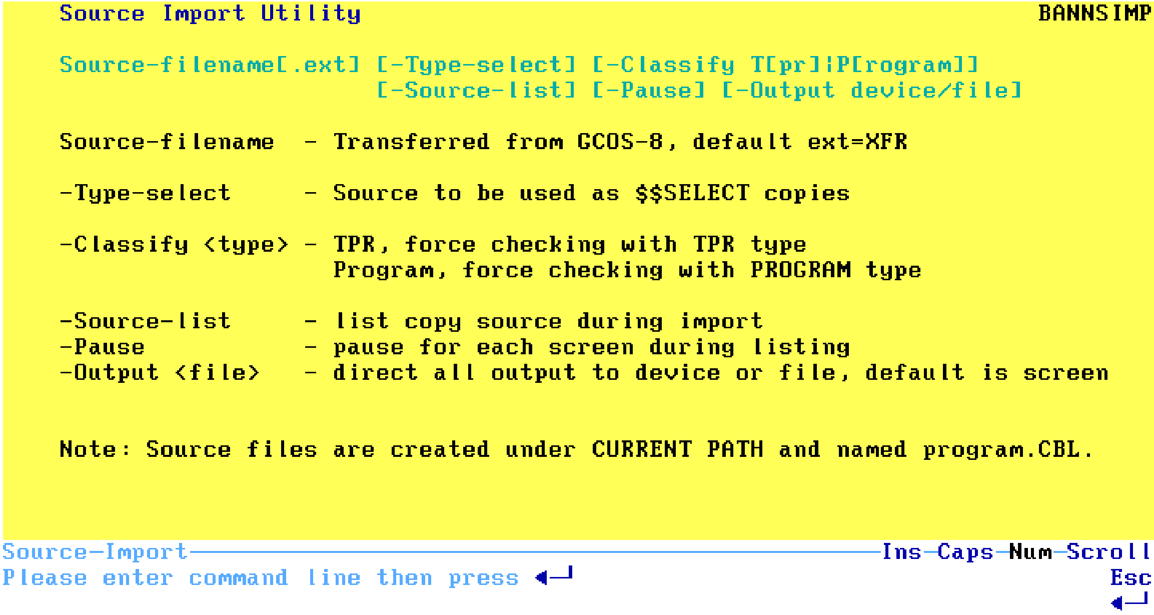
**CPAC A  SP=N  /SOURCEA.XFR  /CPACWORK**

## Source Import

Once you have created the host export file, you need to transfer this into the G8WB environment. The file should be placed in the directory where the individual source files are going to be placed.

Next you invoke Source Import Utility (W8SIMP), which will read the host export file and and split it into separate PC files according to APPEND directives. The PC files are given the extension CBL. W8SIMP will maintain a W8SIMP.LBT control file (create or amend) with information about each source, it's name, host catalog-file-string and a time stamp for the created PC file.

If you import an existing source, W8SIMP will verify whether the source has been changed since last import. If this is the case, W8SIMP will prompt you with a warning, so you do not accidentally override changes made within G8WB environment.



```
    Source Import Utility                                                  BANNSIMP

    Source-filename[.ext] [-Type-select] [-Classify T[pr]|P[rogram]]
                          [-Source-list] [-Pause] [-Output device/file]

    Source-filename   - Transferred from GCOS-8, default ext=XFR

    -Type-select      - Source to be used as $$SELECT copies

    -Classify <type> - TPR, force checking with TPR type
                       Program, force checking with PROGRAM type

    -Source-list      - list copy source during import
    -Pause            - pause for each screen during listing
    -Output <file>    - direct all output to device or file, default is screen


    Note: Source files are created under CURRENT PATH and named program.CBL.



Source-Import──────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                          Esc
                                                                                 ◄┘
```

$$SELECT files are exported from the host as normal source files, however, during import you need to specify the -T(ype-Select) argument. This will cause W8SIMP to create PC files with the extension CPY, which will allow $$SELECT Preprocessor to locate the files. You may need to update Copy Map configuration with information on new GCOS-8 catalog-string. Ref. Configuration chapter for details.

-C(lassify) TPR|PROGRAM, may be used to insert a classification SET directive during import. Ref. Configuration chapter for details.

W8SIMP will automatically insert a comment line with information about time of import, and a COPY SET directive, which will trigger generation of COPY Mapping file. Ref. Configuration chapter for details.

*Note: COBOL line numbers are of little or no use within G8WB. You may configure G8WB, so W8SIMP will remove COBOL line numbers during import. Likewise you may configure, so W8SEXP will re-insert COBOL line numbers during export. Ref. Configuration chapter for details.*

## Host Copy

For copy source no utility is required, as we assume you either have a file containing all copy source in SRCLIB format, or you perform an extract of copy source either with SRCLIB or SSTLIB. Any JCL in the file will be ignored during import and need not be removed.

## CrossePAC

With the CrossePAC option you are able to compress files before file transfer. This will reduce the file size by 70-90%. CrossePAC will create a text file archive, which you are able to transfer as a non-binary file.

You need to enable CrossePAC manually or through other automated procedures. The G8WB Copy Import Utility will automatically recognize a CrossePAC archive file and decompress the file before importing it. It does, however, assume that the embedded file in the CrossePAC archive file is named CPACWORK.

If, as an example, you were to build a file named COPYL1.XFR on GCOS-8 for transfer to G8WB, you would instead build a file named CPACWORK and create a CrossePAC archive file named COPYL1.XFR. The CrossePAC command, TSS or Batch, would look like:

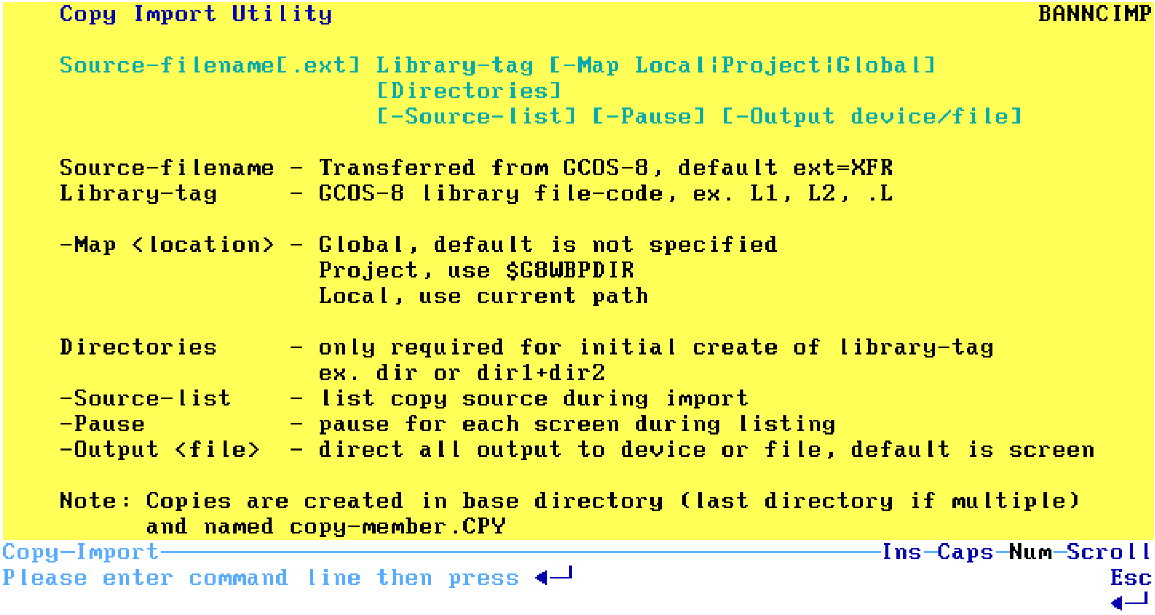
```
CPAC A  SP=N  /COPYL1.XFR /CPACWORK
```

## Copy Import

Once you have created the host export file, you need to transfer this into the G8WB environment. The file should be placed in any directory, however, for practical reasons you may want to place in the directory where the individual copy are going to be placed.

G8WB Preprocessor utilize a Copymap configuration file to process COPY statements. The library qualification (COPY ... [OF library]) is used to logically group copy source, into something similar to a GCOS-8 COPY library. This library qualification is referred to as a Library-Tag, where default is .L like on GCOS-8.

Next you invoke Copy Import Utility (W8CIMP), which will read the host export file and and split it into separate PC files according to APPEND directives. The PC files are given the extension CPY. W8CIMP will maintain a W8CIMP.LBT control file (create or amend) with information about each copy, it's name and a time stamp for the created PC file.



```
    Copy Import Utility                                                 BANNCIMP

    Source-filename[.ext] Library-tag [-Map Local|Project|Global]
                          [Directories]
                          [-Source-list] [-Pause] [-Output device/file]

    Source-filename - Transferred from GCOS-8, default ext=XFR
    Library-tag     - GCOS-8 library file-code, ex. L1, L2, .L

    -Map <location> - Global, default is not specified
                      Project, use $G8WBPDIR
                      Local, use current path

    Directories     - only required for initial create of library-tag
                      ex. dir or dir1+dir2
    -Source-list    - list copy source during import
    -Pause          - pause for each screen during listing
    -Output <file>  - direct all output to device or file, default is screen

    Note: Copies are created in base directory (last directory if multiple)
          and named copy-member.CPY
Copy-Import──────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                      Esc
                                                                              ◄┘
```

Library-tag, must be equal to the first two characters of the library qualification (similar to file-code technique used by GCOS-8).

Directories, specifies the directory or directories (separated by +), where the last directory (base) is used for storage of the PC files. This entry is ONLY required the first time copy sources are imported to a Library-Tag. Multiple directories may be used to support development stages..

If you import an existing copy, W8CIMP will verify whether the copy has been changed since last import. If this is the case, W8CIMP will prompt you with a warning, so you do not accidentally override changes made within G8WB environment.

Copy member may use an identification of up to 30 characters, however, G8WB cannot create a PC file with a name greater than 8 characters, plus the CPY extension. To solve such potential problem, W8CIMP will assign a reference file name for any identification which exceed 8 characters. This information, as well as creation library-tag and associated directory specification the first time you import, is automatically update in the Copymap configuration during import. Ref. Configuration chapter for further details on Copymap.

-M(ap) GLOBAL|PROJECT|LOCAL, is optional and like Directory only used the first time you import to a Library-Tag. Normally only one COPYMAP.CFG is required (Global), however if Library-Tag's are not unique, i.e. more than one project use the same Library-Tag, you need multiple COPYMAP.CFG files. The MAP argument may be used to direct creation of the Library-Tag into either a PROJECT or LOCAL (workstation level) COPYMAP.CFG. Ref. Configuration for more details on Copymap.

Use of multiple directories allows you to maintain working versions of copy members in other than the base directory - copied from base when required. G8WB preprocessor will search directories left-to-right for the copy member and use the first available version.

*Note: Each Library-Tag must be placed in a separate directory to avoid conflicts between member naming.*

*Note: COBOL line numbers are of little or no use within G8WB. You may configure G8WB, so W8CIMP will remove COBOL line numbers during import. Likewise you may configure, so W8CEXP will re-insert COBOL line numbers during export. Ref. Configuration chapter for details.*

# File Export

The File Export Utilities has been designed to assist you and to make export of changed application system components as easy and secure as possible.

The File Export Utilities rely on information in the W8SIMP/W8CIMP.LBT files, which are automatically updated  during Import. The default action for export is to export only changed application system components, however, you may also request export of components regardless of whether they have been changed or not.

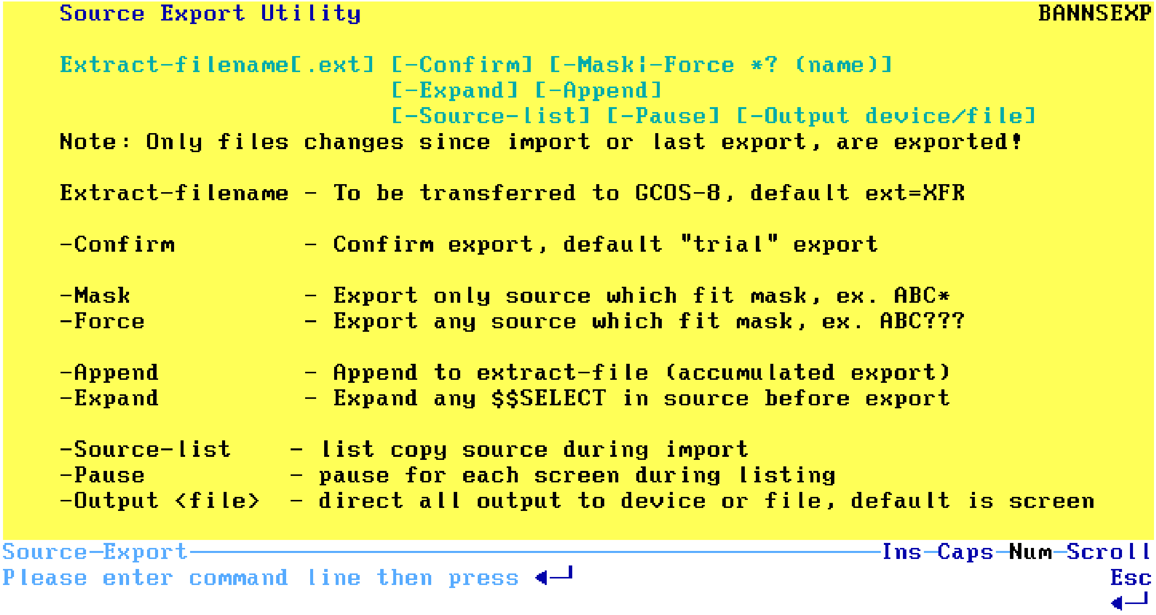
*Note: The CHANGE condition is determined from the OS time stamp on the file. This time stamp does not change when you copy a file, but it does change when you write back into the file, i.e. Save in MF Editor.*

## Source Export

Source Export Utility (W8SEXP) will collect source files based on the W8SIMP.LBT file in current directory, and combine these into a single export file for transfer to GCOS-8. Each source file is preceded by an APPEND directive with source filename and the host catalog-file-string, as specified during import. By default, W8SEXP will export only CHANGED source files.

New source files must be registered for export via Source Create Utility, ref. TP development chapter for details.

-C(onfirm), will make W8SEXP update the time stamp information for source files being exported. If you do not specify Confirm, W8SEXP will export the same files again the next time you export. Export without Confirm may be used to perform a “trial” export.



```
    Source Export Utility                                                   BANNSEXP

    Extract-filename[.ext] [-Confirm] [-Mask¦-Force *? (name)]
                           [-Expand] [-Append]
                           [-Source-list] [-Pause] [-Output device/file]
    Note: Only files changes since import or last export, are exported!

    Extract-filename - To be transferred to GCOS-8, default ext=XFR

    -Confirm          - Confirm export, default "trial" export

    -Mask             - Export only source which fit mask, ex. ABC*
    -Force            - Export any source which fit mask, ex. ABC???

    -Append           - Append to extract-file (accumulated export)
    -Expand           - Expand any $$SELECT in source before export

    -Source-list     -  list copy source during import
    -Pause           -  pause for each screen during listing
    -Output <file>   -  direct all output to device or file, default is screen
Source-Export────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                      Esc
                                                                              ◄┘
```

-F(orce) <mask>, may be used to force export of source files, even though these have not been changed. The <mask>, which may contain * and ? mask character, define which source files to export, i.e. ABC*, ABC??? or * for all files.

-M(ask) <mask>, works like Force, except W8SEXP will only export CHANGED files that fit the given <mask>.

-A(ppend), may be used to append to an existing export file. This may be relevant if you use Force or Mask with different <mask> settings, i.e. accumulated export.

-E(xpand), is only relevant if the source contain $$SELECT statements. With Expand, $$SELECT statements are expanded before the source file is exported.

Export makes no distinction between normal source and $$SELECT source files, which have been imported. If you use Expand, you should not import both normal and $$SELECT into the same directory, as you would then export the $$SELECT mutiple times, i.e. explicit and embedded.

If CrossePAC is enabled, the export file will automatically be compressed.

Once you have created the export file, you need to transfer the file to GCOS-8 for import processing. Export files are always text files and may be transferred as non-binary files.

## Host Source

Source Export Utility (W8SEXP) creates an export file containing multiple program sources in the following format:

**APPEND name catalog-file-string**
**...**
***END***

You may manually process such a file or use the provided Host Source Import Utility (TEX procedure).

```
TEX CALL /G8SIMP.TEX
G8SEXP - Host Source Import Utility
Enter cat-file for Import-File
/PROJ.XFR
...
Importing ...
```

Each program source or $$SELECT source is extracted and re-saved into the designated host catalog-file-string.

*Note: There is no time stamp checking during host import.*

## CrossePAC

G8SIMP will treat the export file as a CrossePAC archive and decompress it before use, if you have enabled this through a switch setting in the TEX procedure.

If you do not use G8SIMP, you need to enable CrossePAC manually or through other automated procedures. The W8SEXP will always name the embedded file CPACWORK, within the CrossePAC archive file.

If, as an example, you have used W8SEXP to build an export file named SOURCEA.XFR, then you would need to extract the CPACWORK from SOURCEA.XFR and use CPACWORK for processing during import. The CrossePAC command, TSS or Batch, would look like:

**CPAC X /SOURCEA.XFR**

## Copy Export

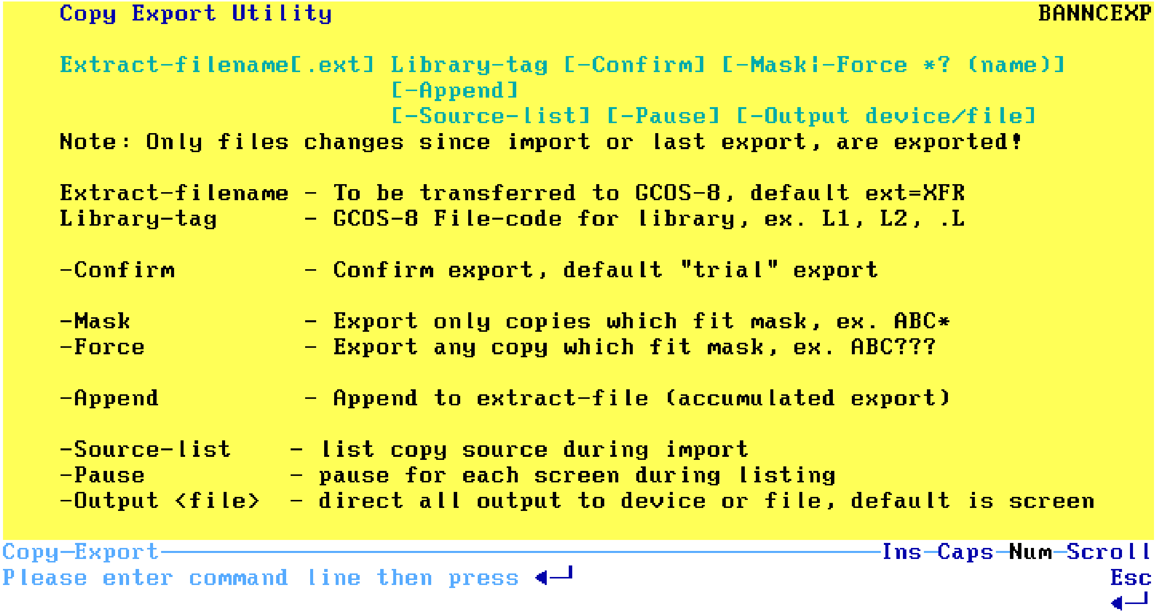
Copy Export Utility (W8CEXP) will collect copy files for a given Library-Tag and combine these into a single export file for transfer to GCOS-8. Each copy file is preceded by a DELETE and APPEND directive with the original copy name, as specified during import. By default, W8CEXP will export only CHANGED copy files.

New copy files must be registered for export via Copy Create Utility, ref. TP development chapter for details. New copy files only have APPEND directive.

Library-Tag, define which copy library to export from. Directory information is derived from the Copymap configuration.

*Note: Export will only take place from base directory, regardless of the existence of other copy member versions in other directories.*

-C(onfirm), will make W8CEXP update the time stamp information for copy files being exported. If you do not specify Confirm, W8CEXP will export the same files again the next time you export. Export without Confirm may be used to perform a “trial” export.



```
    Copy Export Utility                                                  BANNCEXP

    Extract-filename[.ext] Library-tag [-Confirm] [-Mask|-Force *? (name)]
                           [-Append]
                           [-Source-list] [-Pause] [-Output device/file]
    Note: Only files changes since import or last export, are exported!

    Extract-filename - To be transferred to GCOS-8, default ext=XFR
    Library-tag      - GCOS-8 File-code for library, ex. L1, L2, .L

    -Confirm         - Confirm export, default "trial" export

    -Mask            - Export only copies which fit mask, ex. ABC*
    -Force           - Export any copy which fit mask, ex. ABC???

    -Append          - Append to extract-file (accumulated export)

    -Source-list    - list copy source during import
    -Pause          - pause for each screen during listing
    -Output <file>  - direct all output to device or file, default is screen
Copy-Export──────────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                          Esc
                                                                                 ◄┘
```

-F(orce) <mask>, may be used to force export of copy files, even though these have not been changed. The <mask>, which may contain * and ? mask character, define which copy files to export, i.e. ABC*, ABC??? or * for all files.

-M(ask) <mask>, works like Force, except W8CEXP will only export CHANGED files that fit the given <mask>.

-A(ppend), may be used to append to an existing export file. This may be relevant if you use Force or Mask with different <mask> settings, i.e. accumulated export.

If CrossePAC is enabled, the export file will automatically be compressed.

Once you have created the export file, you need to transfer the file to GCOS-8 for import processing. Export files are always text files and may be transferred as non-binary files.

## Host Copy

Copy Export Utility (W8CEXP) creates an export file containing multiple copy sources in the following standard SRCLIB format:

**DELETE name**<br>
**APPEND name**<br>
**...**<br>
***END***

The export file may be used as input to SRCLIB or SSTLIB for update of your host copy library.

*Note: If you use SSTLIB, the SSTLIB setting will make W8CEXP insert UPDATE directives instead of DELETE/APPEND.*

## CrossePAC

If you have CrossePAC enabled, W8CEXP will automatically compress the export file and you need to decompress the file before use by SRCLIB or SSTLIB. W8CEXP will always name the embedded file CPACWORK, within the CrossePAC archive file.

If, as an example, you have used W8CEXP to build an export file named COPYL1.XFR, then you would need to extract the CPACWORK from COPYL1.XFR and use CPACWORK as input to SRCLIB or SSTLIB. The CrossePAC command, TSS or Batch, would look like:

```
CPAC X  /COPYL1.XFR
```

# Maintenance

The Maintenance Utilities has been designed to assist you and to make maintenance of application components as easy and secure as possible.

G8WB maintenance is based on a relatively simple Checkout/Checkin concept, which will provide you with a two staged environment, with security features to prevent concurrent update of application components. The two stages are referred to as:

- User Directory or Unit test

- Reference directory or System test

Maintenance or development work ongoing in the User Directory will only affect the developer or developers, which currently work within the User Directory. All other developers will continue to work with application components from the Reference Directory.

You may implement further stages as required to complement G8WB maintenance. Refer to User Guide Getting organized for further details.

## User Directory

User directory is controlled by $USERDIR setting, which may either define a local (non-shared) directory or a global (shared) directory.

Local setting, which is limited to one workstation, is typically used for maintenance or new development done by a single developer. In principle, USERDIR and G8WBLDIR would be similar, although they may not have same local directory setting.

Global setting, which allow multiple workstations, is typically used for maintenance or new development done by a team of developers, but not all developers within an organization. USERDIR setting would the be a global directory, designated for the project/team.

Actually you should always try to use a setting with an informative directory name, i.e. C:\WORK is not informative, whereas C:\JOHNDOE or E:\ACCOUNT\REL5 are.

User Directory may be changed dynamically, either through different workbench startup command files, or via Project object settings in GUI workbench. This way a developer may concurrently work on multiple projects, and easily change between projects as required.

## Reference Directory

Reference directory is the common base where application components are retrieved from. A reference directory is by definition shared. There is no generic setting for reference directory, i.e. no $REFDIR, as different techniques apply for program and copy source.

Use of reference directories is greatly simplified if you use symbolic instead of explicit references to directories.

As an example, you have placed all program source components into E:\SYSTEM\SOURCE directory on a File Server, and you have placed all copy source components into F:\SYSTEM\COPY\L1 and F:\SYSTEM\COPY\F2 on a File Server.

To use symbolic references, you could use $SOURCE, $COPYL1 and $COPYL2, instead of the explicit directory references. To do thus, you must define the symbols in your environment configuration as follows:

**SET SOURCE=E:\SYSTEM\SOURCE**
**SET COPYL1=F:\SYSTEM\COPY\L1**
**SET COPYL2=F:\SYSTEM\COPY\L2**

Although various LAN software may provide similar facilities, above technique is completely LAN independent and usable with all G8WB utilities, except for XDB.

## COBDIR

COBDIR setting is used during execution to locate application components, i.e. TPR's, programs and modules. In oder to use G8WB Maintenance, you need to define your two stages - User and Reference Directory, in the COBDIR setting, otherwise application components are only found if they are located under your current path.

Using the example from above, your COBDIR could look like:

**SET COBDIR=$USERDIR;$SOURCE;C:\COBOL\LBR;C:\COBOL\EXEDLL**

## COPY Library

When you create COPY libraries, ref. Copy Import above, each library will be identified trough a Library-Tag. For each Library-Tag, a Default-Library setting define, like COBDIR, how to locate application components. For copies, this takes place during checking of a program.

When you initially create the COPY library, the directory setting will probably be an explicit reference to a directory, i.e. F:\SYSTEM\COPY\L1. In order to use G8WB Maintenance, you need to define your two stages - User and Reference Directory, in the library setting, otherwise application components that are checked out cannot be found. Ref. Configuration chapter for details on Copy Map configuration.

Using the examples from above, your Copy Map library definitions could look like:

```
[L1]
DEFAULT-LIBRARY : $USERDIR+$COPYL1
[L2]
DEFAULT-LIBRARY : $USERDIR+$COPYL2
```

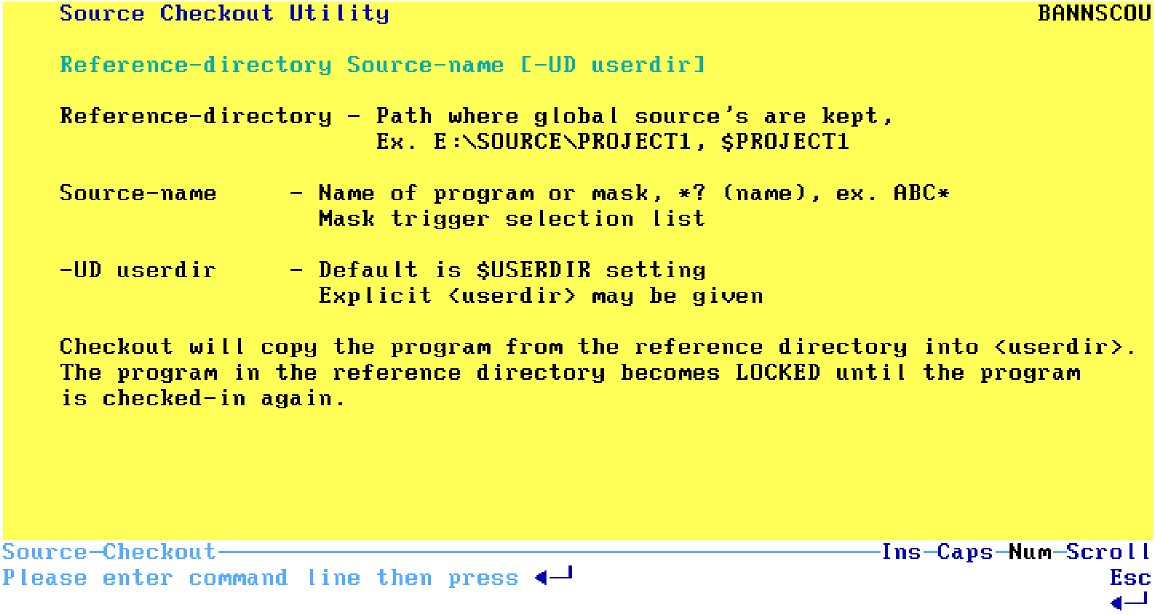
## Source Checkout

Source Checkout Utility (W8SCOUT) may be used to copy a program from the Reference Directory into the User Directory. Only the source file program.CBL is copied. Once copied, a lock file is created in the Reference Directory. This file is named program.LCK and will contain information about the User Directory that has checked out the program.

Any attempt to check out the same program will be refused, however, information is given as to which User Directory currently have the program locked for maintenance.

Reference Directory, define where the program source should be copied from, i.e. $SOURCE as used in example above.

Source-name, define the name of the program or a <mask>, which will trigger a selection list. You may only checkout one program at a time, however you may use a batch file with multiple checkout commands to perform bulk checkouts.



```
    Source Checkout Utility                                                   BANNSCOU

    Reference-directory Source-name [-UD userdir]

    Reference-directory - Path where global source's are kept,
                          Ex. E:\SOURCE\PROJECT1, $PROJECT1

    Source-name     - Name of program or mask, *? (name), ex. ABC*
                      Mask trigger selection list

    -UD userdir     - Default is $USERDIR setting
                      Explicit <userdir> may be given

    Checkout will copy the program from the reference directory into <userdir>.
    The program in the reference directory becomes LOCKED until the program
    is checked-in again.




Source-Checkout──────────────────────────────────────────────────────Ins-Caps-Num-Scroll
Please enter command line then press ◄┘                                                Esc
                                                                                        ◄┘
```

-U(serdir) <userdir>, may be used define User Directory on the fly or change the current USERDIR setting for this checkout operation only.

You are now ready to modify the program, check and test it through G8WB.

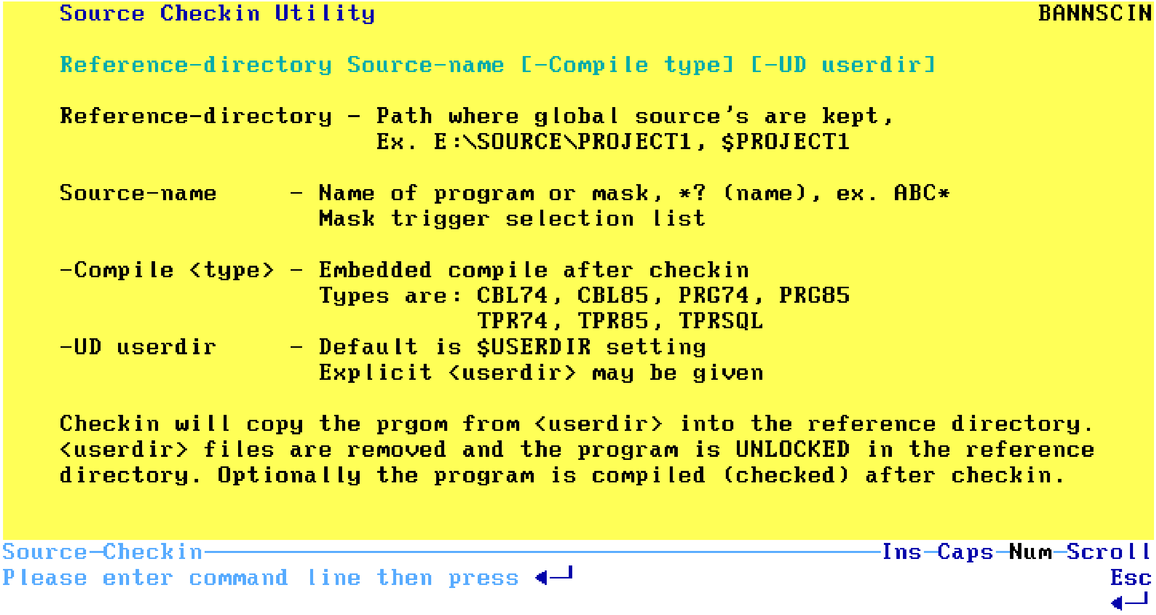
*Note: $$SELECT files are checked out like normal program source.*

## Source Checkin

Source Checkin Utility (W8SCIN) may be used to copy a program back from User Directory to the Reference Directory, after you have finished your testing activities. Only the source file program.CBL is copied, however, you may request renewed checking of the program as part of the checkin activity. Once copied back to Reference Directory, the lock file is removed and the program is open for checkout again.

Reference Directory, define where the program source should be copied to, i.e. $SOURCE as used in example above.

Source-name, define the name of the program or a <mask>, which will trigger a selection list. You may only checkin one program at a time, however you may use a batch file with multiple checkin commands to perform bulk checkins.



```
    Source Checkin Utility                                                 BANNSCIN

    Reference-directory Source-name [-Compile type] [-UD userdir]

    Reference-directory - Path where global source's are kept,
                          Ex. E:\SOURCE\PROJECT1, $PROJECT1

    Source-name      - Name of program or mask, *? (name), ex. ABC*
                       Mask trigger selection list

    -Compile <type> - Embedded compile after checkin
                       Types are: CBL74, CBL85, PRG74, PRG85
                                  TPR74, TPR85, TPRSQL
    -UD userdir      - Default is $USERDIR setting
                       Explicit <userdir> may be given

    Checkin will copy the prgom from <userdir> into the reference directory.
    <userdir> files are removed and the program is UNLOCKED in the reference
    directory. Optionally the program is compiled (checked) after checkin.

Source-Checkin──────────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                          Esc
                                                                                  ◄┘
```

-C(ompile) <type>, define whether and how the program should be checked (compiled) under Reference Directory after checkin. Although optional, it is recommended that you always request compile during checkin, otherwise the previous version of the program will still be current.

-U(serdir) <userdir>, may be used define User Directory on the fly or change the current USERDIR setting for this checkin operation only.

Once checkin is complete, all source and execution files, related to the program, are removed from User Directory.

*Note: $$SELECT files are checked in like normal program source, except you do not request compile.*

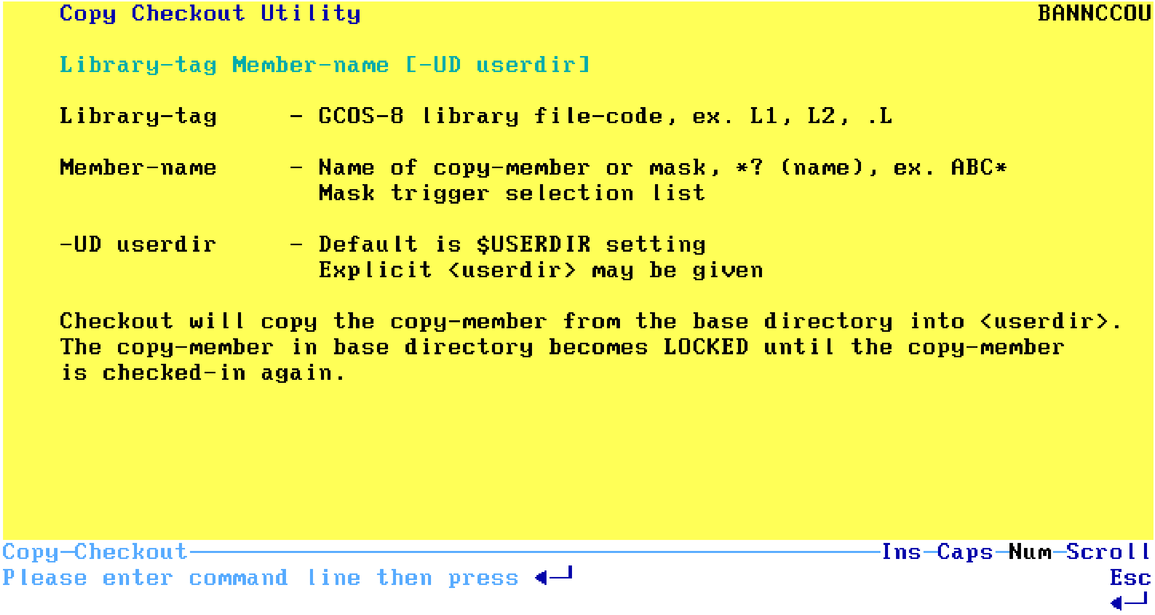
## Copy Checkout

Copy Checkout Utility (W8CCOUT) may be used to copy a copy member from the Reference Directory into the User Directory. Once copied, a lock file is created in the Reference Directory. This file is named copyname.LCK and will contain information about the User Directory that has checked out the copy member.

Any attempt to check out the same copy member will be refused, however, information is given as to which User Directory currently have the copy member locked for maintenance.

Library-Tag, define which COPY library to copy from, i.e. L1 or L2 as used in examples above.

Member-name, define the name of the copy member or a <mask>, which will trigger a selection list. You may only checkout one copy member at a time, however you may use a batch file with multiple checkout commands to perform bulk checkouts.



```
    Copy Checkout Utility                                                    BANNCCOU

    Library-tag Member-name [-UD userdir]

    Library-tag     - GCOS-8 library file-code, ex. L1, L2, .L

    Member-name     - Name of copy-member or mask, *? (name), ex. ABC*
                      Mask trigger selection list

    -UD userdir     - Default is $USERDIR setting
                      Explicit <userdir> may be given

    Checkout will copy the copy-member from the base directory into <userdir>.
    The copy-member in base directory becomes LOCKED until the copy-member
    is checked-in again.
```

-U(serdir) <userdir>, may be used define User Directory on the fly or change the current USERDIR setting for this checkout operation only.

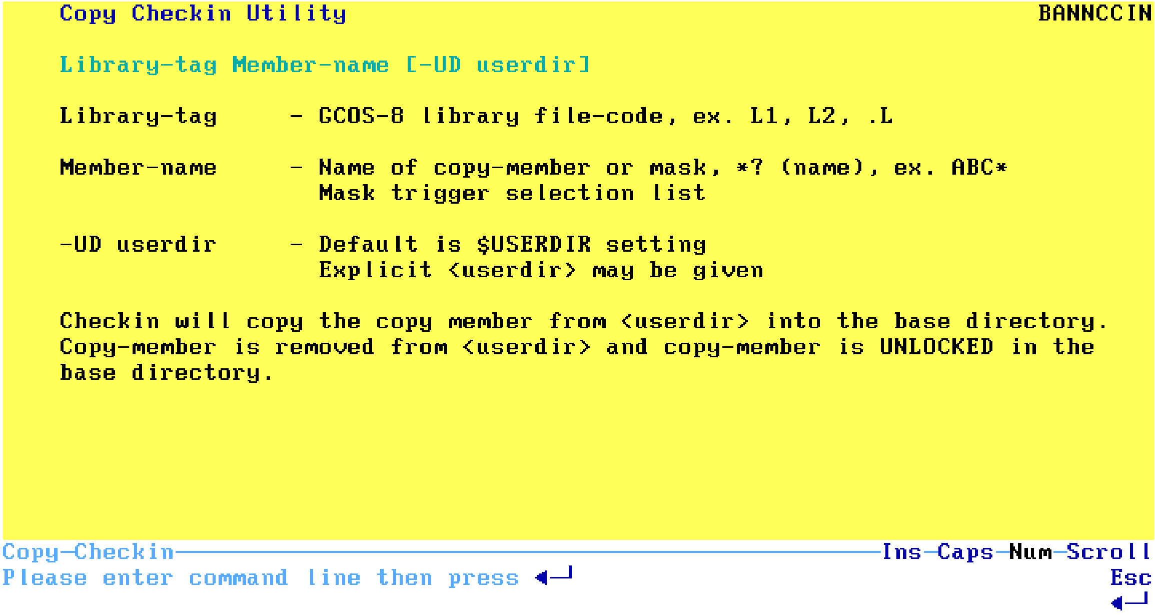
You are now ready to modify the copy member and check/test programs, which use the copy member.

## Copy Checkin

Copy Checkin Utility (W8CCIN) may be used to copy a copy member back from User Directory to the Reference Directory, after you have finished your testing activities. Once copied back to Reference Directory, the lock file is removed and the copy member is open for checkout again.

Library-Tag, define which COPY library to copy to, i.e. L1 or L2 as used in examples above.

Member-name, define the name of the copy member or a <mask>, which will trigger a selection list. You may only checkin one copy member at a time, however you may use a batch file with multiple checkin commands to perform bulk checkins.



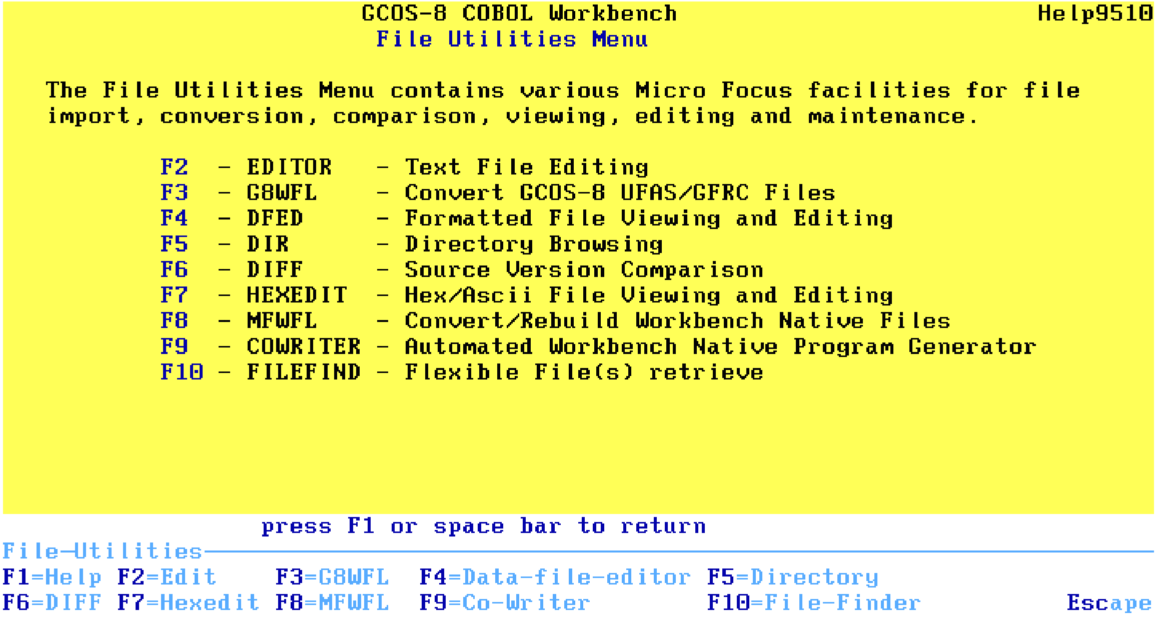
```
    Copy Checkin Utility                                               BANNCCIN

    Library-tag Member-name [-UD userdir]

    Library-tag     - GCOS-8 library file-code, ex. L1, L2, .L

    Member-name     - Name of copy-member or mask, *? (name), ex. ABC*
                      Mask trigger selection list

    -UD userdir     - Default is $USERDIR setting
                      Explicit <userdir> may be given

    Checkin will copy the copy member from <userdir> into the base directory.
    Copy-member is removed from <userdir> and copy-member is UNLOCKED in the
    base directory.





Copy─Checkin──────────────────────────────────────────────Ins─Caps─Num─Scroll
Please enter command line then press ◄┘                                   Esc
                                                                           ◄┘
```

-U(serdir) <userdir>, may be used define User Directory on the fly or change the current USERDIR setting for this checkout operation only.

Once checkin is complete, the copy member is removed from User Directory.

# Chapter 9 FILES MENU

The Files Menu is your place of work for conversion of GCOS-8 test files, file editing, file comparison and file retrieval.



```
                         GCOS-8 COBOL Workbench                       Help9510
                          File Utilities Menu

   The File Utilities Menu contains various Micro Focus facilities for file
   import, conversion, comparison, viewing, editing and maintenance.

           F2  - EDITOR   - Text File Editing
           F3  - G8WFL    - Convert GCOS-8 UFAS/GFRC Files
           F4  - DFED     - Formatted File Viewing and Editing
           F5  - DIR      - Directory Browsing
           F6  - DIFF     - Source Version Comparison
           F7  - HEXEDIT  - Hex/Ascii File Viewing and Editing
           F8  - MFWFL    - Convert/Rebuild Workbench Native Files
           F9  - COWRITER - Automated Workbench Native Program Generator
           F10 - FILEFIND - Flexible File(s) retrieve




                  press F1 or space bar to return
File-Utilities────────────────────────────────────────────────────────────────
F1=Help F2=Edit    F3=G8WFL  F4=Data-file-editor F5=Directory
F6=DIFF F7=Hexedit F8=MFWFL  F9=Co-Writer        F10=File-Finder         Escape
```

## G8WFL Utility

GCOS-8 native file formats (GFRC and UFAS) are not supported by the Micro Focus COBOL/2 system, which provide other native file format implementations. The GCOS-8 Workbench File Loader may however be used to convert existing GCOS-8 test files into COBOL/2 native file formats. G8WFL is in principle and operation very similar to the Micro Focus Workbench File Loader.

### File Formats

The G8WB preprocessor will change GCOS-8 native files (SELECT and FD clause) into following COBOL/2 native file formats:

| Native File Format | GCOS-8 File Format |
|---|---|
| Record Sequential | GFRC or UFAS sequential or tape format |
| Print file | GFRC SSF with PRINTER specification |
| Line Sequential | GFRC SSF with CARD-READER/PUNCHER specification |
| Line Sequential | I* file (ACCEPT) |
| Relative | UFAS relative |
| Indexed Sequential | UFAS indexed |

All COBOL/2 native files use ASCII code-set and any GCOS-8 code-set specification is ignored.

GFRC SSF always result in a variable record length file (length minimum 1 and maximum 1272), whereas other files are made fixed or variable record length according to GCOS-8 rules (SELECT, FD and/or record specification).

## Data Formats

Data formats for non-DISPLAY (ASCII) are implemented differently in G8WB. Functionally this has little or no impact, however it does require fields are transformed when GCOS-8 test files are converted for use in the G8WB environment. Also when viewing and editing files through Hexedit, you must keep such difference of implementation in mind.

| Usage | GCOS-8 | G8WB |
|---|---|---|
| COMP | Packed Decimal | Binary COMP |
| COMP-1 | Binary signed 15 bits | Binary PIC S9(4) COMP-5 |
| COMP-2 | Binary signed 31 bits | Binary PIC S9(9) COMP-5 |
| COMP-3 | Packed decimal | Binary COMP |
| COMP-4 | Packed decimal, half-byte align | Binary COMP |
| COMP-5 | Packed decimal | Binary COMP |
| COMP-6 | Binary signed 35 bits | Binary PIC S9(9) COMP-5<br>- signed 31 bits |
| DB-KEY | Binary signed 35 bits | Binary PIC S9(9) COMP-5<br>- signed 31 bits |

Although Packed decimal fields are implemented as binary, they work functionally equivalent, however the physical size of the field may be different.

Please refer to MFWB documentation for details on native data formats.

## GFRC SSF Sequential

GFRC SSF Sequential files, containing DISPLAY data only with no overpunch, need not be converted through G8WFL. Instead such files should be transferred as plain text files to the G8WB environment. Files with BCD code-set should be changed to ASCII, using CONVERT, before transfer.

The transferred file will be in Micro Focus Line Sequential format and you may convert the file to Record Sequential format (variable length - minimum 1 maximum 1272) using MFWFL.

## UFAS Indexed

UFAS Index file format is not directly supported by G8WFL and you are required to convert the Indexed file to a UFAS Sequential file with UTL2, before file transfer.

After conversion via G8WFL, use MFWFL to convert the Record Sequential file to Indexed Sequential.

## File Transfer

GCOS-8 UFAS and GFRC files must be transferred as BINARY files to the G8WB environment, except as mentioned above. G8WFL will read the original GCOS-8 file as a bit-stream, and decode Block/CI and record header information. Any inconsistencies, i.e. file-transfer somehow failed or corrupted data, will be documented by G8WFL.

When creating (or re-creating) GCOS-8 files, you should start with minimal size and allow files to grow as required. Using this technique, a binary file transfer will only transfer relevant data. If, as an example, you had created a 100 llink file and then only write records that fill up the first two llinks, a binary file transfer would transfer all 100 llinks, because it does not recognize a logical end-of-file.

## CrossePAC

With the CrossePAC option you are able to compress files before file transfer. Besides reducing the file size by 70-90%, CrossePAC will also create a text file archive, which you are able to transfer as a non-binary file.

G8WFL will automatically recognize a CrossePAC archive file and decompress the file before converting it. G8WFL will, however, assume that the embedded file in the CrossePAC archive file is named CPACWORK.

If, as an example, you were to build a file named UFFV.G8F on GCOS-8 for transfer to G8WB, you would instead build a file named CPACWORK and create a CrossePAC archive file named UFFV.G8F. If the UFFV.G8F already exist, you would rename it to CPACWORK, and then create a CrossePAC archive file named UFFV.G8F. The CrossePAC command, TSS or Batch, would look like:

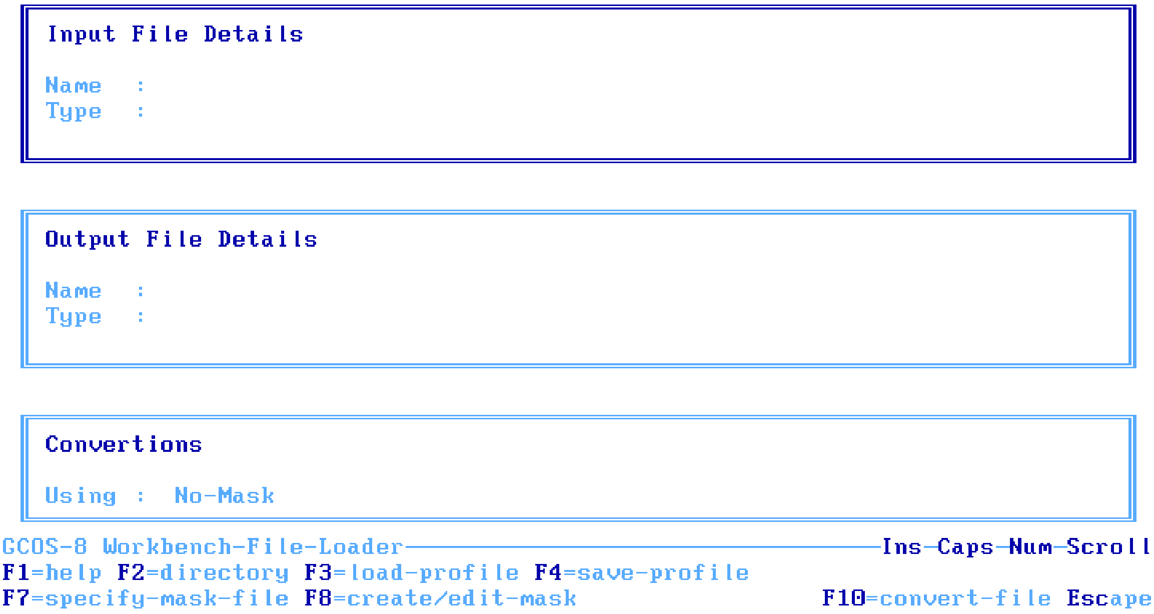
```
CPAC A  SP=N  /UFFV.G8F  /CPACWORK
```

## Profile

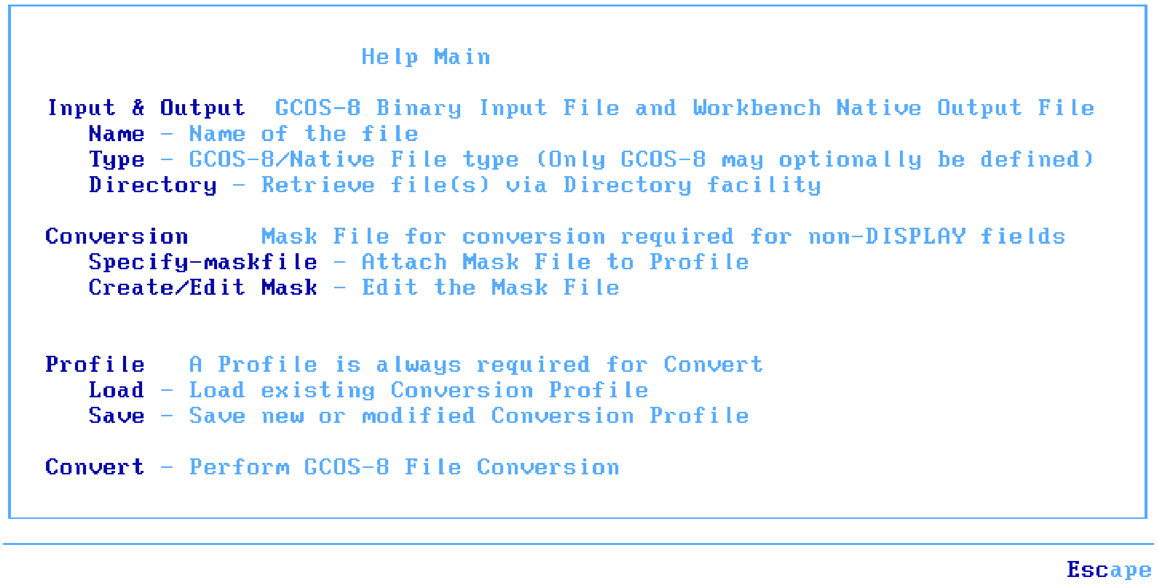
G8WFL provides a Profile feature, where conversion information may be saved in a named profile to ease repeated conversion operations. Such information include input and output file specification, and a mask-set specification.

*Note: You must save the profile, if changed or newly created, before you initiate the file conversion.*

## How to convert

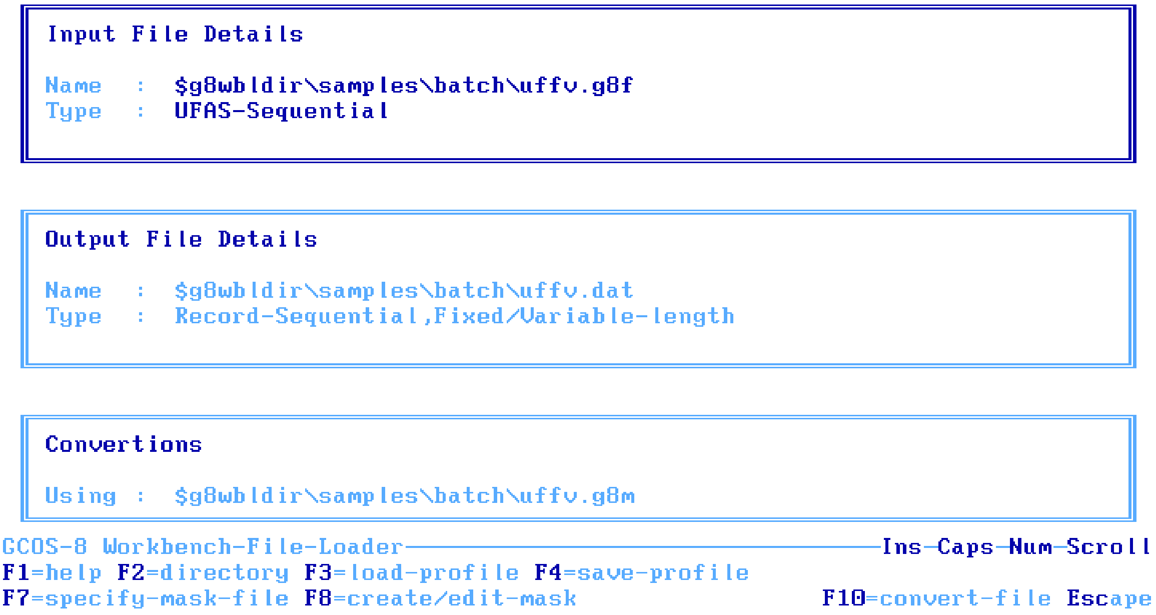


```
Input File Details

Name   :
Type   :

Output File Details

Name   :
Type   :

Convertions

Using  :  No-Mask
GCOS-8 Workbench-File-Loader─────────────────────────────────Ins─Caps─Num─Scroll
F1=help F2=directory F3=load-profile F4=save-profile
F7=specify-mask-file F8=create/edit-mask                   F10=convert-file Escape
```

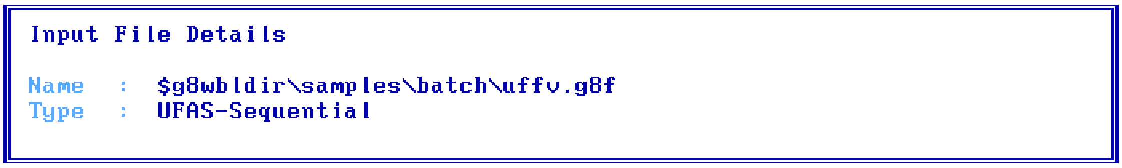


```
                          Help Main

 Input & Output  GCOS-8 Binary Input File and Workbench Native Output File
    Name - Name of the file
    Type - GCOS-8/Native File type (Only GCOS-8 may optionally be defined)
    Directory - Retrieve file(s) via Directory facility

 Conversion     Mask File for conversion required for non-DISPLAY fields
    Specify-maskfile - Attach Mask File to Profile
    Create/Edit Mask - Edit the Mask File


 Profile   A Profile is always required for Convert
    Load - Load existing Conversion Profile
    Save - Save new or modified Conversion Profile

 Convert - Perform GCOS-8 File Conversion

                                                                       Escape
```

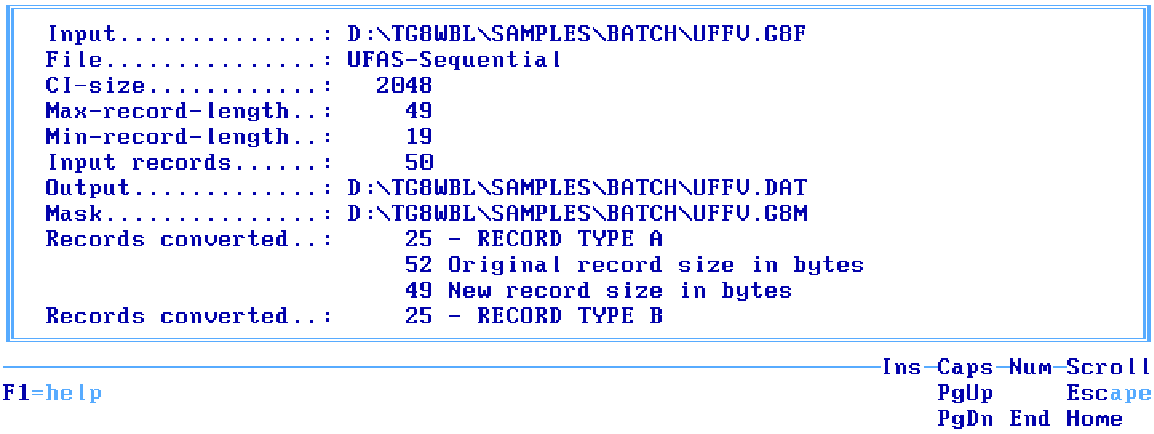
In the following example we load an existing profile and perform the file conversion. G8WFL will communicate conversion statistics or errors messages through a pop-up window.



```
  Input File Details

  Name   :  $g8wbldir\samples\batch\uffv.g8f
  Type   :  UFAS-Sequential


  Output File Details

  Name   :  $g8wbldir\samples\batch\uffv.dat
  Type   :  Record-Sequential,Fixed/Variable-length


  Conver tions

  Using  :  $g8wbldir\samples\batch\uffv.g8m
GCOS-8 Workbench-File-Loader──────────────────────────────────Ins─Caps─Num─Scroll
F1=help F2=directory F3=load-profile F4=save-profile
F7=specify-mask-file F8=create/edit-mask                    F10=convert-file Escape
```



```
Input File Details

Name  :  $g8wbldir\samples\batch\uffv.g8f
Type  :  UFAS-Sequential
```



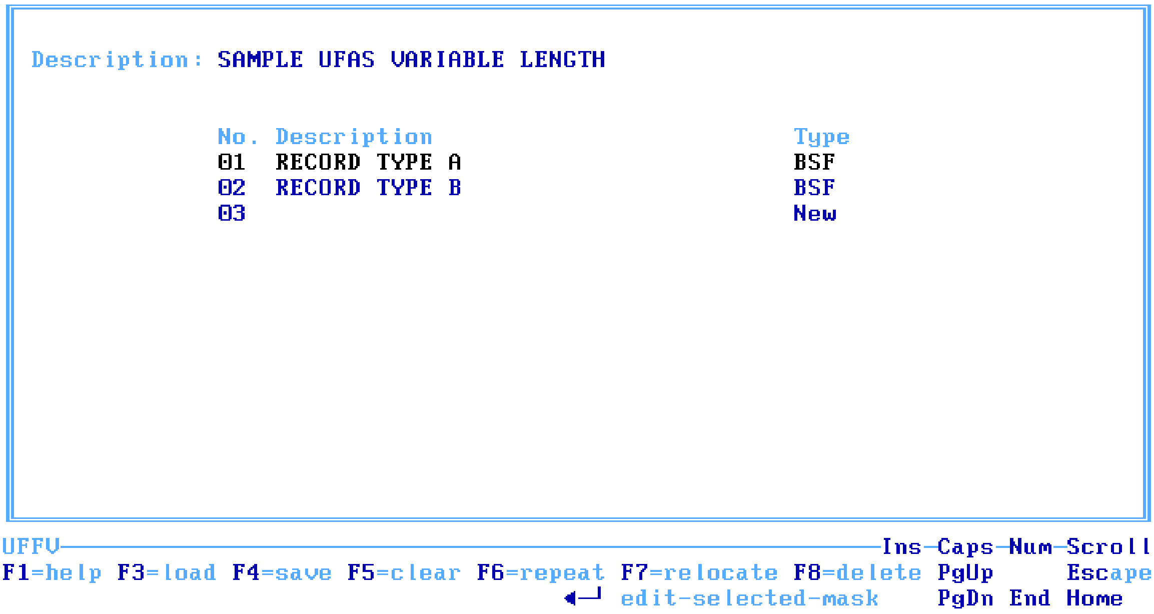
```
Input..............: D:\TG8WBL\SAMPLES\BATCH\UFFV.G8F
File...............: UFAS-Sequential
CI-size...........:   2048
Max-record-length..:     49
Min-record-length..:     19
Input records......:     50
Output.............: D:\TG8WBL\SAMPLES\BATCH\UFFV.DAT
Mask...............: D:\TG8WBL\SAMPLES\BATCH\UFFV.G8M
Records converted..:     25 - RECORD TYPE A
                         52 Original record size in bytes
                         49 New record size in bytes
Records converted..:     25 - RECORD TYPE B

                                                         Ins-Caps-Num-Scroll
F1=help                                                     PgUp     Escape
                                                            PgDn End Home
```

## Mask-set

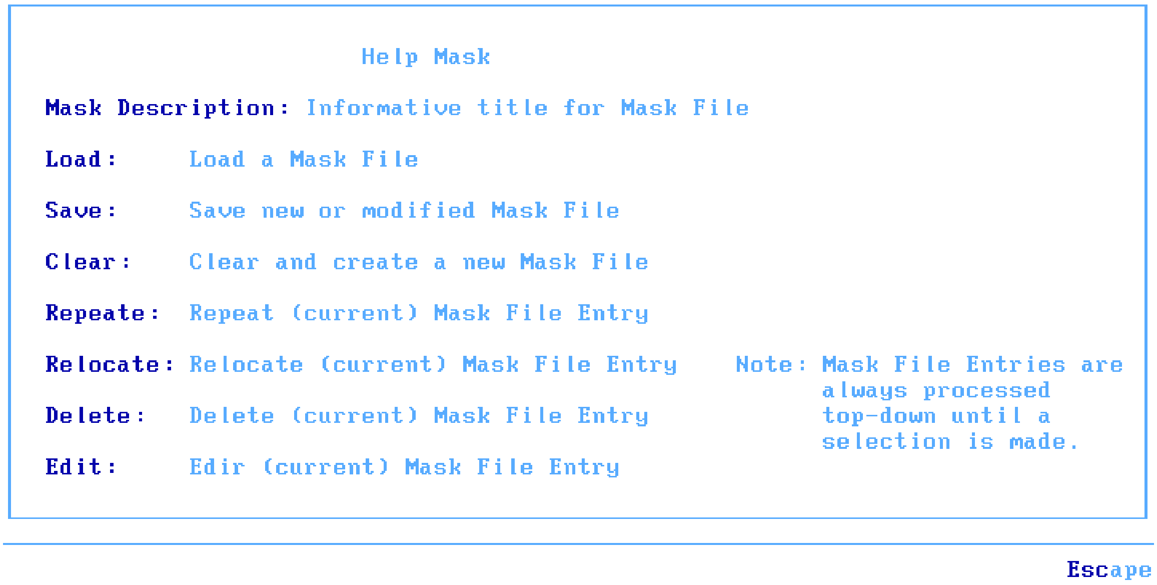
You are required to specify a mask-set for conversion of files containing non-DISPLAY data. The mask-set is used to define one or more masks and hereby associate one or more record descriptions, which are used to perform field transformation for non-DISPLAY data.

Such record descriptions will be generated by the G8WB preprocessor, when requested. Ref. G8WFL-ENABLE preprocessor directive for details. You need to Check an appropriate program first, before building a mask-set, in order to use the generated BSF file.

In the following example we use an existing mask-set with two records types.



```
Description: SAMPLE UFAS VARIABLE LENGTH


             No. Description                        Type
             01  RECORD TYPE A                      BSF
             02  RECORD TYPE B                      BSF
             03                                     New

UFFV---------------------------------------------------------Ins-Caps-Num-Scroll
F1=help F3=load F4=save F5=clear F6=repeat F7=relocate F8=delete PgUp     Escape
                                        ◄┘ edit-selected-mask    PgDn End Home
```



```
                     Help Mask

Mask Description: Informative title for Mask File

Load:      Load a Mask File

Save:      Save new or modified Mask File

Clear:     Clear and create a new Mask File

Repeate:   Repeat (current) Mask File Entry

Relocate: Relocate (current) Mask File Entry    Note: Mask File Entries are
                                                      always processed
Delete:    Delete (current) Mask File Entry           top-down until a
                                                      selection is made.
Edit:      Edir (current) Mask File Entry

                                                                     Escape
```

## Mask

The mask is used to associate a record description, which may be selected from File, Working-Storage or Linkage section of the BSF file. The record description contains all necessary information to support field transformation during conversion.

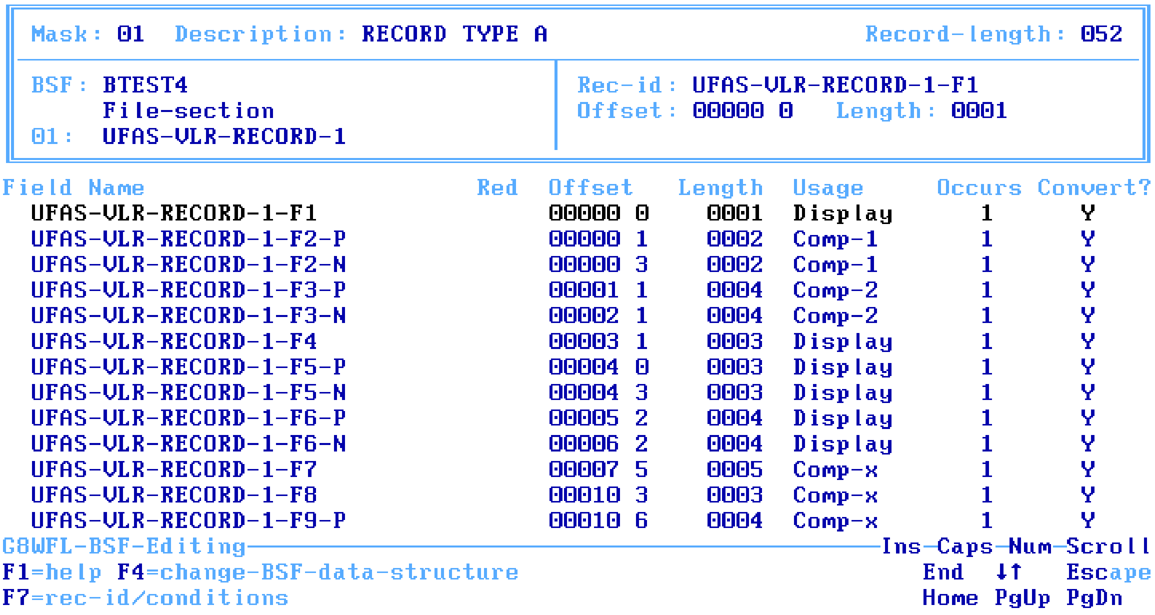
Multiple masks must be specified, when the file has more than one record description. Each mask must then specify a Record-Selection-Field, within the record description, and one or more selection conditions.

| | Condition | Value/Range begin | Range end |
|---|---|---|---|
| Equal | E | ascii-value | |
| Not equal | N | ascii-value | |
| Within range | R | ascii-value | ascii-value |
| Outside range | O | ascii-value | ascii-value |

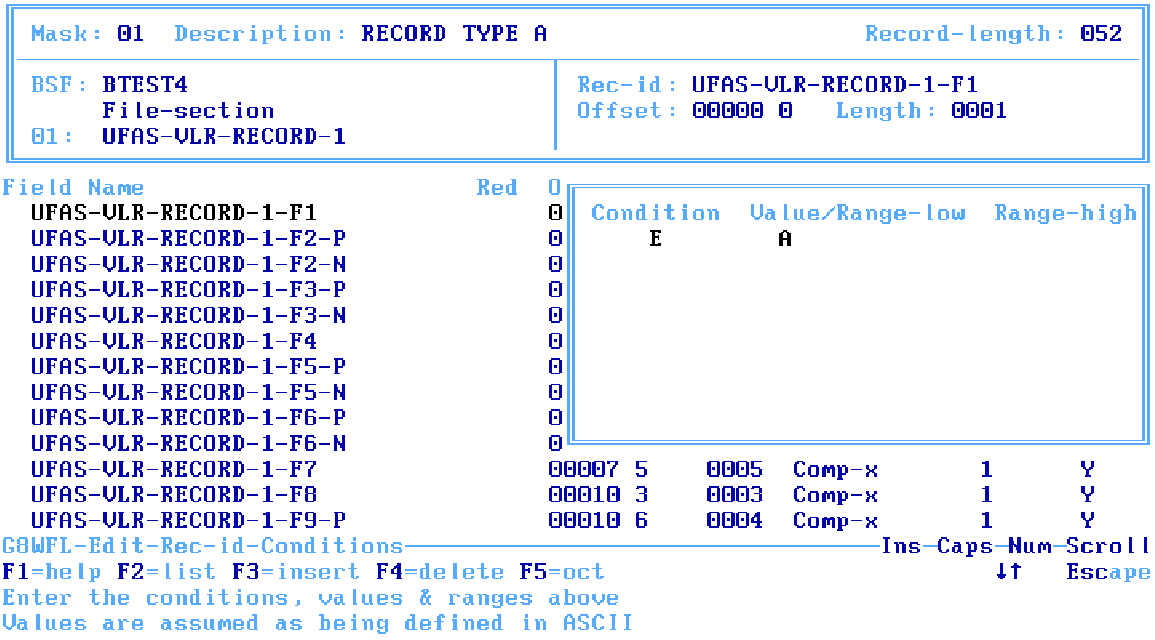
ASCII-values may also be entered as OCTAL-values. The Record-Selection-Field is compared as a PIC X field, regardless of its usage, based on the GCOS-8 representation.

The Mask-set and Masks are evaluated top-down until a selection has been made.

Each record type define which fields should be converted and how the record type is identified.



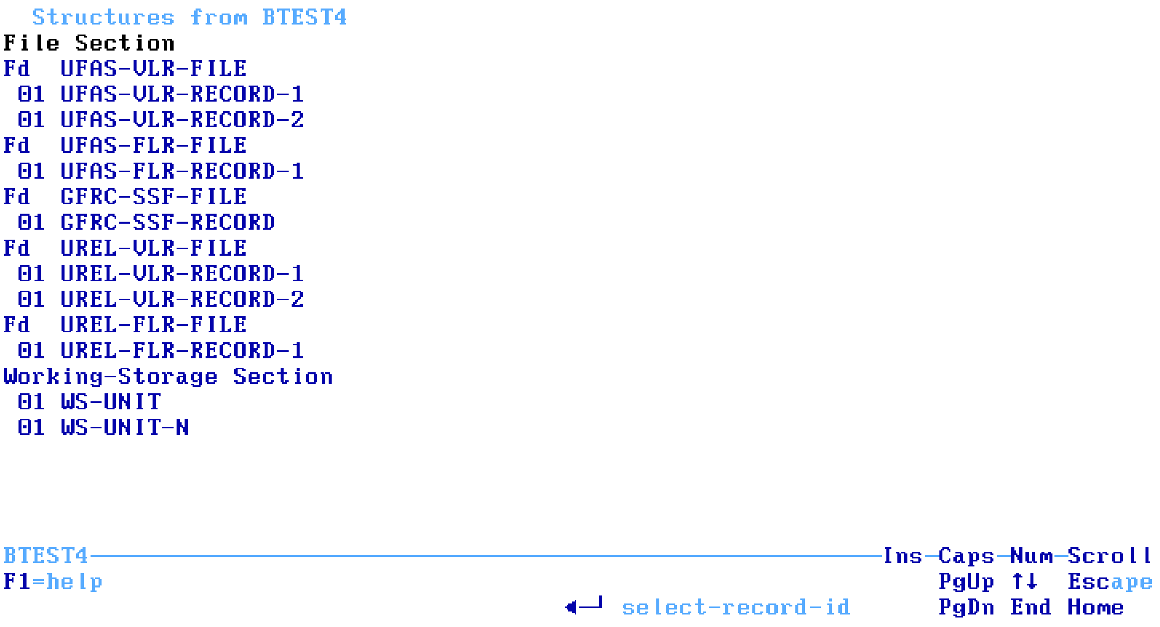
```
 Mask: 01  Description: RECORD TYPE A                  Record-length: 052
 BSF: BTEST4                             Rec-id: UFAS-VLR-RECORD-1-F1
      File-section                       Offset: 00000 0   Length: 0001
 01:  UFAS-VLR-RECORD-1
Field Name                       Red  Offset    Length  Usage     Occurs Convert?
  UFAS-VLR-RECORD-1-F1                00000 0     0001  Display      1      Y
  UFAS-VLR-RECORD-1-F2-P              00000 1     0002  Comp-1       1      Y
  UFAS-VLR-RECORD-1-F2-N              00000 3     0002  Comp-1       1      Y
  UFAS-VLR-RECORD-1-F3-P              00001 1     0004  Comp-2       1      Y
  UFAS-VLR-RECORD-1-F3-N              00002 1     0004  Comp-2       1      Y
  UFAS-VLR-RECORD-1-F4                00003 1     0003  Display      1      Y
  UFAS-VLR-RECORD-1-F5-P              00004 0     0003  Display      1      Y
  UFAS-VLR-RECORD-1-F5-N              00004 3     0003  Display      1      Y
  UFAS-VLR-RECORD-1-F6-P              00005 2     0004  Display      1      Y
  UFAS-VLR-RECORD-1-F6-N              00006 2     0004  Display      1      Y
  UFAS-VLR-RECORD-1-F7                00007 5     0005  Comp-x       1      Y
  UFAS-VLR-RECORD-1-F8                00010 3     0003  Comp-x       1      Y
  UFAS-VLR-RECORD-1-F9-P              00010 6     0004  Comp-x       1      Y
G8WFL-BSF-Editing------------------------------------------Ins-Caps-Num-Scroll
F1=help F4=change-BSF-data-structure                          End  ↓↑   Escape
F7=rec-id/conditions                                          Home PgUp PgDn
```



```
 Mask: 01  Description: RECORD TYPE A                  Record-length: 052
 BSF: BTEST4                             Rec-id: UFAS-VLR-RECORD-1-F1
      File-section                       Offset: 00000 0   Length: 0001
 01:  UFAS-VLR-RECORD-1
Field Name                       Red  O
  UFAS-VLR-RECORD-1-F1                0  Condition  Value/Range-low  Range-high
  UFAS-VLR-RECORD-1-F2-P              0      E          A
  UFAS-VLR-RECORD-1-F2-N              0
  UFAS-VLR-RECORD-1-F3-P              0
  UFAS-VLR-RECORD-1-F3-N              0
  UFAS-VLR-RECORD-1-F4                0
  UFAS-VLR-RECORD-1-F5-P              0
  UFAS-VLR-RECORD-1-F5-N              0
  UFAS-VLR-RECORD-1-F6-P              0
  UFAS-VLR-RECORD-1-F6-N              0
  UFAS-VLR-RECORD-1-F7                00007 5     0005  Comp-x       1      Y
  UFAS-VLR-RECORD-1-F8                00010 3     0003  Comp-x       1      Y
  UFAS-VLR-RECORD-1-F9-P              00010 6     0004  Comp-x       1      Y
G8WFL-Edit-Rec-id-Conditions-------------------------------Ins-Caps-Num-Scroll
F1=help F2=list F3=insert F4=delete F5=oct                         ↓↑   Escape
Enter the conditions, values & ranges above
Values are assumed as being defined in ASCII
```

## Building a new Mask-set

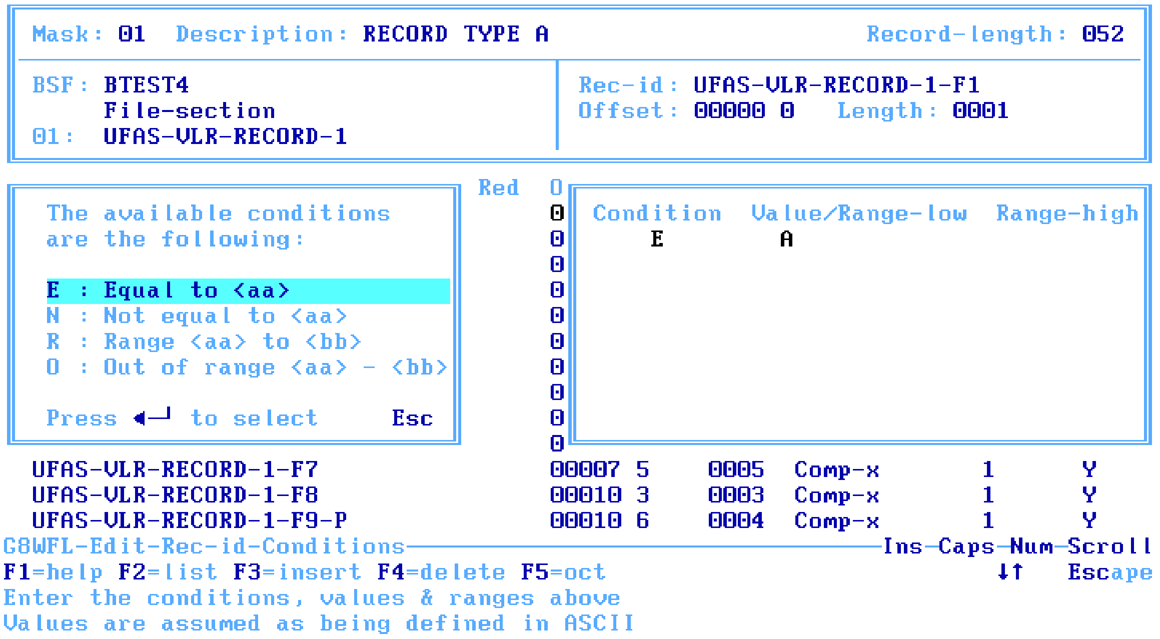
When you initially build a mask-set, you need first to select the BSF structure file, which is generated when your program is checked. This gives you a list of the available record strutures within the program (File, Working or Linkage section).



```
  Structures from BTEST4
File Section
Fd  UFAS-VLR-FILE
 01 UFAS-VLR-RECORD-1
 01 UFAS-VLR-RECORD-2
Fd  UFAS-FLR-FILE
 01 UFAS-FLR-RECORD-1
Fd  GFRC-SSF-FILE
 01 GFRC-SSF-RECORD
Fd  UREL-VLR-FILE
 01 UREL-VLR-RECORD-1
 01 UREL-VLR-RECORD-2
Fd  UREL-FLR-FILE
 01 UREL-FLR-RECORD-1
Working-Storage Section
 01 WS-UNIT
 01 WS-UNIT-N



BTEST4────────────────────────────────────────────────────────Ins─Caps─Num─Scroll
F1=help                                                           PgUp ↑↓  Escape
                                          ◄┘ select-record-id    PgDn End Home
```

After you select a structure, default conversion rules are setup. If the record description contains redefines, or the record description is somehow used for multiple purpose, you may need to modify the Conversion Yes/No for some fields.

If you have multiple records (using the same or other structure), you must specify how the record can be recognized. In the following example a RECORD TYPE A is recognized if the UFAS-VLR-RECORD-1-F1 is Equal to the value A.



```
  Mask: 01  Description: RECORD TYPE A                     Record-length: 052

  BSF: BTEST4                              Rec-id: UFAS-VLR-RECORD-1-F1
       File-section                        Offset: 00000 0   Length: 0001
  01:  UFAS-VLR-RECORD-1

                                      Red  0
   The available conditions                0  Condition  Value/Range-low  Range-high
   are the following:                      0      E          A
                                           0
   E  : Equal to <aa>                      0
   N  : Not equal to <aa>                  0
   R  : Range <aa> to <bb>                 0
   O  : Out of range <aa> - <bb>           0
                                           0
   Press ◄┘ to select        Esc           0
                                           0
  UFAS-VLR-RECORD-1-F7                     00007 5    0005  Comp-x       1       Y
  UFAS-VLR-RECORD-1-F8                     00010 3    0003  Comp-x       1       Y
  UFAS-VLR-RECORD-1-F9-P                   00010 6    0004  Comp-x       1       Y
G8WFL-Edit-Rec-id-Conditions──────────────────────────────────Ins─Caps─Num─Scroll
F1=help F2=list F3=insert F4=delete F5=oct                         ↓↑    Escape
Enter the conditions, values & ranges above
Values are assumed as being defined in ASCII
```

*Note: You may define multiple conditions, which are evaluated top-down, however, you cannot apply conditions to multiple fields.*

### Conversion

File conversion may be requested after you completed definition of profile and optionally a mask-set specification. The profile must be saved before you request conversion.

## MFWFL Utility

May be used to invoke the standard Micro Focus Workbench File Loader utility for file conversion and rebuild of file indexes. Please refer to MFWB documentation for details.

## Editor Utility

May be used to invoke the standard Micro Focus Editor utility for text file editing. Please refer to MFWB documentation for details.

## DFED Utility

May be used to invoke the standard Micro Focus Data File Editor utility for formatted file viewing and editing. Please refer to MFWB documentation for details.

## Hexedit Utility

May be used to invoke the standard Micro Focus Hexadecimal Editor utility for advanced file viewing and editing. Please refer to MFWB documentation for details.

## DIFF Utility

May be used to invoke the standard Micro Focus Source Comparison utility for comparison of source or data files. Please refer to MFWB documentation for details.

## FileFinder Utility

May be used to invoke the standard Micro Focus FileFinder utility for flexible retrieval files. Please refer to MFWB documentation for details.

## Directory Utility

May be used to invoke the standard Micro Focus Directory utility for flexible browsing of directories. Please refer to MFWB documentation for details.

## Co-Writer Utility

May be used to invoke the standard Micro Focus Co-Writer System for generation ad-hoc report or maintenance programs (not GCOS-8 compatible). Please refer to MFWB documentation for details.

# Chapter 10 CONFIGURATION MENU

The Configuration Menu is used to access and edit the various configuration files, and to perform customization of G8WB menu's.

G8WB provides two levels of configuration:

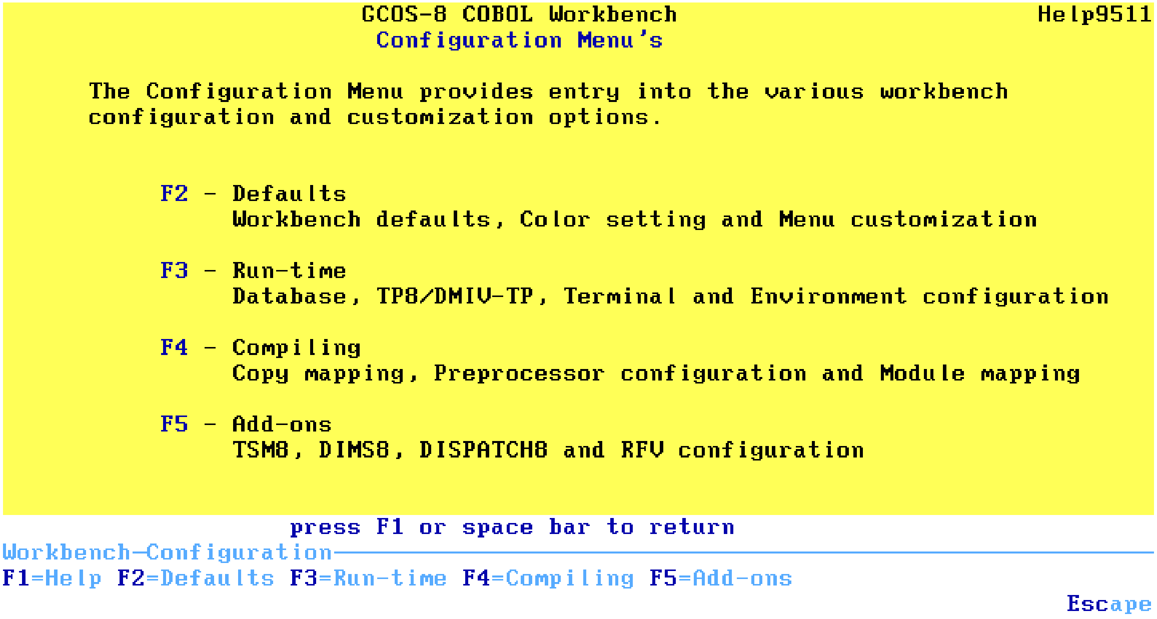
- Global (Server)
- Local (Workstation)

If you have installed Server & Workstation environment, only the Global level is available. As such both Server and Workstation level maintenance reference the same configuration files.

For COPY and EQUATE configuration a third level of level of configuration is provided:

- Project (Group)

You invoke configuration maintenance either through F9-Configuration from main menu, or ALT-F9 from next level menu's. The selected configuration is made available for maintenance through MF Editor.



```
                              GCOS-8 COBOL Workbench                        Help9511
                               Configuration Menu's

      The Configuration Menu provides entry into the various workbench
      configuration and customization options.


           F2 - Defaults
                Workbench defaults, Color setting and Menu customization

           F3 - Run-time
                Database, TP8/DMIV-TP, Terminal and Environment configuration

           F4 - Compiling
                Copy mapping, Preprocessor configuration and Module mapping

           F5 - Add-ons
                TSM8, DIMS8, DISPATCH8 and RFV configuration


                    press F1 or space bar to return
Workbench-Configuration────────────────────────────────────────────────────────────
F1=Help F2=Defaults F3=Run-time F4=Compiling F5=Add-ons
                                                                            Escape
```

## Server Configuration

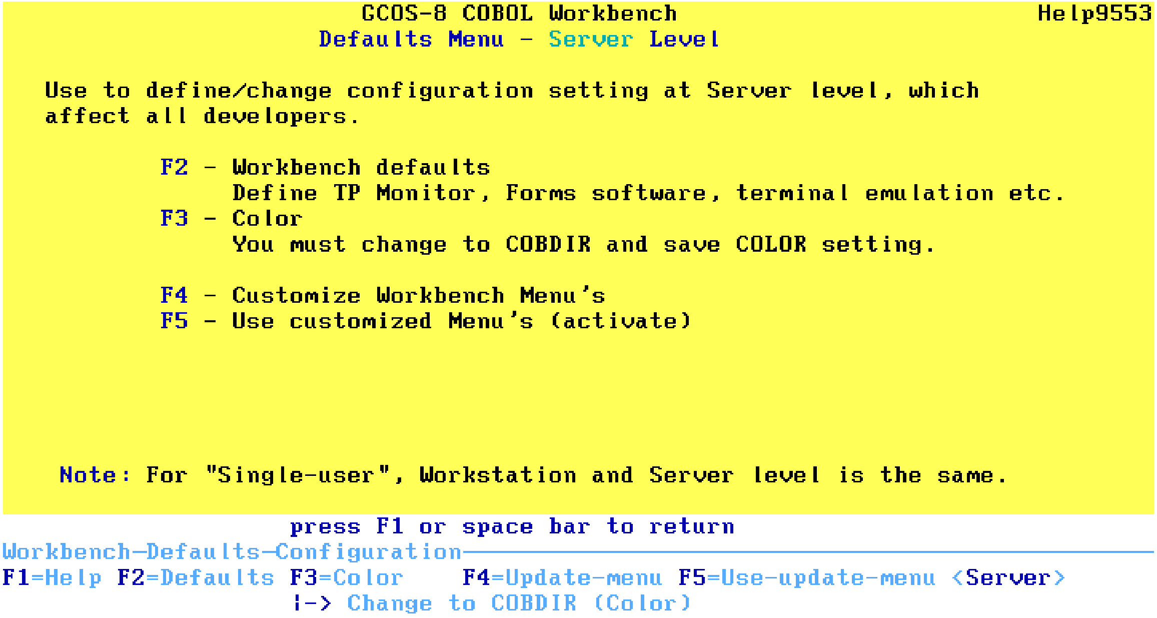
The Server configuration is shared by multiple workstations, thus any changes applied will impact all workstations within the network. The Server is identified via G8WBGDIR and G8WBDDIR (database) setting.

You may have multiple configuration versions, however only one G8WBGDIR/G8WBDDIR setting will be current at any given time. This feature may used when each or some projects need to have a different configuration setup.

G8WBGDIR control access to Server configuration for everything except database.

G8WBDDIR control access to Server configuration for database - IDS-II and INTEREL. By default G8WBDDIR also control all database support files, i.e schema, subschema, however, such support files may also me placed in other shared directories. Please to refer Database configuration for details on how multiple configuration versions can be established.

Server configuration is accessed through ALT-Fn within any of the configuration menu's. Following illustrate a Server configuration menu with F1-Help enabled:



```
                         GCOS-8 COBOL Workbench                        Help9553
                      Defaults Menu - Server Level

  Use to define/change configuration setting at Server level, which
  affect all developers.

        F2 - Workbench defaults
             Define TP Monitor, Forms software, terminal emulation etc.
        F3 - Color
             You must change to COBDIR and save COLOR setting.

        F4 - Customize Workbench Menu's
        F5 - Use customized Menu's (activate)




   Note: For "Single-user", Workstation and Server level is the same.

                    press F1 or space bar to return
Workbench-Defaults-Configuration------------------------------------------------
F1=Help F2=Defaults F3=Color    F4=Update-menu F5=Use-update-menu <Server>
                    !-> Change to COBDIR (Color)
```

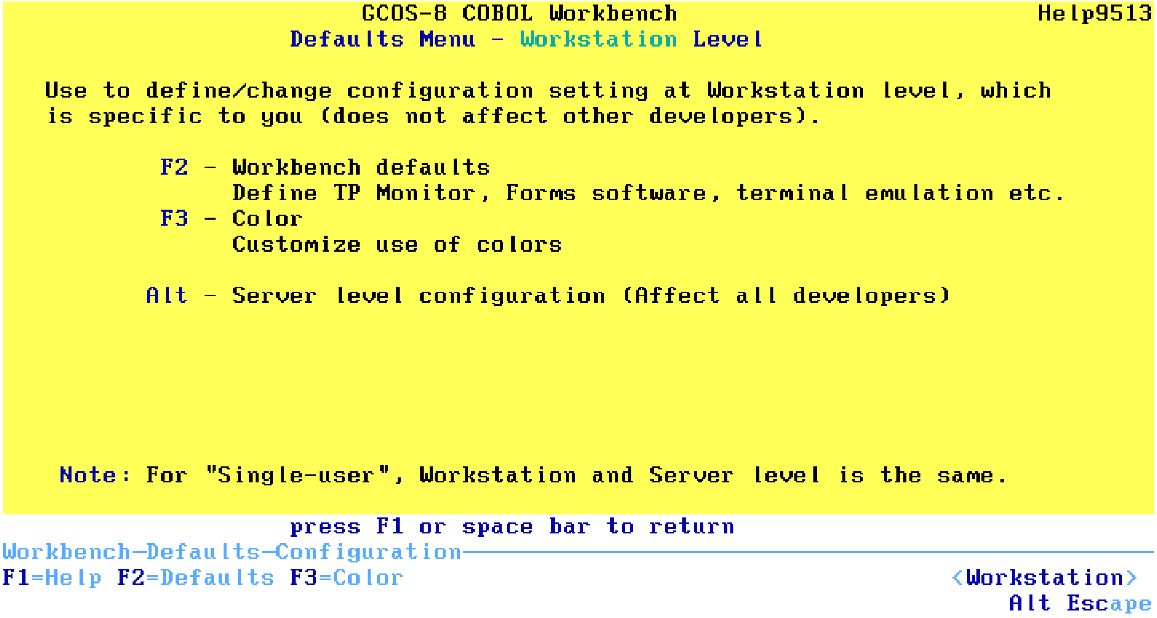
## Workstation Configuration

The Workstation configuration apply only to the specific workstation, thus each workstation in the network may use their "own" configuration setup. The Workstation is identified via G8WBLDIR setting.

Workstation configuration is not complete, thus you are only allowed to override some (most) of the Server settings and/or amend/redefine settings in the Server configuration, i.e. define new commands to TP8.

A sample Workstation configuration file, with no actual settings, is automatically created during installation (Workstation only).

Workstation configuration is accessed through Fn within any of the configuration menu’s. Following illustrate a Workstation configuration menu with F1-Help enabled:



```
                        GCOS-8 COBOL Workbench                            Help9513
                   Defaults Menu - Workstation Level

   Use to define/change configuration setting at Workstation level, which
   is specific to you (does not affect other developers).

           F2 - Workbench defaults
                Define TP Monitor, Forms software, terminal emulation etc.
           F3 - Color
                Customize use of colors

          Alt - Server level configuration (Affect all developers)





    Note: For "Single-user", Workstation and Server level is the same.
                   press F1 or space bar to return
Workbench-Defaults-Configuration------------------------------------------------
F1=Help F2=Defaults F3=Color                                         <Workstation>
                                                                        Alt Escape
```

## Project Configuration

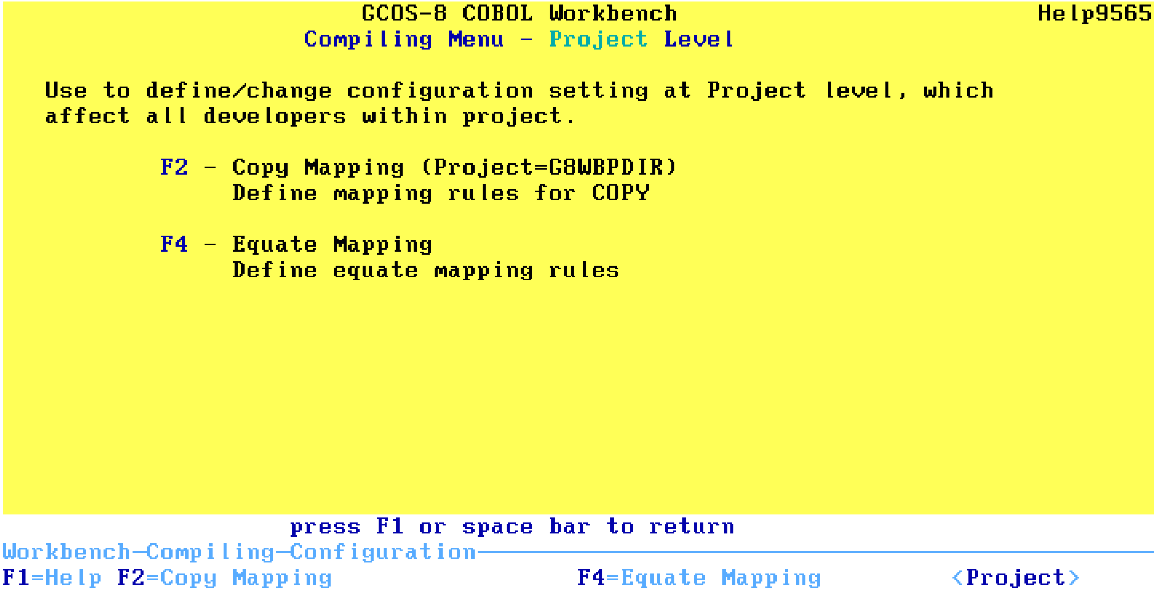
The Project configuration provides an alternative to the Workstation configuration in situation where multiple, but not all workstations, need to share a local configuration. Project configuration is only available for Copy and Equate Mapping, and need only be used when Server configuration cannot be used due to conflicting information.

For Copy Mapping, such a conflict would occur if non-unique libraries are used, i.e. System/Project A use a COPY .. OF L1 library and System/Project B use a COPY ... OF L1 also, however the library is NOT the same. Server configuration cannot identify both libraries at the same time.

Using Project configuration, the configuration for System/Project A would identify L1 and map to the corresponding directory of COPY’s, whereas System/Project B would do likewise.

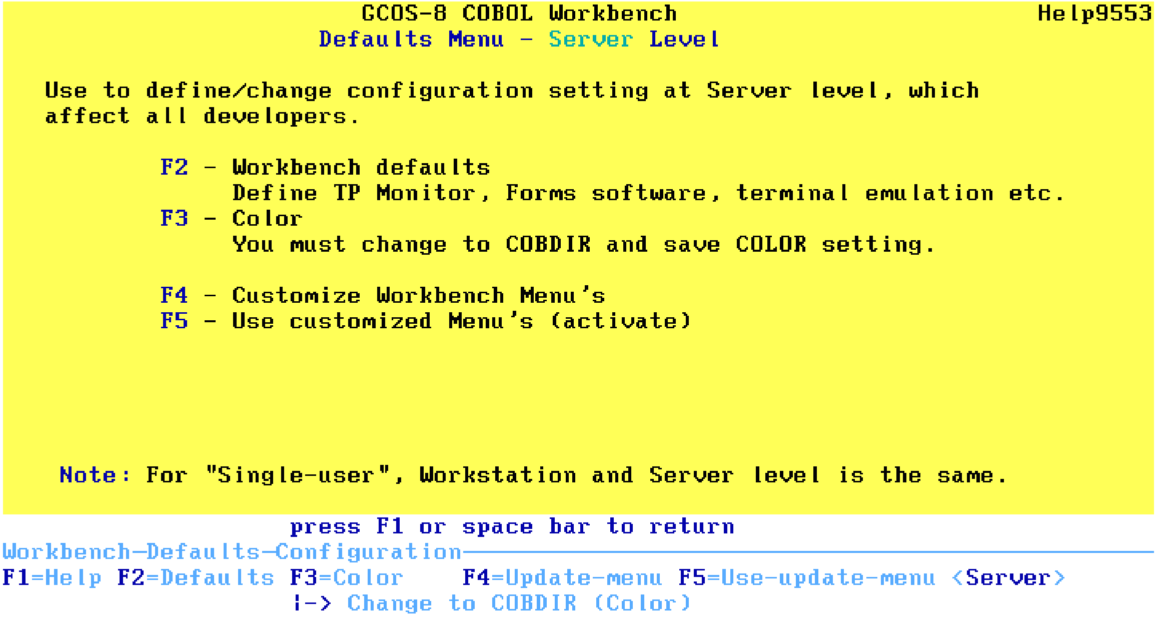
Project configuration is identified through G8WBPDIR setting, or current path (no setting). In above example System/Project A would thus have G8WBPDIR setting different from System/Project B.

Project configuration is accessed through CTRL-Fn within the Compilation menu. Following illustrate a Project configuration menu with F1-Help enabled:



```
                         GCOS-8 COBOL Workbench                         Help9565
                     Compiling Menu - Project Level

  Use to define/change configuration setting at Project level, which
  affect all developers within project.

          F2 - Copy Mapping (Project=G8WBPDIR)
               Define mapping rules for COPY

          F4 - Equate Mapping
               Define equate mapping rules









                    press F1 or space bar to return
Workbench-Compiling-Configuration-----------------------------------------------
F1=Help F2=Copy Mapping                F4=Equate Mapping         <Project>
```

## Defaults Configuration



```
                         GCOS-8 COBOL Workbench                         Help9553
                      Defaults Menu - Server Level

  Use to define/change configuration setting at Server level, which
  affect all developers.

          F2 - Workbench defaults
               Define TP Monitor, Forms software, terminal emulation etc.
          F3 - Color
               You must change to COBDIR and save COLOR setting.

          F4 - Customize Workbench Menu's
          F5 - Use customized Menu's (activate)





   Note: For "Single-user", Workstation and Server level is the same.
                    press F1 or space bar to return
Workbench-Defaults-Configuration------------------------------------------------
F1=Help F2=Defaults F3=Color    F4=Update-menu F5=Use-update-menu <Server>
                    |-> Change to COBDIR (Color)
```

### Workbench Defaults

The Workbench Default configuration is used to define the operation mode of various G8WB components and default values for easy of operation. The provided configuration includes comments which describe the use and possible settings.

Defaults configured at Workstation level, superseed any defaults configuration at Server level.

### Defaults

```
;
; GCOS-8 COBOL Workbench Configuration Version 1.2.00 (Server Config)
;
[G8WB-DEFAULTS]                          ; Defaults configuration
SYSTEM-DATE           : CURRENT          ; CURRENT or YYMMDD
SYSTEM-DAY            : CURRENT          ; CURRENT or YYDDD
SYSTEM-TIME           : CURRENT          ; CURRENT or HHMMSSHS
```

## System-Date

SYSTEM-DATE setting may be used to define a specific date for ACCEPT statement operation. Default is current-date.

## System-Day

SYSTEM-DAY setting may be used to define a specific julian day for ACCEPT statement operation. Default is current julian day.

## System-Time

SYSTEM-TIME setting may be used to define a specific starting time for ACCEPT statement operation. Default is current time.

HHMMSSHS setting will set the starting time upon first ACCEPT and then increment time concurrently with the system clock.

```
TP-MONITOR            : TP8              ; DMIV-TP or TP8
TERMINAL-EMULATION    : VIP78XX V78      ; VIP77XX  V77 call prefix
                                         ; VIP78XX  V78
                                         ; DKU71XX  DKU
                                         ; IBM327X  IBM
                                         ; Other = Name of own Program
CASE                  : LOWER            ; LC ON  = LOWER (Default mode)
                                         ; LC OFF = UPPER
LOGICAL-ID            : MAST             ; Default LID for TP Log-on
MESSAGE-ID            : $NULL            ; Default Message-id/Command for TP
LOGON-MESSAGE         : $NULL            ; Automatic Log-on message for TP
HELLO-MESSAGE         : "THIS IS TP HELLO"
```

## TP-Monitor

TP-MONITOR setting define whether G8WB should emulate a DMIV-TP or TP8 environment. Such setting is both used at compile and run-time, thus when changing setting all TPR's must be re-compiled.

## Terminal-Emulation

TERMINAL-EMULATION setting define the type of Terminal emulation required. Either one of the standard settings (VIP77XX, VIP78XX, DKU71XX or IBM32XX), NONE or the name of an own developed program for Terminal emulation. Both a type or program name, and a prefix must be specified. The prefix is used for calling specific entry-points in the Terminal emulator program and must either be set to the corresponding standard setting (V77, V78, DKU or IBM) or the prefix required by the own developed program for Terminal emulation.

## Case

CASE setting define whether the Terminal emulator should operate in upper-case only or mixed-case (lower-case), when the TP-Monitor is initially started. Such setting may also be changed during run-time.

## Logical-Id

LOGICAL-ID setting define whether TP-Monitor will prompt for a Logical-Id when initially started. When a LID is set, the TP-Monitor will use this without prompting. Logical-Id may also be changed during run-time.

## Message-Id

MESSAGE-ID setting define the default MID or COMMAND to be executed, when an empty transmit is given. The $NULL default setting will just ignore such empty transmit, which would otherwise lead to a transaction abort (X01).

## Logon-Message

LOGON-MESSAGE setting define the default MID or COMMAND to be executed, upon logon to TP. The $NULL default setting disable the feature.

## Hello-Message

HELLO-MESSAGE setting may be used to change the standard “THIS IS TP HELLO” message, you receive after logon to TP.

```
FORMS-SOFTWARE        : F               ; F = TPFF
                                        ; D = DIMS8
                                        ; T = TSM8
                                        ; A = FORMAT
                                        ; P = PILSPRAAK (RFV specific)
                                        ; S = SOLD (BVG specific)
                                        ; NONE
                                        ; Other = Name of own Program
MAX-FORM-LINES        : 24              ; Workbench Painter
MAX-FORM-COLUMNS      : 80              ;
PAINTER-MAX-FIELDS    : 300             ; Maximum 500
PAINTER-MAX-EXTENSIONS: 450             ; Maximum 750 (>max fields)
TPFF-GENERATE         : NORMAL          ; COPY Generation NORMAL/SINGLE/DUAL
```

## Forms-Software

FORMS-SOFTWARE setting define the type of Forms Software emulation required. Either one of the standard settings (F, D, T, A, P or S), NONE or the name of an own developed program for Forms emulation.

## Max-Form-Lines/Columns

MAX-FORMS-LINES/-COLUMNS settings define the maximum size of a forms in terms of lines and columns to the Screen Painter.

## Painter-Max-Fields/Extensions

PAINTER-MAX-FIELDS/-EXTENSIONS settings define table sizes allocated by the Screen painter. Max-extensions must be greater than Max-fields as each field has a minimum of one extention. If the maximum setting is exceeded, the Screen painter will give an appropriate error message, requesting a change of this settiing.

## TPFF-Generate

TPFF-GENERATE setting may be used to control how TPFF will generate form COPY’s.

NORMAL (default) setting, will result is generation of three separate copies - <form>I, <form>R and <form>V.

SINGLE setting, will result is generation of one single copy - <form>, with Image, Record and Vector embedded.

DUAL setting, will result in generation of both NORMAL and SINGLE form copies.

```
CROSSEPAC             : OFF            ; OFF
                                       ; ON  Default export CrossePac format
CROSSEPAC-WORK        : $G8WBLDIR      ; Path for CrossePAC work file or
                                       ; CURRENT (where PAC file is)
SSTLIB                : ON             ; OFF
                                       ; ON  Default UPDATE directive export
SOURCE-IMPORT-XSN     : OFF            ; OFF
                                       ; STRIP Remove COBOL line numbers
COPY-IMPORT-XSN       : OFF            ; OFF
                                       ; STRIP Remove COBOL line numbers
SOURCE-EXPORT-XSN     : OFF            ; OFF
                                       ; ON Make COBOL line numbers 100000,100
                                       ; ssssss iiii Make COBOl line numbers
COPY-EXPORT-XSN       : OFF            ; OFF
                                       ; ON Make COBOL line numbers 100000,100
                                       ; ssssss iiii Make COBOl line numbers
```

## CrossePAC

CROSSEPAC setting define whether you have CrossePAC for GCOS-8 installed.

ON setting will cause all G8WB Export utilities to create files for host transfer in CrossePAC format.

## CrossePAC-work

CROSSEPAC-WORK setting define where temporary files should be placed during CrossePAC operation. This setting is only relevant if CROSSEPAC setting is ON.

$G8WBLDIR setting will place temporary files in your local G8WB directory.

CURRENT setting will place temporary files in the same directory as the Import or Export file being processed.

## Source-XSN

SOURCE-IMPORT-XSN setting define whether COBOL line number should be stripped during import.

SOURCE-EXPORT-XSN setting define whether COBOL line number should be inserted during export.

ON setting will cause COBOL line number sequencing 100000,100 to be inserted. Alternatively, specific start and increment values may be given.

## Copy-XSN

COPY-IMPORT-XSN setting define whether COBOL line number should be stripped during import.

COPY-EXPORT-XSN setting define whether COBOL line number should be inserted during export.

ON setting will cause COBOL line number sequencing 100000,100 to be inserted. Alternatively, specific start and increment values may be given.

## SSTLIB

SSTLIB setting define whether you have SSTLIB installed on GCOS-8.

ON setting will cause G8WB Export Copy utility to use UPDATE directives, instead of APPEND.

```
CHARACTER-SET-ADJUST  : NONE           ; Adjust for national characters
                                       ; during GCOS-8 source/data import &
                                       ; export.
                                       ; NONE
                                       ; DENMARK
                                       ;   GCOS-8 x'7b' G8WB x'91'
                                       ;   GCOS-8 x'5b' G8WB x'92'
                                       ;   GCOS-8 x'7c' G8WB x'9b'
                                       ;   GCOS-8 x'5c' G8WB x'9d'
                                       ;   GCOS-8 x'7d' G8WB x'86'
                                       ;   GCOS-8 x'5d' G8WB x'8f'
                                       ; SWEDEN
                                       ;   GCOS-8 x'23' G8WB x'8e'
                                       ;   GCOS-8 x'24' G8WB x'8f'
                                       ;   GCOS-8 x'40' G8WB x'99'
                                       ;   GCOS-8 x'5b' G8WB x'23'
                                       ;   GCOS-8 x'5c' G8WB x'90'
                                       ;   GCOS-8 x'5d' G8WB x'9a'
                                       ;   GCOS-8 x'60' G8WB x'82'
                                       ;   GCOS-8 x'7b' G8WB x'84'
                                       ;   GCOS-8 x'7c' G8WB x'94'
                                       ;   GCOS-8 x'7d' G8WB x'86'
                                       ;   GCOS-8 x'7e' G8WB x'81'
                                       ; NORWAY
;
PRINT-SOFTWARE        : DISPATCH8      ; DISPATCH8
                                       ; NONE
                                       ; Other = Name of own Program
;
PAPER-LENGTH          : 48             ; Length in lines for reporting
PRINTER-DEVICE        : LPT1:          ; Default device/file for printing
PRINTER-MODEL         : IBM-PROIII     ; Default printer model
```

## Character-Set-Adjust

CHARACTER-SET-ADJUST setting define whether adjustment for national characters should take place during G8WB Import and Export of application components.

NONE (default) setting disables this feature.

DENMARK, SWEDEN, NORWAY settings are currently recognized. The associated national character rules are documented in the configuration file, however, these are not configurable.

## Print-Software

PRINT-SOFTWARE setting define the type of Print Software emulation required. Either one of the standard settings (DISPATCH8), NONE or the name of an own developed program for Print emulation.

## Paper-Length

Not currently used.

## Printer-Device

Not currently used.

### Printer-Model

Not currently used.

## Trace

The TRACE setting may be used to control the amount of information written to the TP-Monitor Journal-Log.

```
[G8WB-TRACE]                                   ; Trace configuration
TP-TX-SEQUENCE        : ON                     ; Trace TX/TPR state
TP-EXEC-CALL          : ON                     ; Trace TP Interface/procedure calls
TP-COMM-DETAIL        : ON                     ; Trace SEND/RECEIVE details
```

### TP-TX-Sequence

The TP-TX-Sequence setting enable/disable logging of Transaction sequence information like Start-of-Transaction, Start-of-TPR etc.

### TP-Exec-Call

The TP-Exec-Call setting enable/disable logging of interface information from executive calls like Send, Receive, Conversation, Spawn-Batch-Job etc.

### TP-Comm-Detail

The TP-Comm-Detail setting enable/disable logging of Message text information for SEND and RECEIVE operation.

### COLOR

The Color Function may be used to invoke the interactive Color Configuration Utility. You may try different colors and update the workbench to use the new color configuration. Please refer to MFWB Focus documentation for details.

For Server level, you must change to $COBDIR before invoking COLOR, otherwise changes will be similar to Workstation level.

For Workstation level, the COLOR setting is saved under current path. When G8WB is subsequently started from the same path, the COLOR configuration is applied.

## Workbench Menu

G8WB utilize the standard MFWB Menu system, which operate based on a Menu Description File. Please refer to MFWB documentation for details on the Menu System and the Menu Description syntax.

The standard G8WB menu may be customized in different ways to suit specific site requirements. Areas of customization could be:

- Amend Host Menu with your preferred communication software.

- Amend File Transfer Menu with your preferred communication software and method of doing file transfer.

- Change default settings for checking.

- Limit functionality in a Menu.
- Reorganize Menu’s in general.
- Change HELP screens.

*Note: You should always keep backup copies of the customized menus. Also, you should keep a copy of the original G8WB menu, so you can use MF DIFF to compare new versions, and migrate changes into your customized menus.*

## Customization

When you select the Update Menu function, the entire G8WB Menu file is made available for editing via the MF Editor.

Once you complete editing, you should save the menu file (F4) and exit the editor using Escape.

To activate the changed menu file, you select the Use Update Menu function.

## Host Menu

Following example illustrate the insertion of GLINK communication software in the Host menu.

```
********************************************************************************
:new-menu: HOST
********************************************************************************
:title:
Host Communication
:line1:
F1=Help                                                                  Escape
hh                                                                       hhh
:efuncs:
F1 :help: 9508 :library: $G8WBDIR\LBR\G8WBHELP
Esc :menu: :lastexitq:
:end-menu:
```

GLINK has been included as F2, which will escape to OS and start GL command.

```
********************************************************************************
:new-menu: HOST
********************************************************************************
:title:
Host Communication
:line1:
F1=help F2=GLINK                                                         Escape
hh      hh                                                               hhh
:efuncs:
F1 :help: 9508 :library: $G8WBDIR\G8WBHELP
F2 :dos: GL
Esc :menu: :lastexitq:
:end-menu:
```

## TP Development (Check) Menu

Following example illustrate the change of defaults in the Check menu of TP development.

```
********************************************************************************
:new-menu: TPCHECK
```

```
********************************************************************************
:title:
Check
:line1:
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
hh      hh     hh       hh     hh           hh      hh     hh     hh hhh     hhh
:name: CBL
:library: $COBDIR\CHECK.LBR
:library: $COBDIR\ADVANIM.LBR
:set-linein: SETDIRC
:efuncs:
F1 :help: 9006 :library: $G8WBDIR\LBR\G8WBHELP
F3 :toggle: (u)Pause [] :at: 11
   :return: "errq editor(mf)" editor(mf)
F4 :toggle: (u)List-Con (u)nolist (u)List-File (u)Print :at: 17
   :return: "list noform" nolist [] list(lst)
F5 :toggle: [] (u)Struct (u)Analyze "(u)Strc+Anlz" :at: 27
   :return: [] struct analyze "struct analyze"
F6 :toggle: "(u)COBOL-74"
            "(u)COBOL-85"
            "(u)COBOL-SQL"
            "(u)TPR-74"
            "(u)TPR-85"
            "(u)TPR-SQL" :at: 37
   :return: use($G8WBDIR\MFI\G8WBCBL7.DIR)
            use($G8WBDIR\MFI\G8WBCBL8.DIR)
            use($G8WBDIR\MFI\G8WBCBLQ.DIR)
            use($G8WBDIR\MFI\G8WBTPR7.DIR)
            use($G8WBDIR\MFI\G8WBTPR8.DIR)
            use($G8WBDIR\MFI\G8WBTPRQ.DIR)
F7 :toggle: [] (u)Xref (u)Ref (u)Xref+Ref :at: 46
    :return: [] xref ref "xref ref"
F8 :toggle: [] (u)CSI :at: 55
   :return: [] csi
F9 :toggle: [] (u)On :at: 59 :return: confirm "confirm %L%"
F10 :menu: SETDIRC
Esc :menu: :lastexitq:
Ret :runsys: CHECK %F% anim ensuite(2)
:message: Checking TPR/Module %NE%, Ctrl+Break to Stop
:menu: :lastexitq:
:end-menu:
```

The language type has been changed as follows: SQL types have been removed, and COBOL-85 has been made the default language.

```
********************************************************************************
:new-menu: TPCHECK
********************************************************************************
:title:
Check
:line1:
F1=help F2=dir F3=pause F4=lst F5=strc/anlz F6=lang F7=ref F8=csi F9/F10=opt Esc
hh      hh     hh       hh     hh           hh      hh     hh     hh hhh     hhh
:name: CBL
:library: $COBDIR\CHECK.LBR
:library: $COBDIR\ADVANIM.LBR
:set-linein: SETDIRC
:efuncs:
F1 :help: 9006 :library: $G8WBDIR\LBR\G8WBHELP
F3 :toggle: (u)Pause [] :at: 11
   :return: "errq editor(mf)" editor(mf)
F4 :toggle: (u)List-Con (u)nolist (u)List-File (u)Print :at: 17
   :return: "list noform" nolist [] list(lst)
F5 :toggle: [] (u)Struct (u)Analyze "(u)Strc+Anlz" :at: 27
   :return: [] struct analyze "struct analyze"
F6 :toggle: "(u)COBOL-85"
            "(u)COBOL-74"
            "(u)TPR-85"
            "(u)TPR-74" :at: 37
   :return: use($G8WBDIR\MFI\G8WBCBL8.DIR)
            use($G8WBDIR\MFI\G8WBCBL7.DIR)
            use($G8WBDIR\MFI\G8WBTPR8.DIR)
            use($G8WBDIR\MFI\G8WBTPR7.DIR)
F7 :toggle: [] (u)Xref (u)Ref (u)Xref+Ref :at: 46
   :return: [] xref ref "xref ref"
F8 :toggle: [] (u)CSI :at: 55
   :return: [] csi
F9 :toggle: [] (u)On :at: 59 :return: confirm "confirm %L%"
F10 :menu: SETDIRC
Esc :menu: :lastexitq:
Ret :runsys: CHECK %F% anim ensuite(2)
:message: Checking TPR/Module %NE%, Ctrl+Break to Stop
:menu: :lastexitq:
:end-menu:
```

## Run-time Configuration Menu

Following example illustrate limitation of functionality for the Run-time configuration menu.

```
********************************************************************************
:new-menu: CONFRUN
********************************************************************************
:title:
Workbench Run-time Configuration
:line1:
F1=Help F2=IDS-II    F3=TP8     F4=Terminal F5=Environment         <Workstation>
hh      hh          hh         hh          hh                       hhhhhhhhhhh
:line2:
        F6=INTEREL  F7=DMIV-TP                                          Alt Escape
        hh          hh                                                  hhh hhh
:alt-line1:
F1=Help F2=IDS-II    F3=TP8     F4=Terminal F5=Environment         <Server>
hh      hh          hh         hh          hh                       hhhhhh
:alt-line2:
        F6=INTEREL  F7=DMIV-TP
        hh          hh
:library: $COBDIR\BLDMENU.LBR
:efuncs:
F1 :help: 9514 :library: $G8WBDIR\LBR\G8WBHELP
F2 :call: EDIT {2} $G8WBLDIR\DBMS.CFG
F3 :call: EDIT {2} $G8WBLDIR\TP8.CFG
F4 :call: EDIT {2} $G8WBLDIR\TERMINAL.CFG
F5 :call: EDIT {2} $G8WBLDIR\ENV.CFG
F6 :call: EDIT {2} $G8WBLDIR\INTEREL.CFG
F7 :call: EDIT {2} $G8WBLDIR\DMIVTP.CFG
F31 :help: 9554 :library: $G8WBDIR\LBR\G8WBHELP
F32 :call: EDIT {2} $G8WBDDIR\DBMS.CFG
F33 :call: EDIT {2} $G8WBGDIR\TP8.CFG
F34 :call: EDIT {2} $G8WBGDIR\TERMINAL.CFG
F35 :call: EDIT {2} $G8WBGDIR\ENV.CFG
F36 :call: EDIT {2} $G8WBGDIR\INTEREL.CFG
F37 :call: EDIT {2} $G8WBGDIR\DMIVTP.CFG
Esc :menu: :lastexitq:
:end-menu:
```

The Server level configuration has been removed.

```
********************************************************************************
:new-menu: CONFRUN
********************************************************************************
:title:
Workbench Run-time Configuration
:line1:
F1=Help F2=IDS-II   F3=TP8     F4=Terminal F5=Environment         <Workstation>
hh      hh          hh         hh          hh                      hhhhhhhhhhh
:line2:
        F6=INTEREL  F7=DMIV-TP                                            Escape
        hh          hh                                                    hhh
:library: $COBDIR\BLDMENU.LBR
:efuncs:
F1 :help: 9514 :library: $G8WBDIR\LBR\G8WBHELP
F2 :call: EDIT {2} $G8WBLDIR\DBMS.CFG
F3 :call: EDIT {2} $G8WBLDIR\TP8.CFG
F4 :call: EDIT {2} $G8WBLDIR\TERMINAL.CFG
F5 :call: EDIT {2} $G8WBLDIR\ENV.CFG
F6 :call: EDIT {2} $G8WBLDIR\INTEREL.CFG
F7 :call: EDIT {2} $G8WBLDIR\DMIVTP.CFG
Esc :menu: :lastexitq:
:end-menu:
```

## Help Screen

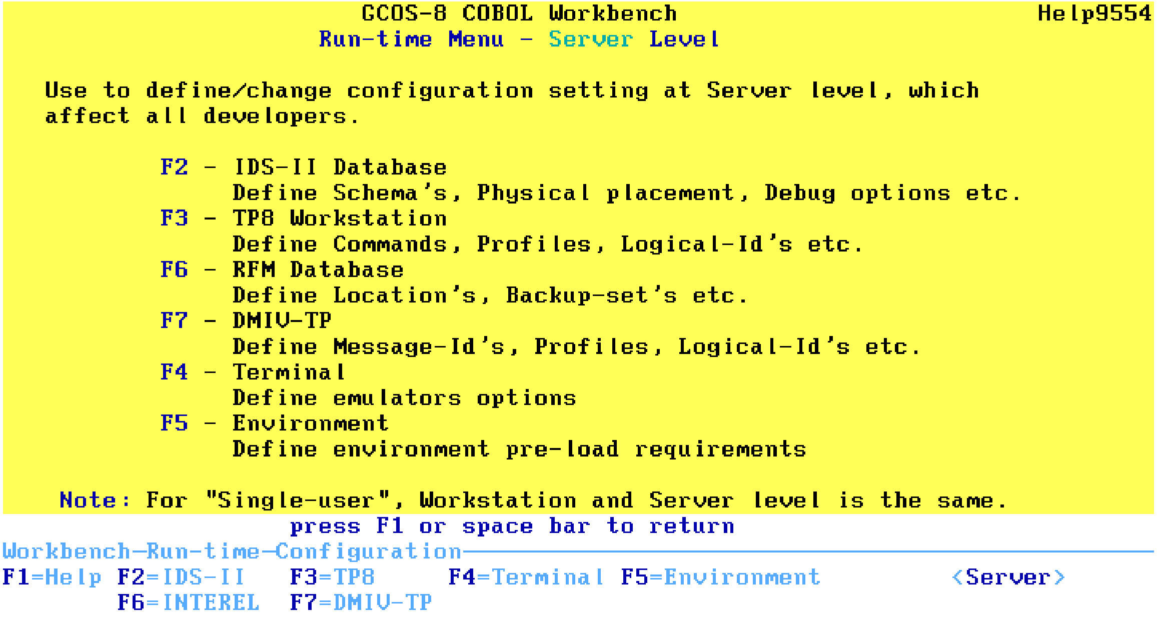
Continuing above example with Run-time configuration menu, the help screen for the menu has been replaced with a customized help screen.

```
********************************************************************************
:new-menu: CONFRUN
********************************************************************************
:title:
Workbench Run-time Configuration
:line1:
F1=Help F2=IDS-II   F3=TP8     F4=Terminal F5=Environment         <Workstation>
hh      hh          hh         hh          hh                      hhhhhhhhhhh
:line2:
        F6=INTEREL  F7=DMIV-TP                                            Escape
        hh          hh                                                    hhh
:library: $COBDIR\BLDMENU.LBR
:efuncs:
F1 :help: S001 :library: $G8WBDIR\LBR\SHELP
F2 :call: EDIT {2} $G8WBLDIR\DBMS.CFG
F3 :call: EDIT {2} $G8WBLDIR\TP8.CFG
F4 :call: EDIT {2} $G8WBLDIR\TERMINAL.CFG
F5 :call: EDIT {2} $G8WBLDIR\ENV.CFG
F6 :call: EDIT {2} $G8WBLDIR\INTEREL.CFG
F7 :call: EDIT {2} $G8WBLDIR\DMIVTP.CFG
Esc :menu: :lastexitq:
:end-menu:
```

G8WB standard menu help screens are located under $G8WBGDIR\HELP. Help screens are build using MF Editor (forms feature) and collected into the $G8WBDIR\LBR\G8WBHLP.LBR library file.

When building customized help screen, it is recommended that you establish new files and collect these into another library file, i.e. $G8WBDIR\LBR\SHELP.LBR in above example.

# Run-time Configuration



```
                         GCOS-8 COBOL Workbench                        Help9554
                      Run-time Menu - Server Level

  Use to define/change configuration setting at Server level, which
  affect all developers.

          F2 - IDS-II Database
               Define Schema's, Physical placement, Debug options etc.
          F3 - TP8 Workstation
               Define Commands, Profiles, Logical-Id's etc.
          F6 - RFM Database
               Define Location's, Backup-set's etc.
          F7 - DMIV-TP
               Define Message-Id's, Profiles, Logical-Id's etc.
          F4 - Terminal
               Define emulators options
          F5 - Environment
               Define environment pre-load requirements

   Note: For "Single-user", Workstation and Server level is the same.
                    press F1 or space bar to return
Workbench-Run-time-Configuration------------------------------------------------
F1=Help F2=IDS-II   F3=TP8     F4=Terminal F5=Environment          <Server>
        F6=INTEREL  F7=DMIV-TP
```

Workstation level configuration, superseed or amend configuration at Server level. For tables, duplicate entries may possibly be created, however only the first entry (workstation) is actually used.

## IDS-II DBMS

### Schema-Section

The Schema section defines all IDS-II schema's available within G8WB. During initial translation and validation of a schema, the schema configuration is updated automatically. Each schema is assigned a unique number (from 99 to 1 until a free number is found) during automatic configuration. Once a schema number has been assigned, it MUST not be changed, unless you delete the schema first. You may control sequence numbering by configuring new schema's manually, before running the initial translation and validation of schema.

Location Schema setting define where the Schema support files are stored.

Location Files setting define where the database files are stored.

Permission setting define whether database is to be used in Read-only or Update mode.

Concurrent setting define whether concurrency control should be applied to the database.

Protect setting define whether integrity control should be applied to the database.

```
;
; IDS-II Database Management Configuration Version 1.2.00 (Server Config)
;
;                    Number Location Location            [Perm Concurrent Protect]
;                    (1-99) Schema   files               [R/W  OFF/ON/FS2 OFF/ON ]
[SCHEMA-SECTION]
DIMS                 : 23   CURRENT  $$LOCAL\$G8WBLDIR\DBMS W OFF OFF; DIMS8 Only
TELEX8               : 24   CURRENT  $$LOCAL\$G8WBLDIR\DBMS W OFF ON;  DIMS8 Only
TESTSCM              : 25   CURRENT  $$LOCAL\$G8WBLDIR\DBMS W OFF ON;  SAMPLES
CLASS                : 26   CURRENT  $$LOCAL\$G8WBLDIR\DBMS W OFF ON;  SAMPLES
```

A maximum of 99 schema's may be defined.

Schema numbers 23 and 24 are reserved for DIMS8 (if installed).

Schema numbers 25 and 26 are reserved for SAMPLES (if installed).

*Note: During DMIV-TP/TP8 start-up all configured schema's and subschema's are logically opened. Allocation and load of DBMS control tables and physical open of database files are deferred until a TPR/Module request use of a shema.*

## Location-Schema

CURRENT setting will place all schema support files under $G8WBDDIR\SCHEMA, which is the default shared database directory.

If you need to work with multiple versions of a schema, you may use <Path> setting , where <path> can be any valid directory (shared or non-shared). All schema support files are then placed under <path>\SCHEMA. You will notice under Backup-set configuration, that a location setting of SCHEMA will ensure backup-set are store together with the schema support files, and you thus have multiple versions of backup-sets.

$G8WBDDIR will as a minimum contain your database configuration file. When you are working with multiple schema versions you have two options for configuration:

Workstation-level - if acces to an alternate schema version is limited a one or few developers, you may choose to re-configure the specific schema in the workstation level configuration file. This is done by duplicating the schema condiguration entry and changing the location schema setting to <path>. Once the alternate version is ready for general use, you MOVE the schema support files back into $G8BWDDIR and remove your workstation level configuration entry.

G8WBDDIR - if access to alternate schema version(s) is more complex, you may instead choose to use different G8WBDDIR setting for each enviroment. You will then have to duplicate the DBMS.CFG configuration, so each G8WBDDIR has a complete configuration. For each schema configuration, within each G8WBDDIR\DBMS.CFG, you may then use different location schema setting to suit your needs for acces to different versions of the same schema.

*Note: Although above allows for very flexible database configuration, without the need for copying schema support files, you should keep in mind that the schema number MUST be correct in all configuration files, otherwise results are unpredictable.*

## Location-Files

Default setting for automatic schema configuration is $$LOCAL\$G8WBLDIR\DBMS. This setting will allow each developer to work with a local single-user database, with or without integrity control enabled. This setting is generally recommended, because it allows each developer in a networked environment to work independently with a database or a set of databases. Shared backup-set may be used to establish (restore) pre-defined versions of the database. Physical database files are thus placed under $G8WBLDIR\DBMS and when integrity is enabled, $$LOCAL setting direct File Share 2 to acces and protect files through the LOCAL server.

If you need to share the physical database, you can specify any setting that reference a shared file server directory. Such setting could be $G8WBDDIR\DBMS, which is readily available. If you intend to use Concurrency or Integrity on the shared database, you may need also to specify a File Share 2 server name, ref. below for details.

## Permission

Default is Write permission. R(ead) setting may be used if a database, usually shared, is to be accessed in Read-only mode. DBMS will then open the database as read-only, and provided all access is read-only, no concurrency control is required. If, however, some acces with write and some with read-only, through different DBMS configuration, then you would must enable some form of concurrency control, otherwise concurrent acces is rejected.

## Concurrency

Concurrency control setting may be used to control whether multi-user database access is to be allowed and which method of concurrency control to apply.

OFF setting (default), database files are opened exclusive (no concurrency). If you are running DOS SHARE, each database file is locked on File level either as read-only or write, depending on permission setting.

ON setting, database files are opened non-exclusive (concurrent access). If you are running DOS SHARE or other form of concurrency control software, each database file is locked at Record level.

*Note: This setting may require a substantial amount of Lock resources and is generally not recommended.*

FS2 setting, database files are opened non-exclusive (concurrent acceess) and I/O requests are re-directed to Micro Focus File Share 2. This setting is recommended for multi-user database access.

A File Share 2 Server (or more) must be started before database is used, otherwise open of the physical files will fail. Database file allocation is prepared for File Share 2 use and you need to change location files setting for the schema.

A default server $$GLOBAL is included with G8WBDDIR, and location files setting could thus be $$GLOBAL\$G8WBDDIR\DBMS. You need to start File Share 2 on the Server with the name GLOBAL. The corresponding Database Reference File is located under $G8WBDDIR\DBMS\GLOBAL.

*Note: FSCOMMS is default set for local use of File Share 2 ($LOCAL). To use a File Share 2 server, i.e. FSCOMMS must be changed to accordingly. You CANNOT simultaniously use File Share 2 in $LOCAL mode and as client towards Server on network.*

*Note: Please refer to MFWB documentation for details on File Share 2.*

## Integrity

Integrity control setting may be used to control whether recovery (commit/rollback) should be applied to the database files. Integrity control is based on File Share 2 Transaction Logging feature.

OFF setting (default), database files opened with recovery disabled and any partial updates are permanent, i.e. abort transaction or program.

ON setting, database files are opened with recovery enabled and all I/O activity is re-directed to File Share 2. Integrity control may be used both in single- and multi-user enviroment.

Single-user: G8WB is prepared for single-user operation and you need only use ON setting to enable both File Share 2 and Transaction Logging. A default $LOCAL Database Reference File is located under $G8WBLDIR\DBMS.

*Note: Transaction Logging write database updates to $G8WBLDIR\DBMS\RECOVER.LOG and this file may become excessively large after a while. You should occationally delete this file, when a clean-point has been reached during testing.*

Multi-user: Integrity control implies concurrency control via File Share 2 also, regardless of concurrency setting. Same comments as mentioned under concurrency control apply. The default GLOBAL server writes database updates to $G8WBDDIR\DBMS\RECOVER.LOG.

*Note: FSCOMMS is default set for local use of File Share 2 ($LOCAL). To use a File Share 2 server, i.e. FSCOMMS must be changed to accordingly. You CANNOT simultaniously use File Share 2 in $LOCAL mode and as client towards Server on network.*

*Note: FSCOMMS is default set for local use of File Share 2 only ($LOCAL). To use a File Share 2 server, i.e. FSCOMMS must be changed to accordingly.*

*Note: Please refer to MFWB documentation for details on File Share 2.*

## Subschema-Section

The Subchema section defines all IDS-II subschema's available within G8WB. During initial translation and validation of a subschema, the subschema configuration is updated automatically. Each subschema is assigned a unique short name (first 6 characters of subschema name) during automatic configuration. Once a subschema short name has been assigned, it MUST not be changed, unless you delete the subschema first. You may control short name assignment by configuring new subschema's manually, before running the initial translation and validation of subschema.

Location Subschema setting define where the Subschema support files are stored.

G8WB will automatically create and configure a subschema during schema validation. This subschema, which is named SS-<schema-name>, represents all full view of the schema. Unload and Load utilities will use this subschema as default. The -AUTO setting indicate the subschema has been generated from schema validation. TP will ignore any subschema with -AUTO setting.

```
;                     Short   Location  Auto
;                     name    Subschema generated
[SUBSCHEMA-SECTION]
DIMS                : DIMS     CURRENT           ; DIMS8 Only
SS-DIMS             : 23SCHS   CURRENT  -AUTO    ; DIMS8 Only
TELEX8              : TELEX8   CURRENT           ; DIMS8 Only
SS-TELEX8           : 24SCHS   CURRENT  -AUTO    ; DIMS8 Only
CLASS               : CLASS    CURRENT           ; SAMPLES Only
SS-CLASS            : 26SCHS   CURRENT  -AUTO    ; SAMPLES Only
TS                  : TS       CURRENT           ; SAMPLES Only
SS-TESTSCM          : 25SCHS   CURRENT  -AUTO    ; SAMPLES Only
```

A maximum of 500 subschema's may be defined.

## Location-Subschema

CURRENT setting will place all subschema support files under $G8WBDDIR\SSCHEMA, which is the default shared database directory.

If you need to work with multiple versions of a subschema, you may use <Path> setting , where <path> can be any valid directory (shared or non-shared). All subschema support files then placed under <path>\SSCHEMA. Please refer to Location-Schema above for more details.

## Default-Section

```
[DEFAULT-SECTION]
DYNAMIC-DEBUG          : ON          ; Dynamic Debug Mode ON/OFF (Drop-in)
DEBUG                  : OFF         ; Debug Mode ON/OFF (Trace)
CHECK                  : ON          ; Subschema Stamp Check ON/OFF
CONCURRENCY-CONTROL    : OFF         ; DB Concurrency Control OFF/ON/FS2
;                                      ON - Standard Record Locking
;                                      FS2 - Use File Share 2
INTEGRITY-CONTROL      : OFF         ; Transaction logging (FS2) OFF/ON
COMP-36-BIT-SUPPORT    : OFF         ; Subschema default OFF/ON/FORCE
DBK-36-BIT-SUPPORT     : OFF         ; Subschema default OFF/FORCE
```

### Dynamic-Debug

DYNAMIC-DEBUG ON will enable the Dynamic IDS-II Debugger (Interactive IIDS drop-in). The Exception-Filter configuration, ref. below, is used to control Dynamic Debugger invocation.

### Debug

DEBUG ON will trigger DBMS to give information for each IDS-II statement (currencies and status). For TP such information will be directed to the Journal Log and for Batch such information will be directed to the screen (with pause prompts).

### Check

CHECK ON will trigger DBMS to perform a Date & Time check similar to GCOS-8. The check will verify inconsistencies in Subschema Translation and Program compilation date & time, which could otherwise lead to unpredictable results, if Subschema is changed without re-compilation of programs.

### Concurrency

Concurrency control setting may be used to define a default concurrency mode for schema's configured without any concurrency setting.

### Integrity

Integrity control setting may be used to define a default integrity mode for schema's configured without any integrity setting.

### COMP-36-bit-Support

COMP-36-BIT-SUPPORT setting defines how COMP-6 fields should be translated for a subschema. This setting apply to all subschema's, however you may also specify setting in the DDL source via SET directives.

Subschema translator, by default, treat COMP-6 fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate translation mode, i.e. *W8PREP ALT active in DDL source, COMP-6 fields are implemented as 32 bit unsigned binary fields.

ON setting may be used to change alternate translation, so COMP-6 fields are implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physical size of a COMP-6 field will however change from 4 to 5 bytes.

FORCE setting may be used to change default translation, so COMP-6 fields are implemented as 39 bit signed binary fields. In alternate translation, COMP-6 fields are implemented as 31 bit signed binary fields.

*Note: You should only enable 36-bit-support if you really need it, and you must pay attention to potential problems with redefines in Working storage of programs, due to the change in physical field size.*

### DBK-36-bit-Support

DBK-36-BIT-SUPPORT setting defines how DB-KEY fields should be translated for a subschema. This setting apply to all subschema's, however you may also specify setting in the DDL source via SET directives.

Subschema translator, by default, treat DB-KEY fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate translation, i.e. *W8PREP ALT active in DDL source, DB-KEY fields are implemented likewise.

FORCE setting may be used to change translation, so DB-KEY fields are by default implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physizal size of a DB-KEY field will howver change from 4 to 5 bytes. In alternate translation, DB-KEY fields are implemented as 31 bit signed binary fields.

*Note: 36-bit-support is ONLY relevant, although this is not very likely, if you unload from host databases with VERY large DBK values, because original DBK's are re-used within G8WB. New DBK's are assigned as highest DBK + 1. A locally created database will thus start with DBK = 1.*

## Exception-Filter

The Exception-Filter configuration may be used to specify the DB-STATUS values, which should not invoke the Dynamic Debugger (exclude) or alternately the DB-STATUS values, which should (include).

```
 [EXCEPTION-FILTER]
;DB-Status to Include/Excluded in dynamic debug (Max. 10)
exclude                : 0102100                ; End-of-Set
exclude                : 0502100                ; End-of-Set
exclude                : 0502400                ; Record-not-found
exclude                : 0502401                ; Record-greater-found
exclude                : 1505100                ; Duplicate-exist
```

A maximum of 10 values may be configured.

### Backup-Section

The Backup section defines the the backup-sets available for backup and restore of Schema's (databases). Each backup-set is identified by a meaningful name (external), which in turn is assigned to a unique number (internal). A maximum of 99 backup-sets may be defined.

Location setting define where backup-set are stored and whether backup-sets can be shared or not.

Compression setting define whether database should be compressed during backup.

```
[BACKUP-SECTION]
;Backup-Set               No.   Location          [Compression]
;                        (1-99) LOCAL|SCHEMA|path   -PAC|-ZIP
REFERENCE              : 1      SCHEMA                  ; Reference set
WORKPAC                : 2      LOCAL               -PAC; Working set
WORKZIP                : 3      LOCAL               -ZIP; Working set
DIMS8-REFERENCE        : 98     SCHEMA              -PAC; DIMS8 Only
SAMPLES-REFERENCE      : 99     SCHEMA              -PAC; SAMPLES Only
```

In above example, REFERENCE is a shared backup-set which have been prepared by one developer or administrator and made available for restore by any other developers in a networked G8WB environment. WORKPAC and WORKZIP are examples of non-shared local backup-set, which can be used by all developers, however each developer have their own physical backup-set.

SAMPLES-REFERENCE and DIMS8-REFERENCE are predefined backup-set for use with SAMPLES and/or DIMS8.

## Location

LOCAL setting will place the backup-set under $G8WBLDIR\DBMS, where you usually have your working database (non-shared).

SCHEMA setting will place the backup-set under $G8WBDDIR\DBMS or the directory where Schema is located (shared).

Path setting will place the backup-set under the specified path (shared or non-shared).

## Compression

Default setting (non-compressed) will backup the database (4 indexed files - 8 files in total) using OS COPY command.

-PAC setting will backup the database using CrossePAC.

-ZIP setting will backup the database using PKZIP.

A reduction in size of 70-80% can be achieved with compressed backup-set. Elapsed time for compressed backup operation may be is 50-100% longer than non-compressed backup, whereas elapsed time for restore operation may be 25-50% longer than non-compressed restore.

*Note: Compression setting is ONLY used during backup operation, and setting may thus be changed at any time. Restore operation will automatically determine which type of restore to perform.*

## Maximum-Section

The Maximum section document the maximum values supported by DBMS. All memory allocation is dynamic, however DBMS may fail to allocate memory before below limits are reached, if insufficient memory is available to G8WB.

```
[MAXIMUM-SECTION]
;                        Not configurable
SCHEMA                 : 99             ; Maximum Schema's
SCHEMA                 : 500            ; Maximum Subschema's
KEY-LENGTH             : 200            ; Maximum Schema Key Length (CALC/INDEX)
RECORD-LENGTH          : 4082           ; Maximum Schema Record Length
UWA-LENGTH             : 64512          ; Maximum UWA size in bytes
REALMS                 : 500            ; Maximum Schema Realms
RECORDS                : 1000           ; Maximum Schema Records
SETS                   : 1000           ; Maximum Schema Sets
KEYS                   : 100            ; Maximum Schema Keys
FIELDS                 : 10000          ; Maximum Schema Fields approx.
SS-FIELDS              : 6000           ; Maximum Subschema Fields (<> Schema)
SET-LEVELS             : 25             ; Maximum Schema Set Selection Levels
VERB-FIELDS            : 100            ; Maximum Fields in DML verb
```

# INTEREL DBMS

## Location-Section

The Location section defines all INTEREL Models available within G8WB. An INTEREL Model is referred to as a Location within G8WB. Whenever you create a new Location within XDB, you need to include the Location in the location section configuration.

```
;
; INTEREL/XDB Configuration Version 1.2.00 (Server Config)
;
[LOCATION-SECTION]
;Location/Model name     [Documentation only]
CLASS                  : "Workshop Tutorial"
TUTORIAL               : "XDB Tutorial"
```

## Backup-Section

The Backup section defines the the backup-sets available for backup and restore of Locations (databases). Each backup-set is identified by a meaningful name (external), which in turn is assigned to a unique number (internal). A maximum of 99 backup-sets may be defined.

Backup of INTEREL databases include both the model description, i.e. tables, columns, indexes, and the database content.

Location setting define where backup-set are stored and whether backup-sets can be shared or not.

Compression setting define the type of compression to be used during database backup.

```
[BACKUP-SECTION]
;Backup-Set               No.   Location          [Compression]
;                       (1-99)                     -PAC|-ZIP
;                            $G8WBLDIR\DBMS (example of local path) (default)
;                            $G8WBDDIR\DBMS (example of global path)
REFERENCE              : 1   $G8WBLDIR\DBMS -PAC ; Reference set
WORK                   : 2   -ZIP                ; Working set
SAMPLES-REFERENCE      : 99                      ; SAMPLES Only
```

## Location

$G8WBLDIR\DBMS (default) setting will place the backup-set under your local G8WB directory, thus the backup-set will be non-shared.

$G8WBDDIR\DBMS setting will place the backup-set under your global G8WB database directory, thus the backup-set can be shared.

Path setting will place the backup-set under the specified path (shared or non-shared).

*Note: Since backup-sets include both model description and data, a restore from a shared backup-set may be used to establish a new location, however, BEFORE performing the restore you MUST create the Location in XDB, otherwise XDB will not recognize the new location.*

## Compression

The XDB structure for a Model/Location is quite complex and involves quite a number of files. As such, backup-set are always created through compression software.

-PAC (default) setting will backup the database structure using CrossePAC.

-ZIP setting will backup the database structure using PKZIP.

*Note: Compression setting is ONLY used during backup operation, and setting may thus be changed at any time. Restore operation will automatically determine which type of restore to perform.*

## GCOS8-Catalogs

The GCOS8-Catalog section may be used to define host catalog-strings, which are used by G8WB when building unload/load scripts for the host.

MODEL setting define how logon to INTEREL ISQL should be done. Default is <model-name>.

SAVE setting define how SAVE files should be accessed within INTEREL ISQL. Default is /<save-file-name>.

```
[GCOS8-CATALOGS]
;Type                    Catalog-string (used for Import/Export only)
;MODEL                 : UMC/CAT
;SAVE                  : UMC/CAT/SAVE
```

## TP8 Workstation

The TP8 Workstation configuration is very much similar to a GCOS-8 Workstation, however the directives have been reduced to a minimum and a new PROFILE concept is introduced to ease configuration (reduce duplication of information). In the following the configuration has been split into sections for easy of reading.

During TP Monitor start-up, the configuration is processed and tables are allocated dynamically. Any error detected in the configuration is logged on the screen with a pause prompt.

## Create_Workstation

```
;
; TP8 System Generation Information Version 1.2.00 (Server Config)
;
[CREATE_WORKSTATION]
WS_NAME                 : TESTTP
```

WS_NAME define the workstation name, which is communicated to TPR's via TP-STORAGE and the .WSNAM executive call.

## Create_TP8_Extension

```
[CREATE_TP8_EXTENSION]
MAX_COMMAND_NAME_SIZE   : PROCEDURE             ; 1-8 or PROCEDURE
COMMON-UWA              : OFF                   ; OFF/DMIV/ON
OUTPUT_MSG_SIZE         : 6144                  ; 100-6144
MAX_FORM_FIELDS         : 128                   ; TPFF only
PRG_PROCEDURE_PROGRAM   : W8CXPP                ; Standard CXPP
;PRG_PROCEDURE_PROGRAM   : RFVCXPP               ; Pilspraak/RFV CXPP
ADMIN_PROCEDURE_PROGRAM : W8CXAP                ; Standard CXAP
;ADMIN_PROCEDURE_PROGRAM : RFVCXAP               ; Pilspraak/RFV CXAP
STARTUP_FILE            : $G8WBDIR\TPMS\STARTUP ; Standard Startup
TERMINATION_FILE        : $G8WBDIR\TPMS\TERM    ; Standard Termination
TPR_POOL                : 3                     ; # TPRs (0-9) [FIFO|RANDOM]
CBL85-TPR-CANCEL        : ON                    ; COBOL-85 TPR reuse
```

## Command_Max_Size

MAX_COMMAND_NAME_SIZE setting define the size in characters of the Command (leading characters of input message). Fixed length of 1 to 8 characters or PROCEDURE may be specified for variable length Command. PROCEDURE setting require the standard or own developed Admin_Procedure_Program is enabled. Variable length recognizes space or Terminal specific tabulator as delimiters.

## Common_UWA

COMMON_UWA setting define TP8 handling of UWA whenever a subschema change occur within a phase.

OFF setting (default), will initialize all current UWA's whenever a change of subschema occur.

DMIV setting, will initialize all but the last used UWA whenever a change of subschema occur.

ON setting, will prevent current UWA's from be initialized whenever a change of subschema occur.

*Note: All current UWA's are always initialized at Start of TX or Phase.*

## Output_Msg_Size

OUTPUT_MSG_SIZE setting define the size in characters of the message buffer. Length of 100 to 6144 characters may be specified. Setting control both maximum output and input message size.

## Max_Forms_Fields

MAX_FORMS_FIELDS setting define the maximum number of fields for a TPFF form.

## PRG_Procedure_Program

PRG_PROCEDURE_PROGRAM setting define the name of the CXPP procedure program. The default setting W8CXPP has no functionality except for trapping and documenting any procedure calls from TPR's.

For Pilspråk, you must enable the RFVCXPP module.

## Admin_Procedure_Program

ADMIN_PROCEDURE_PROGRAM setting define the name of the CXAP procedure program. The default setting W8CXAP has no functionality except for the standard variable length command process, and trapping and documenting any other procedure calls defined for Command(s).

For Pilspråk, you must enable the RFVCXAP module.

### Startup_File

STARTUP_FILE setting define a transaction file to be executed during Workstation startup. Such file must contain one or more messages in the same format as typed on the screen. The default setting $G8WBDIR\TPMS\STARTUP is an empty file.

### Termination_File

TERMINATION_FILE setting define a transaction file to be executed during Workstation termination. Such file must contain one or more messages in the same format as typed on the screen. The default setting $G8WBDIR\TPMS\TERM is an empty file.

### TPR_Pool

TPR_POOL setting may be used to aid testing. In TP8 a common cause of error is the difference between execution within a fresh or re-used copy of a TPR. The TPR_POOL specifies how many entries will be left "Un-canceled" in memory and the FIFO (First-in-First-Out) or RANDOM setting control how entries are entered into the TPR-POOL. Let’s look at the three cases.

### Fresh TPR Copy always

This is achieved by setting TPR_POOL to zero.

### Re-used TPR Copy Always

This is achieved by setting the TPR_POOL number high enough to accommodate the number of TPR your are testing within your transaction. FIFO must also be set.

*Note: The TPR_POOL number combined with your TPR sizes defines the additional memory required for operation. The easiest way is simply to try your setting and if you experience run-time errors (not enough memory), you either decrease your setting or upgrade you memory configuration.*

### Random TPR Copy

This will resemble conditions in a real production environment, where the allocation of fresh or re-used copies are unpredictable, unless controlled via explicit CANCL.

This is achieved by setting the TPR_POOL number to a reasonable number together with RANDOM.

*Note: The TPR load status is indicated in the Journal Log if trace is activated.*

### CBL85-TPR-Cancel

CBL85-TPR-CANCEL setting define whether COBOL-85 TPR’s should be cancelled after use (default).

ON setting will enable same handling as for COBOL-74 TPR’s, including TPR-Pool handling.

### Create_Global_Storage

Each configuration line define a Global_Storage with Name and Size specification.

```
 [CREATE_GLOBAL_STORAGE]
;Name                    Size in bytes
CNST                   : 200
```

Each Global_Storage is allocated during Workstation startup and initialized to low-values. Initialization with application values must be performed via the $RDY command, which must call .DEFNG for each Global_Storage.

## Create_TX_Storage

Each configuration line define a TX_Storage with Name, Size and optionally implicit send/recieve specification.

```
 [CREATE_TX_STORAGE]
;                        Only required if Implicit Receive/Send used
;            Tx-size  [R offset size] [S offset size]
;Name      : in bytes    in bytes        in bytes
TEST3      : 160       R 0      80     S 80     80
```

TX_Storage configuration is only mandatory when implicit send/receive specification is required. All sizes are given in bytes and offsets are relative to zero.

## Create_Profile

The PROFILE may be used to define common attributes for a transaction and you may then reference the profile name when defining your transactions.

Administrative User Procedures are implemented using a unique number, which will invoke the required functionality in the Admin_Procedure_Program. Please refer to Technical Guide for details on how to implement such procedures.

```
[CREATE_PROFILE]
;                        Define profiles for CREATE_COMMAND
;Profile   [Abort-TPR] TX-size   [No-edit] [Constant] [Procedure no. 1-999]
;Name      Def=TP-ABT  bytes/Name           Name      Rcv Snd TXi PHi PHt TXt
******** :              500                 CNST   ; Default Profile
TEST3    :              TEST3   NO-EDIT     CNST   ; SAMPLES Only
RFV      :                500   NO-EDIT               122 132 300 500 ;Pilspraak
TPFF     :                500   NO-EDIT     CNST   ; SAMPLES/TPFF Profile
FORMAT   :                500   NO-EDIT     CNST   ; SAMPLES/FORMAT Profile
TSM8     :              10200   NO-EDIT     CNST   ; SAMPLES/TSM8   Profile
DIMS8    :     DMS8AB   15000   NO-EDIT     CNST   ; SAMPLES/DIMS8 Profile
```

Each configuration line define a Profile with Name, optional Abort-TPR, TX-storage size in bytes or TX_Storage name, optional No-edit, optional Global_Storage name and optional Procedure-Setting.

For Pilspråk, procedures must be given, ref. example RFV profile.

### Abort-TPR

Abort-TPR must be specified when TX_Storage name is used.

### No-Edit

No-edit may be used to suppress insertion of Carriage-Return & Line-Feed sequences in the output message.

### Procedures

When procedure number are specified, all types (Rcv, Snd, TXi, PHi, PHt and TXt) must be given. Procedure number must either correspond to a procedure defined in the Admin_Procedure_Program or set to 0 (zero).

## Default Profile

"*******" designates the default profile, which will be assigned if no explicit profile name is given in the Create_Command configuration.

## Create_Command

Each configuration line define a Command with Name, First TPR and optional Profile specification.

```
[CREATE_COMMAND]
;
;Message-id   First-TPR Profile
********    :                  ; Default for TPR started as TX
;$RDY        : WRDYTPR          ; Required for Global-Storage initialize
DIMS8       : DIMSTX DIMS8     ; Required for DIMS8
FCT         : TSBASE TSM8      ; Required for TSM8
TSM         : TSBASE TSM8      ; Required for TSM8
TEST3       : WTEST3    TEST3  ; SAMPLES Only
TM001       : TM001     TPFF   ; SAMPLES Only
TM002       : TM002     TPFF   ; SAMPLES Only
TM003       : TM003     TPFF   ; SAMPLES Only
;TM001       : TM001     FORMAT; SAMPLES Only
;TM002       : TM002     FORMAT; SAMPLES Only
;TM003       : TM003     FORMAT; SAMPLES Only
;TM001       : TM001     RFV   ; SAMPLES Only
;TM002       : TM002     RFV   ; SAMPLES Only
;TM003       : TM003     RFV   ; SAMPLES Only
;TM001       : TM001     DIMS8 ; SAMPLES Only
;TM002       : TM002     DIMS8 ; SAMPLES Only
;TM003       : TM003     DIMS8 ; SAMPLES Only
;TM001       : TM001     TSM8  ; SAMPLES Only
;TM002       : TM002     TSM8  ; SAMPLES Only
;TM003       : TM003     TSM8  ; SAMPLES Only
```

When no profile is given the default (********) profile is used.

For DIMS8 the DIMS8 command is required.

For TSM8 the FCT and TSM commands are required.

## Default Command

"*******" designates the default Command, which will be assigned if a non-configured Command is entered. The First TPR is then assumed to be equal to the entered Command, i.e. TEST1 transaction could thus be run either by entering TEST1 or WTEST1.

## $RDY Command

The $RDY command must be defined when you require Global_Storage(s) to be initialized during Workstation startup (CALL .DEFNG). READY/FINISH of database is not required, however if you execute READY TPR's, these may execute correctly provided they can accept that the database is already open.

*Note: Use of READY TPR's are not recommend as they will trigger load of Schema and Subschema structures, and a physical open of the database. These activities are otherwise deferred until the first TPR/Program which require access. This deferred technique will make TP8 startup faster and optimize use of memory.*

## Create_Source_Lid

Each configuration line define a source Logical-Id by it's Name (4 characters). Source LID's are used for log-on verification.

```
[CREATE_SOURCE_LID]
;Name
MAST
```

*Note: The default Logical-ID, ref. above, will be inserted automatically and you may have an empty Create_Source/Destination_Lid configuration.*

## Create_Destination_Lid

As G8WB is a single user environment there is little need for describing a network of terminals, however when your application utilize spawn transaction, send to slave or ROP printers, such Logical-Id's must be defined.

```
[CREATE_DESTINATION_LID]
;Name                     [SLAVE_D_LID|SPAWN_LID]
SLAV                   :  SLAVE_D_LID
POUT                   :
SPWN                   :  SPAWN_LID
```

Each configuration line define a destination Logical-Id by it's Name (4 characters) and optionally it's type.

SLAVE_D_LID designates the Slave Destination LID for the workstation.

SPAWN_LID designates a Spawn LID for TPR Transaction spawn use.

When no type is given, the LID is by default a Receive-Only destination LID.

## Create_TPR

Each configuration line define a TPR by Name or Path to be loaded as a resident TPR.

```
[CREATE_TPR]
;TPR or pathname       Loaded as resident
;DIMSTX                                       ; DIMS8 Optimization
;DIMSOT                                       ; DIMS8 Optimization
;DMEXTP                                       ; DIMS8 Optimization
```

For DIMS8, you should enable the DIMSTX, DIMSOT, DMEXTP configuration for optimal performance.

## Associate_TPR

Each configuration line define a TPR association rule. This feature may be used to load an alternate (associated) TPR, while retaining the original TPR name in CURRENT-TPR.

```
[ASSOCIATE_TPR]
;TPR Name       Associate TPR Name
;T1             : T          ; Example
;T2             : T          ; Example
```

In above example, the “T” TPR is loaded when “T1” or “T2” TPR is requested.

## Create_LIBRARY

Each configuration line define a TPR library to be opened during TP8 Start-up.

```
[CREATE_LIBRARY]
;Library pathname      Opened during TP8 Start-up
```

# DMIV-TP SYSGEN

The DMIV-TP SYSGEN configuration is very much similar to a GCOS-8 SYSGEN, however the directives have been reduced to a minimum and a new PROFILE concept is introduced to easy configuration (reduce duplication of information). In the following the configuration has been split into sections for easy of reading.

During TP Monitor start-up, the configuration is processed and tables are allocated dynamically. Any error detected in the configuration is logged on the screen with a pause prompt.

## TP-Section

```
;
; DMIV-TP System Generation Information Version 1.2.00 (Server Config)
;
[TP-SECTION]
SYSTEM-NAME            : TESTTP
MESSAGE-ID-SIZE        : VARYING               ; 1-8 or VARYING
COMMON-UWA             : OFF                   ; OFF/ON
MESSAGE-BUFFER-SIZE    : 2000                  ; 100-6144
MAX-FORM-FIELDS        : 64                    ; TPFF only
USER-DERAIL-PROGRAM    : W8USRDRL              ; Standard ZE64
USER-PROCEDURE-PROGRAM : W8USRPRC              ; Standard ZE64
STARTUP-FILE           : $G8WBGDIR\TPMS\STARTUP ; Standard Startup
TERMINATION-FILE       : $G8WBGDIR\TPMS\TERM    ; Standard Termination
TPR-POOL               : 3                     ; # TPRs (0-9) [FIFO|RANDOM]
```

### System-Name

SYSTEM-NAME setting define the DMIV-TP system name, which is communicated to TPR's via TP-STORAGE.

### Message-Id-Size

MESSAGE-ID-SIZE setting define the size in characters of the Message-Id (leading characters of input message). Fixed length of 1 to 8 characters or PROCEDURE may be specified for variable length Command. PROCEDURE setting require the standard or own developed User-Procedure-Program is enabled. Variable length recognizes space or Terminal specific tabulator as delimiters.

### Common-UWA

COMMON-UWA setting defines whether Common-Uwa feature should be enabled for DMIV-TP.

### Message-Buffer-Size

MESSAGE-BUFFER-SIZE setting define the size in characters of the message buffer. Length of 100 to 6144 characters may be specified. Setting control both maximum output and input message size.

## Max-Forms-Fields

MAX-FORMS-FIELDS setting define the maximum number of fields for a TPFF form.

## User-Derail-Program

USER-DERAIL-PROGRAM setting define the name of the User-Derail program. The default setting W8USRDRL has no functionality except for trapping and documenting any procedure calls from TPR's.

## User-Procedure-Program

USER-PROCEDURE-PROGRAM setting define the name of the User-Procedure program. The default setting W8USRPRC has no functionality except for the standard variable length command process, and trapping and documenting any other procedure calls defined for Message-id(s).

## Startup-File

STARTUP-FILE setting define a transaction file to be executed during DMIV-TP startup. Such file must contain one or more messages in the same format as typed on the screen. The default setting $G8WBDIR\TPMS\STARTUP is an empty file.

## Termination-File

TERMINATION-FILE setting define a transaction file to be executed during DMIV-TP termination. Such file must contain one or more messages in the same format as typed on the screen. The default setting $G8WBDIR\TPMS\TERM is an empty file.

## TPR-Pool

TPR-POOL setting may be used to aid testing. In DMIV-TP a common cause of error is the difference between execution within a fresh or re-used copy of a TPR. The TPR-POOL specifies how many entries will be left "Un-canceled" in memory and the FIFO (First-in-First-Out) or RANDOM setting control how entries are entered into the TPR-POOL. Let’s look at the three cases.

## Fresh TPR Copy always

This is achieved by setting TPR-POOL to zero.

## Re-used TPR Copy Always

This is achieved by setting the TPR-POOL number high enough to accommodate the number of TPR your are testing within your transaction. FIFO must also be set.

*Note: The TPR-POOL number combined with your TPR sizes defines the additional memory required for operation. The easiest way is simply to try your setting and if you experience run-time errors (not enough memory), you either decrease your setting or upgrade you memory configuration.*

## Random TPR Copy

This will resemble conditions in a real production environment, where the allocation of fresh or re-used copies are unpredictable, unless controlled via explicit CANCL.

This is achieved by setting the TPR-POOL number to a reasonable number together with RANDOM.

*Note: The TPR load status is indicated in the Journal Log if trace is activated.*

---

## Constant-Storage

Each configuration line define a Constant-Storage with Name and Size specification.

```
[CONSTANT-STORAGE]
;                        Only required if Constant Storage is used
;Name                    Size in bytes [Value load program]
CNST                   : 200            WCNST
```

Each Constant-Storage is allocated during DMIV-TP startup and initialized to low-values. Initialization with application values may either be performed thorugh a Value Load Program or via TPR calls to .CHANC.

## TX-Storage

Each configuration line define a TX-Storage with Name, Size and optionally implicit send/recieve specification.

```
[TX-STORAGE-SECTION]
;                        Only required if Implicit Receive/Send used
;            Tx-size  [R offset size] [S offset size]
;Name      : in bytes    in bytes        in bytes
TEST3      : 160       R 0      80     S 80     80
```

TX-Storage configuration is only mandatory when implicit send/receive specification is required. All sizes are given in bytes and offsets are relative to zero.

## Profile

Each configuration line define a Profile with Name, optional Abort-TPR, TX-storage size in bytes or TX-Storage name, optional No-edit, optional Constant-Storage name and optional Procedure-Setting.

```
[PROFILE-SECTION]
;                        Define profiles for TRANSACTION-CONTROL
;Profile   [Abort-TPR] TX-size   [No-edit] [Constant] [Procedure no. 1-999]
;Name      Def=TP-ABT  bytes/Name           Name      Rcv Snd TXi PHi PHt TXt
******** :              10000               CNST
TEST3    :              TEST3                      ; SAMPLES Only
TPFF     :                500   NO-EDIT     CNST   ; SAMPLES/TPFF Profile
```

## Abort-TPR

Abort-TPR must be specified when TX-Storage name is used.

## No-Edit

No-edit may be used to suppress insertion of Carriage-Return & Line-Feed sequences in the output message.

## Procedures

When procedure number are specified, all types (Rcv, Snd, TXi, PHi, PHt and TXt) must be given. Procedure number must either correspond to a procedure defined in the User-Procedure-Program or set to 0 (zero).

## Default Profile

"*******" designates the default profile, which will be assigned if no explicit profile name is given in the Transaction-Control configuration.

## Transaction Control

Each configuration line define a Message-Id with Name, First TPR and optional Profile specification.

```
[TRANSACTION-CONTROL]
;
;Message-id   First-TPR Profile
********    :
TEST1       : WTEST1              ; SAMPLES Only
TEST2       : WTEST2              ; SAMPLES Only
TEST3       : WTEST3    TEST3     ; SAMPLES Only
WTEST3      : WTEST3    TEST3     ; SAMPLES Only
TEST4       : WTEST4              ; SAMPLES Only
TM001       : TM001     TPFF      ; SAMPLES Only
TM002       : TM002     TPFF      ; SAMPLES Only
TM003       : TM003     TPFF      ; SAMPLES Only
```

When no profile is given, the default (********) profile is used.

### Default Message-Id

"*******" designates the default Message-Id, which will be assigned if a non-configured Message-Id is entered. The First TPR is then assumed to be equal to the entered Message-Id, i.e. TEST1 transaction could thus be run either by entering TEST1 or WTEST1.

## Communication

As G8WB is a single user environment there is little need for describing a network of terminals, however when your application utilize spawn transaction, send to slave or ROP printers, such Logical-Id's must be defined.

```
[COMMUNICATION-SECTION]
[OPERATOR-CONTROL]
;                         [SLV|ROP|SPN]
MAST                    :
SLAV                    :  SLV
POUT                    :      ROP
SPWN                    :          SPN
```

Each configuration line define a source or destination Logical-Id by it's Name (4 characters) and optionally it's type.

SLV designates the Slave Destination LID for DMIV-TP.

SPN designates a Spawn LID for TPR Transaction spawn use.

ROP designates a Receive-Only destination LID.

When no type is specified, the LID is by default a Source LID. Source LID's are used for log-on verification.

*Note: The default Logical-ID, ref. above, will be inserted automatically and you may have an empty Communication-Section configuration.*

## TPR-Section

Each configuration line define a TPR by Name or Path to be loaded as a resident TPR.

```
[TPR-SECTION]
;TPR or pathname       Loaded as resident
```

## LIBRARY-Section

Each configuration line define a TPR library to be opened during DMIV-TP Start-up.

```
[LIBRARY-SECTION]
;Library pathname      Opened during DMIV-TP Start-up
```

# Terminal Emulator Configuration

The Terminal Emulator configuration may be used to control the functionality of the standard terminal emulators provided with G8WB.

```
;
;  Terminal Emulator Configuration Version 1.2.00 (Server Config)
;
[W8V77XXE]
;                                        VIP77XX Emulator
ATTRIBUTE              : EXTENDED      ; STANDARD/EXTENDED Visual attributes

[W8V78XXE]
;                                        VIP78XX Emulator

[W8DKUXXE]
;                                        DKU71XX Emulator
MODE                   : SDP           ; Default SDP or VIP Mode

[W8IBMXXE]
;                                        IBM327X Emulator
```

## VIP77XX Emulator

### Attribute

ATTRIBUTE setting define whether standard (Blink & Blank only) or extended (QUESTAR Phase 1) visual attributes are being used.

## DKU71XX Emulator

### Mode

MODE setting define whether default mode is VIP or SDP.

# Environment Configuration

The Environment configuration may be used to request programs/modules to be loaded automatically during TP Monitor or Batch program start-up.

An example of use is a migrated GCOS-8 GMAP module, which contain multiple Entry-points. After migrating the module functionality into a MF COBOL/2 module, with a primary entry-point and multiple secondary entry-points (referenced by application modules/TPRs/programs), the primary Entry-point (program name) need to be specified in the environment configuration.

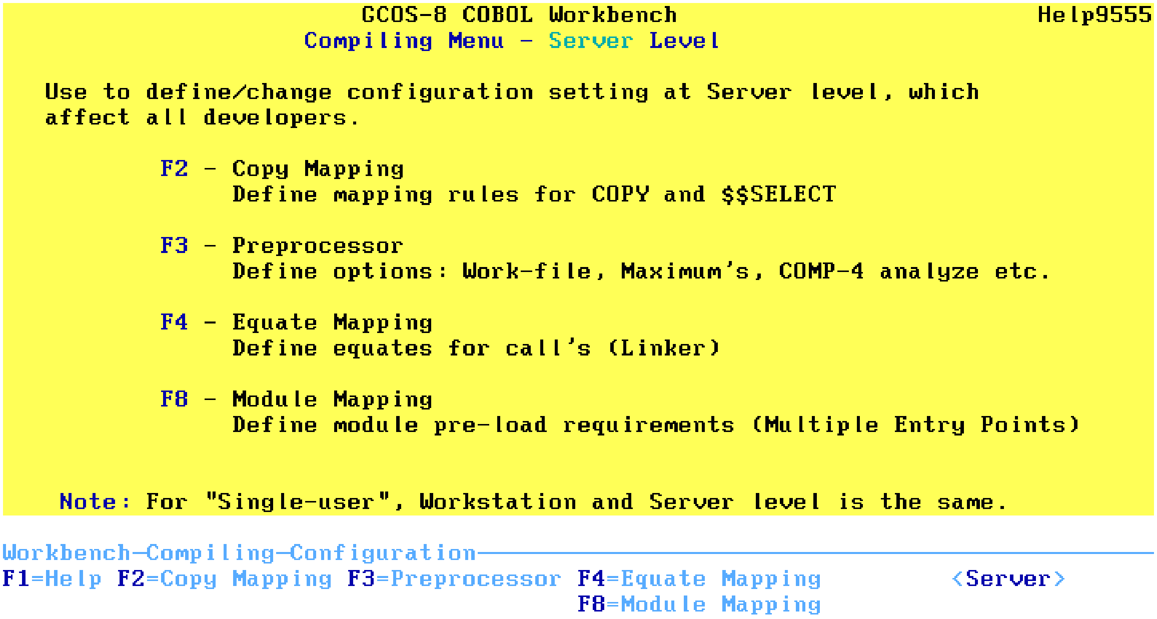
Another example is application calls to modules, which are only relevant in the GCOS-8 enviroment. A dummy program with corresponding entry-points, may then be used to ignore or emulate such module calls.

```
;
; Environment Configuration Version 1.2.00 (Server Config)
;
[TP-SECTION]
; Program, pathname or library  Pre-loaded during TP Start-up

[BATCH-SECTION]
; Program, pathname or library  Pre-loaded during Program Start-up
```

DIMS8 will use environment configuration to open TP and Batch libraries containing the DIMS8 software (host version running under G8WB).

```
;
; Environment Configuration Version 1.2.00 (Server Config)
;
[TP-SECTION]
; Program, pathname or library  Pre-loaded during TP Start-up
$G8WBDIR\LBR\DIMS8TP.LBR ; DIMS8 Only
$G8WBDIR\LBR\DIMS8SI.LBR ; DIMS8 Only

[BATCH-SECTION]
; Program, pathname or library  Pre-loaded during Program Start-up
$G8WBDIR\LBR\DIMS8BA.LBR ; DIMS8 Only
```

## Compiling Configuration



```
                          GCOS-8 COBOL Workbench                           Help9555
                      Compiling Menu - Server Level

   Use to define/change configuration setting at Server level, which
   affect all developers.

           F2 - Copy Mapping
                Define mapping rules for COPY and $$SELECT

           F3 - Preprocessor
                Define options: Work-file, Maximum's, COMP-4 analyze etc.

           F4 - Equate Mapping
                Define equates for call's (Linker)

           F8 - Module Mapping
                Define module pre-load requirements (Multiple Entry Points)


    Note: For "Single-user", Workstation and Server level is the same.

Workbench—Compiling—Configuration————————————————————————————————————————————————
F1=Help F2=Copy Mapping F3=Preprocessor F4=Equate Mapping          <Server>
                                        F8=Module Mapping
```

### Copy Mapping

Copy Mapping configuration is used by the G8WB Preprocessor to resolve COPY statements as follows:

COPY name [OF Library]

- The first two characters of the library is used as a Library-Tag ([xx]), which the G8WB Preprocessor will search for. The default Library-Tag is [.L], when no library is specified.

- Next the G8WB Preprocessor will search for a name match within the Library-Tag. If found, the associated file is searched for within the DEFAULT directories, and then expanded.

- If the name is longer than 8 characters, the search will stop and an error message will be given.

- Next the G8WB Preprocessor will search, in the DEFAULT directories associated with the Library-Tag, for file <name.CPY>. If found, the file is expanded.

- Next the G8WB Preprocessor will search $G8WBFDIR\FMS\CPY for  file <name.CPY>, which hold TPFF, FORMAT and Pilspråk generated copy members. If found, the file is expanded.

- As it's final resort, the G8WB Preprocessor will search the current directory for file <name.CPY>. If found, the file is expanded, otherwise an error message in given.

COPY Mapping is normally only done at Server level, however, if non-unique library-tag’s are used, you may use Project level configuration. Please refer to details described above under Project level configuration.

DEFAULT may specify one or more directories. For multiple directories the search takes place left-to-right until the file is located.

Automatic maintenance of the Copy Mapping configuration may be achieved through the use of Copy Import Utility, ref. File-Transfer chapter for details.

```
;
;  Preprocessor Copy Name => File Configuration Version 1.2.00 (Server)
;
; ------------------------------------------------------------------------
;  GCOS-8 COBOL-74/COBOL-85 Copy support feature
;
;  Tag = first two characters of COPY <name> OF xx....
;        .L default for COPY <name>.
;
;  Copy Names of max. 8 characters need not be specified, when
;  placed in the TAG default-directory or current directory
;  with extension = .CPY
;
;  DEFAULT-DIRECTORY           : <directory> or
;                                <dir1+dir2...> Searched left-to-right
;
;  AUTOREF-CONTROL             : nnnn Used/Created by Copy Import
;
```

## DMIV-TP

A Standard pre-configured DMIV-TP COPY library is provided. These standard copies are pre-preprocessed, however they may customized as required.

```
[TP-DMIV]
;                                Standard DMIV-TP copies
DEFAULT-DIRECTORY              : $g8wbdir\tpmscpy\dmiv
AUTOREF-CONTROL                : 0000
TP-STORAGE                     : tpstorag.cpy
LID-DATA                       : liddata.cpy
SLEEP-DATA                     : slpdata.cpy
WAKE-DATA                      : wakedata.cpy
TP-INFO                        : tpinfo.cpy
INPUT-MSG                      : cdinput.cpy
EXPLICIT-INPUT-MSG             : xcdinput.cpy
OUTPUT-MSG                     : cdoutput.cpy
EXPLICIT-OUTPUT-MSG            : xcdoutpu.cpy
```

## TP8

Standard pre-configured TP8 COPY libraries are provided. Separate libraries are used to support COBOL-74 and COBOL-85. These standard copied are pre-preprocessed, however they may customized as required. For COBOL-74 you may want to change default TP-STORAGE mapping, which use the Extended verison of TP-STORAGE.

```
[TP-C74]
;                                Standard TP8 copies (COBOL-74)
;          TP8     TP-STORAGE -> ..\AREF0016.CPY
;          TP8 Ext TP-STORAGE -> $g8wbdir\tpmscpy\c85\AREF0037.CPY (Default)
;
DEFAULT-DIRECTORY              : $g8wbdir\tpmscpy\c74
AUTOREF-CONTROL                : 0019
ABT-CODES                      : AREF0001.CPY
AJ-GLOBAL-STORAGE              : AREF0002.CPY
EXPL-RECV-CD                   : AREF0003.CPY
EXPL-SEND-CD                   : AREF0004.CPY
EXPLICIT-INPUT-MSG             : AREF0005.CPY
EXPLICIT-OUTPUT-MSG            : AREF0006.CPY
INPUT-MSG                      : AREF0007.CPY
LVL3ID-A1                      : AREF0008.CPY
OUTPUT-MSG                     : AREF0009.CPY
SCRB-BLOCK                     : AREF0010.CPY
SIRB-BLOCK                     : AREF0011.CPY
SLEEP-DATA                     : AREF0012.CPY
SRRB-BLOCK                     : AREF0013.CPY
STS-ERR-MSGS                   : AREF0014.CPY
SWTM-DATA                      : AREF0015.CPY
;TP-STORAGE                     : AREF0016.CPY ; Normal COBOL-74
TP-STORAGE                     : $g8wbdir\tpmscpy\c85\AREF0037.CPY ; Extended C
OBOL-85
TPDF-DATA                      : AREF0017.CPY
TQSVC-DATA                     : AREF0018.CPY
WAKE-DATA                      : AREF0019.CPY
DBWFLREC                       : $g8wbdir\dbmscpy\dbwflrec.cpw ; Support DBWFL
```

```
[TP-C85]
;                                Standard TP8 copies (COBOL-85)
DEFAULT-DIRECTORY              : $g8wbdir\tpmscpy\c85
AUTOREF-CONTROL                : 0046
DELAY-DATA                     : AREF0001.CPY
DEQUE-DATA                     : AREF0002.CPY
ENQUE-DATA                     : AREF0003.CPY
EXPL-RECV-CD                   : AREF0004.CPY
EXPL-SEND-CD                   : AREF0005.CPY
EXPLICIT-INPUT-MSG             : AREF0006.CPY
EXPLICIT-OUTPUT-MSG            : AREF0007.CPY
INPUT-MSG                      : AREF0008.CPY
LID-SC-DATA                    : AREF0009.CPY
LVL3ID-LI                      : AREF0010.CPY
OUTPUT-MSG                     : AREF0011.CPY
POST-DATA                      : AREF0012.CPY
SLEEP-DATA                     : AREF0013.CPY
SPAWN-TX-DATA                  : AREF0014.CPY
STANDARD-INPUT-CD              : AREF0015.CPY
STANDARD-OUTPUT-CD             : AREF0016.CPY
STS-ERR-MSGS                   : AREF0017.CPY
SWTM-DATA                      : AREF0018.CPY
TEST-IC-DATA                   : AREF0019.CPY
TP-ABORT-DATA                  : AREF0020.CPY
TP-BROADCAST-DATA              : AREF0021.CPY
TP-CALL-TPR-DATA               : AREF0022.CPY
TP-DB-COMMITMENT-DATA          : AREF0023.CPY
TP-FILE-SERVICES-DATA          : AREF0024.CPY
TP-GLOBAL-STORAGE-DATA         : AREF0025.CPY
TP-GPCL-DATA                   : AREF0026.CPY
TP-IC-SERVICES-DATA            : AREF0027.CPY
TP-JOURNALIZE-DATA             : AREF0028.CPY
TP-LID-SERVICES-DATA           : AREF0029.CPY
TP-PROCESS-SERVICES-DATA       : AREF0030.CPY
TP-RECEIVE-MESSAGE-DATA        : AREF0031.CPY
TP-RESOURCE-SERVICES-DATA      : AREF0032.CPY
TP-SEND-MESSAGE-DATA           : AREF0033.CPY
TP-SLEEP-SERVICES-DATA         : AREF0034.CPY
TP-SPAWN-BATCH-DATA            : AREF0035.CPY
TP-SPAWN-TRANSACTION-DATA      : AREF0036.CPY
TP-STORAGE                     : AREF0037.CPY
TP-TPR-SERVICES-DATA           : AREF0038.CPY
TP-TQ-SERVICES-DATA            : AREF0039.CPY
TP-USER-PROCEDURE-DATA         : AREF0040.CPY
TP-USER-TRACE-DATA             : AREF0041.CPY
TP-WORKSTATION-SERVICES-DATA   : AREF0042.CPY
TPDF-DATA                      : AREF0043.CPY
TQSVC-DATA                     : AREF0044.CPY
WAKE-DATA                      : AREF0045.CPY
WS-NAME-DATA                   : AREF0046.CPY
DBWFLREC                       : $g8wbdir\dbmscpy\dbwflrec.cpw ; Support DBWFL
```

## VMPS

Standard pre-configured VMPS COPY libraries are provided. These standard copies are pre-preprocessed, however they may customized as required.

```
[VS]
;                                Standard VMPS copies (VS Library)
DEFAULT-DIRECTORY              : $g8wbdir\vmpscpy\s
AUTOREF-CONTROL                : 0017
COMMENT_INFO_PACKED            : AREF0001.CPY
COMMENT_INFO_UNPACKED          : AREF0002.CPY
COMMENT_MESSAGE_INFO           : AREF0003.CPY
COMMENT_PACKED_STATUS          : AREF0004.CPY
COMMENT_STATUS_RETURN          : AREF0005.CPY
COMMENT_TOKEN_INFO             : AREF0006.CPY
COMMENT_UNPACKED_STATUS        : AREF0007.CPY
LVL3ID-ST                      : AREF0008.CPY
MESSAGE_INFO_TYPE              : AREF0009.CPY
PACKED_STATUS_TYPE             : AREF0010.CPY
PRIMARY_STATUS_VALUES          : AREF0011.CPY
STATUS_INFO_PACKED_TYPE        : AREF0012.CPY
STATUS_INFO_UNPACKED_TYPE      : AREF0013.CPY
STATUS_RETURN_TYPE             : AREF0014.CPY
TOKEN_INFO_TYPE                : AREF0015.CPY
UNPACKED_STATUS_TYPE           : AREF0016.CPY
VMPS_VERSION                   : AREF0017.CPY
```

```
[VF]
;                                Standard VMPS copies (VF Library)
DEFAULT-DIRECTORY              : $g8wbdir\vmpscpy\f
AUTOREF-CONTROL                : 0009
CAT_ATTR_TYPE                  : AREF0001.CPY
CAT_QUERY_TYPE                 : AREF0002.CPY
EXTENDED_ATTR_TYPE             : AREF0003.CPY
EXTENDED_QUERY_TYPE            : AREF0004.CPY
FILE_ATTR_TYPE                 : AREF0005.CPY
FILE_QUERY_TYPE                : AREF0006.CPY
LVL3ID-FM                      : AREF0007.CPY
PASSWORD_TYPE                  : AREF0008.CPY
SPECIFIC_PERM_TYPE             : AREF0009.CPY
```

```
[VB]
;                                Standard VMPS copies (VB Library)
DEFAULT-DIRECTORY              : $g8wbdir\vmpscpy\b
AUTOREF-CONTROL                : 0002
LVL3ID-BA                      : AREF0001.CPY
OPTION_TYPE                    : AREF0002.CPY
```

```
[VQ]
;                                Standard VMPS copies (VQ Library)
DEFAULT-DIRECTORY              : $g8wbdir\vmpscpy\q
AUTOREF-CONTROL                : 0004
DATE_STRUCTURE                 : AREF0001.CPY
LVL3ID-QR                      : AREF0002.CPY
SECURITY_OPTIONS_STRUCTURE     : AREF0003.CPY
TIME_STRUCTURE                 : AREF0004.CPY
```

## DISPATCH8

A standard pre-configured DISPATCH8 COPY library is provided. These standard copies are pre-preprocessed, however they may customized as required.

```
[D8]
;                                 Standard DISPATCH8 copies
DEFAULT-DIRECTORY               : $g8wbdir\pmscpy
AUTOREF-CONTROL                 : 0000
DISPATCH8-INTERFACE             : d8api.cpy
D8-TP-BATCH-PRINT-RECORDS       : d8prtrec.cpy
D8-SPECIAL-REQUESTS             : d8spcreq.cpy
D8-CONFIRM-RECORDS              : d8cfmrec.cpy
D8-PRINTVELDEN                  : d8prtvld.cpy
D8-PRINTVELDEN2                 : d8prtvl2.cpy
D8-FIELD-INDICATORS             : d8fldind.cpy
DISPATCH8-CHECK-PERM            : d8chkprm.cpy
D8-NY-CHECK-PERMISSION          : d8chkprx.cpy
D8-TP-PRINT                     : d8tpprt.cpy
DISPATCH8-TP-PRINT              : d8tpprtx.cpy
```

## DIMS8

A standard pre-configured DIMS8 COPY library is provided. These standard copies are pre-preprocessed, however they may customized as required.

```
[DI]
;                                Standard DIMS8 copies
DEFAULT-DIRECTORY              : $g8wbdir\dims8cpy
AUTOREF-CONTROL                : 0025
TELE-PR-BATCH-INPUT            : AREF0025.CPY
TELE-WS-INPUT                  : AREF0024.CPY
TELE-PR-BATCH-PRINT            : AREF0023.CPY
D8-DIMS-INTERFACE              : AREF0001.CPY
D8-DIMS8-INTERFACE             : AREF0002.CPY
D8-NATIVE-DIMS-INTERFACE       : AREF0003.CPY
DECLARATIVES-DB-EXCEPTION      : AREF0004.CPY
DIMS-EXTENDED-CALLS            : AREF0005.CPY
DIMS-FIXED-TX-STORAGE          : AREF0006.CPY
DIMS-INTERFACE-CALLS           : AREF0007.CPY
DIMS-TX-STORAGE                : AREF0008.CPY
EDS-ARGUMENT                   : AREF0009.CPY
EDS-TP-CALLS                   : AREF0010.CPY
TELE-PR-OUTPUT                 : AREF0011.CPY
TELE-WS-OUTPUT                 : AREF0012.CPY
TELE-PR-PRINT                  : AREF0013.CPY
TELE-WS-PRINT                  : AREF0014.CPY
TELE-PRKOM1                    : AREF0016.CPY
TELE-WSKOM1                    : AREF0015.CPY
TELE-DAN-TX                    : AREF0017.CPY
TELE-PR-BATCH-OUTPUT           : AREF0018.CPY
DIMS-EDS-FIXED-TX-STORAGE      : AREF0019.CPY
DIMS-EDS-TX-STORAGE            : AREF0020.CPY
DIMS-EDS-TP-TX-STORAGE         : AREF0021.CPY
EDS-BATCH-CALLS                : AREF0022.CPY
TELE-PR-BATCH-PRINT            : AREF0023.CPY
TELE-WS-INPUT                  : AREF0024.CPY
TELE-PR-BATCH-INPUT            : AREF0025.CPY
OPERATOR-BLOCK-OPTION          : $g8wbgdir\dims8\copy\AREF0070.CPY
```

## TSM8

A standard pre-configured TSM8 COPY library is provided. These standard copies are pre-preprocessed, however they may customized as required.

```
[TS]
;                                Standard TSM8 copies
DEFAULT-DIRECTORY              : $g8wbdir\tpmscpy\tsm8
AUTOREF-CONTROL                : 0004
CONSTANT-STG                   : AREF0001.CPY
TSMAIL00-FORM                  : AREF0002.CPY
TSSIGN-ARGS                    : AREF0003.CPY
XSIGNONX-FORM                  : AREF0004.CPY
```

## $$Select

G8WB support use of $$SELECT style COPY handling through the W8SELCPY Preprocessor. Mapping of GCOS-8 filenames is done through COPYMAP.CFG, like standard COPY handling. Mappings are grouped under the SELECT tag, which must be configured with appropriate mapping information. As an example a $$SELECT(SPS/TA4.1/FILE1) would required mapping of SPS/TA4.1 to the directory where the file has been placed.

Like COPY libraries, mapping may specify one or more directories. For multiple directories the search takes place left-to-right until the file is located.

Automated import of $$SELECT files can be achieved through Source Import Utility, ref. File-Transfer chapter for details, which will create separate <name.CPY> files for each $$SELECT file.

$$SELECT copies are mapped in the preprocessor COPY Mapping file, ref. below, and you are thus able to dynamically open $$SELECT files like COPY member from MF Editor.

```
;
;  Preprocessor Copy Name => File Configuration Version 1.2.00 (Server)
;
; ----------------------------------------------------------------------
;  GCOS-8 $$SELECT support feature
;
;  Place any and all GCOS-8 catalog strings referenced in $$SELECT
;  statement under the SELECT tag.
;
;  Each entry map the GCOS-8 catalog string to the directory where
;  the source members are imported to.
;
[SELECT]
;GCOS-8 Catalog                  G8WB <directory> or
;                                <dir1+dir2...> Searched left-to-right
sps/ta4.1                      : $g8wbgdir\sps
```

## Preprocessor Configuration

The Preprocessor configuration may be used to control the functionality of the G8WB Preprocessor. Although such configuration may easily be changed through Workstation level configuration, G8WB also provides an ad-hoc configuration capability through SET directives, which temporarily override any current configuration.

Configuration at Workstation level, superseed configuration at Server level.

### COMP Analyzer

The implementation of COMP, COMP-3, COMP-4 and COMP-5 fields are different for GCOS-8 and G8WB.

On GCOS-8 these fields are implemented as Packed-Decimal fields which require one half-byte for each digit or sign. COMP-4 may align to a half-byte boundary, whereas for COMP, COMP-3 and COMP-5 slack-half-bytes are inserted to force alignment to byte boundary.

On G8WB these fields are implemented as signed binary fields. The ANSI behaviour applied, make such binary fields functionaly work as the Packed-Decimal equivalents. The physical size of the fields may wary (both greater and less) between GCOS-8 and G8WB, however this represent no problem, unless redefines are involved.

The COMP Analyzer is designed to automatically make adjustment of field usage and/or size to overcome such potential problems. COMP Analyzer introduces substantial overhead during preprocessing and should only be used when such automatic adjustment is required.

*In short: COMP Analyzer must be enabled when such COMP-x fields are being redefined/sub-divided, otherwise G8WB will use same implementation throughout all program source and subschema definition, and no special adjustment is required. Such Redefine/Sub-divide is referred to as compound fields*

## Default-Section

```
;
; Preprocessor Configuration Version 1.2.00 (Server Config)
;
[DEFAULT-SECTION]
DEBUG                  : OFF            ; Source Debug switch OFF/ON
PREP-WORK-FILE         : DELETE         ; DELETE/KEEP program.WRK file
GNT-DELETE             : OFF            ; Delete GNT file OFF/ON
AIF-DELETE             : OFF            ; Delete AIF file OFF/ON
PREP-UPPER-CASE        : OFF            ; Convert to upper-case OFF/ON
COMP-36-BIT-SUPPORT    : OFF            ; COMP-6 36 bit OFF/ON/FORCE
DB-KEY-36-BIT-SUPPORT  : OFF            ; DB-KEY 36 bit OFF/FORCE
G8WFL-BSF-GENERATE     : OFF            ; G8WFL program.BSF generate ON/OFF
MODULE-ENTRY-MAPPING   : OFF            ; Enable Module Entry Mapping OFF/ON
EQUATE-MAPPING         : OFF            ; Enable CALL Equate Mapping OFF/ON
COMP-REDEFINE-ANALYZER : OFF            ; COMP Redefine Analyzer OFF/FLAG/ADJUST
FLAG-ILLEGAL-REDEFINE  : OFF            ; Flag illegal redefines OFF/ON
COMP-MOVE-ANALYZER     : OFF            ; COMP Move Analyzer OFF/FLAG/ADJUST
COMP-TX-STORAGE-ADJUST : OFF            ; COMP TX Storage Adjust OFF/nnnn[nnnnn]
COMP-TX-STORAGE-INIT   : OFF            ; COMP TX Storage Init   OFF/nnnn
COMP-LINKAGE-CHECK     : OFF            ; COMP Linkage Check OFF/ON
EXIT-PROGRAM-TRAP      : OFF            ; Stop Run after exit program OFF/ON
EXAMINE-TALLYING       : OFF            ; Include TALLY definition OFF/ON
RCM                    : OFF            ; RCM Enable OFF/ON
SITE-PREPROCESSOR      : OFF            ; Site Preprocessor program or OFF
;                                         W8SELCPY enables $$SELECT processing
SELECT-EXPAND-FILE     : OFF            ; $$SELECT program.XBL file ON/OFF
BLOCK-UNIFY            : OFF            ; BLOCK Unify names OFF/ON
XDB                    : "VALIDATE"     ; XDB Default directives
SPZERO                 : ON             ; Use SPZERO directive ON/OFF
DEFAULTBYTE            : 0              ; Set defaultbyte (uninitialized data)
                                        ; 0 (default is low-value)
                                        ; nnn (ASCII char decimal value
                                        ; ex. 32 = SPACE
```

## Debug

DEBUG setting is equivalent to use of compile DEBUG option on GCOS-8 $ CBL74 or $ CBL85 JCL card.

OFF setting will remove any DEBUGGING-MODE clause and change all debug lines to comment lines.

ON setting will leave program as is, and Checker will determine whether debug lines and debug sections should be processed based on the presence of DEBUGGING-MODE in Source-Computer clause.

## Prep-Work-File

PREP-WORK-FILE setting defines whether Preprocessor work file(s) should be deleted or retained after preprocessing. When reporting problems, you may be requested to supply such work file(s), or you may want to browse the work file(s) to view the resulting COBOL/2 coding from the Preprocessor.

G8WB will use a program.WRK file during preprocessing. If COMP Analyzer is enabled, an additional work file program.KRW is used. If both work files are present, the program.KRW work file represent the source being passed to COBOL/2. If XDB preprocessing (INTEREL) is enabled, an additional work file program.XWK is used. In this case program.WRK contain the preprocessed source before SQL preprocessing and program.XWK represent the source being passed to COBOL/2 after SQL preprocessing.

Normally the DELETE setting is used to avoid waste of disk space.

## GNT-Delete

GNT-DELETE setting defines whether any program.GNT file should be deleted as the program is being re-checked. This technique of deletion works independent of MFWB and it’s possibilities for automatic GNT file deletion.

## AIF-Delete

AIF-DELETE setting defines whether any program.AIF file should be deleted as the program is being re-checked. This technique of deletion works independent of MFWB and it’s possibilities for automatic AIF file deletion.

*Note: G8WB will force Animator to place AIF files under $G8WBLDIR.*

## Prep-Upper-Case

PREP-UPPER-CASE setting defines whether source should be converted to upper-case before preprocessing. ON setting work functionally like $$SELECT of source on GCOS-8, without $ ASCII.

## Comp-36-bit-Support

COMP-36-BIT-SUPPORT setting defines how COMP-6 fields should be preprocessed.

G8WB, by default, treat COMP-6 fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate preprocessing, i.e. *W8PREP ALT active, COMP-6 fields are implemented as 32 bit unsigned binary fields, which allow redefintion with PIC X.

ON setting may be used to change alternate preprocessing, so COMP-6 fields are implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physical size of a COMP-6 field will however change from 4 to 5 bytes.

FORCE setting may be used to change default preprocessing, so COMP-6 fields are implemented as 39 bit signed binary fields. In alternate preprocessing, COMP-6 fields are implemented as 31 bit signed binary fields.

*Note: You should only enable 36-bit-support if you really need it, and you must pay attention to potential problems with redefines due to the change in physical field size.*

## DB-Key-36-bit-Support

DB-KEY-36-BIT-SUPPORT setting defines how DB-KEY fields should be preprocessed.

G8WB, by default, treat DB-KEY fields like COMP-2 fields, which are implemented as 31 bit signed binary fields. In alternate preprocessing, i.e. *W8PREP ALT active, DB-KEY fields are implemented likewise.

FORCE setting may be used to change preprocessing, so DB-KEY fields are by default implemented as 39 bit signed binary fields, which support 35 bit precision like on GCOS-8. The physizal size of a DB-KEY field will howver change from 4 to 5 bytes. In alternate preprocessing, i.e. *W8PREP ALT active, DB-KEY fields are implemented as 31 bit signed binary fields.

*Note: 36-bit-support is ONLY relevant, although this is not very likely, if you unload from host databases with VERY large DBK values, because original DBK’s are re-used within G8WB. New DBK’s are assigned as highest DBK + 1. A locally created database will thus start with DBK = 1.*

## G8WFL-bsf-enable

G8WFL-BSF-ENABLE setting defines whether the Preprocessor should generate information to support GCOS-8 file conversion via G8WFL. When enabled (ON) an additional program.BSF is generated.

*Note: ON setting is only relevant for batch programs.*

## Module-entry-mapping

MODULE-ENTRY-MAPPING setting defines whether the Preprocessor should scan the MODULE.CFG for entry-point mapping, during preprocessing of CALL statements, ref. Module Mapping configuration.

## Equate-mapping

EQUATE-MAPPING setting defines whether the Preprocessor should scan the EQUATE.CFG for call mapping, during preprocessing of CALL statements, ref. Equate Mapping configuration.

## COMP-Redefine-Analyzer

COMP-REDEFINE-ANALYZER setting may be used to enable COMP Analyzer for explicit redefines. Automatic adjustment may involve either change of USAGE or adjustment of LENGTH, depending on the redefinition type.

The ADJUST setting enable automatic adjustment, whereas FLAG setting will detect and flag such occurences as errors.

## Usage Adjustment

```
01      Field1.
  05    Field1-1        PIC 9(04) COMP-4.
  05    Field1-2 REDEFINES Field1-1.
    10  Field1-2-1      PIC 9(02) COMP-4.
    10  Field1-2-2      PIC 9(02) COMP-4.
01      Field1.
  05    Field1-1        PIC 9(04) COMP-4.
01      Field2 REDEFINES Field1.
    10  Field2-1        PIC 9(02) COMP-4.
    10  Field2-2        PIC 9(02) COMP-4.
```

The COMP Analyzer will automatically change such involved COMP-x fields to USAGE DISPLAY. Proper information message or given by the Preprocessor and optionally logged to the CINFO.LOG also.

## Length Adjustment

```
01      Field1.
  05    Field1-1        PIC X(02).
  05    Field1-2 REDEFINES Field1-1.
    10  Field1-2-1      PIC 9(03) COMP-4.
    10  Field1-2-2      PIC 9(01) COMP-4.
01      Field1.
  05    Field1-1        PIC X(02).
01      Field2 REDEFINES Field1.
    10  Field2-1        PIC 9(03) COMP-4.
    10  Field2-2        PIC 9(01) COMP-4.
```

The COMP Analyzer will automatically change such involved COMP-x fields to USAGE DISPLAY and in addition change the LENGTH of X(nn) fields. Proper information message or given by the Preprocessor and optionally logged to the CINFO.LOG also.

### Unrelated

Redefines may be related or unrelated, meaning the same space is simply being re-used with a different definition and as such does not represent a true redefinition of a field. This is referred to as an unrelated redefine and the COMP Analyzer will detect such situations and leave fields unchanged.

*Note: A redefinition where a sign is involved is always considered unrelated.*

### Flag-Illegal-Redefine

FLAG-ILLEGAL-REDEFINE setting defines whether COMP Analyzer should flag illegal redefines. GCOS-8 will flag redefines if definition and re-definition have different length, provided the redefine is at elementary level, i.e. > 01.

ON setting, in connection with COMP Analyzer, will enable similar redefine length checking, and cause illegal redefines to be flagged.

### COMP-Move-Analyzer

COMP-MOVE-ANALYZER setting may be used to enable COMP Analyzer for implicit redefines. Automatic adjustment may involve either change of USAGE or adjustment of LENGTH, depending on the redefinition type, similar to explicit redefines. Imlicit redefines are ONLY recognized at 01 level and are based on analysis of MOVE statements within the program source.

The ADJUST setting enable automatic adjustment, whereas FLAG setting will detect and flag such occurences as errors.

```
01      Field1.
  05    Field1-1        PIC X(02).

...

01      Field2.
    10  Field2-1        PIC 9(03) COMP-4.
    10  Field2-2        PIC 9(01) COMP-4.

...

        MOVE Field1 TO Field2
```

The COMP Analyzer will automatically change such involved COMP-x fields to USAGE DISPLAY and in addition change the LENGTH of X(nn) fields. Proper information message or given by the Preprocessor and optionally logged to the CINFO.LOG also.

### Subschema

When implicit redefines involve subschema records, the COMP Analyzer is unable to make an automatic adjustment of the subschema definition, but is limited to making a verification of the implicit redefine.

In the example below the COMP Analyzer will adjust the LENGTH of WSRecord-Field-1 automatically.

```
01      Subschema-Record1.
    10  SSRecord1-Field-1 PIC 9(03) COMP-4.
    10  SSRecord1-Field-2 PIC 9(01) COMP-4.

...

01      Working-Storage-Record1.
  05    WSRecord-Field-1  PIC X(02).

...

        MOVE Subschema-Record1 TO Working-Storage-Record1
```

In the example below, the COMP Analyzer is unable to change the USAGE for the subschema fields involved, and you are required to change the subschema DDL manually. G8WB Subschema Translator support embedded directives (like G8WB Preprocessor), and the DDL may thus be changed without loosing compatibility with GCOS-8.

In the example below, the subschema now change dynamically from COMP-4 to numeric DISPLAY format, and COMP Analyzer will recognize this and change WSRecord-Field-1 to numeric DISPLAY format also.

```
01      Subschema-Record1.
*w8prep alt
    10  SSRecord1-Field-1 PIC 9(03) COMP-4.
    10  SSRecord1-Field-2 PIC 9(01) COMP-4.
*w8prep on

...

01      Working-Storage-Record1.
  05    WSRecord-Field-1  PIC 9(04) COMP-4.

...

        MOVE Subschema-Record1 TO Working-Storage-Record1
```

## I/O Modules

When a subschema I/O module technique is being used, you may have program source with Working-Storage definitions which are actually redefines of subschema records. This fact cannot be detected by the COMP Analyzer, since you neither have the subschema definition present nor the MOVE to the subschema record.

In this situation you are required to include a Preprocessor directive in the source to establish such implicit linkage/redefinition. Although this does require changes to the source, such changes are usually contained within COPY-members.

Following example illustrates how such linkage is established:

```
 01      Working-Storage-Record1.
*W8PREP LINK subschema-name record-name
   05    WSRecord-Field-1  PIC X(02).
```

## COMP-TX-Storage-Adjust

COMP-TX-STORAGE-ADJUST setting may be used, in connection with the COMP Analyzer, to verify and optionally adjust fields to a fixed offset within TX-Storage.

Following example illustrates when this may be required:

```
01      TX-Storage.
  05    Site-Fixed-area-1      PIC X(200).
  05    TX-Use-of-Area REDEFINES Site-Fixed-Area.
    10  FILLER                 PIC X(100).
    10  TX-Field-1             PIC 9(03) COMP-4.
    10  TX-Field-2             PIC 9(01) COMP-4.
    10  FILLER                 PIC X(98).
  05    Site-Fixed-Area-2      PIC X(2000).
  05    TX-Field-3             PIC 9(01).
```

As a Site-Standard the first 200 bytes and the following 2000 bytes of TX-Storage are the same for all TPR's, however byte 101-200 may be used (redefined) as required by the individual TPR's.

Due to the difference in implementation between GCOS-8 and G8WB, the redefine in above example would become 101 bytes on G8WB as opposed to 100 bytes on GCOS-8. In other situations, depending on field usage etc., the length could have been less on G8WB. Problem being the offset of Site-Fixed-Area-2 is not constant (201) as on GCOS-8.

The COMP-TX-Storage-Adjust setting may be used to fix this problem. A setting of 200, would in above example result in an automatic LENGTH adjustment of the FILLER just in front of Site-Fixed-Area-2 (-1 adjustment). Proper information message is given by the Preprocessor and optionally logged to the CINFO.LOG also.

If such adjustment is impossible (fall below zero or no FILLER present), the COMP-TX-Storage-Adjust setting of 200 300, would force the previously fixed location (201) to a new fixed location (301), and thus overcome such adjustment problem. New ALIGN-FILLER is inserted as required and proper information message is given by the Preprocessor and optionally logged to the CINFO.LOG also.

G8WB will detect TX-Storage from the TPR USING sequence (third argument), however when TX-Storage is passed to modules, this automatic detection cannot be made and you are required to insert a Preprocessor directive to indentify TX-Storage.

```
 LINKAGE SECTION.
 01  TX-Storage-passed-to-module.
*W8PREP TX-STORAGE
   05 ...
```

*Note: Offset specification is relative to zero.*

## COMP-TX-Storage-Init

Due to the difference in handling of Numeric DISPLAY fields on GCOS-8 and G8WB, where GCOS-8 will treat both SPACES and LOW-VALUES as ZERO, whereas G8WB will treat only SPACES as ZERO and give a run-time error for LOW-VALUES, the COMP-TX-Storage-Init setting may be used to request proper initialize of TX-Storage before entering the first TPR.

COMP-TX-STORAGE-INIT setting must specify an offset within TX-Storage, where the initialize should commence. Pertaining to above example a setting of 2200, would force the Preprocessor to insert initialize code for any Numeric DISPLAY fields starting above this offset, at the beginning of the TPR.

The initialize code is inserted for all TPR's based on it's description of TX-Storage, however only the first TPR in a Transaction will actually execute this initialize code.

If you have generalized Front-/Back-end TPR's, you may suppress such initialize by inserting a Preprocessor directive in the source. This way initialize is delayed until the first real application TPR.

```
...

*W8PREP NO-TX-INIT
...
 PROCEDURE DIVISION ...
```

*Note: Offset specification is relative to zero.*

*Note: This solution does not handle all potential situations involving LOW-VALUES in Numeric DISPLAY fields.*

## COMP-Linkage-Check

COMP-LINKAGE-CHECK setting may be used, in connection with the COMP Analyzer, to enable a Length-Verification check for Data Definitions passed through Linkage Section.

The Preprocessor will include Length-Setup for all arguments wherever a CALL statement is processed. This is performed regardless of the COMP-Linkage-Check setting.

When COMP-Linkage-Check is enabled (ON), the Preprocessor will also include a Length-Verification check of all USING arguments in the beginning of the module. If unequal lengths are detected, the module is aborted and proper debug information is either written to the TP Journal LOG or to the screen (Batch).

## RCM

RCM setting defines whether you have installed and wish to enable the Micro Focus Reusable Code Manager. On setting will automatically stack the RCM Preprocessor and no further Preprocessor setup is required.

RCM directives may be configured in RCM.CFG (ref. RCM documentation for details) and/or through G8WB SET directives, ref. below.

## Site-preprocessor

SITE-PREPROCESSOR setting defines whether you wish to enable your own Preprocessor. Such Preprocessor must be implemented according the Micro Focus stacked preprocessor specification, ref. Micro Focus documentation and G8WB Technical Guide for details.

Directives for the site preprocessor may be passed through G8WB SET directives, ref. below, or any other means as provides by the SITE Preprocessor.

*Note: A site preprocessor executes as "Bottom-of-Stack", thus source lines from site preprocessor will pass RCM and/or XDB (if applicable) and finally G8WB preprocessor.*

## $$SELECT-preprocessor

SITE-PREPROCESSOR setting of W8SELCPY may be used to enable $$SELECT preprocessing.

## SELECT-expand-file

When set ON, the $$SELECT preprocessor will save the expanded source file with extension XBL.

## Block-Unify

BLOCK-UNIFY setting defines how G8WB Preprocessor should implement the GCOS-8 specific BLOCK feature.

With OFF setting (default), G8WB Preprocessor will assign the EXTERNAL attribute to any File-name(s) and/or Data-name(s) referenced in BLOCK clause of SPECIAL-NAMES. Use of EXTERNAL will ensure File- and Data-name(s) are shared between different compile units, however it does require the use of same name(s) within all involved compile units.

With ON setting, G8WB Preprocessor will, in addition to assign of EXTERNAL attribute, rename all Fiel-name(s) and/or Data-name(s) referenced in BLOCK clause of SPECIAL-NAMES. A pre-defined naming convension is used, as follows:

```
W8UBLK-bb-ss

   where   bb = BLOCK number
           ss = Sequence number within block

           01 for first or only
           02-98 for intermediate within through
           99 for last within through
```

*Restriction: The same BLOCK number may not be repeated, i.e. File(s) or Data are accumulated within a BLOCK through separate BLOCK clauses.*

## XDB

XDB setting define directives to be passed to XDB Preprocessor when COBOL-85/SQL type program are checked. The default setting “VALIDATE” will make XDB check SQL syntax through the SQL Server, thus providing a complete validation of embedded SQL.

## SPZERO

SPZERO setting defines whether the MF SPZERO directive should be inserted during preprocessing. The default On setting will ensure GCOS-8 compatible handling of numeric DISPLAY fields containing SPACES, i.e. treated like ZEROES.

## DEFAULTBYTE

DEFAULTBYTE setting defines the attribute to be used with the MF DEFAULTBYTE directive, which is always inserted during preprocessing. The default 0 setting will ensure GCOS-8 compatible results, as LOW-VALUE is the default value on GCOS-8 for un-initialized fields.

## Maximum-Section

This Maximum-Section settings are used to allocate tables within the G8WB Preprocessor. In general you should not increase such settings, unless the Preprocessor reports an error where adjustment is requested, otherwise memory is just wasted.

```
[MAXIMUM-SECTION]
;                        Configurable -  Standard Preprocessor
REPLACING              : 5       ; Max. (470) Replacing arguments
CALL-NAMES             : 100     ; Max. (700) Service call names in TBL
ENTRIES                : 150     ; Max. (700) Division Entries in TBL
NAMES                  : 50      ; Max. (2000) Dynamic Names
IDS-WORDS              : 50      ; Max. (700) IDS-II Words in TBL
IDS-NAMES              : 9       ; Max. (1000) IDS-II Names in Verb
IDS-ARGS               : 120     ; Max. (1000) IDS-II Arguments in Verb
IDS-USE                : 5       ; Max. (100) IDS-II Use Procedures
IDS-DB-STATUS          : 10      ; Max. (100) IDS-II DB-Status's in Use Proc.
MAX-BLOCK-UNIFY        : 10      ; Max. (500) BLOCK's for unification
;                        Not configurable
;                                        COMP Redefinition Analyzer
01-IDENTIFIER          : 1478          ; Max. 01 Identifier's
01-ITEMS               : 868           ; Max. Items within 01
```

## Trace-Section

```
[TRACE-SECTION]
COMP-ANALYZE-LOG       : OFF           ; COMP Analyzer Logging OFF/ON
COMP-ANALYZE-TRACE     : OFF           ; COMP Analyzer Tracing OFF/ON
```

### COMP-Analyze-Log

The COMP-Analyze-Log setting may be used to enable/disable the logging feature of the COMP Analyzer. When enabled, both the CINFO.LOG and CERROR.LOG log files are being updated.

### COMP-Analyze-Trace

The COMP-Analyze-Trace setting may be used to enable/disable a trace logging from COMP Analyzer on the Preprocessor work file (.WRK). This should normally only be enabled on request, when reporting problems in relation to the COMP Analyzer.

## Preprocessor SET Directives

SET directives are very similar to the Micro Focus $SET directives and may be used to temporarily override G8WB Preprocessor configuration and/or pass directives to other Preprocessors (XDB, RCM and Site Preprocessor) and the MF Checker itself.

SET directive(s) are formatted as comment lines and must be placed before any non-comment COBOL line in the source. The format is as follows:

***SET>G8WB configuration-entry**

***SET>XDB parameter-1 [...]**

***SET>RCM parameter-1 [...]**

***SET>SITE parameter-1 [...]**

***SET>MFWB parameter-1 [...]**

***SET>CPM parameter-1 [...]**

## G8WB

Any of the Preprocessor configuration specifications may be overridden via a SET directive. Only one configuration entry is allowed per directive and the format is equivalent to the PREP.CFG configuration file entry. Multiple SET directives may be given. Example:

***SET>G8WB prep-work-file:keep**
***SET>G8WB g8wfl-bsf-enable:on**

## Source Classify

When checking programs through G8WB, you have to specify whether the source is COBOL-74, COBOL-85 or COBOL-85 with SQL. You may be also specify whether the source is a TPR or a PROGRAM, which is used by G8WB Preprocessor to enforce further logic checks.

Specification of COBOL dialect MUST be given externally, i.e. either through language-type in character-mode workbench or specific project attribute in GUI workbench, the TPR/PROGRAM classification may, however, be embedded in the source, so you need not specify this externally.

This may be done through the a SET directive as follows:

***SET>G8WB TPR**

or

***SET>G8WB PROGRAM**

Above classification may be inserted automatically during Source Import, ref. File-Transfer chapter for details.

## Copy Mapping

MF Editor and Animator2 (GUI) includes a nice feature, which will allow you to open COPY member dynamically. In MF Editor, you position yourself to the source line with a COPY statement and open the COPY member through Alt-F2. This feature is NOT sentive to GCOS-8 specific COPY syntax, and will use the COPY <name> to try and locate a file named <name>.CPY, either in your current directory or via directories defined in COBCPY. This technique works ok as long as you only use short COPY names, i.e. < 9 characters, and you include all your COPY libraries (directories) in COBCPY setting.

G8WB Preprocessor use COPYMAP.CFG to find COPY members, in order to fully support GCOS-8 specific COPY syntax, and this technique is not directly compatible with the described Alt-F2 feature. You may, however, obtain a similar functionality by including a SET directive in your program source as follows:

***SET>G8WB COPY program.CPM**

Program.CPM must identify the actual name of the program, i.e. p1.CPM would be specified in P1.CBL source file.

This SET directive will trigger G8WB Preprocessor to generate a COPY Map file, which will contain a list of all COPY statement in your program with an associated COPY line identifying the actual location of the COPY member. For a program with subschema, the mapping will also include the UWA (all fields) for the subschema. Instead of using Alt-F2 directly on the source line with the COPY statement, you have to return to the beginning of the source and use Alt-F2 on the directive line. This will open the COPY Map, and you may then use Alt-F2 again to open any of the COPY members associated with the program.

This directive is automatically inserted during Source Import, ref. File-Transfer chapter for details.

## XDB

One or more XDB directives may be passed to the XDB Preprocessor via a SET directive. Multiple SET directives may be given. Example:

***SET>XDB nologon declare**

## RCM

One or more RCM directives may be passed to the RCM Preprocessor via a SET directive. Multiple SET directives may be given. Example:

***SET>RCM debug(on) beauty(on)**

## SITE

One or more SITE defined directives may be passed to a SITE Preprocessor via a SET directive. Multiple SET directives may be given. Example:

***SET>SITE turn-something-on-or-off**

## MFWB

One or more MFWB defined directives may be passed to the MF Checker via a SET directive. Multiple SET directives may be given. Example:

***SET>MFWB remove"upsi"**

## CPM

May be used to place entries in the COPY Mapping file. The directive line is copied asis to the associated COPY Mapping file, if COPY Mapping has previously been enabled. Example:

***SET>CPM copy "college" Example of XDB DCL file reference**

# Preprocessor Embedded Directives

G8WB provides a few embedded Preprocessor directives, which may be used to control the behaviour of the G8WB Preprocessor. Embedded directives are formatted as comments, have a fixed format and may be inserted anywhere in the COBOL source. The format is as follows:

***W8PREP keyword [parameter-1 [...]]**

| Keyword | Parameter(s) |
|---|---|
| OFF | none |
| ALT | none |
| ON | none |
| LINK | subschema-name record-name |
| TX-STORAGE | none |
| NO-TX-INIT | none |

## OFF

May be used to suppress G8WB preprocessing.

## ALT

May be used to change Data Format implementation for COMP, COMP-3, COMP-4, COMP-5, COMP-6 or DB-KEY usage types.

For COMP, COMP-3, COMP-4 and COMP-5, alternate preprocessing will change USAGE to DISPLAY, thus forcing fields to numeric DISPLAY format.

For COMP-6, alternate preprocessing in dependent of COMP-36-BIT-SUPPORT setting, ref. above.

For DB-KEY, alternate preprocessing in dependent of DB-KEY-36-BIT-SUPPORT setting, ref. above.

## ON

Used to cancel OFF or ALT directives and return to normal preprocessing.

## LINK

May be used to link a record description to a subschema (only used with COMP-Analyzer).

## TX-STORAGE

May be used to identify TX-Storage description in modules (only used when TX-STORAGE-INIT enabled).

## NO-TX-INIT

May be used to suppress initialize of TX-Storage for this TPR (only used when TX-STORAGE-INIT enabled).

## Module Configuration

Module configuration is quite similar to Environment configuration, however, whereas environment configuration takes place at run-time, module configuration takes place at time of preprocessing. Any CALL reference to a configured entry-point, will cause the associated module to be loaded in advance.

The functionality of Module configuration is thus functionally equivalent to Environment, however, module configuration is generally recommended, when many entry/module entries are involved (only modules required being loaded).

```
;
; Module Entry Mapping Configuration Version 1.2.00 (Server Config)
;
; GCOS-8
;   Use static linking. Reference to an "entry-point" will cause implicit
;   link of "module-name", if embedded in object library or explicitly
;   loaded with $ USE "module-name".
;
; G8WB
;   Use dynamic linking. Reference to an "entry-point" cannot be resolved
;   unless "module-name" is explicitly loaded first. Module Entry Mapping
;   provides a way of explicitly loading "module-name", without changing
;   program source. Effective during Checking.
;
; Any call "entry-point" will have call "module-name" inserted in front.
;
; Note; MODULE-ENTRY-MAPPING must be enabled through PREP.CFG, otherwise
;       this configuration file is NOT searched.
;
[MODULE]
; "entry-point"        : "Module-name" to be preloaded
```

Configuration at Workstation level, superseed or amend configuration at Server level.

## Equate Configuration

Equate configuration may be used to implement a similar functionality as GCOS-8 General Loader provides with the $ EQUATE loader directive.

GCOS-8 use static linking and General Loader will allow you remap external references through $ EQUATE loader directive(s). As an example a CALL “MOD1” can be equated to “MOD1N”, so MOD1N is called instead of MOD1, as described in the calling program.

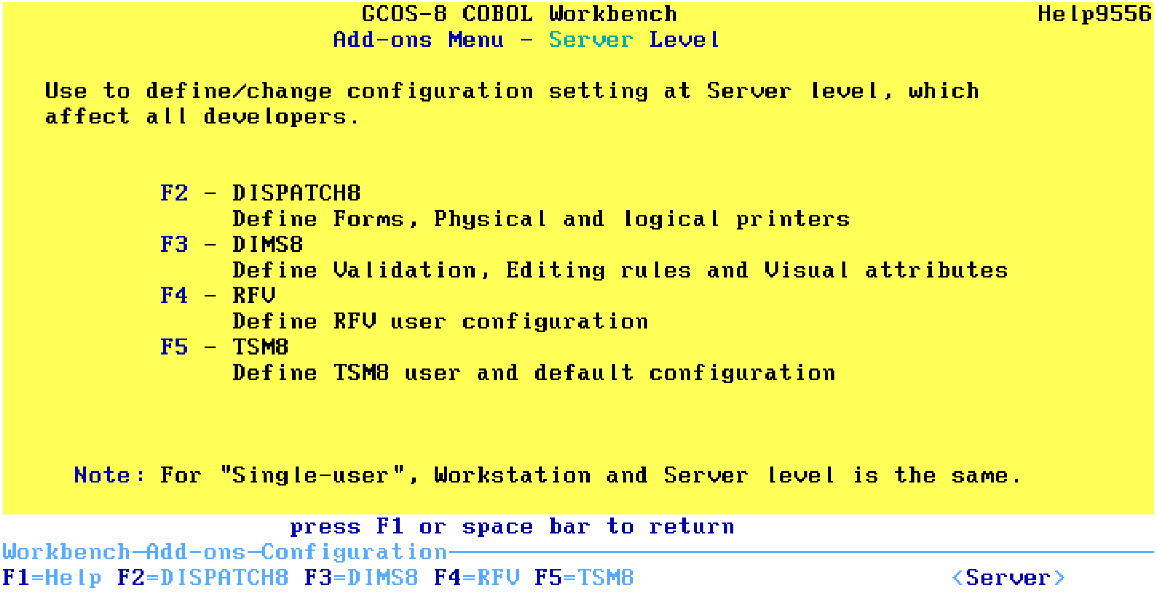
G8WB use dynamic linking, and programs are searched for dynamically as the call is executed. Equate Mapping may be used to remap a CALL at time of preprocessing. As an example a CALL “MOD1” can be equated to “MOD1N”, so MOD1N is called instead of MOD1, as described in the calling program. Unlike GCOS-8, programs must be re-checked if you change the equate configuration.

Equate rules are program specific, so equate rules must be placed after an identifying program tag in the configuration file.

```
;
; Equate Module Mapping Configuration Version 1.2.00 (Server Config)
;
; GCOS-8
;   Use static linking. A reference may be remapped through $ EQUATE.
;   A call to "original-name" could thus be remapped to "equate-name".
;
; G8WB
;   Use dynamic linking.  A reference is solved through search of
;   libraries and/or directories for a file (INT or GNT) with the same
;   name. Equate Module Mapping provides a way of remapping, without
;   changing program source. Effective during Checking.
;
; For a given "program-name", any call "original-name" will be changed
; to call "equate-name".
;
; Note; EQUATE-MAPPING must be enabled through PREP.CFG, otherwise
;       this configuration file is NOT searched.
;
[program-name]
"original-name"        : "equate-name"
```

Configuration at Workstation level, superseed or amend configuration at Server level.

# Add-ons Configuration



```
                            GCOS-8 COBOL Workbench                    Help9556
                          Add-ons Menu - Server Level

  Use to define/change configuration setting at Server level, which
  affect all developers.


        F2 - DISPATCH8
             Define Forms, Physical and logical printers
        F3 - DIMS8
             Define Validation, Editing rules and Visual attributes
        F4 - RFV
             Define RFV user configuration
        F5 - TSM8
             Define TSM8 user and default configuration



    Note: For "Single-user", Workstation and Server level is the same.

                    press F1 or space bar to return
Workbench-Add-ons-Configuration-------------------------------------------------
F1=Help F2=DISPATCH8 F3=DIMS8 F4=RFV F5=TSM8                       <Server>
```

Workstation level configuration, superseed or amend configuration at Server level. For tables, duplicate entries may possibly be created, however only the first (workstation) entry is actually used.

## DISPATCH8 Configuration

G8WB's implementation of DISPATCH8 does not include the host control database and collection file, and the associated administrative service transactions. Instead, Forms, Logical and Physical printers are configured through the D8.CFG configuration file. The print reports are collected in separate file under the $G8WBGDIR\PMS directory. Permission checking of LID/Logical-printer relation is not enforced by G8WB. G8WB provides no automated tool for unloading/loading information from an existing DISPATCH8 control database, however within G8WB you need only configure entities that relate to your current test environment, thus your entire network of Logical and Physcial printers need not be configured.

### Physical Printers

Physical printer configuration is used by the D8 Executive utility to direct printing to an appropriate device or file, and edit report lines in accordance with the FF- and CC-Code specification. Customized CCHAR and NFORM modules are supported, thus you are able to perform a complete test and verify correct print formatting within G8WB environment.

```
;
; DISPATCH8 Print Configuration Version 1.2.00 (Server Config)
;
[PRINTER-SECTION]
; Physical Printer(xx)   FF-Code CC-Code Device or File Comments
AA                     : 22      13      LPT1:          HP Laser Portrait
```

## Logical Printers

Logical Printer (Queue) configuration is used both by D8 Application interfaces for collection and by the D8 Executive utility for mapping Logical to Physical printer. D8 application interfaces require the destination Logical printer to be defined, otherwise an exception is returned.

```
[QUEUE-SECTION]
; Logical Printer(nnn)   Physical Printer
001                    : AA
002                    : AA
012                    : AA
015                    : AA
```

## Forms

Forms configuration is used both by the D8 Application interfaces for collection and by the D8 Executive utility for printing control.

```
[FORM-SECTION]
; Form Code(nnnn)        Lines/Page Lines/Inch Mount  Paper-text
0001                   : 48         8          No     Plain-paper
0002                   : 54         8          Yes    Invoice
0019                   : 48         8          No     Plain-paper
```

*Note: Mount Yes will make Executive utility prompt for Retry or Continue, after printing the first page of the first report, thus allowing adjustment of paper.*

# DIMS8 Configuration

DIMS8 configuration has been simplied and does not directly correspond to the implementation on GCOS-8. DIMS8 configuration file replace the GCOS-8 Configuration Definition and the RDYDMS, USRVAL, USREDT and USRTTY configuration modules. These various configuration options correspond to GCOS-8 options, and you should consult DIMS8 Installation Manual for details on the various options.

Following entries correspond to DIMS8 Configuration Definition options.

```
;
; DIMS8 Forms Configuration Version 1.2.0 (Server Config)
;
[CONFIGURATION]
SOFTWARE-VERSION             : "3.1"
SITE                         : "DAN"
COMPANY                      : "DanTransport"
SAME                         : "-"
OVERRIDE                     : "%"
DEFAULT                      : "!"
HELP                         : "?"
DECIMAL                      : "."
CONFIG-SWITCH-1              : 0 ; 1=Access control using LID
CONFIG-SWITCH-2              : 1 ; 1=Access control using Initials
CONFIG-SWITCH-3              : 1 ; 1=User-Broadcast Facility
CONFIG-SWITCH-4              : 0 ; 1=Dynamic Authority/User-group check
CONFIG-SWITCH-6              : 1 ; 1=Extended Help Facility
CONFIG-SWITCH-7              : 1 ; 1=Extended Menu Facility
CONFIG-SWITCH-8              : 1 ; 1=Force Logon
CONFIG-SWITCH-9              : 0 ; 1=Underline 3270
CONFIG-SWITCH-22             : 1 ; 1=TELEX8 Facility Enable
CONFIG-SWITCH-23             : 1 ; 1=DISPATCH8 Facility Enable
CONFIG-SWITCH-24             : 1 ; 1=EDS8 Facility Enable
SITE-OPTION-3                : 1 ; 1=Suppress Company-name lookup
SITE-OPTION-8                : 1 ; 1=DAN-DK Special TX-Swapping SITENEXT
SITE-OPTION-10               : 1 ; 1=Retain PAGE-display when only one page
SITE-OPTION-11               : 0 ; 1=Continue-return if FK present
SITE-OPTION-13               : 0 ; 1=Use forward-space on QUESTAR
SITE-OPTION-14               : 0 ; 1=Request same=empty-field if no permission
SITE-OPTION-15               : 0 ; 1=Update same value on non-empty field only
SITE-OPTION-19               : 0 ; 1=Department numeric (need DISPLAY changes)
SITE-OPTION-20               : 0 ; 1=TEC-DK Special FK handling
SITE-OPTION-21               : 0 ; 1=Immediate broadcast
SITE-OPTION-22               : 0 ; 1=HOKI-DK No same-playback for Default=Request-
same
SITE-OPTION-23               : 0 ; 1=SVB-NL Extended Menu Spawn
SITE-OPTION-24               : 0 ; 1=SVB-NL UPD-LID/INIT company & supervisor check
SITE-OPTION-30               : 0 ; Password security check
;                                  0=Normal password check
;                                  1=Extended password check (refuse logon)
;                                  2=Extended password change (request change)
SITE-OPTION-31               : 0 ; First digit of password change interval
SITE-OPTION-32               : 0 ; Second digit of password change interval
;                              nn=Password change interval in days
SITE-OPTION-34               : 0 ; Minimum password length (0=any)
SITE-OPTION-35               : 0 ; Set TELEX mail indication
SITE-OPTION-42               : 0 ; First digit DISPATCH8
SITE-OPTION-43               : 1 ; Second digit         Forms
SITE-OPTION-44               : 9 ; Third digit               Number
SITE-OPTION-46               : 0 ; 1=Suppres (DTHXXX) field content update optimize
SITE-OPTION-47               : 0 ; 1=Force SECurity attribute during error
correction
SITE-OPTION-48               : 0 ; 1=VIP7800 FK only = no fields changed else all
empty
SITE-OPTION-50               : 1 ; 1=VIP7800 modified transmit off
```

Following entries correspond to configuration options within the RDYDMS module.

```
MAXIMUM-SAME-VALUES          : 15 ; Max. concurrent same values
MAXIMUM-PAGES                : 150 ; Max. page reservations
DEFAULT-MID                  : MENU ; Default MID/Command
BREAK-FC                     : B ; Break Standard Function Key (Code)
DELIMITER-WITHOUT-SKIP       : O"003" ; ETX
DELIMITER-WITHOUT-SKIP       : O"004" ; EOT
DELIMITER-WITHOUT-SKIP       : O"011" ; HT
DELIMITER-WITHOUT-SKIP       : O"012" ; LF
DELIMITER-WITHOUT-SKIP       : O"015" ; CR
DELIMITER-WITHOUT-SKIP       : O"021" ; SBA
DELIMITER-WITHOUT-SKIP       : O"037" ; US
DELIMITER-WITH-SKIP          : O"040" ; SPACE
DELIMITER-WITHOUT-SKIP       : O"054" ; COMMA
DELIMITER-WITHOUT-SKIP       : O"057" ; SLASH
DELIMITER-WITHOUT-SKIP       : O"072" ; COLON
DELIMITER-WITHOUT-SKIP       : O"073" ; SEMI-COLON
```

Following entries correspond to configuration options within the USRVAL and USREDT modules. G8WB Screen Painter will use the Validation/Editing configuration actively, thus only configured rules may be applied to form fields.

```
[VALIDATION]
; Validation Rule-code  Descriptive-text                 Module
ALP                   : "Alphabetic validation"           NONE
NUM                   : "Numeric validation (+ editing)"  NONE
NUV                   : "Numeric validation only"         NONE
DAT                   : "Date validation"                 NONE
CDA                   : "YYMMDD date validation"          NONE
ADA                   : "MMDDYY date validation"          NONE
TIM                   : "Time validation"                 NONE

[EDITING]
; Editing Rule-code     Descriptive-text                 Module
NUM                   : "Numeric editing"                 NONE
NUV                   : "Numeric editing suppress"        NONE
DAT                   : "Date editing"                    NONE
CDA                   : "YYMMDD date, no editing"         NONE
ADA                   : "MMDDYY date  editing"            NONE
TIM                   : "Time editing"                    NONE
```

Visual configuration has no correspondance with DIMS8. G8WB Screen Painter will use the Visual configuration actively, thus only configured rules may be applied to form fields.

```
[VISUAL]
; Visual Attribute-code Descriptive-text
SMA                   : "Upper and lower case"
LOW                   : "Low Intensity"
BLI                   : "Blinking"
REV                   : "Reverse Video"
SEC                   : "Concealed/Security"
RES                   : "Reset standard"
```

Following entries correspond to configuration options within the USRTTY module.

```
[TERMINAL]
; Terminal emulation    Handler-module Terminal-type
W8V77XXE              : DTHQP1         O"015"
W8V78XXE              : DTHV78         O"022"
W8DKUXXE              : DTHQP2         O"006"
W8IBMXXE              : DTHI32         O"021"
```

## RFV Configuration

RFV configuration is specific to Pilspråk forms support.

### Forms

GENERATE-COPY may be used to control generation of Form Copy for Pilspråk forms. Extended setting will assign labels to cursor and attribute entries, and generate OCCURS construct for ROW/ARRAY.

### TX-Information

Option settings may be used to prime corresponding TX-storage fields, whenever a new transaction start.

```
;
; RFV Configuration Version 1.2.00 (Server Config)
;
[FORMS]
GENERATE-COPY                : EXTENDED ; EXTENDED or NORMAL COPY syntax

[TX-INFORMATION]
OPERATOR-ID                  : "ABCDEFGH" ; X(08) "CKLKLPNR"
BADGE-CARD                   : O"101101101101" ; COMP-6 Octal
;BADGE-CARD                   : 0               ; COMP-6 Numeric
TP8-SYSTEM-ID                : "X" ; X(01)
;BST-KAT                      : O"102102102102"  ; COMP-6 Octal
BST-KAT                      : 52               ; COMP-6 Numeric
BST-OMR                      : "B" ; X(01)
```

## TSM8 Configuration

G8WB’s implementation of TSM8 does not include the host control database, and the associated TSM8 transactions. Instead, Forms, Menus, Messages etc. are maintained through G8WB Screen Painter, which store these entities within the $G8WBFDIR directory. TSM8 configuration may be used to set various default for TSM8 execution.

### User

Used to assign USERID, DISTRICT and STATION for TSM8 session.

```
;
; TSM8 Configuration Version 1.2.00 (Server Config)
;
[USER]
USERID                 : ADMIN
DISTRICT               : LOCAL
STATION                : MAST
```

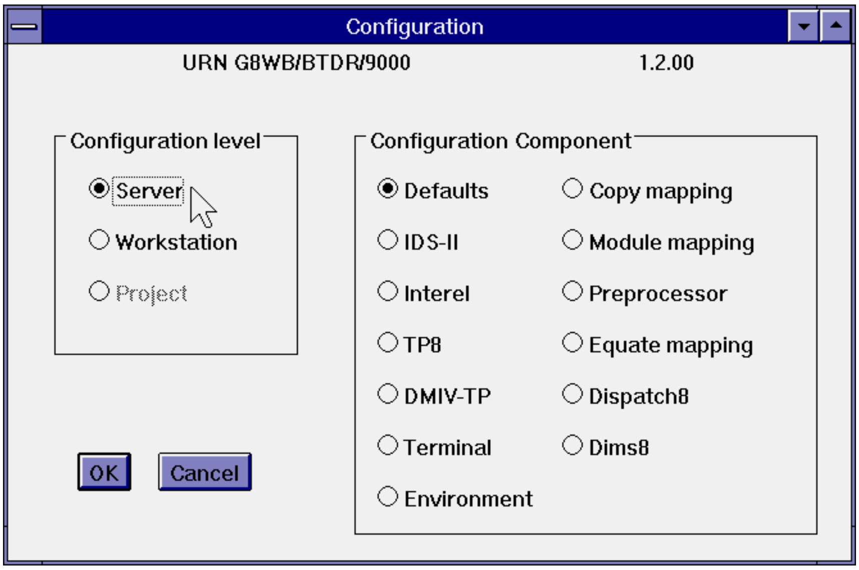
### Header

Used to define DEFAULT header for TSM8 template.

```
[HEADER]
DEFAULT                : "<< TSM8 >>"
```

# GUI Workbench Configuration

For GUI Workbench, you invoke configuration maintenance through the Configure Tool icon.



URN G8WB/BTDR/9000 1.2.00

Configuration level

- ● Server
- ○ Workstation
- ○ Project

Configuration Component

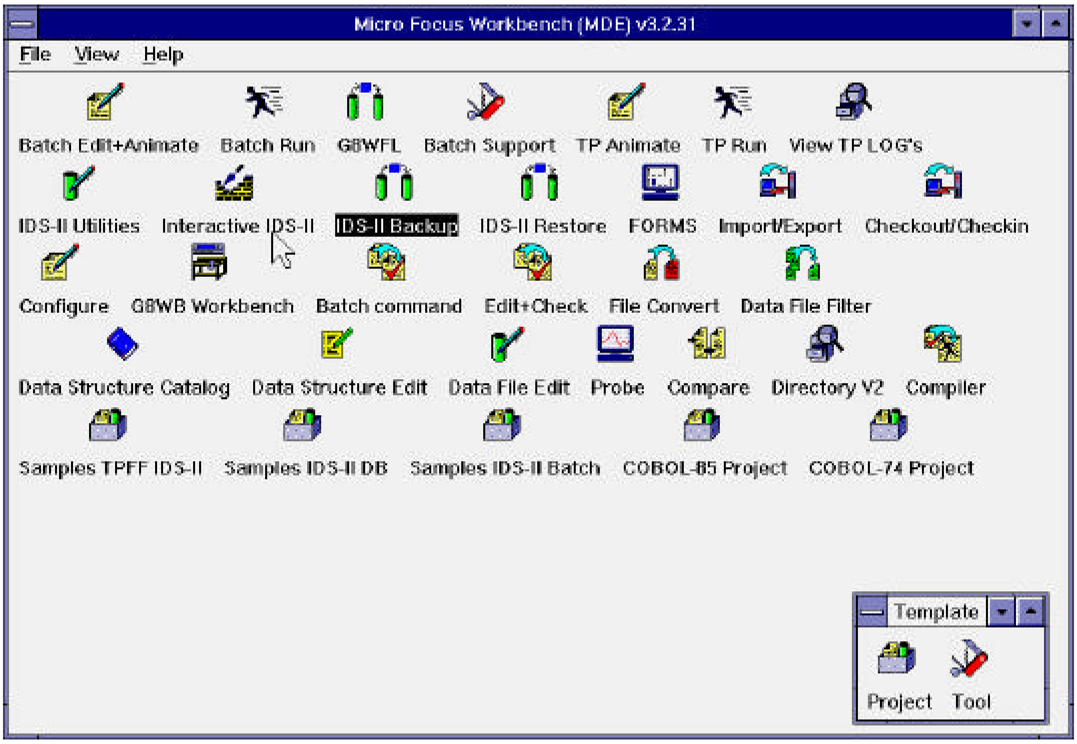
| | |
|---|---|
| ● Defaults | ○ Copy mapping |
| ○ IDS-II | ○ Module mapping |
| ○ Interel | ○ Preprocessor |
| ○ TP8 | ○ Equate mapping |
| ○ DMIV-TP | ○ Dispatch8 |
| ○ Terminal | ○ Dims8 |
| ○ Environment | |

The selected configuration is made available for maintenance through MF Editor.

## Workbench Organisor

The Workbench Organisor replace the Menu’s of the character-mode workbench. The use of Tool and Project objects and drap-and-drop technique, provides a very easy and flexible user interface. The Organisor is (currently) limited to one level, so all Tool and Project icons required must be immediately present. Assigning a Tool icon to each of the available G8WB utilities would create a VERY crowded organisor, so many of the G8WB utilities have been grouped under a single Tool icon. Some of the less frequently used tools like IDS-II Schema Translator etc. are only avialable through character-mode workbench, which can be invoked through the G8WB Workbench icon.

You would normally customize the organisor to suit your specific needs for Tools and Projects. You may also use separate organisor's, through different MFORG setting. Please refer to MFWB documentation for details.

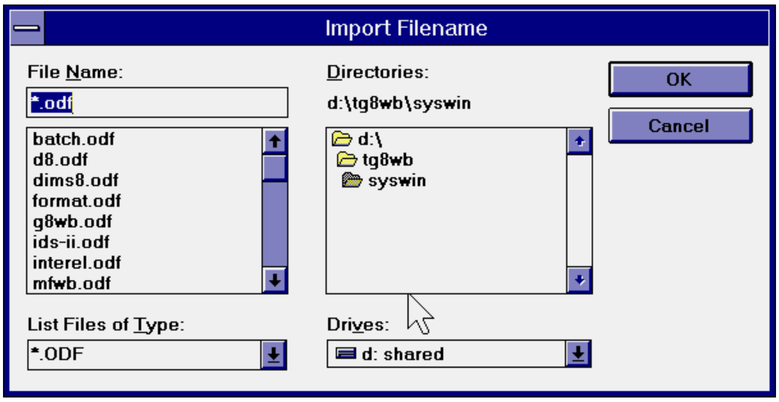


You have different ways of customizing the Organisor.

## Import

Import may be used to import a text file describing tools and/or projects. The import function is invoked through File menu of organisor. G8WB provides a number of import files, which may be used to complement or re-build the organisor. These files are located under $G8WBDIR\SYSWIN for Windows and $G8WBDIR\SYSPM for OS/2.

| Import file | Content |
|---|---|
| MFWB.ODF | Standard MFWB tools and projects |
| G8WB.ODF | G8WB basic tools and projects |
| BATCH.ODF | G8WB Batch tools and projects |
| TP.ODF | G8WB TP tools and projects |
| IDS-II.ODF | G8WB IDS-II tools and projects |
| INTEREL.ODF | G8WB INTEREL tools and projects |
| D8.ODF | G8WB DISPATCH8 tool |
| TPFF.ODF | Samples TPFF projects |
| FORMAT.ODF | Samples FORMAT projects |
| DIMS8.ODF | Samples DIMS8 projects |
| PIL.ODF | Samples Pilspråk projects |
| TSM8.ODF | Samples TSM8 projects |
| READY.ODF | G8WB default organisator |



Import Filename

File Name: *.odf

Directories: d:\tg8wb\syswin

List Files of Type: *.ODF

Drives: d: shared

Once you have completed your customization, you may use the Export function to save the entire organisor on a text file (ODF). This may be used for backup purposes or used to distribute an organisor model.

## Project

Project object may be used to establish proper settings for one or more project. You thus apply such setting when you drag the project icon and drop it on a specific tool. The settings related to a Project object are as follows:

Name, which makes the project easily recognized within the organisor.

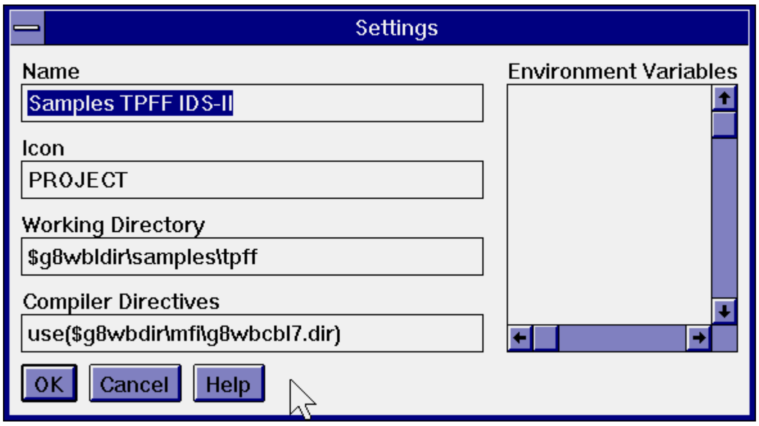
Working directory, which direct tool(s) to use the appropriate directory as current path during execution.

Icon, default is the MFWB Project icon.

Compiler directives, which define which source type checking to be applied by Checker or Animator. For G8WB you ALWAYS need to specify compiler directives, otherwise G8WB preprocessing will not be enabled. G8WB specific compiler settings are as follows:

| Source type | Compiler directives |
|---|---|
| COBOL-74 | use($g8wbdir\mfi\g8wbcbl7.dir) |
| COBOL-85 | use($g8wbdir\mfi\g8wbcbl8.dir) |
| COBOL-SQL | use($g8wbdir\mfi\g8wbcblq.dir) |
| TPR-74 | use($g8wbdir\mfi\g8wbtpr7.dir) |
| TPR-85 | use($g8wbdir\mfi\g8wbtpr8.dir) |
| TPR-SQL | use($g8wbdir\mfi\g8wbtprq.dir) |
| PRG-74 | use($g8wbdir\mfi\g8wbprg7.dir) |
| PRG-85 | use($g8wbdir\mfi\g8wbprg8.dir) |
| PRG-SQL | use($g8wbdir\mfi\g8wbprgq.dir) |

Environment variables, which may be used to environment variables before execution. An example could be setting of G8WBPDIR.



Name: Samples TPFF IDS-II

Icon: PROJECT

Working Directory: $g8wbldir\samples\tpff

Compiler Directives: use($g8wbdir\mfi\g8wbcbl7.dir)

You have different ways of creating new Project objects.

Organisor Template, selecting Project icon will create a new project.

G8WB template(s), select Settings on either COBOL-74, COBOL-85 or COBOL-85 SQL, will allow you to copy the project with predefined compiler directives.

Project, select Settings on any Project, will allow you to copy the project with current settings.

## Tool

Tool object may be used to invoke any of the MFWB or G8WB utilities. As an example you can create a separate tool for a frequently used G8WB utility, which is otherwise only indirectly available.

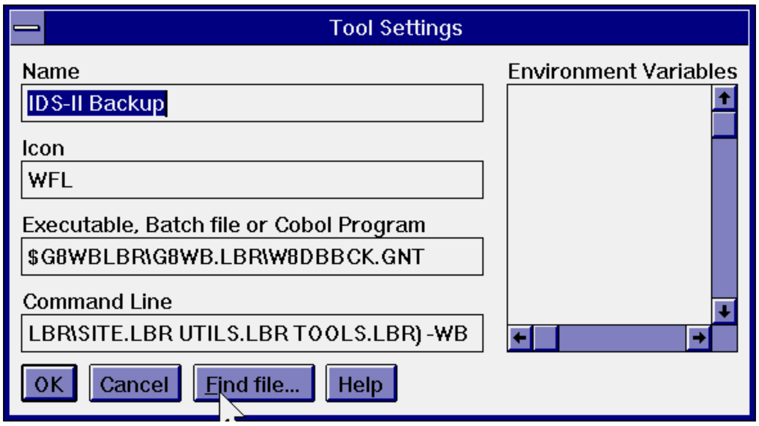
Name, which makes the tool easily recognized within the organisor.

Icon, select of the standard MFWB icons or create your own specific icons.

Executable, defines which program to launch.

Command Line, define libraries to open and command line option to pass to the program.

Please refer to MFWB documentation for details on setting options.



Name: IDS-II Backup

Icon: WFL

Executable, Batch file or Cobol Program: $G8WBLBR\G8WB.LBR\W8DBBCK.GNT

Command Line: LBR\SITE.LBR UTILS.LBR TOOLS.LBR) -WB

You have different ways of creating new Tool objects.

Organisor Template, selecting Tool icon will create a new tool.

Tool, select Settings on any Tool, will allow you to copy the project with current settings.

# Chapter 11 GUIDE LINES

G8WB was designed with the intention to provide a completely GCOS-8 compatible environment, both in terms of compiler and execution behaviour. This has been achieved to a very high degree, however G8WB is NOT 100% compatible with GCOS-8.

This chapter describes the differences and limitations between the GCOS-8 and G8WB environment.

## Preprocessing

Micro Focus COBOL/2 compiler system does not support BULL's COBOL-74/COBOL-85 adaptions and G8WB provides an integrated Preprocessor, which will migrate specific BULL syntax into COBOL/2 syntax. G8WB Preprocesor will attempt both to make the integrated compiling system behave functionally equivalent to BULL compilers, and at the same time ensure that the migrated syntax will execute functionally equivalent to GCOS-8. Preprocessing of IDS-II DML verbs are built into the G8WB Preprocessor, whereas INTEREL SQL is handled by a combination of XDB-DB2 preprocessor and G8WB Preprocessor.

In general G8WB environment may be more restrictive than BULL compilers in a few specific areas. This behaviour has little, if any, impact on new development, however for maintenance of some existing application systems, you may be required to make minor adjustments of existing coding. Such adjustements will, however be compatible with GCOS-8.

In some specific areas, G8WB environment may be less restrictive than BULL compilers. This behaviour, which affect only new or changed coding, could result in failure to clean-compile application programs, once they are returned to GCOS-8. The number of "less restrictive" incompatibilities are limited and have little impact in practice. Most of the incompatibilities are related to use of COBOL-85 features in COBOL-74 programs.

### JCL

GCOS-8 JCL within source is removed by G8WB Preprocessor. Leading JCL (before first non comment line of source) is ignored. Any subsequent JCL is treated as end of source and all remaining lines (JCL or other) are ignored.

## COBOL

### Format

G8WB does not allow neither keywords nor arguments to be split between lines via continuation in column 7.

Missing "." delimiter on data-field definition, section names etc. are generally recovered by GCOS-8 (warning given), however, G8WB will flag this as an error (various types of messages given). Insert of "." is required.

### Reserved words

The word AUTO is reserved in G8WB.

### Control division

Control Division is partially supported.

Default clauses for DISPLAY and COMP-4 sign are ignored and a warning is given.

Substitution REPLACE clauses are ignored and a warning is given.

Default clauses SYMBOLIC and USE are flagged as errors.

## Program-id

GCOS-8: Program-id is used by loader to resolve linkage.

G8WB: Program-id is ignored, instead the filename, i.e. program.CBL is used to resolve dynamic linkage. If program-id differ from filename, a warning is given.

## Source-computer

Source and Object computer must at least include LEVEL-66-ASCII, if specified.

## Block

BLOCK nn IS file-name-1 is converted to obtain similar functionality, through the use of EXTERNAL on FD and all FD record(s). This technique require that FD and FD record(s) have equal names in all programs, unless BLOCK-UNIFY is used. Ref. Configuration chapter for details.

BLOCK nn IS data-name-1 [THRU data-name-2 ] is converted to obtain similar functionality, through the use of EXTERNAL on 01 definitions referenced. This technique require that 01 definitions have equal names in all programs, unless BLOCK-UNIFY is used. Ref. Configuration chapter for details.

VALUE clause is not allowed for data-name-1 or any of its subordinate identifiers. VALUE clause must be replaced by initialize coding in Procedure Division to make G8WB compatible with GCOS-8.

REDEFINE clause for data-name-1 is not allowed. The involved 01 identifiers must be joined and the REDEFINE moved to a subordinate level, to make G8WB compatible with GCOS-8.

Block structures must have same size in all programs, otherwise G8WB will give a run-time error "External data size inconsistent". GCOS-8 does allow unequal sizes, and will only give a warning during GELOAD.

## Collating

COLLATING SEQUENCE alphabet-name|ASCII|EBCDIC is allowed by GCOS-8, however G8WB only support the alphabet-name syntax. If a predefined alphabet, i.e. EBCDIC, is given, you need to change your source as follows:

```
   COLLATING SEQUENCE EBCDIC
```

becomes

```
   COLLATING SEQUENCE AEBCDIC (or other name)
```

and the following clause must be added to SPECIAL-NAMES

```
   AEBCDIC IS EBCDIC
```

This source change is compatible with GCOS-8.

## Decimal point

GCOS-8 will accept a period (.) as decimal point in numeric literals, even if “Decimal point is comma” is specified, i.e.COMPUTE field = field * 1.5.

G8WB will, in a similar situation, terminate the numeric literal at the point of the period (.) If this occur in the middle of a statement, i.e. MULTIPLY 1.5 BY field, it is flagged, however if this occur at the end of a statement, i.e. COMPUTE field = field * 1.5., it is NOT flagged, but will give a different result.

## Select

If SELECT file-name-1 ASSIGN TO fc-PRINTER/-CARD-READER/-CARD-PUNCHER is used, then ASSIGN must immediately follow SELECT file-name-1, otherwise the organization for the file will not be correct.

SELECT file-name-1 ASSIGN TO fc....-PRINTER. The -PRINTER suffix must be present to trigger print formatting. This apply also for Report-Code and Report-Writer files, otherwise the Expand Print utility will fail.

*Note: GCOS-8 COBOL-74 documentation specify that -PRINTER must be given (proper media code assigned), however on GCOS-8 a direct allocation to SYSOUT will work even though -PRINTER is not specified.*

## Copy

COPY .. REPLACING is restricted to ==...== and identifier formats.

COPY ... OF Library-tag. The Library-tag must not be a reserved word as this will conflict with preprocessor conversion rules, i.e. COPY ABC OF SD may be flagged with “TPR not allowed” or “COPY incomplete” because SD is recognized as an SD (Sort Description entry). Amend the Library-tag so this conflict is removed, i.e. SD changed to SD-LIB.

When COPY syntax is used to for a part of a COBOL verb, the COPY will not be expanded during Animation, i.e. Animator will not show the content of <arguments> for CALL "abc" USING COPY arguments.

Use of COPY ... REPLACING will not make changed indentifiers recognizable to Animator, which will show original (before replace) source, however you are not able to query any changed identifiers.

## Picture

Numeric fields are limited to 18 digits. PICTURE > 18 is valid on GCOS-8, although a warning is given.

GCOS-8 will accept a PIC Z(5)9.Z(3), although this is incorrect use of LEADING ZERO SUPRESSION. G8WB will flag this as an error.

PIC clause for a field with usage COMP-1, COMP-2 or COMP-6 (documentary on GCOS-8) is removed by G8WB, so it does not conflict with the PIC clause inserted during editing of COMP-1, COMP-2 or COMP-6 usage. This removal only work when both PIC and USAGE clause are on the same source line. Avoidance: Make sure PIC and USAGE are on the same line or remove the PIC clause.

## Value

VALUE clause with non-matching type, i.e. DISPLAY numeric field with alphanumeric value or reverse, is accepted by GCOS-8, however G8WB will flag this as an error.

## Literal

Use of '...' literal syntax is supported, except for ASCII literal, i.e. ""28"", which must use standard COBOL quotes.

G8WB will not flag a literal declaration like - "literal".. - where double period is invalid.

## Redefine

Redefine at elementary level (>01) - G8WB will not flag a redefine description which is longer than the original.

## Usage

### Numeric Display

GCOS-8 generally treat low-values and spaces in numeric display fields as zero, except for numeric test.

G8WB does not allow low-values in numeric display fields.

G8WB treat spaces in numeric display fields as zero, including numeric test. This behaviour is default, however you may change this so spaces are treated like low-values. Ref. Configuration chapter for details.

### COMP-1

USAGE at group level is not supported.

Dynamic change to COMP-2 based on PIC is not supported, i.e. always treated as COMP-1.

### COMP-2

USAGE at group level is not supported.

### COMP, -3, 4 & 5

Redefine of packed decimal fields is not supported. Please refer to COMP Analyzer in Configuration chapter for details on how to cope with this limitation.

### COMP-6

USAGE at group level is not supported.

Default implementation for COMP-6 is similar to COMP-2 (31 bit signed). Alternate implementation to support redefine or 35 bit signed is possible. Ref. Configuration chapter for details.

### INDEX

You cannot redefine an INDEX field, because the implementation on GCOS-8 and G8WB is different.

### DB-Key

USAGE at group level is not supported.

G8WB will not flag MOVE to a field or INITIALIZE of a field with usage DB-Key.

Default implementation for DB-Key is similar to COMP-2 (31 bit signed). Alternate implementation to support 35 bit signed is possible. Ref. Configuration chapter for details.

## Tables

G8WB will flag any explicit boundary violation for a table, i.e. FIELD OCCURS 5 -> MOVE ZERO TO FIELD(6). Change table size or coding to correct syntax.

G8WB will not flag use of index (INDEXED BY) from another table.

## Report Writer

G8WB will flag any invalid LINE specification for Report Writer, i.e. FIRST DETAIL LINE 4 -> DETAIL LINE IS 3. Change either value to the logical correct value.

## Call

In CALL "literal", G8WB will replace any period with underscore in literal to avoid conflict with DOS or OS/2 naming conventions. The called module must be renamed accordingly.

G8WB use dynamic linking, and implicit linkage to entry points performed by GCOS-8 loader, must be configured to G8WB, i.e. CALL to entry-point in program without any explicit call to the program containing the entry-point. Also use of GCOS-8 $ EQUATE loader directives need to be configured to G8WB. Please refer to Configuration chapter for details on Module and Equate configuration.

## Go to

Statement out of reach, i.e. GO TO followed by another statement, is not flagged.

## Initialize

GCOS-8 will allow INITIALIZE identifier without subscript and default to initialize all occurences. G8WB does require a subscript or to obtain same functionality, the INITIALIZE should reference a superior group level identifier (correct syntax).

## Move

For MOVE numeric-display TO comp-x you must ensure the move is numeric, otherwise you get incorrect result.

GCOS-8 will, in most instances, accept either an alphanumeric, i.e. PIC X to COMP, or a numeric, i.e. PIC 9 to COMP, so you may see this problem in existing application programs.

## Next sentence

G8WB will not flag when a statement is followed by NEXT SENTENCE.

## Paragraph

G8WB will not flag duplicate paragraph names unless these are referenced.

### Set

G8WB does not support the SET switch-nn TO switch-nn syntax.

## COBOL-74

### Add

ADD field-1 TO field-2 GIVING field-3 is not flagged for COBOL-74 program, even though "TO" is ANS85 syntax.

### Examine

EXAMINE field REPLACING UNTIL FIRST ... syntax is not supported.

## COBOL-85

### Names

GCOS-8 will allow the use of "_" and "$" in COBOL names. This is not ANSI85 standard, however G8WB Preprocessor will accept this and change source naming as follows:

leading "_" to "0"

any other "_" to "-"

leading "$" to "9"

any other "$" to "-"

to make it compatible with GCOS-8. This automatic adjustment of names is not recognized by Animator, which will show original COBOL names, however you are not able to query such COBOL names. It is recommended to avoid use of such non standard naming.

### Usage

#### COMP-7

Default implementation for COMP-7 is similar to COMP-1 (15 bit signed). Alternate preprocessing implementation is 23 bit signed (3 bytes).

#### COMP-8

COMP-8 is implemented similar to COMP-4.

### X9MA

CALL "X9MA$ALLOCATE" is supported, with following restrictions:

- Size is limited to 64KB vs. 1024KB on GCOS-8

- "BY CONTENT LENGTH OF identifier-3" format cannot be used (length is not recognized correctly).

*Note: Heap space is limited by available free (real) memory within G8WB vs virtual memory on GCOS-8.*

CALL "X9MA$FREE" is supported.

*Note: If you attempt to "X9MA$FREE" using a pointer, which was not allocated through “X9MA$ALLOCATE", G8WB will crash within W8VMPS.*

## Batch COBOL-74

Following table illustrates the common COBOL-74 batch service calls and how they are processed within G8WB.

| Function | Process | Remarks |
|---|---|---|
| .DCKPT | Yes | Only database files |
| .DCKPF | Yes | Only database files |
| .DROLF | Yes | Only database files |
| .IPSTI | Partial | Defaults are set-up |
| WRAP | No | Ignored |
| USERID | No | Ignored |
| SETL | No | Ignored |
| DEACCS | No | Ignored |
| ACCESS | No | Ignored |
| TSCOPT | No | Always return “Batch environment” |
| FASTIO | No | Ignored |

## Batch COBOL-85

Following table illustrates the common COBOL-85 batch service calls and how they are processed within G8WB.

| Function | Process | Remarks |
|---|---|---|
| .DCKPT | Yes | Only database files |
| .DCKPF | Yes | Only database files |
| .DROLF | Yes | Only database files |
| CHECKPOINT | Yes | Only database files |
| CHECKPOINT_FILES | Yes | Only database files |
| ROLLBACK | Yes | Only database files |
| ROLLBACK_FILES | Yes | Only database files |
| X$CKPT_PROCESS_STATUS | No | Defaults set-up |
| FASTIO | No | Ignored |

## TP

Following table illustrates the common DMIV-TP and TP8 service calls and how they are processed within G8WB. The TP Journal Log will document all calls.

| Function | Process | Remarks |
|---|---|---|
| .ABORT | Yes | |
| .CANCL | Yes | |
| .SLEEP | Partial | Verified and documented (Sleep period is ignored) |

| | | |
|---|---|---|
| .TPREX | Yes | |
| .BRDCT | Partial | Verified and documented |
| .CHANC | Yes | |
| .LOCK | Partial | No effect |
| .LSTAT | Yes | |
| .PURGE | Yes | |
| .SPWNB | Partial | Verified and Documented |
| .UJRNL | Yes | |
| .UNLCK | Partial | No effect |
| .UNLKA | Yes | |
| .UNLKV | Partial | No effect |
| .WAKE | Partial | Verified and documented |
| .FREE | Yes | |
| .FREER | Yes | |
| .LOGON | No | Ignored |
| .NABRT | Partial | Work as .ABORT |
| .DHCLR | Yes | |

## DMIV-TP

Following table illustrates the various DMIV-TP service calls and how they are processed within G8WB. The TP Journal Log will document all calls, if trace is activated.

| Function | Process | Remarks |
|---|---|---|
| .NCMIT | No | Ignored |
| .DISTR | No | Ignored |
| .LNKST | No | Ignored |
| .SECHD | No | Ignored |
| .SEQNO | Yes | |
| .TPINF | Partial | Limited information |
| .TPRNM | Yes | |

## TP8

Following table illustrates the various TP8 unique service calls and how they are processed within G8WB. The TP Journal Log will document all calls, if trace is activated.

| Function | Process | Remarks |
|---|---|---|
| .ABTTN | No | Ignored |
| .AJCMT | No | Ignored |
| .ALPRM | No | Ignored |
| .BAKDR | No | Ignored |
| .CAPRM | No | Ignored |
| .CATMP | No | Ignored |
| .CRCLG | No | Ignored |
| .DAPRM | No | Ignored |
| .DEFNG | Yes | |
| .FLINF | No | Ignored |
| .DLTMP | No | Ignored |
| .FNLCP | No | Ignored |
| .GRPRM | No | Ignored |
| .GRTMP | No | Ignored |
| .ILINK | Yes | |
| .LID | No | Ignored |
| .LIDEX | No | Ignored |
| .LOGOT | No | Ignored |

| | | |
|---|---|---|
| .PGPRM | No | Ignored |
| .RABRT | No | Ignored |
| .SRCVM | No | Ignored |
| .SWTM | Yes | |
| .RST | No | Ignored |
| .TQSVC | No | Ignored |
| .TRACE | Yes | |
| .UCASE | Yes | |
| .UCNCL | No | Ignored |
| .WRUP | No | Ignored |
| .WSNAM | Yes | |

## .SWTM

.SWTM to originator LID will result in immediate execution of next TX, i.e. any sleep period is ignored.

.SWTM to originator LID is limited to one message.

.SWTM to non-originator LID is logged on the Spawn Log.

## VMAPI

Following table illustrates the various TP8 Virtual-Mode Application Programmatic Interfaces (VMAPI) and how they are processed within G8WB. The TP Journal Log will document all calls, if trace is activated.

| Function | Process | Remarks |
|---|---|---|
| _TP_ABORT_TRANSACTION | Yes | |
| _TP_CALL_TPR | Yes | |
| _TP_USER_PROCEDURE | Yes | |
| _TP_BROADCAST | Ignored | |
| _TP_CANCEL_TPR | Ignored | |
| _TP_DB_COMMITMENT | Ignored | |
| _TP_FILE_SERVICES | Ignored | |
| _TP_GET_PROCESS_CONFORMANCE | Ignored | |
| _TP_GLOBAL_STORAGE | Ignored | |
| _TP_IC_SERVICES | Ignored | |
| _TP_JOURNALIZE | Ignored | |
| _TP_LID_SERVICES | Ignored | |
| _TP_PROCESS_SERVICES | Ignored | |
| _TP_PURGE_MESSAGES | Ignored | |
| _TP_RECEIVE_MESSAGE | Ignored | |
| _TP_RESOURCE_SERVICES | Ignored | |
| _TP_SEND_MESSAGE | Ignored | |
| _TP_SET_PRIORITY | Ignored | |
| _TP_SLEEP_SERVICES | Ignored | |
| _TP_SPAWN_BATCH | Ignored | |

| Function | Process | Remarks |
|---|---|---|
| _TP_SPAWN_TRANSACTION | Ignored | |
| _TP_TPR_SERVICES | Ignored | |
| _TP_TQ_SERVICES | Ignored | |
| _TP_USER_TRACE | Ignored | |
| _TP_WORKSTATION_SERVICES | Ignored | |

*Note: "Ignored" will return PRIMARY-STATUS = 3.*

*Note: VMAPI calls are not allowed in COBOL-74, and call names must be given in UPPER-CASE.*

### Send

Multi buffered SEND is only partially supported, as TP only use one messag-buffer.

## BATCH

### Sort

GCOS-8 SORT (GMAP sort routines) is not directly supported, however you are able to migrate existing sort routines quite easily to Micro Focus SORT Command Line. Please refer to MFWB documentation for details.

## VMPS

Following table illustrates the various GCOS-8 Virtual-Mode Programmatic Services (VMPS) and how they are processed within G8WB. When performed under TP8, the TP Journal Log will document all calls, if trace is activated.

| Function | Process | Remarks |
|---|---|---|
| X_UNPACK_STATUS | Yes | |
| X_ABORT | Yes | |
| X_CONTINUE_CATALOGS | Ignored | |
| X_CONTINUE_FILES | Ignored | |
| X_CONTINUE_FMS_QUERY | Ignored | |
| X_COUNT_CATALOG_PERMISSIONS | Ignored | |
| X_COUNT_CATALOGS | Ignored | |
| X_COUNT_FILE_PERMISSIONS | Ignored | |
| X_COUNT_FILES | Ignored | |
| X_CREATE_CATALOG | Ignored | |
| X_CREATE_FILE | Ignored | |
| X_DELETE_CATALOG | Ignored | |
| X_DELETE_FILE | Ignored | |
| X_GENERATE_SNUMB | Ignored | |
| X_INIT_CATALOG_ATTRIBUTES | Ignored | |
| X_INIT_FILE_ATTRIBUTES | Ignored | |
| X_LIST_CATALOGS | Ignored | |
| X_LIST_FILES | Ignored | |
| X_MODIFY_CATALOG | Ignored | |
| X_MODIFY_FILE | Ignored | |
| X_QUERY_BASE_RELEASE | Ignored | |
| X_QUERY_CATALOG_ATTRIBUTES | Ignored | |
| X_QUERY_CATALOG_PERMISSIONS | Ignored | |

| Function | Process | Remarks |
|---|---|---|
| X_QUERY_DATE_TIME_NUMERIC | Ignored | |
| X_QUERY_DATE_TIME_STRING | Ignored | |
| X_QUERY_ENVIRONMENT | Ignored | |
| X_QUERY_FILE_ATTRIBUTES | Ignored | |
| X_QUERY_FILE_PERMISSIONS | Ignored | |
| X_QUERY_IDENT_CARD | Ignored | |
| X_QUERY_IDENTIFIERS | Ignored | |
| X_QUERY_IMMEDIATE_STATUS | Ignored | |
| X_QUERY_JOB_STATUS | Ignored | |

| | | |
|---|---|---|
| X_QUERY_MESSAGE | Ignored | |
| X_QUERY_MESSAGE_DATA | Ignored | |
| X_QUERY_MODULE_ID | Ignored | |
| X_QUERY_ORGINAL_STATUS | Ignored | |
| X_QUERY_PROCESSOR_TIME | Ignored | |
| X_QUERY_SECURITY_OPTIONS | Ignored | |
| X_QUERY_SERVICE_CALL | Ignored | |
| X_QUERY_SNUMB_ACTIVITY | Ignored | |
| X_QUERY_SOFTWARE_SET | Ignored | |
| X_QUERY_SYSTEM_ID | Ignored | |
| X_QUERY_SYSTEM_PRODUCTS_USERID | Ignored | |
| X_QUERY_SYSTEM_TYPE | Ignored | |
| X_SORT_CATFILE_NAMES | Ignored | |
| X_SPAWN_JOB_FROM_FILE | Ignored | |
| X_SPAWN_JOB_FROM_MEMORY | Ignored | |
| X_WAIT | Ignored | |
| X_FINAL_CLOSE | Ignored | |
| X_USOF | Ignored | |
| X_USGN | Ignored | |
| X_USPN | Ignored | |
| X_USBS | Ignored | |
| X_USFS | Ignored | |
| X_USRW | Ignored | |
| X_USCP | Ignored | |
| X_USCF | Ignored | |
| X_UIOF | Ignored | |
| X_UIGN | Ignored | |
| X_UIPN | Ignored | |
| X_UIGR | Ignored | |
| X_UIPR | Ignored | |
| X_UIRW | Ignored | |
| X_UIPF | Ignored | |
| X_UIDL | Ignored | |
| X_UICF | Ignored | |
| X_UROF | Ignored | |
| X_URGN | Ignored | |
| X_URPN | Ignored | |
| X_URGR | Ignored | |
| X_URPR | Ignored | |
| X_URRW | Ignored | |
| X_URPF | Ignored | |
| X_URDL | Ignored | |
| X_URCF | Ignored | |
| X_GSOF | Ignored | |
| X_GSGN | Ignored | |
| X_GSPN | Ignored | |
| X_GSBS | Ignored | |
| X_GSFS | Ignored | |
| X_GSRW | Ignored | |
| X_GSCP | Ignored | |

| Function | Process | Remarks |
|---|---|---|
| X_GSCF | Ignored | |
| X_RIOF | Ignored | |
| X_RIGN | Ignored | |
| X_RIPN | Ignored | |
| X_RIGR | Ignored | |
| X_RIPR | Ignored | |
| X_RIRW | Ignored | |
| X_RIPF | Ignored | |

| | |
|---|---|
| X_RIDL | Ignored |
| X_RICF | Ignored |
| X_INFO | Ignored |
| X_CLOSCN | Ignored |
| X_OPENCN | Ignored |
| X_RCVINT | Ignored |
| X_RECVDT | Ignored |
| X_REQSTAT | Ignored |
| X_SENDDT | Ignored |
| X_SNDINT | Ignored |
| X_TESTCN | Ignored |

*Note: "Ignored" will return PRIMARY-STATUS = 3.*

*Note: VMPS calls are not allowed in COBOL-74.*

# IDS-II

## Schema

### Unspecified

Due to difference in implementation of non-DISPLAY format fields, unspecified field types, which are later redefined in subschema or application, may have insufficient length within G8WB. Please refer to IDS-II Database chapter for details.

### Character

Due to difference in implementation of non-DISPLAY format fields, character field types, which are later redefined in subschema or application, may have insufficient length within G8WB. Please refer to IDS-II Database chapter for details.

### Database procedures

Only procedures implemented in COBOL-74 or COBOL-85 are supported.

## Subschema

### Display

Due to difference in implementation of non-DISPLAY format fields, PIC X field types, which are later redefined in the application, may have insufficient length within G8WB. Please refer to IDS-II Database chapter for details.

### COMP, -3, 4 & 5

Use of COMP Analyzer feature may require change of USAGE for COMP, COMP-3, COMP-4 or COMP-5 fields in the subschema. Please refer to IDS-II Database and Configuration chapters for details.

### COMP-6

Default implementation of COMP-6 is similar to COMP-2 (31 bit signed), whereas schema implementation is 35 bit signed. Field transformation is automatic, however truncation is possible. Alternate implementation to support 35 bit signed is possible. Ref. IDS-II Database chapter for details.

### DB-Key

Default implementation of DB-Key is similar to COMP-2 (31 bit signed), whereas schema implementation is 35 bit signed. Field transformation is automatic, however truncation is possible. Alternate implementation to support 35 bit signed is possible. Ref. IDS-II Database chapter for details.

## DML

### Find

GCOS-8 will accept FIND LAST .. USING key-name, although this is not documented. G8WB will flag this as an error.

## DB Unload

GCOS-8 may allow duplicate records even when duplicates are not allowed, i.e. DUPLICATE clause changed after load of some records. If such records are unloaded with DBWFL, WDBLnn will abort with a duplicate error status.

# INTEREL

GCOS-8 INTEREL is generally compatible with ANSI and DB2 SQL standards, however there are several differences in the area of host adaption and SQL features. G8WB INTEREL Support, which is based on XDB-DB2 Workbench running in DB2-mode, will attempt to make such differences transparent, so GCOS-8 INTEREL application development can be performed within G8WB, without or with limited adaption requirements.

G8WB currently makes no attempt to prevent use of XDB/DB2 features, so SQL command files or programs with embedded SQL could very well function within G8WB, but turn out to be incompatible with INTEREL, once it is transferred back to GCOS-8.

INTEREL Support is limited to "Host Specific" implementation, i.e. RDBC not supported.

INTEREL Utilities are not supported, however XDB provides utilities with similar capabilities.

## Naming

XDB will allow use of #$%&{} characters in identifiers, however INTEREL will not. This has little practical impact, except that if you mistakenly use these characters, XDB will not flag it.

INTEREL will allow up to 30 characters for identifiers, except for Table, Tablespace, View, Snapshot and Model, which allow up to 12 characters. This is slightly different in XDB which will allow up to 18 characters for identifiers, except for Location (Model) which allow up to 16 characters. This difference has little practical impact, unless you use more than 16 character naming within your INTEREL application.

XDB will allow both (') and (") to define literals, however INTEREL will only allow ('). This has little practical impact, except that if you mistakenly use ("), XDB will not flag it.

## Data types

INTEREL and XDB basically support the same data types, however due to differences in the IBM/BULL Host platforms, there are differences/limitations in relation to the precision of some data types.

| Data Type | INTEREL | XDB/G8WB |
|---|---|---|
| DECIMAL | Max. integer 59<br>Max. Fraction 31 | Max. Integer/fraction 31<br>Max. Integer/fraction 18 (COBOL) |
| | Sign optional | Sign mandatory<br>(adapted by G8WB, if unsigned) |
| | Host variable must<br>DISPLAY, if signed<br>LEADING SEPARATE | Host variable must be COMP-3<br>(adapted by G8WB) |
| | Host variable does not allow<br>A and P in picture. | |
| NUMERIC | Like DECIMAL, except<br>for physical storage | Like DECIMAL, physical storage<br>always PACKED |
| INTEGER | 35 bit signed | 31 bit signed<br>(like COMP-2) |
| SMALLINT | 17 bit signed | 15 bit signed<br>(like COMP-1) |
| FLOAT | 1-27 single prec.<br>28-63 double prec. | 1-21 single precision<br>22-53 double precision |
| CHAR | 4095 | 254<br>>254 require change to<br>VARCHAR, where 4056 is max. |
| TIME | hh:mm:ss | hh.mm.ss (ISO) |

*Note: Value for time format is different. XDB use hh.mm.ss, whereas INTEREL use hh:mm:ss. This difference is handled automatically during Import/Export of tables, however for host variables used in embedded SQL there is no automated solution.*

*Note: The G8WB 36-BIT-SUPPORT feature cannot be used with INTEREL integer (COMP-6) fields.*

## SQL DCL

CREATE/ALTER/DROP MODEL: Cannot be used in G8WB, where a MODEL is mapped to an XDB LOCATION. Use XDB SQL and Utilities to maintain XDB Location's as required.

CREATE/ALTER TABLE: Following adaption rules must be applied:

VOLUME ... parameter cannot be used (comment out)

IN tsp-name parameter cannot be used (comment out).

CREATE/ALTER/DROP TABLESPACE: Although XDB does support Tablespaces, the implementation is quite different and it should not be used within G8WB. This has limited practical impact, i.e. databases are of limited size anyway, except that the SQL commands and any references to Tablespace must be commented out.

GRANT/REVOKE: Although XDB does support Security, this is not currenlty supported by G8WB, and should not be used. This has limited practical impact, i.e. single-user environment, except that the SQL commands must be commented out.

CREATE/ALTER/DROP SNAPSHOT: Not supported by XDB.

CREATE/ALTER INDEX: The syntax for PART specification of CLUSTER is not compatible between INTEREL and XDB, and you need to adapt SQL command, so you only use the plan CLUSTER specification. This has limited practical impact, i.e. databases are of limited size, except that the SQL commands may need to be changed.

## Embedded SQL

### Preprocessing

XDB will give warning flag for any SQL statement starting before column 12. This has no practical impact, except warnings are more visible in G8WB.

XDB Preprocessor will give warning and error messages on-screen ONLY, so you will not be able to browse XDB error through the F2 COBOL feature of the editor. Instead you should stop checking as at the first XDB error, so Editor is called with proper positioning.

*Note: The line numbers indicated by XDB are not exact because of G8WB preprocessing, i.e. line are deleted/inserted before XDB, which is not properly recognized by XDB.*

Use of underscore (_) in name(s) within SQL data declaration or SQL verbs is not supported by XDB, and G8WB will change underscore to dash (-), so they are properly processed by XDB Preprocessor. It is recommended not to make use of underscore (_) or dollar ($) characters COBOL identifier and SQL host variables, because this is not ANSI standard (BULL specific). Although G8WB will accomodate the use, you cannot use Animator properly because changed names are not recognized by Animator.

### Host variable

The host variable structure is NOT equivalent to a similar Working-Storage declaration, so you cannot move a 01 host variable structure to a 01 working storage structure, because of differences in the USAGE implementation. You MUST use MOVE CORR or multiple elementary field MOVE's instead.

### SQLCA

SQLCA MUST be included through EXEC SQL syntax. INTEREL will allow you to define SQLCA or parts of it in Working Storage, i.e. 01 SQLCODE COMP-6, however this technique is not supported by G8WB.

INTEREL use an extended version of SQLCA and G8WB will automatically include the extended version during preprocessing, however only some of the fields within the extended version are actually used.

SQL-DB-NAME: Used for auto-connect to database (Model).

SQL-DB-OWNER-ID: Ignored, since G8WB does not currently support security.

SQL-ROW-COUNT: Updated for INSERT, UPDATE and DELETE.

SQL-MSG-LENGTH: Updated with length of error message when exceptions occur, i.e. SQLCODE NOT = ZERO, otherwise set to zero.

SQL-ERROR-MESSAGE: Updated with one or more lines, separated by CR-LF, of error text when exceptions occur, otherwise set to spaces.

### Fetch

The INDICATOR field feature is not supported by XDB.

## Select

INDICATOR field feature is not supported by XDB.

INTER|MINUS syntax is not supported by XDB.

## Where

Operators NOT >, < or = are implemented as <=, >= and <> by XDB. Syntax is adapted by G8WB, i.e. NOT > changed to <=.

## Delete

FROM is optional in INTEREL, whereas mandatory in XDB. Syntax is adapted by G8WB if required, i.e. FROM inserted.

## Insert

INTO is optional in INTEREL, whereas mandatory in XDB. Syntax is adapted by G8WB if required, i.e. INTO inserted.

## Connect/Disconnect

CONNECT syntax, which is different for XDB, is adapted by G8WB and any 'Userid' specification is ignored.

DISCONNECT syntax, which is not supported by XDB, is adapted by G8WB, which set a logical disconnect status and connect XDB back to the default Location.

INTEREL's concept "Only one database open at a time" is NOT enforced by G8WB/XDB. XDB will allow two (or more if configured) databases to be opened concurrently. In practice this could mean that an SQL statement referencing a non-current database could execute successfully within G8WB, whereas INTEREL would give an exception.

Connection to database(s) is handled as follows:

IMPLICIT: Before each SQL statement, the SQL-DB-NAME if checked. If this is not = SPACES, and if a previous implicit connect has not been performed, the database (Location) specified is connected. The connection is recorded in SQLCA, thus when located in WS a new connection is established after any EXIT TPR.

EXPLICIT: An explicit CONNECT will close connection with current database and establish a new connection to the given database. An explicit DISCONNECT will close connection with current database.

## Whenever

PERFORM syntax is not supported by XDB, however the syntax is adapted by G8WB.

## SQLCode

SQLCODE exception values are different, except for ZERO and 100. G8WB adaption will attempt to return SQLCODE’s

-2 No database connected

-3 Host variable mismatch

-6 Host variable truncated

like INTEREL, however due to differences between INTEREL and XDB, the conditions that cause these exception are not 100% compatible.

Although other SQLCODE values are different, this has limited practical impact because you would either view the equivalent XDB Error Message via SQL-ERROR-MESSAGE and/or consult XDB Error Message documentation. Since XDB is working in DB2 mode you must use following technique to find proper documentation.

Example: SQLCODE = -603

Use table in Appendix C of "XDB COBOL Precompiler Manual" to convert DB2 code to XDB code, i.e. -603 = -51.

Find the Xnnn error message in chapter 22 of "XDB Error Message Manual", where nnn = 051 (equivalent positive value).

*Note: Do not include hard-coded exception values, except ZERO and 100, into your program, because these values will not be compatible.*

*Note: -1100, which is not documented in XDB, indicate the XDB Server is not accessible.*

CALL 'X7EX$CURRENT_SQL_MESSAGE' is currently not supported.

## Dynamic SQL

Dynamically prepared SQL is currently not supported (qualified).

## Command files

When you download or upload SQL command files, you must take into account that G8WB INTEREL support is NOT 100% compatible, so either way SQL command files may need to be changed. G8WB Provides a Migration utility to automate this task, ref. INTEREL chapter for details.

When working within INFOEDGE/INTEREL-SQL, a number of auxillary SQL command are available. These commands are not supported by G8WB/XDB, however this will only have a practical impact if you include such commands in command files and transfer these to G8WB. The various XDB utilities, i.e. Interactive SQL, provides a range of commands and features with similar functionality, however as with Host command files, you should take care not to include such environment specify commands in any command file transferred to GCOS-8.

Following list the INTEREL-SQL commands, which are not supported:

DEFER
DEFINE PROGRAM
DONE
EDIT
ERASE PROGRAM
END
FORWARD
FORMAT
HELP
HOLD
IGNORE
INPUT
LEFT
LIST
NEW
RECALL
REFRESH
REPORT

PRINT
ROLLBACK PROGRAM
RUN
SAVE
SET
SHIFT
SHOW TABLE
START
STORE
TSS
VOID

### System tables

INTEREL and XDB system tables are substantially different, thus use of system tables is not compatible.

### Security

Security (access control) is currently not supported and you need only work with one userid (authID) within XDB. The default authID has all permission and should be used as is. Any references to userid within embedded SQL will be removed or ignored by G8WB. Use of security with INTEREL SQL (script/command files) should be commented out.

*Note: Since XDB is used only in a single-user environment the use of security has little meaning anyway.*

## Terminal emulation

Underline attribute is currently treated as INVERSE and inverse attribute as INVERSE + RED.

The cursor is not allowed to move outside variable fields when in forms-mode. Also any cursor movement outside of a variable field is treated as a TAB to next field always. Cursor movement should be “free” as in normal-mode.

### VIP78XX

Concealed attribute cannot be used in combination with other visual attributes (Concealed is not enforced then).

## TPFF

Only VIP78XX mode supported.

## FORMAT

Field sequencing is not supported.

Only VIP77XX-mode supported.

## Pilspråk

Only IBM327X-mode supported.

# Micro Focus Workbench

## Structure animation

Structure Animation does not reflect any source line, which have been modified by G8WB Preprocessor. Since all CALL statements are modified, structure animator will not show this, even when you request CALL documentation. G8WB preprocessor will also change any source line with multiple statements, i.e. IF A > 1 GO TO LABEL. In such situations, structure animator will not show the GO TO.

## CSI

CSI is, unfortunately, not really useable with G8WB, since any statement modified by G8WB Preprocessor will not be considered by CSI. Since almost all data definitions are modfied during G8WB preprocessing, CSI information could be actually be misleading, i.e. dead data.

# G8WFL

G8WFL cannot handle 01 structures with more than 200 fields (it ignores the rest). As such you may not be able to build a proper mask for file conversion. To work around this limitation you should copy the program and change the File/Working storage structure(s) into multiple 01's, check the changed program and use the generated BSF for G8WFL. Be careful not to override such BSF file when you check the original program, as this is used at run-time to support file conversion.

RELATIVE Variable length files are not supported.

Conversion problem(s) at field level is documented with a general occurence count (for each record type). This indicate how many fields that had conversion problems. Each is occurence is documented on a SYSOUT file (in current directory). Each line specify the RECORD name, the relative record number (within the type), the FIELD name and the cause. Possible causes are:

- COMP-11/COMP-12 or INDEX fields are ignored (not converted)

- Overpunch sign invalid - forced to +

- Non-numeric DISPLAY field - forced to 0

- COMP-X sign invalid - forced to +

- Non-numeric COMP-X field - forced to 0

- > 31 significant bits - truncated (COMP-6 or DB-KEY)

# Special features

Following describe special GCOS-8 features available with G8WB.

## RCOPY

The ALFTRAN syntax

**RCOPY name (R in column 7)**

is treated by G8WB Preprocessor as

```
COPY name OF AL
```

Extract of COPY's from ALFTRAN should thus be imported in G8WB under the AL Library-tag.

## TDS-Storage

All references to TDS-STORAGE are changed to TP-STORAGE.

## Soundex

A G8WB version of the SOUNDEX module from BULL Norway is available. Please refer to Technical Guide for details.

## DBMS Query

A DBMS Query routine (not compatible with GCOS-8) may be used to determine DBMS allocation mode within G8WB.

```
CALL "W8DBALCQ" USING schema-name
                      allocation-mode

01  schema-name       pic x(30). *> Only First 8 chars used

01  allocation-mode    comp-6.
    88  schema-not-found value -1.
    88  dbms-read-mode   value 0.
    88  dbms-write-mode  value 1.
```

## Abort

A sample module for migration of GCOS-8 abort routines is provided. Please refer to $G8WBGDIR\SITE\BATCH for details.

## TPLOAD

G8WB support NOVIT TPLOAD, TPRET and TPGOTO routines. The implementation is compatible with GCOS-8, except for following restriction:

CALL "TPLOAD" must be the last verb and terminated by a period, if within a conditional statement. Ex.

```
IF something
....
CALL "TPLOAD".
```

Number of TPLOAD's within one TPR is limited to 10.

TPLOAD, TPRET and TPGOTO calls are documented on the journal.

Following abort codes may occur:

UU1 - TPRET without anything in TPLOAD stack

UU4 - TPLOAD stack overflow (max. 5 levels)

UU5 - TPGOTO return point not found

G8WB provides a migrated version of the NOVIT OBSUTX, OBSINN, OBSPRI, OBSPRO and OBSOPP subroutines (migrated from SUB005). Please refer to $G8WBGDIR\SITE\NOVIT for details.

## INSTALL/1

G8WB support INSTALL1/PC. Please refer to Technical Guide for details.

# Chapter 12 ERROR MESSAGES

## IDS-II Schema Translate

### Information

| | |
|---|---|
| I-0001 | INCREMENT applied |

### Warnings

| | |
|---|---|
| W-0001 | PRIVACY feature not enforced within G8WB |
| W-0002 | TEMPORARY type not supported |
| W-0003 | AREA x.x has no RECORD's |
| W-0004 | SCHEMA (DMCL) not defined |
| W-0005 | AREA (DMCL) not declared |
| W-0006 | PRIVACY feature not enforced within G8WB |
| W-0007 | KEY FILE_CODE not used |
| W-0008 | RANGE feature not applied within G8WB |
| W-0009 | RANGE PROCEDURE not used |
| W-0010 | PROCEDURE not referenced |
| W-0011 | ASSEMBLER not supported |
| W-0012 | GMAP not supported |
| W-0013 | PL1 not supported |
| W-0014 | FORTRAN not supported |
| W-0015 | FORT77 not supported |
| W-0016 | 'Entry' termination not found |
| W-0017 | invalid COMMENT |
| W-0018 | (-) is illegal character |

### Errors

| | |
|---|---|
| E-0001 | SCHEMA entry duplicated |
| E-0002 | AREA entry duplicated |
| E-0003 | RECORD entry duplicated |
| E-0004 | Duplicate LOCATION clause |
| E-0005 | LOCATION mode invalid error-text |

| | |
|---|---|
| **E-0006** | **SET (VIA) x.x not defined** |
| **E-0007** | **PROCEDURE (DMCL) x.x not declared** |
| **E-0008** | **KEY field x.x not defined in RECORD** |
| **E-0009** | **LOCATION key x.x not defined in RECORD** |
| **E-0010** | **DIRECT key x.x must have type DB_KEY** |
| **E-0011** | **CALC field x.x not defined in RECORD** |
| **E-0012** | **No fields giving** |
| **E-0013** | **DUPLICATE clause missing** |
| **E-0014** | **AREA x.x ORGANIZATION conflict** |
| **E-0015** | **AREA x.x ORGANIZATION conflict** |
| **E-0016** | **Duplicate WITHIN clause** |
| **E-0017** | **AREA not defined** |
| **E-0018** | **AREA not defined** |
| **E-0019** | **PARAMETER x.x not defined as AREA_ID** |
| **E-0020** | **PROCEDURE use conflict** |
| **E-0021** | **KEY entry duplicated** |
| **E-0022** | **No fields giving** |
| **E-0023** | **DUPLICATE clause missing** |
| **E-0024** | **Fields must have same level** |
| **E-0025** | **FIELD entry duplicated** |
| **E-0026** | **TYPE clause missing** |
| **E-0027** | **BINARY integer invalid** |
| **E-0028** | **FLOAT BINARY not supported** |
| **E-0029** | **BINARY integer invalid** |
| **E-0030** | **INTEGER > 18 not supported** |
| **E-0031** | **INTEGER invalid (1-4095)** |
| **E-0032** | **TYPE invalid** |
| **E-0033** | **PROCEDURE use conflict** |
| **E-0034** | **SET entry duplicated** |
| **E-0035** | **PRIOR clause missing** |
| **E-0036** | **OWNER clause missing** |

| | |
|---|---|
| **E-0037** | **ORDER clause missing** |
| **E-0038** | **Duplicate OWNER clause** |
| **E-0039** | **RECORD not defined** |
| **E-0040** | **Duplicate PRIOR clause** |
| **E-0041** | **Duplicate ORDER clause** |
| **E-0042** | **DUPLICATES invalid** |
| **E-0043** | **ORDER invalid** |
| **E-0044** | **MEMBER entry duplicated** |
| **E-0045** | **INSERTION clause missing** |
| **E-0046** | **RETENTION clause missing** |
| **E-0047** | **OWNER clause missing** |
| **E-0048** | **Duplicate INSERTION clause** |
| **E-0049** | **INSERTION invalid** |
| **E-0050** | **Duplicate RETENTION clause** |
| **E-0051** | **RETENTION invalid** |
| **0052** | **Duplicate OWNER clause** |
| **E-0053** | **Duplicate DUPLICATES clause** |
| **E-0054** | **FIELD not defined in RECORD** |
| **E-0055** | **Duplicate KEY clause** |
| **E-0056** | **FIELD not defined in RECORD** |
| **E-0057** | **DUPLICATE invalid** |
| **E-0058** | **Duplicate SELECTION clause** |
| **E-0059** | **SET not current** |
| **E-0060** | **SET not defined** |
| **E-0061** | **PARAMETER must be type DB_KEY** |
| **E-0062** | **Not MEMBER of current SET** |
| **E-0063** | **IDENTIFIED BY invalid** |
| **E-0064** | **THRU illegal (nothing lower)** |
| **E-0065** | **SET not defined** |
| **E-0066** | **FIELD x.x TYPE mismatch** |
| **E-0067** | **PARAMETER x.x TYPE mismatch** |

| | |
|---|---|
| E-0068 | FIELD not defined in OWNER |
| E-0069 | RECORD not LOCATION DIRECT |
| E-0070 | RECORD not LOCATION CALC |
| E-0071 | No of EQUAL fields < CALC fields |
| E-0072 | FIELD x.x TYPE mismatch |
| E-0073 | PARAMETER x.x TYPE mismatch |
| E-0074 | No of EQUAL fields > CALC fields |
| E-0075 | SCHEMA (DDL) not defined |
| E-0076 | SCHEMA (DDL) not current |
| E-0077 | FILE_CODE invalid |
| E-0078 | ALLOCATE invalid |
| E-0079 | RESERVE invalid |
| E-0080 | PAGE_INTERVAL invalid |
| E-0081 | CALC_INTERVAL invalid |
| E-0082 | PAGE_SIZE invalid |
| E-0083 | INVENTORY invalid |
| E-0084 | LOAD_LIMIT invalid |
| E-0085 | ORGANIZATION conflict |
| E-0086 | ORGANIZATION conflict |
| E-0087 | ORGANIZATION invalid |
| E-0088 | AREA not defined |
| E-0089 | AREA DBK's > 34,359,738,367 |
| E-0090 | INDEXED FILE_CODE's > 8 |
| E-0091 | INTEGRATED FILE_CODE's > 1 |
| E-0092 | KEY FILE_CODE missing |
| E-0093 | TYPE invalid |
| E-0094 | RECORD not defined |
| E-0095 | FIELD must be type BINARY 35 |
| E-0096 | PARAMETER must be type BINARY 35 |
| E-0097 | FIELD must be type BINARY 35 |
| E-0098 | PARAMETER must be type BINARY 35 |

| | |
|---|---|
| **E-0099** | **PROCEDURE use conflict** |
| **E-0100** | **RANGE invalid** |
| **E-0101** | **SET not defined** |
| **E-0102** | **POINTERS invalid** |
| **E-0103** | **SIZE invalid** |
| **E-0104** | **MODE invalid** |
| **E-0105** | **PA_TYPE invalid** |
| **E-0106** | **ALLOW invalid** |
| **E-0107** | **KEY not defined** |
| **E-0108** | **TYPE invalid** |
| **E-0109** | **Duplicate USING clause** |
| **E-0110** | **Invalid AREA argument** |
| **E-0111** | **Invalid CHECK argument** |
| **E-0112** | **Invalid USING argument** |
| **E-0113** | **USING without argument(s)** |
| **E-0114** | **Duplicate GIVING clause** |
| **E-0115** | **Invalid GIVING argument** |
| **E-0116** | **FIELD/PARAMETER not defined** |
| **E-0117** | **FIELD must qualified** |
| **E-0118** | **FIELD not defined** |
| **E-0119** | **Expecting 'Entry' keyword** |
| **E-0120** | **'Entry' termination not found** |
| **E-0121** | **Expecting '"** |
| **E-0122** | **Unexpected keyword/syntax found** |
| **E-0123** | **SCALE is invalid** |
| **E-0124** | **SORTED invalid** |
| **E-0125** | **FILE_CODE missing** |

## Fatal Errors

| | |
|---|---|
| **F-0001** | **Cannot access Schema file <file>** |
| **F-0002** | **Schema file specification illegal <file>** |

| | |
|---|---|
| F-0003 | Schema <name> configuration not valid<br>Change of Schema configuration required |

# IDS-II Schema Validation

## Warnings

| | |
|---|---|
| W-0001 | This schema exceed current G8WB limitations<br>Record size <nnn> > <max> (current limit)<br>Calc Key size <nnn> > <max> (current limit)<br>Key size <nnn> > <max> (current limit)<br>You may continue using schema, however you<br>cannot create any database file(s)<br>Please contact Technical Support" |
| W-0004 | Subschema configuration failed |
| W-0005 | Subschema translation failed |
| W-0006 | Subschema validation failed |
| W-0007 | Unload/load program generation failed |

## Fatal Errors

| | |
|---|---|
| F-0001 | Schema <name> not configured<br>Prior translate is required |
| F-0002 | Schema must be translated first |
| F-0003 | The existing schema does not match with the translated schema<br><name> Validation has failed<br>Probable cause: Schema number has been changed |
| F-0008 | Failed to allocate <nnn> bytes for Schema Control Table |
| F-0009 | Failed to allocate <nnn> bytes for Schema Field Segment(a) |
| F-0010 | Failed to allocate <nnn> bytes for Schema Field Segment(b) |
| F-0011 | Failed to allocate <nnn> bytes for Schema Field Segment(c) |
| F-0012 | Failed to allocate <nnn> bytes for Schema Field Name Segment(a) |
| F-0013 | Failed to allocate <nnn> bytes for Schema Field Name Segment(b) |
| F-0014 | Failed to allocate <nnn> bytes for Schema Extension Segment |
| F-0015 | Failed to allocate <nnn> bytes for Schema Other Segment |
| F-0016 | Failed to allocate <nnn> bytes for Schema Other Name Segment |
| F-0017 | BSF-I/O-Status => 9/nnn on <file> |
| F-0018 | Name Segment overflow<br>Schema is too big, validation failed |

| | |
|---|---|
| **F-0019** | **Field Segment overflow**<br>**Schema is too big, validation failed** |
| **F-0020** | **Name Other Segment overflow**<br>**Schema is too big, validation failed** |
| **F-0021** | **Pointer Other Segment overflow**<br>**Schema is too big, validation failed** |
| **F-0022** | **Extension Segment overflow**<br>**Schema is too big, validation failed** |

# IDS-II Subschema Translate

## Information

| | |
|---|---|
| **I-0001** | **Transformation: x.x Possible truncation** |
| **I-0002** | **INCREMENT applied** |
| **I-0003** | **changed to NUMERIC DISPLAY** |
| **I-0004** | **32 bit unsigned format applied** |
| **I-0005** | **35 bit signed format applied** |
| **I-0006** | **35 bit signed format applied** |

## Warnings

| | |
|---|---|
| **W-0001** | **END missing** |
| **W-0002** | **PRIVACY feature not enforced within G8WB** |
| **W-0003** | **WITHIN feature ignored within G8WB** |
| **W-0004** | **ALIAS active treated as SS field** |
| **W-0005** | **Transformation: x.x COMP use not supported by G8WB** |
| **W-0006** | **Transformation: x.x Content must be numeric** |
| **W-0007** | **Transformation: x.x Possible loss of sign** |
| **W-0008** | **Transformation: x.x Possible loss of fraction** |
| **W-0009** | **Transformation: x.x Possible truncation** |
| **W-0010** | **Transformation: x.x Possible truncation** |
| **W-0011** | **'Entry' termination not found** |
| **W-0012** | **(_) is illegal character** |
| **W-0013** | **BSF table truncated - exceed max fields, Record <name>** |

## Errors

| | |
|---|---|
| E-0001 | REALM SECTION missing |
| E-0002 | RECORD SECTION missing |
| E-0003 | Subschema configuration not valid |
| E-0004 | Schema not found .. |
| E-0005 | PRIVACY type invalid |
| E-0006 | Subschema not defined |
| E-0007 | MAPPING DIVISION misplaced |
| E-0008 | REALM not defined |
| E-0009 | SET not defined |
| E-0010 | RECORD not defined |
| E-0011 | KEY not defined |
| E-0012 | FIELD not unique |
| E-0013 | FIELD, REALM x.x not defined |
| E-0014 | AD clause invalid |
| E-0015 | Subschema not defined |
| E-0016 | MAPPING DIVISION missing |
| E-0017 | STRUCTURE DIVISION misplaced |
| E-0018 | REALM SECTION misplaced |
| E-0019 | REALM not defined |
| E-0020 | REALM SECTION missing |
| E-0021 | SET SECTION misplaced |
| E-0022 | SET not defined |
| E-0023 | REALM SECTION missing |
| E-0024 | KEY SECTION misplaced |
| E-0025 | KEY not defined |
| E-0026 | REALM SECTION missing |
| E-0027 | RECORD SECTION misplaced |
| E-0028 | RECORD not defined |
| E-0029 | REALM not included |
| E-0030 | REALM not defined |

| | |
|---|---|
| **E-0031** | **Level invalid** |
| **E-0032** | **Clause duplicated** |
| **E-0033** | **USAGE invalid** |
| **E-0034** | **Clause duplicated** |
| **E-0035** | **Clause duplicated** |
| **E-0036** | **SIGN invalid** |
| **E-0037** | **Previous field missing PICTURE** |
| **E-0038** | **Transformation: x.x, illegal** |
| **E-0039** | **USAGE clause mismatch GROUP USAGE** |
| **E-0040** | **SIGN clause mismatch PIC or USAGE** |
| **E-0041** | **USAGE clause mismatch PIC** |
| **E-0042** | **Numeric > 18 digits** |
| **E-0043** | **Numeric > 18 digits** |
| **E-0044** | **SIGN clause without PICTURE** |
| **E-0045** | **Clause duplicated** |
| **E-0046** | **S must be first** |
| **E-0047** | **X when already numeric** |
| **E-0048** | **9 when already alphanumeric** |
| **E-0049** | **V or P already present** |
| **E-0050** | **V when already alphanumeric** |
| **E-0051** | **Invalid character** |
| **E-0052** | **V or P already present** |
| **E-0053** | **P when already alphanumeric** |
| **E-0054** | **() without context** |
| **E-0055** | **() not numeric** |
| **E-0056** | **) missing** |
| **E-0057** | **Expecting 'Entry' keyword** |
| **E-0058** | **'Entry' termination not found** |
| **E-0059** | **Expecting x...x** |
| **E-0060** | **Unexpected keyword/syntax found** |

### Fatal Errors

| | |
|---|---|
| F-0001 | Cannot access subschema file <file> |
| F-0002 | Subschema file specification Illegal |
| F-0003 | Schema must be validated first |
| F-0004 | Failed to allocate <nnn> bytesfor 01-ITEMS Table |
| F-0005 | BSF-I/O-Status => 9/nnn on <file> |

## IDS-II Subschema Validation

### Warnings

| | |
|---|---|
| W-0001 | Area <name> has been deleted obsolete |
| W-0002 | Set  <name> has been deleted obsolete |
| W-0003 | Key  <name> has been deleted obsolete |
| W-0004 | Record <name> has been deleted obsolete |
| W-0005 | Field <name> has been deleted obsolete |
| W-0006 | Record <name> has FIELDS restriction |
| W-0007 | Record <name> has SETS   restriction |
| W-0008 | Record <name> has MEMBERS restriction |
| W-0009 | Record <name> has KEYS   restriction |

### Fatal Errors

| | |
|---|---|
| F-0001 | Subschema <name> not configured |
| F-0002 | Subschema must be translated first |
| F-0003 | Schema not found or configuration not valid |
| F-0004 | Schema must be validated first |
| F-0005 | Failed to allocate <nnn> bytes for Subschema Control Table |
| F-0006 | Failed to allocate <nnn> bytes for subschema Map Segment |
| F-0007 | Failed to allocate <nnn> bytes for subschema Offset Segment |
| F-0008 | Failed to allocate <nnn> bytes for subschema Other Segment |
| F-0009 | Failed to allocate <nnn> bytes for subschema Field Segment |
| F-0010 | Failed to allocate <nnn> bytes for subschema Name Segment |
| F-0011 | Map Segment overflow<br>Subschema is too big, validation failed |

| | |
|---|---|
| **F-0012** | **Offset Segment overflow<br>Subschema is too big, validation failed** |
| **F-0013** | **BSF-I/O-Status => 9/nnn on <file>** |
| **F-0014** | **Name Segment overflow<br>Subschema is too big, validation failed** |
| **F-0015** | **Field segment overflow<br>Subschema is too big, validation failed** |
| **F-0016** | **Name Segment overflow<br>Subschema is too big, validation failed** |

# G8WB Preprocessor

## Warnings

| | |
|---|---|
| **w-8501** | **Compatibility problem, check results** |
| **w-8502** | **More then one COBOL-statement on a line** |
| **w-8503** | **Period missing at the end of a section** |
| **w-8504** | **Period missing at the end of a level** |
| **w-8505** | **Missing equal (=) sign in a condition** |
| **w-8506** | **Wrong attributes for a COMP-4 picture (SIG** |
| **w-8507** | **Wrong value for an index (SET INDEX TO 0)** |
| **w-8508** | **READY/FINISH use limited to TP8 READY TPR'** |
| **w-8509** | **Field in error x.x** |
| **w-8510** | **BSF Create error x.x** |
| **w-8511** | **'PROGRAM-ID. x.x .' not equal source-name** |
| **w-8512** | **Default clause ignored"** |
| **w-8513** | **Replace clause ignored** |
| **w-8514** | **CALL to x.x has 'x.x' preloaded** |
| **w-8515** | **CALL to x.x equated to 'x.x'** |
| **w-8516** | **END PROGRAM not equal to program-id** |

## Errors

| | |
|---|---|
| **e-8001** | **Flagged by W8PREP: <reason>** |
| **e-8002** | **Sub-schema x.x missing, translate required** |
| **e-8003** | **Search for copy x.x OF x.x failed** |

| | |
|---|---|
| **e-8004** | **CD & 01 not known** |
| **e-8005** | **IDS-Reserved word missing or incorrectly used** |
| **e-8006** | **IDS-Verb without Subschema** |
| **e-8007** | **IDS x.x-name expected** |
| **e-8008** | **IDS-statement not complete** |
| **e-8009** | **CD format wrong or not complete** |
| **e-8010** | **CD syntax error at x.x** |
| **e-8011** | **Reserved word missing or incorrectly used** |
| **e-8012** | **==text== expected** |
| **e-8013** | **COPY-name expected** |
| **e-8014** | **COPY statement not complete** |
| **e-8015** | **COPY file not found** |
| **e-8016** | **Nested COPY not allowed** |
| **e-8017** | **Literal-continuation error** |
| **e-8018** | **Ascii-literal error** |
| **e-8019** | **TX-Storage alignment not possible** |
| **e-8020** | **LINK to x.x in x.x Not current Subschema** |
| **e-8021** | **LINK to x.x in x.x Record not found** |
| **e-8022** | **LINK to x.x in x.x Subschema not found** |
| **e-8023** | **LINK to x.x in x.x Record not found** |
| **e-8024** | **Redefine x.x <> x.x redefine > object** |
| **e-8025** | **Redefine x.x <> x.x not compatible** |
| **e-8026** | **Redefine x.x <> x.x not compound compatible** |
| **e-8027** | **Redefine x.x <> x.x not compound compatible** |
| **e-8028** | **Redefine x.x alignment/compatibility error** |
| **e-8029** | **Redefine x.x alignment/compatibility error** |
| **e-8030** | **Fields x.x <> x.x not compatible** |
| **e-8031** | **Fields x.x <> x.x not compound compatible** |
| **e-8032** | **Fields x.x <> x.x not compound compatible** |
| **e-8033** | **Field x.x subschema need adjust** |
| **e-8034** | **Field x.x subschema need adjust** |

| | |
|---|---|
| **e-8035** | **Redefine x.x > x.x Subschema < Object** |
| **e-8036** | **Redefine x.x > x.x align/comp error** |
| **e-8037** | **Redefine x.x > x.x alignment change error** |
| **e-8038** | **Redefine x.x > x.x Subschema < Object** |
| **e-8039** | **Redefine x.x > x.x align/comp error** |
| **e-8040** | **Redefine x.x > x.x alignment change error** |
| **e-8041** | **__ illegal naming** |
| **e-8042** | **VMAPI call must be UPPER_CASE** |
| **e-8043** | **VMAPI Service illegal COBOL-74** |
| **e-8044** | **VMPS Service illegal COBOL-74** |
| **e-8045** | **Record not associated with Key** |
| **e-8046** | **Duplicates are not allowed for Record** |
| **e-8047** | **Record location not Calc** |
| **e-8048** | **Record not member of Set** |
| **e-8049** | **Record not located in Realm** |
| **e-8050** | **Store/Modify, Field(s) restricted** |
| **e-8051** | **Store, Set ownership(s) restricted** |
| **e-8052** | **Store/Modify, Key(s) restricted** |
| **e-8053** | **Store/Modify, Set membership(s) restricted** |
| **e-8054** | **Store, Realm(s) restricted** |
| **e-8055** | **Field incorrectly qualified** |
| **e-8056** | **Field ambigious** |
| **e-8057** | **END-EXEC missing** |

## Fatal Errors

| | |
|---|---|
| **err-001** | **Cannot load Preprocessor x.x** |
| **err-002** | **ENTRIES (Maximum) must be increased** |
| **err-003** | **Cannot access x.x** |
| **err-004** | **NAMES (Maximum) must be increased** |
| **err-005** | **01-IDENTIFIER (Maximum) must be increased** |
| **err-006** | **01-ITEMS (Maximum) must be increased** |

| | |
|---|---|
| err-007 | REPLACING (Maximum) must be increased |
| err-008 | CALL-NAMES (Maximum) must be increased |
| err-009 | IDS-NAMES (Maximum) must be increased |
| err-010 | IDS-ARGS (Maximum) must be increased |
| err-011 | IDS-WORDS (Maximum) must be increased |
| err-012 | IDS-USE (Maximum) must be increased |
| err-013 | IDS-USE-STATUS (Maximum) must be increased |
| err-014 | IIDS-Verb Table full |
| err-015 | Heap x.x failed - status 9/nnn |
| err-016 | Failed to allocate nnn bytes for NAMES Table |
| err-017 | Failed to allocate nnn bytes for ENTRIES Table |
| err-018 | Failed to allocate nnn bytes for CALL-NAME Table |
| err-019 | Failed to allocate nnn bytes for IDS-USE-STATUS Table |
| err-020 | Failed to allocate nnn bytes for IDS-WORDS Table |
| err-021 | Failed to allocate nnn bytes for IDS-ARGS Table |
| err-022 | Failed to allocate nnn bytes for IDS-ARGS Table |
| err-023 | Failed to allocate nnn bytes for IDS-NAMES Table |
| err-024 | Failed to allocate nnn bytes for REPLACING Tbale |
| err-025 | Failed to allocate nnn bytes for 01-IDENTIFIER Table |
| err-026 | Failed to allocate nnn bytes for 01-ITEMS Table |
| err-027 | Failed to allocate nnn bytes for 01-ITEMS Table |
| err-028 | Failed to allocate nnn bytes for 01-ITEMS Table |
| err-029 | Failed to allocate nnn bytes for 01-ITEMS Table |
| err-030 | Failed to allocate nnn bytes for IDS-USE Table |
| err-031 | IOMS-Status => 9/nnn on <source-file> |
| err-032 | IOMS-Status =>  9/nnn on <table-file> |
| err-033 | IOMS-Status =>  9/nnn on <copy-file> |
| err-034 | IOMS-Status =>  9/nnn on <work-file> |
| err-035 | BSF-I/O-Status => 9/nnn on <file> |
| err-036 | IOMS-Status =>  9/nnn on <cmp-fille> |
| err-037 | Failed to allocate nnn bytes for BLOCK-UNIFY |

# INDEX