

**GCOS-8 COBOL Workbench™**

**Technical Guide**

Version 1.2.00/xx

Revision: October 1994

© Copyright 1994 B & C Solution SARL, France

All rights reserved.



## TECHNICAL 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

GCOS-8 COBOL Workbench is a trademark of B & C Solution SARL. All other trademarks and trade names belong to their respective companies.

® Copyright 1994 B & C Solution S.A.R.L. All Rights Reserved.



# Table of Contents

INTRODUCTION.....	1
Preface .....	1
Intended Audience.....	1
Related Publications.....	1
Operation Environment & Requirements .....	1
Hardware.....	1
Software.....	1
SOFTWARE INTEGRATION .....	3
User Developed Software.....	3
TP .....	3
DMIV-TP Procedures.....	3
TP8 Procedures.....	3
Abort TPR .....	3
Terminal Emulation .....	3
Forms Software .....	3
Toolkit .....	3
FORMAT .....	3
DIMS8 .....	3
Print Software.....	3
Toolkit .....	3
DISPATCH8 .....	4
Preprocessor .....	4
Toolkit .....	4
\$\$SELECT .....	4
Other Software .....	4
Soundex.....	4
Software Maintenance .....	4
Development.....	4
Configure .....	5
Test.....	5
Compile.....	5
Library update .....	5
EXECUTIVE TABLES .....	7
Executive Storage.....	7
G8WB Defaults .....	8
TP Monitor Configuration.....	9
DMIV-TP Configuration .....	10
TP8 Configuration .....	10
Logical-Id Table.....	10
Message-Id/Command Table.....	11
Constant Storage Table.....	11
Implicit Communication Table .....	12
Profile Table.....	12
Message Buffer .....	13
DMIV-TP PROCEDURES .....	15
DMIV-TP User Derail.....	15
External Files .....	15
Data Definition.....	15
Executive Tables.....	16
Interfacing Arguments .....	16
Module Initialize .....	17
Procedure Process .....	17
Trace & Abort service.....	18
DMIV-TP User Procedure.....	19
External Files .....	19
Data Definition.....	20
Executive Tables.....	20
Executive Interfacing.....	21
Module Initialize .....	22
Receive Procedure .....	22
Send Procedure.....	24
Procedure Selection .....	24
TX Init/Term Procedure .....	25

Phase Init/Term Procedure.....	25
Message-Id Procedure .....	26
Trace service .....	27
TP8 PROCEDURES .....	29
TP8 Program Procedures.....	29
External Files .....	29
Data Definition .....	29
Executive Tables.....	30
Interfacing Arguments .....	30
Module Initialize .....	31
Procedure Process.....	31
Trace & Abort service.....	32
TP8 Administrative Procedures .....	33
External Files .....	33
Data Definition .....	34
Executive Tables.....	34
Executive Interfacing.....	35
Module Initialize .....	36
Receive Procedure.....	36
Send Procedure .....	38
Procedure Selection .....	38
TX Init/Term Procedure.....	39
Phase Init/Term Procedure.....	40
Message-Id Procedure .....	40
Trace service .....	41
ABORT TPR .....	42
Terminal Emulation .....	42
FORMS Software.....	43
Toolkit.....	43
Concepts.....	43
Module Initialize .....	43
Service Process .....	45
Send & Receive .....	46
Trace Service.....	47
FORMAT.....	48
Concepts.....	48
Module Initialize .....	48
Editing User Exit .....	49
Validation User Exit.....	50
DIMS8.....	51
PRINT Software.....	53
Toolkit.....	53
Concepts.....	53
Application Interface.....	53
Print Collection.....	53
Print Utility .....	53
Module Initialize .....	53
Service Process .....	55
Trace Service.....	55
DISPATCH8.....	55
Preprocessor.....	57
Toolkit.....	57
Concepts.....	57
Enable SITE Preprocessor .....	57
Source Expansion .....	57
Source Translation .....	57
SITEPREP .....	58
Prepare Phase 1 .....	61
Directives.....	61
Process Phase 1 .....	62
Phase 2.....	63
\$\$Select.....	64
Other Software.....	65
Soundex.....	65
Concepts.....	65

SOUNDX Program .....65  
SOUNDX Entry-point.....67



# INDEX

## \$

\$\$SELECT, 4  
\$\$Select, 64

## A

ABORT TPR, 42  
ABort TPR, 3  
Application Interface, 53

## C

Compile, 5  
Concepts, 43; 48; 53; 57; 65  
Configure, 5  
Constant Storage Table, 11

## D

Data Definition, 15; 20; 29; 34  
Development, 4  
DIMS8, 3; 51  
Directives, 61  
DISPATCH8, 4; 55  
DMIV-TP Configuration, 10  
DMIV-TP PROCEDURES, 15  
DMIV-TP Procedures, 3  
DMIV-TP User Derail, 15  
DMIV-TP User Procedure, 19

## E

Editing User Exit, 49  
Enable SITE Preprocessor, 57  
Executive Interfacing, 21; 35  
Executive Storage, 7  
EXECUTIVE TABLES, 7  
Executive Tables, 16; 20; 30; 34  
External Files, 15; 19; 29; 33

## F

FORMAT, 3; 48  
FORMS Software, 43  
Forms Software, 3

## G

G8WB Defaults, 8

## H

Hardware, 1

## I

Implicit Communication Table, 12  
Intended Audience, 1  
Interfacing Arguments, 16; 30  
INTRODUCTION, 1

## L

Library update, 5  
Logical-Id Table, 10

## M

Message Buffer, 13  
Message-Id Procedure, 26; 40  
Message-Id/Command Table, 11  
Module Initialize, 17; 22; 31; 36; 43;  
48; 53

## O

Operation Environment &  
Requirements, 1  
Other Software, 4; 65

## P

Phase 2, 63  
Phase Init/Term Procedure, 25; 40  
Preface, 1  
Prepare Phase 1, 61  
Preprocessor, 4; 57  
Print Collection, 53  
PRINT Software, 53  
Print Software, 3  
Print Utility, 53  
Procedure Process, 17; 31  
Procedure Selection, 24; 38  
Process Phase 1, 62  
Profile Table, 12

## R

Receive Procedure, 22; 36  
Related Publications, 1

## S

Send & Receive, 46  
Send Procedure, 24; 38  
Service Process, 45; 55  
SITEPREP, 58  
Software, 1  
SOFTWARE INTEGRATION, 3  
Software Maintenance, 4  
Soundex, 4; 65  
SOUNDX Entry-Point, 67  
SOUNDX Program, 65  
Source Expansion, 57  
Source Translation, 57

## T

Terminal Emulation, 3; 42  
Test, 5  
Toolkit, 3; 4; 43; 53; 57  
TP, 3  
TP Monitor Configuration, 9  
TP8 Administrative Procedures, 33  
TP8 Configuration, 10  
TP8 PROCEDURES, 29  
TP8 Procedures, 3  
TP8 Program Procedures, 29  
Trace & Abort service, 18; 32  
Trace Service, 47; 55  
Trace service, 27; 41  
TX Init/Term Procedure, 25; 39

## U

User Developed Software, 3

**V**

Validation User Exit, 50

# 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 intended for use by system programmers or TP/DB administrators knowledgeable regarding the design and coding of DMIV-TP or TP8 User Procedures, and BULL software interfacing in general.

---

## 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 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 and Micro Focus Limited.

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

---

## Operation Environment & Requirements

---

### Hardware

G8WB was designed within the concept and environment of Micro Focus Workbench and will operate on any micro computers supported by Micro Focus Workbench. In general Micro Focus Workbench 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 Micro Focus Workbench documentation for details.

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

---

### Software

G8WB requires Micro Focus Workbench 3.2 (minimum 3.2.31), which in turn requires OS/2 1.2 or DOS 3.1 or greater. Please refer to Micro Focus Workbench 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.

# SOFTWARE INTEGRATION

---

## User Developed Software

---

G8WB has been designed as an open environment, where user developed software components may easily be integrated into the workbench.

---

### TP

---

#### DMIV-TP Procedures

A toolkit for integration of DMIV-TP User Derail and User Procedures is readily delivered. This toolkit includes samples to support porting of any existing procedure. The Message-ID procedure provided in the toolkit, is used as standard in G8WB DMIV-TP Emulation.

---

#### TP8 Procedures

A toolkit for integration of TP8 User Program and Administrative procedures is readily delivered. This toolkit includes samples to support porting of any existing procedure. The Command procedure provided in the toolkit, is used as standard in G8WB TP8 Emulation.

---

#### Abort TPR

The system ABORT TPR (TP-ABT) is delivered in a simple version. This may be customized as required.

---

#### Terminal Emulation

A toolkit for integration of Terminal Emulator software is readily delivered. This toolkit may be used to implement a customized front-end for TP. An example could be an advanced Transaction-Driver, which emulate Two-level transaction processing.

---

## Forms Software

---

### Toolkit

A toolkit for integration of Forms software is readily delivered. This has been used for integration of TPF Forms emulation, which use callable services to initiate Forms send and receive operations.

---

### FORMAT

A toolkit for integration of FORMAT User-exit functionality is readily delivered.

---

### DIMS8

A toolkit for integration of DIMS8 Site configurable software components is readily delivered.

---

## Print Software

---

### Toolkit

A toolkit for integration of Print software is readily delivered. This has been used for integration of the DISPATCH8 Distributed Print System, which operate with a separate Server Program outside of the TP environment.

---

## DISPATCH8

A toolkit for integration of DISPATCH8 Site configurable software components is readily delivered.

---

## Preprocessor

---

### Toolkit

A toolkit for integration of Preprocessor software is readily delivered. G8WB fully support stackable preprocessors with flexible configuration setting possibilities.

---

## \$\$SELECT

The standard G8WB \$\$SELECT preprocessor is delivered in source and may be used as a guide for implementation of your own SITE preprocessor.

---

## Other Software

Although other user developed software components area not readily supported by G8WB, B & C Solution is very open towards this and will , upon request, provide you the toolkit, hooks and handles required, for you to integrate such software components.

---

## Soundex

A toolkit for integration of BULL Norway SOUNDEX software is readily delivered.

---

## Software Maintenance

G8WB utilizes the Library concept of the Micro Focus Workbench to support the run-time environment, and user developed software components are simply inserted into these libraries at the completion of the software migration activity. The G8WB configuration feature may be used to identify the user specific programs containing the migrated software.

---

## Development

The standard toolkits are shipped in the SITE directory of G8WBDIR, which is divided into following sub-directories to ease administration:

- TP
- TP8
- DMIVTP
- FORMS
- DIMS8
- PRINT
- D8
- PREP
- SOUNDEX

- OTHER

You have two options when implementing Site changes: 1) Change the delivered source files directly or 2) Copy the toolkit/program to another file, with the program name you intend to use. Following example illustrates this activity:

- W8USRDL is selected toolkit
- COPY W8USRDL.CBL to MYUSRDL.CBL

Use the Micro Focus Workbench or COBOL Editor for program editing and checking (ANS85 language type), until your program is error-free.

---

## Configure

If you have changed the program name, you then have to change the DMIV-TP configuration to invoke your program version of User Derail processing. The name (MYUSRDL) is specified in the USER-DEMAIL-PROGRAM parameter. Refer to Reference guide chapter 11 for details.

---

## Test

You may of course use the Animator to test any changed program, however you must first amend your COBDIR setting to include \$G8WBDIR\SITE\subdir, so the changed program is found during execution. If you have not changed program name, you must also remove the Library configuration and re-generate libraries before performing the test. Otherwise the changed program will not be found (the Library version is used).

You may now test your program version of User Derail through the Animator. Start an animated TP session from G8WB (non-Zoom) and set your break-points once your program is entered (during initialize). As you are now testing TPR's calling User Derail service, you are able to step through your program, once your break-point setting is reached.

---

## Compile

Once you have completed testing, your program must be compiled and included in the G8WB SITE libraries. By doing this, your program will be executed at full speed and will not interfere with TPR animation.

Each SITE subdirectory has a COMP.ABT file with commands and directives for compile of all programs within the sub-directory. If you have changed program name you must amend/change this file before doing the compilation activity. You have now generated a .GNT version of your program, ex. MYUSRDL.GNT.

---

## Library update

The G8WB SITE library building activity is controlled by the following .LBT files, located in the G8WBDIR\SITE directory:

- SITE.LBT (General programs)
- SITEDMIV.LBT (DMIV-TP specific programs)
- SITETP8.LBT (TP8 specific programs)
- SITEBA.LBT (Batch specific programs)

These libraries are automatically allocated whenever you execute within G8WB.

## 6

If you have changed program name or you need to exclude a program while testing a new version, you must then amend/change the appropriate .LBT file, ex. replace W8USRDL by MYUSRDL.

You then re-generate all the libraries using the GENSITE.BAT file.

## EXECUTIVE TABLES

The various toolkits have fixed calling sequences for initialization, where Executive Tables are passed for access within the toolkit. Although these table references are only passed during initialize (Main program entry), the STICKY-LANGUAGE directive will make these available from within any program entry point.

### Executive Storage

Executive Storage is used for execution control of the TP Monitor. It is divided into five parts:

- System Status Information
- Session Status
- Transaction Status Information
- TPR Status Information
- TPR Pool Control

In general the TRUE-VALUE(1) indicates a condition is set.

The tpes-xxx-ptr field contains the index value for access into other tables.

```

*-W8TPCNFG.CPW-----*
*
*      GCOS-8 COBOL Workbench          Version 1.0.07      *
*
*              Copyright 1993 by B & C Solution - France  *
*-----*
*
*      Tables for TP-Executive Configuration                *
*
*-----*

*-   TP Executive Storage
01   w8tp-executive-storage.
    05   tpes-system.
        10   tpes-tx-serial-number      pic 9(08) comp-5.
        10   tpes-system-seq-no        pic 9(08) comp-5.
        10   tpes-spawn-snumb          pic 9(04) comp-5.
        10   tpes-slave-lid            pic x(04) .
        10   tpes-terminate            pic 9(01) comp-5.
        10   tpes-abort-tpr            pic 9(01) comp-5.
        10   tpes-file-process-status  pic 9(01) comp-5.
        10   tpes-session-max          pic 9(04) comp-5.
    05   tpes-session.
        10   tpes-session-id           pic 9(04) comp-5.
        10   tpes-lid-ptr              pic 9(04) comp-5.
        10   tpes-spawn-file-status    pic 9(01) comp-5.
        10   tpes-sleep-requested      pic 9(01) comp-5.
        10   tpes-lid-storage          pic x(32) .
        10   tpes-send-moved           pic 9(01) comp-5.
        10   tpes-receive-moved        pic 9(01) comp-5.
    05   tpes-transaction.
        10   tpes-mid-ptr              pic 9(04) comp-5.
        10   tpes-prof-ptr             pic 9(04) comp-5.
        10   tpes-txs-ptr              pic 9(04) comp-5.

```

```

10 tpes-phase-number          pic 9(08) comp-5.
10 tpes-send-indicator        pic 9(01) comp-5.
10 tpes-tpr-count             pic 9(08) comp-5.
10 tpes-tp-storage-size       pic 9(04) comp-5.
10 tpes-constant-storage-size pic 9(06) comp-5.
10 tpes-tx-storage-size       pic 9(06) comp-5.
10 tpes-edit-mode             pic 9(01) comp-5.
10 tpes-forms-mode            pic 9(01) comp-5.
10 tpes-stack-level           pic 9(02) comp-5.
   88 tpes-non-stacked         value 0.
10 tpes-stack-current-tpr     pic x(08).
10 tpes-stack-prior-tpr       pic x(08).
10 tpes-stack-tpr             occurs 25 times
                               pic x(008).

05 tpes-tpr.
   10 tpes-tpr-checkpoint      pic 9(001) comp-5.
   10 tpes-tpr-conversation    pic 9(001) comp-5.
   10 tpes-tpr-abort-status    pic 9(001) comp-5.
   10 tpes-tpr-send-indicator  pic 9(001) comp-5.
   10 tpes-tpr-cancel-status   pic 9(001) comp-5.
   10 tpes-tpr-type            pic 9(001) comp-5.
   10 tpes-tpr-chanc           pic 9(001) comp-5.
   10 tpes-tpr-unlka           pic 9(001) comp-5.
05 tpes-tpr-pool.
   10 tpes-tpr-pool-max        pic 9(001) comp-5.
   10 tpes-tpr-pool-current    pic 9(001) comp-5.
   10 tpes-tpr-pool-entry      occurs 9 times
                               pic x(008).

```

## G8WB Defaults

The G8WB defaults reflect the default configuration of G8WB. The Terminal-xxx-Entry may be used for generic call of the terminal emulator. As an example the Terminal-Msg-Entry may be used to update the status line (25).

```

*- g8wb-defaults
01 w8tp-defaults.
   05 w8tp-monitor-pgm         pic x(08).
   88 dmiv-tp-monitor          value "W8DMIVTP".
   88 tp8-monitor              value "W8TP8".
   05 w8tp-terminal-emul-pgm   pic x(08).
   88 vip77-emulator           value "W8V77XXE".
   88 vip78-emulator           value "W8V78XXE".
   88 dku71-emulator           value "W8D71XXE".
   88 ibm327x-emulator         value "W8I327XE".
   05 terminal-get-entry.
   10 terminal-get-prefix      pic x(03).
   10 filler                   pic x(05) value "GET".
   05 terminal-put-entry.
   10 terminal-put-prefix      pic x(03).
   10 filler                   pic x(05) value "PUT".
   05 terminal-clr-entry.
   10 terminal-clr-prefix      pic x(03).
   10 filler                   pic x(05) value "CLR".
   05 terminal-off-entry.
   10 terminal-off-prefix      pic x(03).
   10 filler                   pic x(05) value "OFF".
   05 terminal-on-entry.
   10 terminal-on-prefix       pic x(03).

```

```

10 filler                                pic x(05) value "ON".
05 terminal-msg-entry.
10 terminal-msg-prefix                    pic x(03).
10 filler                                pic x(05) value "MSG".
05 w8tp-terminal-case                     pic x(01).
05 w8tp-default-lid                       pic x(04).
88 no-default-lid                         value spaces.
05 w8tp-default-mid                       pic x(08).
88 no-default-mid                         value spaces.
05 w8tp-forms-software-pgm                pic x(08).
88 tpff-forms-manager                     value "W8TPFF".
88 dims-forms-manager                     value "W8DIMS8".
88 tsm-forms-manager                       value "W8TSM8".
88 format-forms-manager                   value "W8FMAT".
88 sold-forms-manager                     value "W8SOLD".
88 no-forms-manager                       value "NONE".
05 w8tp-print-software-pgm                pic x(08).
88 dispatch8-print-manager                value "W8DPS8".
88 no-print-manager                       value "NONE".
05 w8tp-printer-device                    pic x(114).
05 w8tp-os-type                           pic x(01) comp-x.
88 os-type-msdos                          value 1 2 3.
88 os-type-unix                            value 128 129.
88 os-type-os2                             value 0.
05 w8tp-trace-exec-call                    pic x(01) comp-5.
88 trace-exec-call-on                      value 1.
05 w8tp-trace-tx-sequence                  pic x(01) comp-5.
88 trace-tx-sequence-on                    value 1.
05 w8tp-trace-comm-detail                  pic x(01) comp-5.
88 trace-comm-detail-on                    value 1.
05 w8tp-animate-signal                     pic x(01) comp-5.
88 animation-mode-on                       value 1.

```

## TP Monitor Configuration

The TP Monitor Configuration reflect the configuration of either DMIV-TP or TP8. All common configuration parameters are collected in this table.

```

*-   DMIV-TP & TP8 configuration
01   w8tp-configuration.
05   w8tp-system-name                      pic x(08).
05   w8tp-mid-size                          pic 9(02) comp-5.
88   mid-procedure                          value 0.
05   w8tp-mb-size                            pic 9(04) comp-5.
05   w8tp-startup-file                       pic x(114).
05   w8tp-termination-file                   pic x(114).
05   w8tp-max-cnst                           pic 9(04) comp-5.
05   w8tp-cnst-table-ptr                     usage pointer.
05   w8tp-max-txs                            pic 9(04) comp-5.
05   w8tp-txs-table-ptr                       usage pointer.
05   w8tp-max-prof                           pic 9(04) comp-5.
05   w8tp-prof-table-ptr                     usage pointer.
05   w8tp-max-mid                            pic 9(04) comp-5.
05   w8tp-mid-table-ptr                       usage pointer.
05   w8tp-max-lid                            pic 9(04) comp-5.
05   w8tp-lid-table-ptr                       usage pointer.
05   w8tp-max-rtptr                           pic 9(04) comp-5.
05   w8tp-rtptr-table-ptr                     usage pointer.

```

```

05  w8tp-common-uwa          pic x(01).
05  w8tp-max-tx-size         pic 9(06) comp-5.
05  w8tp-max-constant-size  pic 9(06) comp-5.
05  w8tp-max-form-fields    pic 9(03) comp-5.
05  w8tp-tpr-pool-max       pic 9(01) comp-5.
05  w8tp-tpr-pool-mode      pic 9(01) comp-5.

```

## DMIV-TP Configuration

The DMIV-TP configuration reflect only specific DMIV-TP configuration.

```

*-  DMIV-TP Configuration
01  w8tp-dmiv-tp-configuration.
05  w8tp-user-derail-pgm     pic x(08).
    88  dmiv-tp-dummy-user-derail value "W8USRDRDL".
05  w8tp-user-procedure-pgm  pic x(08).
    88  dmiv-tp-dummy-user-proc  value "W8USRPRC".

```

## TP8 Configuration

The TP8 configuration reflect only specific TP8 configuration.

```

*-  TP8 Configuration
01  w8tp-tp8-configuration.
05  w8tp-cxpp-pgm           pic x(08).
    88  tp8-dummy-cxpp       value "W8CXPP".
05  w8tp-cxap-pgm          pic x(08).
    88  tp8-dummy-cxap       value "W8CXAP".
*-----*

```

## Logical-Id Table

The Logical-Id table contains static and current status information for all Logical-Id's configured in either DMIV-TP or TP8.

```

*-W8TPLIDT.CPW-----*
*
*   GCOS-8 COBOL Workbench           Version 1.0.07
*
*                               Copyright 1991 by B & C Solution - France
*-----*
*
*   TP-Executive Logical-Id Table
*
*-----*
78  w8tp-lid-entry-size           value 11.
01  w8tp-lid-table.
05  w8tp-lid-table-max            pic 9(04) comp-5.
05  w8tp-lid-entry                occurs 5000 times
                                   indexed by lid-ix.
10  w8tp-lid                      pic x(04).
10  w8tp-lid-type                 pic 9(02) comp-5.
88  normal-type-lid              value 0.

```

88	slave-type-lid	value 1.
88	rop-type-lid	value 2.
88	spawn-type-lid	value 3.
10	w8tp-lid-mode	pic 9(02) comp-5.
10	w8tp-lid-input-msg-number	pic 9(08) comp-5.
10	w8tp-lid-status	pic 9(01) comp-5.
88	sleeping-lid	value 1.
88	awaken-lid	value 2.
88	spawn-active	value 3.
*-----*		

## Message-Id/Command Table

The Message-Id/Command table contains information on all Message-Id's or Commands configured in either DMIV-TP or TP8. This table must be used in conjunction with the Profile table (and other tables) to form a complete picture.

*-W8TPMIDT.CPW-----*		
*		
88	GCOS-8 COBOL Workbench	Version 1.0.07
*		
Copyright 1991 by B & C Solution - France		
*-----*		
*		
88	TP-Executive Message-Id Table	
*-----*		
78	w8tp-mid-entry-size	value 18.
01	w8tp-mid-table.	
05	w8tp-mid-table-max	pic 9(04) comp-5.
05	w8tp-mid-entry	occurs 3000 times indexed by mid-ix.
10	w8tp-mid	pic x(08).
10	w8tp-mid-first-tpr	pic x(08).
10	w8tp-mid-prof-ptr	pic 9(04) comp-5.
*-----*		

## Constant Storage Table

The Constant Storage table contains information on all Constant Storage's configured in either DMIV-TP or TP8.

*-W8TPCNST.CPW-----*		
*		
88	GCOS-8 COBOL Workbench	Version 1.0.07
*		
Copyright 1992 by B & C Solution - France		
*-----*		
*		
88	TP-Executive Constant-Storage Table	
*-----*		
78	w8tp-cnst-entry-size	value 25.
01	w8tp-cnst-table.	
05	w8tp-cnst-table-max	pic 9(04) comp-5.
05	w8tp-cnst-entry	occurs 1024 times indexed by cnst-ix.



```

10 w8tp-prof-constant-ptr    pic 9(04) comp-5.
10 w8tp-prof-txs-ptr        pic 9(04) comp-5.
10 w8tp-prof-name           pic x(08).
10 w8tp-prof-abort-tpr      pic x(08).
10 w8tp-prof-tx-size        pic 9(06) comp-5.
10 w8tp-prof-constant-size  pic 9(06) comp-5.
10 w8tp-prof-tx-recv-proc   pic 9(04) comp-5.
10 w8tp-prof-tx-send-proc   pic 9(04) comp-5.
10 w8tp-prof-tx-init-proc   pic 9(04) comp-5.
10 w8tp-prof-ph-init-proc   pic 9(04) comp-5.
10 w8tp-prof-tx-term-proc   pic 9(04) comp-5.
10 w8tp-prof-ph-term-proc   pic 9(04) comp-5.
10 w8tp-prof-edit-mode      pic 9(01) comp-5.
*-----*
```

## Message Buffer

The Message Buffer is used for send and receive communication. The ":xx:" is replaced during copy (refer to specific toolkit use).

```

*-W8TPMBH.CPW-----*
*
*   GCOS-8 COBOL Workbench           Version 1.0.07
*
*
*           Copyright 1991 by B & C Solution - France
*-----*
*
*   TP-Executive Message Buffer
*-----*
78      :xx:buffer-size      value 4196.
78      :xx:buffer-max      value 4096.
01      :xx:message-buffer.
        05      :xx:data.
          10     :xx:data-chr      pic x(01)
                                occurs :xx:buffer-size times
                                indexed by :xx:chr-i.
        05      :xx:ptr        pic 9(04) comp-5.
        05      :xx:length     pic 9(04) comp-5.
        05      :xx:offset     pic 9(04) comp-5.
        05      :xx:fc         pic x(04).
        05      :xx:lid        pic x(04).
        05      :xx:msg        pic x(40).
*-----*
```



## DMIV-TP PROCEDURES

### DMIV-TP User Derail

The Toolkit for implementation of DMIV-TP User Derail provides you with a straight forward path for porting of existing User Derail coding. Although existing code is in GMAP, the Micro Focus COBOL/2 system will provide features very similar to possibilities in GMAP.

The Toolkit is used to define both data areas, external files and procedure coding.

### External Files

You may define external files as required in the File-Control and File Section. Such files should be opened during initialization entry and may then be accessed during process of procedure calls. The use of external (similar to the TP Journal Log), makes such files available both to User Derail and User Procedure programs.

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8USRDL.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00          *
*
*                               Copyright 1992 by B & C Solution - France *
*-----*
*
*      DMIV-TP User Derail Emulation (dummy module to customize)
*
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. w8usrdl.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

      copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****
data division.
file section.

      copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value         value 0.

```

### Data Definition

Common data areas are defined as required in the Data Division. The toolkit contains a default Derail Buffer, which is shared with the User procedure program via it's external definition.

Common data areas should be initialized during the initialization entry.

```

*- Work fields
   copy "$g8wkdir\mfi\w8migflg.cpy".

*- Derail Buffer
01      w8tp-drl-buffer          pic x(01) external.

```

---

## Executive Tables

Access to TP Monitor Executive tables are provide through the initialize entry.

*Note: Do not change the initialize entry using clause and the table definitions, as G8WB is using these as provided in the toolkit.*

```

*-----*
linkage section.

*- TP-STORAGE Definition
   copy "$g8wkdir\tpmscpy\tpstorag.cpy".
   copy "$g8wkdir\tpmscpy\cdinput.cpy".
   copy "$g8wkdir\tpmscpy\cdoutput.cpy".

*- Constant-Storage (current)
01  constant-storage          pic x(00001).

*- TX-Storage (current)
01  tx-storage                pic x(00001).

*- TP-Executive Configuration
   copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

*- Constant-Storage Table
   copy "$g8wkdir\syscpy\w8tpcnst.cpw".

*- Profile Table
   copy "$g8wkdir\syscpy\w8tpprot.cpw".

*- TX-Storage Table
   copy "$g8wkdir\syscpy\w8tptxst.cpw".

*- MID Table
   copy "$g8wkdir\syscpy\w8tpmidt.cpw".

*- LID Table
   copy "$g8wkdir\syscpy\w8tplidt.cpw".

   copy "$g8wkdir\tpmscpy\tpcomflg.cpy".

```

---

## Interfacing Arguments

The TPR procedure interface may differ from your current implementation and the toolkit only contain a sample on how this could be set-up. Additional arguments may be added easily.

```

01      drl-ident          pic x(08).
01      drl-function      pic 9(03).
01      drl-status        pic 9(01).
      88      drl-ok          value zero.

01      drl-argument-1    pic x(01).

```

---

## Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                        w8tp-configuration
                        w8tp-dmiv-tp-configuration
                        w8tp-executive-storage
                        w8tp-cnst-table
                        w8tp-txs-table
                        w8tp-prof-table
                        w8tp-mid-table
                        w8tp-lid-table
                        tp-storage
                        constant-storage
                        tx-storage.

*-----*
main section.
section-entry.

      display "User Derail Initialize"
      move low-values to w8tp-drl-buffer.

section-exit.
      exit program.

```

---

## Procedure Process

The procedure requests from application TPR's are made as follows:

```

CALL ".USER" USING DRL-IDENT,
                  DRL-FUNCTION,
                  DRL-STATUS,
                  ARGUMENT.

```

You may change the interface as required. The toolkit illustrate a technique, which is used throughout G8WB, and which allow you to prevent a crash due to bad TPR interfacing, instead you are able to abort the TPR in a controlled fashion.

```

*
entry "tpUSER" using drl-ident
                    drl-function
                    drl-status
                    drl-argument-1.
*
dmiv-tp-user-derail section.
section-entry.

    if w8-arg-no not = w8-arg-tot
        move "W8E" to abort-code
        move "Argno" to abort-reason
        perform tp-generated-abort
        go to section-exit
    end-if
    if length of drl-ident > w8-arg-lgt ( 1 ) or
        length of drl-function > w8-arg-lgt ( 2 ) or
        length of drl-status > w8-arg-lgt ( 3 ) or
        length of drl-argument-1 > w8-arg-lgt ( 4 )
        move "W8E" to abort-code
        move "Argsiz" to abort-reason
        perform tp-generated-abort
        go to section-exit
    end-if

    move zero to drl-status.

    string "USRDRL Called - Ident("
            drl-ident
            " Function("
            drl-function
            ")")
            delimited by size into journal-log-record
    perform trace-exec-call

    evaluate drl-ident
        when "example"
            evaluate drl-function
                when zero
                    move 1 to drl-status
                when other
                    move 1 to drl-status
            end-evaluate
        when other
            move 1 to drl-status
    end-evaluate.

section-exit.
    exit program.

```

---

## Trace & Abort service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

Use the abort procedure to force a clean abort of the calling TPR.

```

trace-exec-call section.
section-entry.

    if trace-exec-call-on
        write journal-log-record
    end-if
    move spaces to journal-log-record.

section-exit.
    exit.
*-----*
tp-generated-abort section.

    string "TP forced .ABORT ("
        abort-code ") reason ("
        abort-reason ")"
        delimited by size into journal-log-record
    perform trace-exec-call.
    move 1 to tpes-tpr-abort-status
        w8-abort-flg.

section-exit.
    exit.
***** END OF PROGRAM *****

```

---

## DMIV-TP User Procedure

The Toolkit for implementation of DMIV-TP User Procedures provides you with a straight forward path for porting of existing User Procedure coding. Although existing code is in GMAP, the Micro Focus COBOL/2 system will provide features very similar to possibilities in GMAP.

The Toolkit is used to define both data areas, external files and procedure coding.

---

## External Files

You may define external files as required in the File-Control and File Section. Such files should be opened during initialization entry and may then be accessed during process of procedure calls. The use of external (similar to the TP Journal Log), makes such files available both to User Procedure and User Derail programs.

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8USRPRC.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*              Copyright 1993 by B & C Solution - France  *
*-----*
*
*      DMIV-TP User Procedure Emulation (to customize)    *
*
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. w8usrpc.

***** ENVIRONMENT DIVISION *****

```

```

environment division.
input-output section.
file-control.

    copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****
data division.
file section.

    copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value         value 0.

*- Work fields
01      edit-3              pic z(02)9.
01      constant-8         pic 9(08) comp-5
                                value 8.
01      ws-mid-size        pic 9(02) comp-5.
01      ws-count           pic 9(02) comp-5.
01      ws-delimiter       pic x(01).

01      ws-delimiters.
05      etx.
01      filler              pic x(001) comp-x value 3.
05      ht.
01      filler              pic x(001) comp-x value 9.
05      cr.
01      filler              pic x(001) comp-x value 13.

```

---

## Data Definition

Common data areas are defined as required in the Data Division. The toolkit contains a default Derail Buffer, which is shared with the User procedure program via it's external definition.

Common data areas should be initialized during the initialization entry.

```

*- Derail Buffer
01      w8tp-drl-buffer     pic x(01) external.

```

---

## Executive Tables

Access to TP Monitor Executive tables are provide through the initialize entry.

*Note: Do not change the initialize entry using clause and the table definitions, as G8WB is using these as provided in the toolkit.*

```

*-----*
linkage section.

*- TP-STORAGE Definition
    copy "$g8wkdir\tpmscpy\tpstorag.cpy".

```

```

copy "$g8wkdir\tpmscpy\cdinput.cpy".
copy "$g8wkdir\tpmscpy\cdoutput.cpy".

*- Constant-Storage (current)
01 constant-storage          pic x(00001).

*- TX-Storage (current)
01 tx-storage                pic x(00001).

*- TP-Executive Configuration
copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

*- Constant-Storage Table
copy "$g8wkdir\syscpy\w8tpcnst.cpw".

*- Profile Table
copy "$g8wkdir\syscpy\w8tpprot.cpw".

*- TX-Storage Table
copy "$g8wkdir\syscpy\w8tptxst.cpw".

*- MID Table
copy "$g8wkdir\syscpy\w8tpmidt.cpw".

*- LID Table
copy "$g8wkdir\syscpy\w8tplidt.cpw".

*- Input Message-Buffer
copy "$g8wkdir\syscpy\w8tpmbh.cpw"
replacing ==:xx:== by ==i-mbh==.

*- Output Message-Buffer
copy "$g8wkdir\syscpy\w8tpmbh.cpw"
replacing ==:xx:== by ==o-mbh==.

*- Message-Send/Receive-area
01 tpr-message-area          pic x(2400).

*- CD-Input-block
copy "$g8wkdir\syscpy\xcdinput.cpw"
replacing ==:xx:== by ==tpr-cd==.

*- CD-Output-block
copy "$g8wkdir\syscpy\xcdoutpu.cpw"
replacing ==:xx:== by ==tpr-cd==.

copy "$g8wkdir\tpmscpy\tpcomflg.cpy".

```

---

## Executive Interfacing

The TP Monitor Executive will call User Procedures in accordance with the DMIV-TP configuration (ref. PROFILE configuration in Reference Guide, chapter 11).

01	w8tp-procedure-type	pic 9(02) comp-5.
88	receive-procedure	value 1.
88	send-procedure	value 2.
88	tx-init-procedure	value 3.
88	tx-term-procedure	value 4.

88	ph-init-procedure	value 5.
88	ph-term-procedure	value 6.
88	message-id-procedure	value 7.
01	w8tp-procedure-number	pic 9(03) comp-5.

---

## Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                        w8tp-configuration
                        w8tp-dmiv-tp-configuration
                        w8tp-executive-storage
                        w8tp-cnst-table
                        w8tp-txs-table
                        w8tp-prof-table
                        w8tp-mid-table
                        w8tp-lid-table
                        i-mbh-message-buffer
                        o-mbh-message-buffer
                        tp-storage
                        constant-storage
                        tx-storage.

*-----*
main section.
section-entry.

        display "User Procedure Initialize".

section-exit.
        exit program.

```

---

## Receive Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
entry "tpPrcR" using w8tp-procedure-type
                    w8tp-procedure-number
                    tpr-cd-input
                    tpr-message-area.

tpprcr-user-procedure section.
section-entry.

        evaluate w8tp-procedure-number
                when 0 *> never called - replicate standard receive

```

```

string "RECEIVE Procedure 0 activated"
    delimited by size into journal-log-record
perform trace-exec-call
if w8-rxxx = w8-rseg
    move zero to ws-count
    move space to ws-delimiter
    unstring i-mbh-data (1:i-mbh-length)
        delimited by cr
        into tpr-message-area (1:w8-into-lgt)
        delimiter ws-delimiter
        count      ws-count
        pointer i-mbh-offset
if i-mbh-offset > i-mbh-length
    move "3" to tpr-cd-end-key
else
    if ws-delimiter = cr
        add 1 to ws-count
        move cr to tpr-message-area (ws-count:1)
        move "1" to tpr-cd-end-key
    else
        move "0" to tpr-cd-end-key
    end-if
end-if
move ws-count to tpr-cd-text-length-i
else
    if i-mbh-offset > i-mbh-length
        move zero to tpr-cd-text-length-i
    else
        compute tpr-cd-text-length-i =
            i-mbh-length - i-mbh-offset + 1
    end-if
    if tpr-cd-text-length-i > w8-into-lgt
        move w8-into-lgt to tpr-cd-text-length-i
        move "0" to tpr-cd-end-key
    else
        move "3" to tpr-cd-end-key
    end-if
move i-mbh-data
    (i-mbh-offset:tpr-cd-text-length-i) to
    tpr-message-area
    (1:tpr-cd-text-length-i)
add tpr-cd-text-length-i to i-mbh-offset
end-if
move true-value to tpes-receive-moved
when other
    move w8tp-procedure-number to edit-3
    string "USRPRC Receive # "
        edit-3
        " called - ignored"
        delimited by size into journal-log-record
    perform trace-tx-sequence
end-evaluate.

section-exit.
exit program.

```

---

## Send Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
entry "tpPrcS" using w8tp-procedure-type
                    w8tp-procedure-number
                    tpr-cd-output
                    tpr-message-area.
tpprcs-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0 * > never called - replicate standard send
            string "SEND Procedure 0 activated"
                delimited by size into journal-log-record
            perform trace-exec-call
            if tpr-cd-text-length-o > zero
                move tpr-message-area
                    (1:tpr-cd-text-length-o) to
                    o-mbh-data
                    (o-mbh-ptr:tpr-cd-text-length-o)
                add tpr-cd-text-length-o to o-mbh-ptr
            end-if
            move true-value to tpes-send-moved
        when other
            move w8tp-procedure-number to edit-3
            string "USRPRC Send # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
exit program.

```

---

## Procedure Selection

Each procedure is identified by a unique Procedure Number, which is specified in the configuration of DMIV-TP profiles. A common entry point is used to determine procedure type.

```

*-----*
entry "tpProc" using w8tp-procedure-type
                    w8tp-procedure-number.
tpproc-user-procedure section.
section-entry.

    evaluate true
        when tx-init-procedure
            perform process-tx-init-procedure
        when tx-term-procedure
            perform process-tx-term-procedure
        when ph-init-procedure
            perform process-ph-init-procedure
        when ph-term-procedure
            perform process-ph-term-procedure
        when message-id-procedure
            perform process-message-id-procedure
        when other
            display "*-> Invalid User Procedure Type"
            display "W8TPEXEC Aborted from W8USRPRC"

```

```

        stop run
    end-evaluate.

section-exit.
    exit program.

```

---

## TX Init/Term Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
process-tx-init-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "USRPRC TX-Init # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
    exit.
*-----*
process-tx-term-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "USRPRC TX-Term # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
    exit.

```

---

## Phase Init/Term Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
process-ph-init-procedure section.
section-entry.

```

```

        evaluate w8tp-procedure-number
            when 0
                continue
            when other
                move w8tp-procedure-number to edit-3
                string "USRPRC PH-Init # "
                    edit-3
                    " called - ignored"
                    delimited by size into journal-log-record
                perform trace-tx-sequence
            end-evaluate.

    section-exit.
        exit.
*-----*
process-ph-term-procedure section.
section-entry.

        evaluate w8tp-procedure-number
            when 0
                continue
            when other
                move w8tp-procedure-number to edit-3
                string "USRPRC PH-Term # "
                    edit-3
                    " called - ignored"
                    delimited by size into journal-log-record
                perform trace-tx-sequence
            end-evaluate.

    section-exit.
        exit.

```

---

## Message-Id Procedure

Only one (triggered by VARYING configuration) Message-Id procedure is allowed, and the existing coding section should be changed or amended to include new functionality.

*Note: The current coding is used as default Message-Id procedure.*

```

*-----*
process-message-id-procedure section.    *> Only One allowed
section-entry.

    if w8tp-mid-size > 0                    *> Fixed-length
        if w8tp-mid-size > i-mbh-length
            move i-mbh-length to ws-mid-size
        else
            if i-mbh-data(1:1) = "$"
                move i-mbh-length to ws-mid-size
            else
                move w8tp-mid-size to ws-mid-size
            end-if
        end-if
        move i-mbh-data (1:ws-mid-size) to tx-type
    else
        if i-mbh-length > 8                    *> Variable length
            move 9 to ws-mid-size
        else

```

```

        move i-mbh-length to ws-mid-size
    end-if
    move x'ff' to ws-delimiter
    unstring i-mbh-data (1:ws-mid-size)
        delimited by etx or
            ht or
            space
        into tx-type
        delimiter ws-delimiter
        count ws-count
    move ws-count to ws-mid-size
    if ws-delimiter = space           *> remove leading space
        add 1 to ws-mid-size
    end-if
end-if

move ws-mid-size to i-mbh-offset
add 1 to i-mbh-offset
inspect tx-type
    replacing all "*" by " "
call "CBL_TOUPPER" using tx-type by value
    constant-8.

string "USRPRC Message-id called - "
    " MID("
    delimited by size
    tx-type
    delimited by space
    ")"
    delimited by size into journal-log-record
perform trace-tx-sequence.

section-exit.
exit.

```

---

## Trace service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

```

*-----*
trace-tx-sequence section.
section-entry.

    if trace-tx-sequence-on
        write journal-log-record
    end-if
    move spaces to journal-log-record.

section-exit.
exit.
*-----*

trace-exec-call section.
section-entry.

    if trace-exec-call-on
        write journal-log-record
        move spaces to journal-log-record
    end-if.

```

```
section-exit.  
    exit.  
***** END OF PROGRAM *****
```

# TP8 PROCEDURES

## TP8 Program Procedures

The Toolkit for implementation of TP8-TP User Program Procedures provides you with a straight forward path for porting of existing CXPP Procedure coding. Although existing code is in GMAP, the Micro Focus COBOL/2 system will provide features very similar to possibilities in GMAP.

The Toolkit is used to define both data areas, external files and procedure coding.

### External Files

You may define external files as required in the File-Control and File Section. Such files should be opened during initialization entry and may then be accessed during process of procedure calls. The use of external (similar to the TP Journal Log), makes such files available both to CXPP and CXAP programs.

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8CXPP.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00          *
*
*                               Copyright 1993 by B & C Solution - France *
*-----*
*      TP8 Program Procedure Emulation (dummy module to customize) *
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. w8cxpp.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

      copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****
data division.
file section.

      copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value         value 0.

```

### Data Definition

Common data areas are defined as required in the Data Division. The toolkit contains a default "Derail" Buffer, which is shared with the CXAP procedure program via it's external definition.

Common data areas should be initialized during the initialization entry.

```
*- Work fields
   copy "$g8wkdir\mfi\w8migflg.cpy".

*- CXPP Buffer
01      w8tp-drl-buffer          pic x(01) external.
```

---

## Executive Tables

Access to TP Monitor Executive tables are provide through the initialize entry.

*Note: Do not change the initialize entry using clause and the table definitions, as G8WB is using these as provided in the toolkit.*

```
*-----*
linkage section.

*- TP-STORAGE Definition
   copy "$g8wkdir\tpmscpy\tpxstora.cpy".

*- Constant-Storage (current)
01  constant-storage          pic x(00001).

*- TX-Storage (current)
01  tx-storage                pic x(00001).

*- TP-Executive Configuration
   copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

*- Constant-Storage Table
   copy "$g8wkdir\syscpy\w8tpcnst.cpw".

*- Profile Table
   copy "$g8wkdir\syscpy\w8tpprot.cpw".

*- TX-Storage Table
   copy "$g8wkdir\syscpy\w8tptxst.cpw".

*- MID Table
   copy "$g8wkdir\syscpy\w8tpmidt.cpw".

*- LID Table
   copy "$g8wkdir\syscpy\w8tplidt.cpw".

   copy "$g8wkdir\tpmscpy\tpcomflg.cpy".
```

---

## Interfacing Arguments

The TPR procedure interface may differ from your current implementation and the toolkit only contain a sample on how this could be set-up. Additional arguments may be added easily.

```
01      prc-ident              pic x(08).
```

```

01      prc-function          pic 9(03).
01      prc-status           pic 9(01).
      88      prc-ok          value zero.

01      prc-argument-1      pic x(01).

```

## Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                        w8tp-configuration
                        w8tp-tp8-configuration
                        w8tp-executive-storage
                        w8tp-cnst-table
                        w8tp-txs-table
                        w8tp-prof-table
                        w8tp-mid-table
                        w8tp-lid-table
                        tp-storage
                        constant-storage
                        tx-storage.

*-----*
main section.
section-entry.

        display "CXPP User Procedure Initialize"
        move low-values to w8tp-drl-buffer.

section-exit.
        exit program.

```

## Procedure Process

The procedure requests from application TPR's are made as follows:

```

CALL ".USER" USING DRL-IDENT,
                  DRL-FUNCTION,
                  DRL-STATUS,
                  ARGUMENT.

```

You may change the interface as required. The toolkit illustrate a technique, which is used throughout G8WB, and which allow you to prevent a crash due to bad TPR interfacing, instead you are able to abort the TPR in a controlled fashion.

```

*
```

```

entry "tpUSER" using prc-ident
                    prc-function
                    prc-status
                    prc-argument-1.
*
cxpp-user-procedure section.
section-entry.

    if w8-arg-no not = w8-arg-tot
        move "W8E" to abort-code
        move "Argno" to abort-reason
        perform tp-generated-abort
        go to section-exit
    end-if
    if length of prc-ident > w8-arg-lgt ( 1 ) or
        length of prc-function > w8-arg-lgt ( 2 ) or
        length of prc-status > w8-arg-lgt ( 3 ) or
        length of prc-argument-1 > w8-arg-lgt ( 4 )
        move "W8E" to abort-code
        move "Argsiz" to abort-reason
        perform tp-generated-abort
        go to section-exit
    end-if

    move zero to prc-status.

    string "CXPP Called - Ident("
            prc-ident
            " Function("
            prc-function
            ")"
            delimited by size into journal-log-record
    perform trace-exec-call

    evaluate prc-ident
        when "example"
            evaluate prc-function
                when zero
                    move 1 to prc-status
                when other
                    move 1 to prc-status
            end-evaluate
        when other
            move 1 to prc-status
    end-evaluate.

section-exit.
    exit program.

```

---

## Trace & Abort service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

Use the abort procedure to force a clean abort of the calling TPR.

```

*-----*
trace-exec-call section.
section-entry.

```

```

        if trace-exec-call-on
            write journal-log-record
        end-if
        move spaces to journal-log-record.

    section-exit.
        exit.
*-----*
    tp-generated-abort section.

        string "TP forced .ABORT ("
            abort-code ") reason ("
            abort-reason ")"
            delimited by size into journal-log-record
        perform trace-exec-call.
        move 1 to tpes-tpr-abort-status
            w8-abort-flg.

    section-exit.
        exit.
***** END OF PROGRAM *****

```

## TP8 Administrative Procedures

The Toolkit for implementation of TP8 User Administrative Procedures provides you with a straight forward path for porting of existing CXAP Procedure coding. Although existing code is in GMAP, the Micro Focus COBOL/2 system will provide features very similar to possibilities in GMAP.

The Toolkit is used to define both data areas, external files and procedure coding.

### External Files

You may define external files as required in the File-Control and File Section. Such files should be opened during initialization entry and may then be accessed during process of procedure calls. The use of external (similar to the TP Journal Log), makes such files available both to CXAP and CXPP programs.

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8CXAP.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*              Copyright 1993 by B & C Solution - France  *
*-----*
*
*      TP8 Administrative User Procedure Emulation (to customize) *
*
*-----*
***** IDENTIFICATION DIVISION *****
    identification division.
    program-id. w8cxap.

***** ENVIRONMENT DIVISION *****
    environment division.
    input-output section.

```

```

file-control.

    copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****
data division.
file section.

    copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value        value 0.

*- Work fields
01      edit-3              pic z(02)9.
01      constant-8         pic 9(08) comp-5
                                value 8.
01      ws-mid-size        pic 9(02) comp-5.
01      ws-count           pic 9(02) comp-5.
01      ws-delimiter       pic x(01).

01      ws-delimiters.
05      etx.
010     filler              pic x(001) comp-x value 3.
05      ht.
010     filler              pic x(001) comp-x value 9.
05      cr.
010     filler              pic x(001) comp-x value 13.

01      wix                 pic 9(08) comp-5.
01      ws-mbh-char.
05      ws-mbh-char-value   pic 9(02) comp-x.
088     char-value-non-dis value 0 thru 31
                                128 thru 255.

01      ws-mbh-fields.
05      ws-octal-wix        pic 9(02) comp-x.
05      ws-octal-1          pic 9(01).
05      ws-octal-2          pic 9(01).
05      ws-octal-3          pic 9(01).
05      ws-dis-ix           pic 9(02) comp-x.

```

---

## Data Definition

Common data areas are defined as required in the Data Division. The toolkit contains a default "Derail" Buffer, which is shared with the CXPP program via its external definition.

Common data areas should be initialized during the initialization entry.

```

*- CXPP Buffer
01      w8tp-drl-buffer     pic x(01) external.

```

---

## Executive Tables

Access to TP Monitor Executive tables are provide through the initialize entry.

*Note: Do not change the initialize entry using clause and the table definitions, as G8WB is using these as provided in the toolkit.*

```

*-----*
linkage section.

*- TP-STORAGE Definition
copy "$g8wkdir\tpmscpy\tpxstora.cpy".

*- Constant-Storage (current)
01 constant-storage          pic x(00001).

*- TX-Storage (current)
01 tx-storage                pic x(00001).

*- TP-Executive Configuration
copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

*- Constant-Storage Table
copy "$g8wkdir\syscpy\w8tpcnst.cpw".

*- Profile Table
copy "$g8wkdir\syscpy\w8tpprot.cpw".

*- TX-Storage Table
copy "$g8wkdir\syscpy\w8tptxst.cpw".

*- MID Table
copy "$g8wkdir\syscpy\w8tpmidt.cpw".

*- LID Table
copy "$g8wkdir\syscpy\w8tplidt.cpw".

*- Input Message-Buffer
copy "$g8wkdir\syscpy\w8tpmbh.cpw"
  replacing ==:xx:== by ==i-mbh==.

*- Output Message-Buffer
copy "$g8wkdir\syscpy\w8tpmbh.cpw"
  replacing ==:xx:== by ==o-mbh==.

*- Message-Send/Receive-area
01 tpr-message-area          pic x(2400).

*- CD-Input-block
copy "$g8wkdir\syscpy\xcdinput.cpw"
  replacing ==:xx:== by ==tpr-cd==.

*- CD-Output-block
copy "$g8wkdir\syscpy\xcdoutpu.cpw"
  replacing ==:xx:== by ==tpr-cd==.

copy "$g8wkdir\tpmscpy\tpcomflg.cpy".

```

## Executive Interfacing

The TP Monitor Executive will call User Administrative Procedures in accordance with the TP8 configuration (ref. PROFILE configuration in Reference Guide, chapter 11).

```

01      w8tp-procedure-type      pic 9(02) comp-5.
      88      receive-procedure      value 1.
      88      send-procedure      value 2.
      88      tx-init-procedure      value 3.
      88      tx-term-procedure      value 4.
      88      ph-init-procedure      value 5.
      88      ph-term-procedure      value 6.
      88      message-id-procedure    value 7.

01      w8tp-procedure-number      pic 9(03) comp-5.

```

---

## Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                    w8tp-configuration
                    w8tp-tp8-configuration
                    w8tp-executive-storage
                    w8tp-cnst-table
                    w8tp-txs-table
                    w8tp-prof-table
                    w8tp-mid-table
                    w8tp-lid-table
                    i-mbh-message-buffer
                    o-mbh-message-buffer
                    tp-storage
                    constant-storage
                    tx-storage.

*-----*
main section.
section-entry.

        display "CXAP User Procedure Initialize".

section-exit.
        exit program.

```

---

## Receive Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
entry "tpPrcR" using w8tp-procedure-type
                    w8tp-procedure-number

```

```

        tpr-cd-input
        tpr-message-area
        w8-receive-flg.
tpprcr-user-procedure section.
section-entry.

evaluate w8tp-procedure-number
  when 0 *> never called - replicate standard receive
  string "RECEIVE Procedure 0 activated"
    delimited by size into journal-log-record
  perform trace-exec-call
  if w8-rxxx = w8-rseg
    move zero to ws-count
    move space to ws-delimiter
    unstring i-mbh-data (1:i-mbh-length)
      delimited by cr
      into tpr-message-area (1:w8-into-lgt)
      delimiter ws-delimiter
      count ws-count
      pointer i-mbh-offset
  if i-mbh-offset > i-mbh-length
    move "3" to tpr-cd-end-key
  else
    if ws-delimiter = cr
      add 1 to ws-count
      move cr to tpr-message-area (ws-count:1)
      move "1" to tpr-cd-end-key
    else
      move "0" to tpr-cd-end-key
    end-if
  end-if
  move ws-count to tpr-cd-text-length-i
else
  if i-mbh-offset > i-mbh-length
    move zero to tpr-cd-text-length-i
  else
    compute tpr-cd-text-length-i =
      i-mbh-length - i-mbh-offset + 1
  end-if
  if tpr-cd-text-length-i > w8-into-lgt
    move w8-into-lgt to tpr-cd-text-length-i
    move "0" to tpr-cd-end-key
  else
    move "3" to tpr-cd-end-key
  end-if
  move i-mbh-data
    (i-mbh-offset:tpr-cd-text-length-i) to
    tpr-message-area
    (1:tpr-cd-text-length-i)
  add tpr-cd-text-length-i to i-mbh-offset
end-if
move true-value to tpes-receive-moved
when other
  move w8tp-procedure-number to edit-3
  string "CXAP Receive # "
    edit-3
    " called - ignored"
    delimited by size into journal-log-record
  perform trace-tx-sequence
end-evaluate.

```

```

section-exit.
    exit program.

```

## Send Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
entry "tpPrcS" using w8tp-procedure-type
                    w8tp-procedure-number
                    tpr-cd-output
                    tpr-message-area
                    w8-send-flg.

tpprcs-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0 *> never called - replicate standard send
            string "SEND Procedure 0 activated"
                delimited by size into journal-log-record
            perform trace-exec-call
            if tpr-cd-text-length-o > zero
                move tpr-message-area
                    (1:tpr-cd-text-length-o) to
                    o-mbh-data
                    (o-mbh-ptr:tpr-cd-text-length-o)
                add tpr-cd-text-length-o to o-mbh-ptr
            end-if
            move true-value to tpes-send-moved
        when other
            move w8tp-procedure-number to edit-3
            string "CXAP Send # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
    exit program.

```

## Procedure Selection

Each procedure is identified by a unique Procedure Number, which is specified in the configuration of TP8 profiles. A common entry point is used to determine procedure type.

```

*-----*
entry "tpProc" using w8tp-procedure-type
                    w8tp-procedure-number.

tpproc-user-procedure section.
section-entry.

    evaluate true
        when tx-init-procedure
            perform process-tx-init-procedure
        when tx-term-procedure

```

```

        perform process-tx-term-procedure
    when ph-init-procedure
        perform process-ph-init-procedure
    when ph-term-procedure
        perform process-ph-term-procedure
    when message-id-procedure
        perform process-message-id-procedure
    when other
        display "*-> Invalid User Procedure Type"
        display "W8TPEXEC Aborted from W8CXAP"
        stop run
    end-evaluate.

section-exit.
exit program.

```

---

## TX Init/Term Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
process-tx-init-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "CXAP TX-Init # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
exit.
*-----*
process-tx-term-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "CXAP TX-Term # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
exit.

```

---

## Phase Init/Term Procedure

User defined procedures should be amended in the evaluate expression with a perform of the appropriate section.

```

*-----*
process-ph-init-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "CXAP PH-Init # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
exit.
*-----*
process-ph-term-procedure section.
section-entry.

    evaluate w8tp-procedure-number
        when 0
            continue
        when other
            move w8tp-procedure-number to edit-3
            string "CXAP PH-Term # "
                edit-3
                " called - ignored"
                delimited by size into journal-log-record
            perform trace-tx-sequence
        end-evaluate.

section-exit.
exit.

```

---

## Message-Id Procedure

Only one (triggered by VARYING configuration) Command procedure is allowed, and the existing coding section should be changed or amended to include new functionality.

*Note: The current coding is used as default Command procedure.*

```

*-----*
process-message-id-procedure section.    *> Only One allowed
section-entry.

    if w8tp-mid-size > 0                    *> Fixed-length
        if w8tp-mid-size > i-mbh-length
            move i-mbh-length to ws-mid-size
        else
            if i-mbh-data(1:1) = "$"

```

```

        move i-mbh-length to ws-mid-size
    else
        move w8tp-mid-size to ws-mid-size
    end-if
end-if
move i-mbh-data(1:ws-mid-size) to tx-type
else
    if i-mbh-length > 8                *> Variable length
        move 9 to ws-mid-size
    else
        move i-mbh-length to ws-mid-size
    end-if
    move x'ff' to ws-delimiter
    unstring i-mbh-data (1:ws-mid-size)
        delimited by etx or
            ht or
            space
        into tx-type
        delimiter ws-delimiter
        count ws-count
    move ws-count to ws-mid-size
    if ws-delimiter = space            *> remove leading space
        add 1 to ws-mid-size
    end-if
end-if

move ws-mid-size to i-mbh-offset
add 1 to i-mbh-offset
inspect tx-type
    replacing all "*" by " "
call "CBL_TOUPPER" using tx-type by value
    constant-8.

string "CXAP Message-id called - "
    " MID("
    delimited by size
    tx-type
    delimited by space
    ")"
    delimited by size into journal-log-record
perform trace-tx-sequence.

section-exit.
exit.

```

---

## Trace service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

```

*-----*
trace-tx-sequence section.
section-entry.

    if trace-tx-sequence-on
        write journal-log-record
    end-if
    move spaces to journal-log-record.

```

```
section-exit.  
    exit.  
*-----*  
trace-exec-call section.  
section-entry.  
  
    if trace-exec-call-on  
        write journal-log-record  
        move spaces to journal-log-record  
    end-if.  
  
section-exit.  
    exit.  
***** END OF PROGRAM *****
```

---

## ABORT TPR

A very basic ABORT TPR (TP-ABT) is delivered as part of the SITE TP Toolkit. TP-ABT is normally only used as a last resort, if a SITE implemented ABORT TPR should fail. If you have modified standard TP-ABT on GCOS-8 (TP8 or DMIV-TP), you should port such changes into G8WB also.

---

## Terminal Emulation

The Toolkit for implementation of customized Terminal Emulation or Advanced Two-level front-end driver is currently undocumented. Please contact Technical Support if you need to use this Toolkit.

# FORMS Software

## Toolkit

The Forms software Toolkit is in principle very much similar to DMIV-TP and TP8 User Procedures.

### Concepts

A number of user developed Forms Management software solutions exist. Although G8WB will not be able to support such user developed software, it does contain a standard Toolkit, which will allow you to port parts or all of your Forms Management software and integrate this in the workbench environment.

*Note: If your Forms Management is based on service provided through separate TPR's, this toolkit is not required and such TPR's should simply be included in G8WB as any other application TPR.*

As an example the Toolkit has been used to implement TPF Emulation, which perform Forms handling through service calls, ex. CDGET, CSEND.

### Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

$set directives "$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8FXXX.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*              Copyright 1993 by B & C Solution - France  *
*-----*
*      User Developed Forms Run-time Manager                *
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. w8fxxx.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

copy "$g8wkdir\syscopy\w8tpjlse.cpf".

```

```

***** DATA DIVISION *****
data division.
file section.

    copy "$g8wkdir\syscpy\w8tpjld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value        value 0.

*- Work fields
01      msg-ptr             pic 9(04) comp-5.
01      ws-call-name       pic x(08).

*- Forms Receive/Send Buffer
01      message-area       pic x(2400).

*- TPR CD-Output-block
copy "$g8wkdir\syscpy\xcdoutpu.cpw"
    replacing ==:xx:== by ==tpr-cd==.

*- TPR Communication Flags
copy "$g8wkdir\tpmscpy\tpcomflg.cpy".

*- Workbench Migration Flags
copy "$g8wkdir\mfi\w8migflg.cpy".

*-----*
linkage section.

*- TP-Storage
copy "$g8wkdir\tpmscpy\tpstorag.cpy".

*- TP-Executive Configuration
copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

*- TP-Executive Forms Table
copy "$g8wkdir\syscpy\w8tpfft.cpw".

01 user-argument          pic x(010).

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                    w8tp-configuration
                    w8tp-dmiv-tp-configuration
                    w8tp-tp8-configuration
                    w8tp-executive-storage
                    w8tp-tpff-table
                    tp-storage.

*-----*
main section.
section-entry.

    display "FXXX Forms Run-time Manager Initialize".

section-exit.
    exit program.

```

## Service Process

The procedure requests from application TPR's will depend on your current software implementation. Following example illustrates two different service entry points.

```

*-----*
entry "FXXX02" using user-argument.
FXXX02 section.
section-entry.

*   Example of calling sequence if "FXXX01" not in W8PREP table.

move "FXXX01" to ws-call-name

move zero to tpr-cd-status-key-o
move lid to tpr-cd-destination-id
move 100 to tpr-cd-text-length-o
perform send-with-egi.

section-continue.

string "FXXX02"
      " processed"
      delimited by size into journal-log-record
perform trace-exec-call.

section-exit.
exit program.

```

You may use the G8WB interface verification technique (ref. example below), to improve your test of application TPR interfacing.

You are then required to include the service call names in the G8WB Preprocessor conversion tables. These are located under the MFI directory and one conversion table exists for each source and language type (you may have to amend more than one). You may specify a number (01-99) in the alternate part of the conversion table, to force a fixed number of arguments to be passed to your program entry point. If the application is passing less argument, this is reflected and dummy arguments reference are inserted.

*Note: The conversion tables are sorted on keyword within each COBOL division and you must obey this rule when inserting a service call name.*

```

*-----*
entry "FXXX01" using user-argument.
FXXX01 section.
section-entry.

*   Example of calling sequence if "FXXX01" in W8PREP table.

move "FXXX01" to ws-call-name
if w8-arg-no < 1 or
   w8-arg-lgt ( 1 ) <
   length of user-argument
   perform arg-length-error
   go to section-exit
end-if

```

```

        move zero to tpr-cd-status-key-o
        move lid to tpr-cd-destination-id
        move 100 to tpr-cd-text-length-o
        perform send-with-egi.

section-continue.

        if tpr-cd-status-key-o not = zero
            string "FXXX01"
                " status-key ("
                tpr-cd-status-key-o
                ")"
                delimited by size into journal-log-record
            perform trace-exec-call
        else
            string "FXXX01"
                " processed"
                delimited by size into journal-log-record
            perform trace-exec-call
        end-if.

section-exit.
        exit program.
*-----*
arg-length-error section.
section-entry.

        move "F17" to abort-code
        move "Argsiz" to abort-reason
        string ws-call-name
            " Called"
            delimited by size into journal-log-record
        perform trace-exec-call
        perform tp-generated-abort.

section-exit.
        exit.

```

---

## Send & Receive

Following example procedures illustrates how send and receive operation may be implemented within the Forms software Emulation.

```

*-----*
check-buffer-overflow section.
section-entry.

        if msg-ptr > w8tp-mb-size - 100
            perform send-with-por
        end-if.

section-exit.
        exit.
*-----*
send-with-por section.
section-entry.

        if msg-ptr > 1

```

```

        subtract 1 from msg-ptr giving tpr-cd-text-length-o
        move 1 to msg-ptr
    else
        go to section-exit
    end-if

    move w8-por to w8-exi
    move w8-advn to w8-advx
    move tpr-cd-text-length-o to w8-from-lgt
    call "tpSend" using tpr-cd-output
                        message-area
                        w8-send-flg.

section-exit.
    exit.
*-----*
send-with-egi section.
section-entry.

    if msg-ptr > 1
        subtract 1 from msg-ptr giving tpr-cd-text-length-o
        move 1 to msg-ptr
    else
        move zero to tpr-cd-text-length-o
    end-if

    move w8-advn to w8-advx
    move tpr-cd-text-length-o to w8-from-lgt
    call "tpSend" using tpr-cd-output
                        message-area
                        w8-send-flg.

section-exit.
    exit.

```

---

## Trace Service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

```

*-----*
trace-exec-call section.
section-entry.

    if trace-exec-call-on
        write journal-log-record
    end-if
    move spaces to journal-log-record.

section-exit.
    exit.
*-----*
tp-generated-abort section.

string "TP forced .ABORT ("
        abort-code ") reason ("
        abort-reason ")"
        delimited by size into journal-log-record
perform trace-exec-call.

```

```

        move 1 to tpes-tpr-abort-status
            tpes-tpr-cancel-status
            w8-abort-flg.

section-exit.
    exit.
***** END OF PROGRAM *****

```

---

## FORMAT

FORMAT provides a User Exit feature, which allow User Exit routines to be invoked as part of the Forms processing. Such routines are implemented in GMAP on GCOS-8.

The Toolkit for implementation of FORMAT User Exit provides you with a straight forward path for porting of existing User Exit coding. Although existing code is in GMAP, the Micro Focus COBOL/2 system will provide features very similar to possibilities in GMAP.

---

### Concepts

Each User Exit is identified by a single character. All User Exits for validation and editing are handled through one program (W8FMATU), which is automatically enabled whenever FORMAT Forms is configured.

User Exit processing is amended as required by specifying the User Exit character is the related logic processing.

---

### Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message

```

$set directives "$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-W8FMATU.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*              Copyright 1993 by B & C Solution - France  *
*-----*
*      FORMAT User Exit Code Manager      *
*-----*
***** IDENTIFICATION DIVISION *****
    identification division.
    program-id. w8fmatu.

***** ENVIRONMENT DIVISION *****
    environment division.
    input-output section.
    file-control.

    copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****

```

```

data division.
file section.

    copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value         value 0.

*- Workbench Migration Flags
    copy "$g8wkdir\mfi\w8migflg.cpy".

*-----*
linkage section.

*- Receive: Contains field content and length from terminal
*- Send: Contains field content and length to terminal
01 ws-field                pic x(80).
01 ws-length                pic 9(04) comp-5.

01 ws-status                pic 9(01) comp-5.

*- FORMAT Linkage Tables
    copy "$g8wkdir\syscpy\w8falnt.cpw".

*- FORMAT Working Tables
    copy "$g8wkdir\syscpy\w8fawst.cpw".

***** PROCEDURE DIVISION *****
procedure division.
*-----*
main section.
section-entry.

    display "FORMAT User Exit initialized".

section-exit.
    exit program.

```

## Editing User Exit

Following example illustrates how User Exit characters are recognized and how editing is performed during FORMAT send operation.

```

*-----*
entry "W8FMATU_send" using w8a-header
                        w8xa-header
                        w8a-field
                        w8xa-field
                        w8a-field-data
                        ws-length
                        ws-field
                        ws-status.

w8fmatu-send section.
section-entry.

```

```

evaluate w8a-field-user-exit
  when " " *> never used - example move asis
    move w8xa-field-size to ws-length
    move w8a-field-data(1:ws-length) to
      ws-field(1:ws-length)
    *> field will be moved into output buffer
  when other
    move spaces to ws-field
    string ".FORMAT Unknown User Exit code ("
      w8a-field-user-exit
      ")"
      delimited by size
      into ws-field
    move 2 to ws-status
end-evaluate.

section-exit.
  exit program.

```

## Validation User Exit

Following example illustrates how User Exit characters are recognized and how validation is performed during FORMAT receive operation.

```

*-----*
entry "W8FMATU_receive" using w8a-header
                             w8xa-header
                             w8a-field
                             w8xa-field
                             w8a-field-data
                             ws-length
                             ws-field
                             ws-status.

w8fmatu-receive section.
section-entry.

evaluate w8a-field-user-exit
  when " " *> never used - example only
    if ws-field(1:ws-length) = "something-bad"
      move 1 to ws-status *> set not ok
    else
      *> note: field has been moved before call
      move ws-field(1:ws-length) to
        w8a-field-data(1:w8a-field-size)
    end-if
  when other
    move spaces to ws-field
    string ".FORMAT Unknown User Exit code ("
      w8a-field-user-exit
      ")"
      delimited by size
      into ws-field
    move 2 to ws-status
end-evaluate.

section-exit.
  exit program.

***** END OF PROGRAM *****

```

---

## DIMS8

DIMS8 provides a number of software components, which may be customized to meet the installation requirements.

G8WB is delivered with a version of such software components as part of the support Toolkit. These software components may be customized as required or you may download existing versions from your GCOS-8 DIMS8 environment, to obtain a completely equivalent emulation of DIMS8 functionality. Please refer to DIMS8 documentation for details on such software components.



# PRINT Software

---

## Toolkit

The Print software Toolkit is in principle very much similar to DMIV-TP and TP8 User Procedures.

---

### Concepts

A number of user developed Print Management software solutions exist. Although G8WB will not be able to support such user developed software, it does contain a standard Toolkit, which will allow you to port parts or all of your Print Management software and integrate this in the workbench environment.

It may not be required to port all software components, since the development environment, need not reflect all network capabilities. What you do require, is the application interface, collection and a program or tool to locally print the collected output.

An example of such porting activity is the workbench supported DISPATCH8 Distributed Print System.

---

### Application Interface

The application interface is handled through the Print toolkit, which define the service call entry point accessible from the application.

---

### Print Collection

The print reports are collected in separate files (one per report), which are created within a directory structure. The DISPATCH8 solution use the PMS directory of G8WB for collection as follows:

\PMS\Logical-printer\Formcode            Reports with same formcode

\PMS\Logical-printer\Formcode\nnnnnn Report occurrence

Micro Focus Workbench contains callable services, which makes such dynamic directory and file creation quite easy.

---

### Print Utility

The collected report are optionally printed outside the TP environment through a utility, with functionality partially ported from the Print Management software.

---

### Module Initialize

The main entry is called during TP Monitor initialize and this entry should be used to:

- Display an initialize message
- Initialize Common Data Areas
- Open any external files

*Note: Access to Executive Tables will apply to all other entry points after initialize.*

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set sticky-linkage
*-W8PXXX.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*      Copyright 1993 by B & C Solution - France          *
*-----*
*      User Developed Print Management System              *
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. w8pxxx.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

      copy "$g8wkdir\syscpy\w8tpjlse.cpf".

***** DATA DIVISION *****
data division.
file section.

      copy "$g8wkdir\syscpy\w8tpjlfld.cpf".

*-----*
working-storage section.

*- Constants
78      true-value          value 1.

78      false-value        value 0.

*- Work fields

*-----*
linkage section.

*- TP-Executive Configuration
      copy "$g8wkdir\syscpy\w8tpcnfg.cpw".

***** PROCEDURE DIVISION *****
procedure division using w8tp-defaults
                        w8tp-configuration
                        w8tp-dmiv-tp-configuration
                        w8tp-tp8-configuration
                        w8tp-executive-storage.

*-----*
main section.
section-entry.

      display "PXXX Print Interface Manager Initialize".

section-exit.
      exit program.

```

---

## Service Process

The procedure requests from application TPR's will depend on your current software implementation. Following examples illustrate two different service entry points.

```

*-----*
entry "pxxx01".
*
pxxx01-function section.
section-entry.

    string "PXXX01"
           " called from TPR"
           " - ignored"
           delimited by size into journal-log-record
    perform trace-exec-call.

section-exit.
    exit program.
*-----*
entry "pxxx02".
*
pxxx02-function section.
section-entry.

    string "PXXX02"
           " called from TPR"
           " - ignored"
           delimited by size into journal-log-record
    perform trace-exec-call.

section-exit.
    exit program.

```

---

## Trace Service

You are able (recommended) to log information messages to the TP Journal Log, which may be configured independent (ON/OFF) of your program.

```

*-----*
trace-exec-call section.
section-entry.

    if trace-exec-call-on
        write journal-log-record
    end-if
    move spaces to journal-log-record.

section-exit.
    exit.
***** END OF PROGRAM *****

```

---

## DISPATCH8

DISPATCH8 provides a number of software components, which may be customized to meet the installation requirements.

G8WB is delivered with a version of such software components as part of the support Toolkit. These software components may be customized as required or you may download existing versions from your GCOS-8 DISPATCH8 environment, to obtain a completely equivalent emulation of DISPATCH8 functionality. Please refer to DISPATCH8 documentation for details on such software components.

# Preprocessor

---

## Toolkit

The Preprocessor Toolkit may be used to port existing Preprocessor software from GCOS-8 or to development preprocessing features which are unique to the G8WB environment.

---

### Concepts

Micro Focus Workbench provides a standard Preprocessor feature (ref. documentation in Micro Focus Toolset manual), which allow you to integrate one or more preprocessors with the Micro Focus compiling system. When multiple preprocessors are required, these must be stacked. Additional rules apply for stacked preprocessors.

G8WB provides a standard Preprocessor, which translate BULL COBOL-74 and COBOL-85 into a combination of ANS85 and Micro Focus COBOL syntax. INTEREL (SQL) programs are furthermore preprocessed with the XDB DB2 preprocessor, which translate SQL syntax. Use of Micro Focus Reusable Code Manager (RCM) furthermore require preprocessing with the RCM preprocessor.

G8WB provides a standardized method of stacking preprocessors, which make invocation and directive processing more simple and consistent. This method includes a standard for SITE developed preprocessor integration.

Preprocessor stacking takes place in following predefined sequence:

- SITE Preprocessor
- RCM Preprocessor
- XDB Preprocessor
- G8WB Preprocessor
- COBOL compiling system

---

### Enable SITE Preprocessor

A SITE preprocessor must be enabled through the G8WB Preprocessor configuration file. The SITE-PREPROCESSOR directive is used to specify the name of the preprocessor program, i.e. SITEPREP. The preprocessor program must be included in the SITE.LBR (ref. chapter 2).

---

### Source Expansion

A Preprocessor may typically be used for specialized source expansion, i.e non-standard COPY processing. The SITEPREP and W8SELCPY sample preprocessors provides you with a toolkit for implementation of such functionality.

*Note: When replacing the original "COPY verb" with a standard Micro Focus COPY verb, both Checker and Animator will automatically open the file, i.e. Animation is visible and any errors detected will enable Editor to open and place cursor correctly.*

---

### Source Translation

A Preprocessor may also be used to preprocess/translate specialized syntax, i.e. non-COBOL verbs. The SITEPREP and W8SELCPY sample preprocessor does not contain such functionality and you should instead consult the preprocessor samples provided with Micro Focus Workbench.

G8WB Preprocessor will preprocess both original and changed source lines, thus new lines inserted by SITE preprocessor are fully preprocessed by G8WB preprocessor.

## SITEPREP

SITEPREP is an example preprocessor, which has no specific preprocessing logic.

SITEPREP use a two phase technique, where phase 1 is used to perform that actual preprocessing and phase 2 is used to pass the resulting source lines (line by line) to the calling preprocessor.

An alternate technique is to combine the two phases into one, which is more efficient as you save the intermediate file processing. This technique is best suited when one source line may be preprocessed independent of any other source line.

```

$set directives"$g8wbdir\mfi\g8wb.dir"
$set fcdreg
*-SITEPREP.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.1.00      *
*
*-----*
*
*      Example of Site Preprocessor.
*
*      Currently no example function has been implemented.
*
*      A work file (program.SWK) is used for complete expansion
*      of source (Phase 1), before source lines are passed via
*      other preprocessors to the Micro Focus Checker.
*
*      Note: Program.SWK file is deleted or kept in accordance
*            with standard WORK-FILE configuration.
*
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. siteprep.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

        select source-file assign to disk source-file-name
           organization is line sequential.

        select work-file assign to work-file-name
           organization is line sequential.

        select console-file assign to "con:"
           organization is line sequential.

***** DATA DIVISION *****
data division.
file section.

fd      source-file
        data record is source-line.
01      source-line.
03      source-column-1-6          pic x(006).

```

```

03    source-column-7          pic x(001).
03    source-column-8-80      pic x(073).

fd    work-file
      data record is work-linex.
01    work-linex.
03    work-main              pic 9(02) comp-x.
03    work-more              pic 9(02) comp-x.
03    work-line.
05    work-column-1-6        pic x(06).
05    work-column-7-80.
07    work-column-7          pic x(01).
07    work-column-8-80      pic x(73).

fd    console-file
      data record is console-line.
01    console-line          pic x(80).

*-----*
working-storage section.

*-  Constants
78    true-value             value 1.
78    false-value           value 0.

78    sound-beep            value X"E5".
78    keybd-read            value X"83".

01    temp-x                pic x(01).

01    source-file-name      pic x(255).
01    source-file-status    pic 9(001) comp-5.
01    work-file-name        pic x(255).
01    work-file-status      pic 9(001) comp-5.

01    ws-command-line       pic x(255).
01    ws-command-length     pic 9(008) comp-5
                             value 255.

01    ws-time.
05    ws-time-hh            pic 9(02).
05    ws-time-mm            pic 9(02).
05    ws-time-ss            pic 9(02).
05    ws-time-hs            pic 9(02).
01    ws-time-seconds       pic 9(08) comp-5.
01    phase-start-time      pic 9(08) comp-5.
01    phase-stop-time       pic 9(08) comp-5.
01    phase-elapsed-time    pic 9(04).

01    ws-number-lines       pic 9(04) comp-5.
01    phase-number-lines    pic 9(05).

78    mf-main-original-line value 32.
78    mf-main-start-copy    value 3.
78    mf-main-continue-copy value 4.
78    mf-main-end-copy      value 128.
78    mf-main-new-line      value 1.
78    mf-main-delete-line   value 2.
78    mf-main-warning       value 5.
78    mf-main-error         value 7.
78    mf-main-fatal-error   value 6.

```

```

78          mf-main-end                value 0.

*-   Dynamic File Handling
      copy "$g8wkdir\syscpy\w8xpyfn.cpw".

*-----*
linkage section.

01      mf-function                    pic 9(002) comp-x.

01      mf-line.
05      mf-column-1-6                  pic x(06).
05      mf-column-7-80.
05      mf-column-7                    pic x(01).
05      mf-column-8-80                 pic x(73).

01      mf-response.
05      mf-response-status             pic 9(002) comp-x.
05      mf-response-code-1            pic 9(004) comp-x.
05      filler redefines mf-response-code-1.
05      filler                         pic x.
05      mf-response-main               pic 9(002) comp-x.
05      mf-response-code-2            pic 9(004) comp-x.
05      filler redefines mf-response-code-2.
05      filler                         pic x.
05      mf-response-more              pic 9(002) comp-x.
***** PROCEDURE DIVISION *****
procedure division using mf-function
                                mf-line
                                mf-response.

siteprep-main section.
section-entry.
module-entry.

      move zero to mf-response-status.
      if mf-function = 0          *> Start Phase 1
          move zero to ws-number-lines
          perform get-current-time
          move ws-time-seconds to phase-start-time
          perform prep-phase-1
          move ws-number-lines to phase-number-lines
          perform get-current-time
          move ws-time-seconds to phase-stop-time
          compute phase-elapsed-time = phase-stop-time -
                                phase-start-time
          string "* SITE: " phase-elapsed-time
            " seconds elapsed time "
            phase-number-lines
            " lines processed (Phase 1)"
            delimited by size
            into console-line
          perform write-console
          move phase-stop-time to phase-start-time
      else
          perform prep-phase-2      *> Phase 2
      end-if.

module-exit.
      exit program.

```

## Prepare Phase 1

Prepare phase 1 is used to allocate the original source file and check for errors in relation to this.

*Note: SITE preprocessor, being last in the preprocessor stack, is required to open the original source file. All other stacked preprocessor are reading source from intermediate result files.*

```

*-----*
prep-phase-1 section.
section-entry.

    open output console-file
    move spaces to console-line

    string "* SITE: Preprocessor (transparent)"
        delimited by size
        into console-line
    perform write-console

    perform process-directives

    move mf-line to w8xpdyfn-filespec
    perform w8xpdyfn-split-filename
    move "CBL" to w8xpdyfn-extension
    perform w8xpdyfn-join-filename
    perform w8xpdyfn-check-file-exist
    if w8xpdyfn-status-ok
        move w8xpdyfn-filespec to source-file-name
        open input source-file
        move false-value to source-file-status
        perform read-source-line
        perform w8xpdyfn-split-filename
        move "SWK" to w8xpdyfn-extension
        perform w8xpdyfn-join-filename
        move w8xpdyfn-filespec to work-file-name
        open output work-file
        perform process-phase-1
            until source-file-status = true-value
        close source-file
            work-file
        open input work-file
        move false-value to work-file-status
    else
        display "*-> Cannot access "
            w8xpdyfn-filespec
        display "SITEPREP Preprocessor Aborted"
        move 255 to mf-response-status
    end-if.

section-exit.
exit.

```

## Directives

You may use various techniques to control the SITE preprocessing via configuration and directives.

G8WB preprocessor will pass any "**SET>SITE**" directives defined in the source file via command line.

You may open your own configuration file, i.e. SITE.CFG and read configuration information to determine setting.

You may include your own keywords in the PREP.CFG file and read these to determine setting.

*Note: W8SELCPY provides examples of configuration file access.*

```
*-----*
process-directives section.
section-entry.

    accept ws-command-line from command-line
    call "CBL_TOUPPER" using ws-command-line by value
                                ws-command-length.

    *> Include logic to process directives here

section-exit.
    exit.
```

---

## Process Phase 1

Phase 1 is used to preprocess the entire source and write the result to an intermediate file.

*Note: Errors detected are simply written to the intermediate file and later processed in phase 2. This technique allow the developer to see all errors resulting from the preprocessing. Exception being fatal errors, which prevent any further preprocessing.*

```
*-----*
process-phase-1 section.
section-entry.

    if source-file-status = false-value

    *> Place additional preprocessing code here

        write work-linex
        perform read-source-line
    end-if.

section-exit.
    exit.
```

```
*-----*
read-source-line section.
section-entry.

    read source-file
    at end
        move true-value to source-file-status
        move spaces to work-line
        move mf-main-end to work-main
        move 0          to work-more
    not at end
        add 1 to ws-number-lines
```

```

        move source-line to work-line
        move mf-main-original-line to work-main
        move 0                to work-more
    end-read.

section-exit.
exit.

```

---

## Phase 2

Phase 2 is used to pass the preprocessed source lines to the caller and to communicate errors detected during preprocessing.

```

*-----*
prep-phase-2 section.
section-entry.

    read work-file
    at end
        move 0 to mf-response-main
        move 0 to mf-response-more
        move true-value to work-file-status
    not at end
        evaluate work-main
            when mf-main-warning
                perform setup-line    *> Documented later
            when mf-main-error
                move 2 to work-more    *> Error level
                perform setup-line
            when mf-main-fatal-error
                perform fatal-line    *> Stop right here
                perform setup-line
            when other
                perform setup-line
        end-evaluate
    end-read.

section-exit.
exit.
*-----*

setup-line section.
section-entry.

    move work-line to mf-line
    move 0 to mf-response-code-1
        mf-response-code-2
    move work-main to mf-response-main
    move work-more to mf-response-more.

section-exit.
exit.
*-----*

fatal-line section.
section-entry.

    move work-column-8-80 to console-line
    perform write-console

```

```

    move "FATAL ERROR - press any key to return to Editor" to
        console-line
    perform write-console
    call sound-beep
    call keybd-read using temp-x
    display work-column-8-80 upon command-line.

section-exit.
    exit.
*-----*
get-current-time section.
section-entry.

    accept ws-time from time
    compute ws-time-seconds = ws-time-ss +
                                ( ws-time-mm * 60 ) +
                                ( ws-time-hh * 3600 ).

section-exit.
    exit.
*-----*
write-console section.
section-entry.

    write console-line
    move spaces to console-line.

section-exit.
    exit.

*-   Dynamic File Handling
    copy "$g8wkdir\syscpy\w8xpdyfn.cpp".

***** END OF PROGRAM *****

```

---

## \$\$Select

GCOS8 \$\$SELECT handling is implemented as a SITE preprocessor. The source has been included as part of the toolkit to provide you with examples of configuration file handling and the use of stacked files (same FD used to read mutiple file, i.e nested \$\$SELECT's).

## Other Software

### Soundex

The Soundex Toolkit is in an emulation of the SOUNDEX routines developed by BULL Norway. The functionality is currently limited to a dummy emulation with simple "Move-and truncate" logic.

#### Concepts

SOUNDEX operate on the basis of one input argument, which may be logically divided in three parts, and one output argument, which may be logically divided in three parts also. The logical division of input/output argument is directly related to the name of called SOUNDEX routine.

The SOUNDEX Toolkit is implemented as one program with multiple entry points, where each entry point correspond to the equivalent SOUNDEX routine on GCOS-8. No Toolkit initialize is required and the main entry (SOUNDX) may in fact be used as an entry point also.

Input argument is (currently) limited to a maximum of 500 characters. Output argument is (currently) limited to 100 characters.

#### SOUNDX Program

```

$set directives"$g8wkdir\mfi\g8wb.dir"
$set nobound
$set sticky-linkage
*-SOUNDX.CBL-----*
*
*      GCOS-8 COBOL Workbench          Version 1.0.07          *
*
*                               Copyright 1993 by B & C Solution - France *
*-----*
*      BULL Soundex interface          *
*
*-----*
***** IDENTIFICATION DIVISION *****
identification division.
program-id. soundx.

***** ENVIRONMENT DIVISION *****
environment division.
input-output section.
file-control.

***** DATA DIVISION *****
data division.
file section.

*-----*
working-storage section.

*- Constants
78      true-value          value 1.
78      false-value         value 0.

*- Work
01      1-ettnavn           pic 9(04) comp-5.

```

```

01      l-fornavn                pic 9(04) comp-5.
01      l-melnavn                pic 9(04) comp-5.
01      l-settnavn               pic 9(04) comp-5.
01      l-sfornavn               pic 9(04) comp-5.
01      l-smelnavn               pic 9(04) comp-5.

*-   Workbench Migration Flags
      copy "$g8wbdir\mfi\w8migflg.cpy".
*-----*

linkage section.

01  soundex-input                pic x(500).

01  soundex-output               pic x(100).

01  soundex-spraak              pic x(02).
***** PROCEDURE DIVISION *****
procedure division using soundex-input
                        soundex-output
                        soundex-spraak.
*-----*

main section.
section-entry.

      move 20 to l-ettnavn *> Set length of input
      move 20 to l-fornavn
      move 20 to l-melnavn
      move 12 to l-settnavn *> Set length of output
      move 04 to l-sfornavn
      move 00 to l-smelnavn
      perform make-soundex.

section-exit.
      exit program.
*-----*

make-soundex section.
section-entry.

      move spaces to
          soundex-output(1:l-settnavn + l-sfornavn + l-smelnavn)

      if w8-arg-no not = 3          *> Must have three arguments
          display "*-> SOUNDEX: "
              "Incorrect number of arguments"
          go to section-exit
      end-if
      if w8-arg-lgt ( 3 ) < 2      *> Check third argument length
          display "*-> SOUNDEX: "
              "Third argument length error"
          go to section-exit
      end-if
      if w8-arg-lgt ( 1 ) <        *> Check input length
          ( l-ettnavn + l-fornavn + l-melnavn )
          display "*-> SOUNDEX: "
              "First argument length error"
          go to section-exit
      end-if
      if w8-arg-lgt ( 2 ) <        *> Check output length
          ( l-settnavn + l-sfornavn + l-smelnavn )
          display "*-> SOUNDEX: "
              "Second argument length error"
          go to section-exit

```

```

end-if

if l-ettnavn > zero and      *> Dummy soundex
  l-settnavn > zero
  move soundex-input(1:) to
    soundex-output(1:l-settnavn)
end-if
if l-fornavn > zero and      *> Dummy soundex
  l-sfornavn > zero
  move soundex-input(1 + l-ettnavn:) to
    soundex-output(1 + l-settnavn:l-sfornavn)
end-if
if l-melnavn > zero and     *> Dummy soundex
  l-smelnavn > zero
  move soundex-input(1 + l-ettnavn + l-fornavn:) to
    soundex-output(1 + l-settnavn +
      l-sfornavn:l-smelnavn)
end-if.

section-exit.
exit.

```

---

## SOUNDX Entry-point

Following illustrates unique SOUNDEX entry points, each having their implicit sub-division rules clearly described in the source. You may add new/change existing entry points and specifications as required.

*Note: SOUNDX does a logic check on the calling arguments. Any conflict with the implicit sub-division rules are documented as a debugging aid.*

```

*-----*
entry "SDX001" using soundex-input
                    soundex-output
                    soundex-spraak.

main section.
section-entry.

    move 20 to l-ettnavn *> Set length of input
    move 20 to l-fornavn
    move 20 to l-melnavn
    move 12 to l-settnavn *> Set length of output
    move 04 to l-sfornavn
    move 00 to l-smelnavn
    perform make-soundex.

section-exit.
exit program.

```

```

*-----*
entry "SDX002" using soundex-input
                    soundex-output
                    soundex-spraak.

main section.
section-entry.

    move 20 to l-ettnavn *> Set length of input
    move 20 to l-fornavn

```

```
move 20 to l-melnavn  
move 12 to l-settnavn *> Set length of output  
move 04 to l-sfornavn  
move 00 to l-smelnavn  
perform make-soundex.
```

```
section-exit.
```

```
exit program.
```

