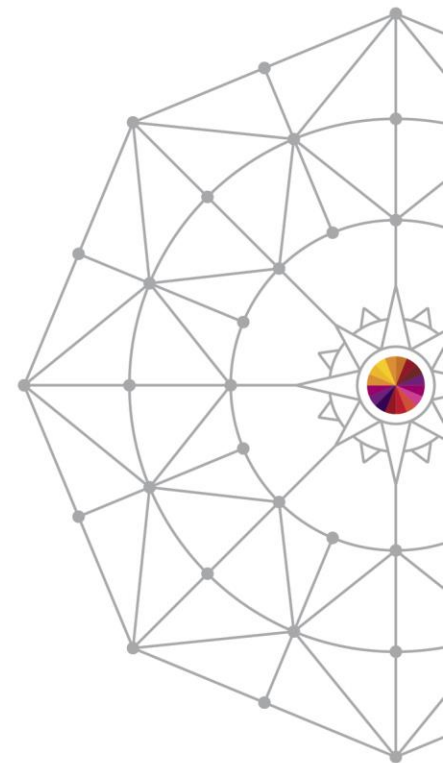


# GTF trace – Swiss army knife in MVS professional's toolbox

*Vlastimil Jedek*  
*CA Technologies*

*August 5, 2014*  
*16147*

[www.SHARE.org](http://www.SHARE.org)



#SHAREorg



# Disclaimer

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. IBM, System z, zEnterprise and z/OS are trademarks of International Business Machines Corporation in the United States, other countries, or both. *Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries* All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

**THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY.** CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages.

# Agenda

- What is GTF?
- Starting and stopping GTF
- Customizing GTF output
- Processing GTF output in IPCS
- Using GTF with SLIP
- Summary
- Reference

# What is GTF?

- GTF stands for **G**eneralized **T**race **F**acility
- It is a service aid that can:
  - Record many different system and program events
  - Greatly help in system and/or program problem diagnosis
- Part of MVS

# What is GTF? (Continued)

- GTF records historical data
- It does not run by default, needs to be activated
- Typically GTF is used for limited periods of time to analyze the specific problem
- Needs to be tailored to record the desired data
- Multiple instances of GTF can be active at the same time

# What you can trace with GTF?

- Channel programs and associated data for various subchannel operations
- I/O interruptions on specific device(s)
- SVC calls
- System recovery routine operations
- Program execution including the contents of registers and storage

# What is GTF good for?

- Identify a sequence of system events leading to a problem
- Confirm a failing scenario
- Improve understanding of how your (or somebody else's) code works and what it is doing

# Comparison to other debugging tools

- Unlike dumps, which contain information for a certain point in time, GTF records historical data
- In contrast to a timeframe of data recorded by SYSTRACE, GTF can record data for significantly longer time periods
- XDC can also be used to track program execution. Unlike XDC, GTF is part of the system – available everywhere



# Steps for obtaining GTF trace

1. Start **GTF** with the trace options that you want
2. Recreate the problem or event(s) to be recorded in the trace
3. Stop **GTF**
4. Format the trace data in **IPCS**

# How to start GTF

- GTF is started using MVS START command:  
**S gtf\_procedure\_name.identifier**
  - **gtf\_procedure\_name** is the name of GTF procedure in **SYS1.PROCLIB**
  - **identifier** is name is 1-8 chars long name of specific GTF session
- Default IBM-supplied GTF procedure is stored in **SYS1.PROCLIB(GTF)**
- Uses default IBM-supplied GTF trace options stored in **SYS1.PARMLIB(GTFPARM)**

# GTF procedure

## Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                Line 00000000 col
Command ==>                                         Scroll ==
***** Top of Data *****
//GTF       PROC MEMBER=GTFPARM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//  TIME=1440,REGION=2880K
//IEFRDOR DD  DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB  DD  DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- Calls **AHLGTF** program to collect the trace

# GTF procedure

## Default IBM-supplied GTF procedure:

```

BROWSE      SYS1.PROCLIB(GTF) - 01.04                               Line 00000000 col
Command ==>                                                              Scroll ==
***** Top of Data *****
//GTF      PROC MEMBER=GTFPARM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//  TIME=1440,REGION=2880K
//IEFRDER DD  DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB  DD  DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****

```

- Specifies general parameters to control GTF operation
  - **MODE** specifies where GTF should store trace data
    - EXT » direct data to data set specified by IEFRDER
    - INT » stores data in trace table in virtual storage
    - DEFER » stores data in trace table, writes the data to the data set after the trace is stopped

# GTF procedure

## Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                      Line 00000000 col
Command ==>                                           Scroll ==
***** Top of Data *****
//GTF      PROC MEMBER=GTFPARM
//IEFPROC EXEC PGM=AHLGTF , PARM='MODE=EXT , DEBUG=NO , TIME=YES ' ,
//  TIME=1440 , REGION=2880K
//IEFRDOR DD  DSNAME=SYS1.TRACE , UNIT=SYSDA , SPACE=(TRK , 20) , DISP=(NEW , KEEP)
//SYSLIB  DD  DISP=SHR , DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- Specifies general parameters to control GTF operation
  - **DEBUG** specifies whether GTF attempts error recovery
    - NO » GTF attempts to recover from an error
    - YES » no recovery, GTF stops after encountering an error

# GTF procedure

## Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                Line 00000000 col
Command ==>                                         Scroll ==
***** Top of Data *****
//GTF       PROC MEMBER=GTFPARM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//  TIME=1440,REGION=2880K
//IEFRDTER DD  DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB  DD   DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- Specifies general parameters to control GTF operation
  - **TIME = YES** specifies that all GTF trace records will be time-stamped

# GTF procedure - region

Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                Line 00000000 col
Command ==>                                         Scroll ==
***** Top of Data *****
//GTF        PROC MEMBER=GTFPARM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
// TIME=1440, REGION=2880K
//IEFRDER DD  DSN=SYS1.TRACE, UNIT=SYSDA, SPACE=(TRK,20), DISP=(NEW,KEEP)
//SYSLIB DD   DISP=SHR, DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- **REGION** specifies maximum storage required by GTF
- This value can be from 832K to 2880K
- Details on how to calculate the storage requirements for GTF are available in the [z/OS MVS Diagnosis: Tools and Service Aids](#) manual

# GTF procedure - IEFORDER

Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                      Line 00000000 col
Command ==>                                           Scroll ==
***** Top of Data *****
//GTF      PROC MEMBER=GTFPARAM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//  TIME=1440,REGION=2880K
//IEFRDER DD  DSN=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB DD  DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- **IEFRDER DD** defines trace output data set when MODE=EXT or MODE=DEFER
- Each GTF instance needs to use its own data set
- When the data set is full, GTF starts writing at the beginning of the data set again, overwriting any previously written data



# GTF procedure - IEFORDER

Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                Line 00000000 col
Command ==>                                         Scroll ==
***** Top of Data *****
//GTF       PROC MEMBER=GTFPARAM
//IEFPROC   EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//          TIME=1440,REGION=2880K
//IEFRDER   DD      DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB    DD      DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- For pre-allocated GTF data sets, it is recommended to use:

**LRECL=27994**

**BLOCKSIZE=27998**

**RECFM=VB**

# GTF procedure - SYSLIB

Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PROCLIB(GTF) - 01.04                Line 00000000 col
Command ==>                                         Scroll ==
***** Top of Data *****
//GTF      PROC MEMBER=GTFPARAM
//IEFPROC EXEC PGM=AHLGTF, PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//  TIME=1440,REGION=2880K
//IEFRDER DD  DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(TRK,20),DISP=(NEW,KEEP)
//SYSLIB  DD  DISP=SHR,DSN=SYS1.PARMLIB(&MEMBER)
***** Bottom of Data *****
```

- **SYSLIB DD** provides a member with GTF trace options
  - » Specifies **what** will be traced

# GTF procedure - SYSLIB

Default IBM-supplied GTF procedure:

```
BROWSE      SYS1.PARMLIB(GTFPARM) - 01.02                Line 00000000 col
Command ===>                                           Scroll ==
***** Top of Data *****
TRACE=SYSM,USR,TRC,DSP,PCI,SRM
***** Bottom of Data *****
```

- Specifies **what** will be traced
- Does not make much sense to use the default values, values need to be tailored to the specific situation
- Details on various GTF trace options are available in the [z/OS MVS Diagnosis: Tools and Service Aids](#) manual

# GTF trace options - interruptions

- SVC or SVCP** – recording for SVC interruptions
- EXT** – recording for all external interruptions
- IO or IOP** – recording of non-program-controlled I/O interrupts
- IOX or IOXP** – recording of non-program-controlled I/O interrupts providing a summary of a complete channel program
- PCI** – recording of intermediate status interruptions
- PCIE** – requests tracing of PCI load and store instructions, adapter interrupts, and PCIE de-multiplexing request
- PFIDP** – limits GTF tracing of PCIE-related events to a subset of the PCIE function identifiers (PFIDs)
- PI or PIP** – comprehensive recording for program interruptions

# GTF trace options - subchannel operations

- CCW or CCWP** – requests tracing of channel programs and associated data for I/O events
- CSCH** – recording for clear subchannel operations
- HSCH** – recording for halt subchannel operations
- SSCH or SSCHP** – recording for start subchannel and resume subchannel operations
- MSCH** – recording for modify subchannel operations
- XSCH** – recording for cancel subchannel operations

# GTF trace options

## DSP

- requests recording for all dispatchable units of work: SRB, LSR, TCB, and SVC prolog dispatch events

## RNIO

- requests recording of all VTAM network activity

## RR

- requests comprehensive recording of data associated with all invocations of recovery routines (such as STAE and ESTAE)

## SRM

- requests recording of trace data each time the system resource manager (SRM) is invoked

# GTF trace options

- ASIDP** – limits GTF tracing to a subset of address spaces
- JOBNAMEP** – limits GTF tracing to a subset of jobs
- SLIP** – requests trace entry be made when:
- a match occurs for a SLIP trap with ACTION=TRACE
  - a SLIP trap with the SLIP DEBUG option is checked
- TRC** – tracing for events that are associated with GTF itself
- USR or USRP** – recording the data that GTRACE macro passes to GTF

# GTF trace options – SYS, SYSM and SYSP

## **SYS**

- requests recording of comprehensive trace data for:
  - clear subchannel operations
  - external interruptions
  - halt subchannel operations
  - I/O interruptions
  - modify subchannel operations
  - program interruptions
  - recovery routines
  - start subchannel and resume channel operations
  - SVC interruptions

## **SYSM**

- records minimal trace data for the same events as SYS

## **SYSP**

- requests recording for the same events as SYS, but prompts for specific SVC, IO, SSCH, and PI events



# Starting GTF – example 1

```

06:08:04.21 JEDVL01 00000290 S GTF.EXAMPLE1
06:08:04.24          00000090 IRR812I PROFILE *.* (G) IN THE STARTED CLASS WAS USED 925
                   925 00000090          TO START GTF WITH JOBNAME GTF.
06:08:04.92 STC00159 00000094 $HASP100 GTF          ON STCINRDR
06:08:05.15 STC00159 00000290 IEF695I START GTF          WITH JOBNAME GTF          IS ASSIGNED TO USER IBMUSER
                   , GROUP SYS1
06:08:05.15 STC00159 00000281 $HASP373 GTF          STARTED
06:08:05.16 STC00159 00000090 IEF403I GTF - STARTED - TIME=06.08.05
06:08:05.16 STC00159 00000090 AHL121I TRACE OPTION INPUT INDICATED FROM MEMBER GTFPARM OF PDS
                   SYS1.PARMLIB
06:08:05.23 STC00159 00000090 TRACE=SYSM,USR,TRC,DSP,PCI,SRM
06:08:05.23 STC00159 00000090 AHL103I TRACE OPTIONS SELECTED --SYSM,USR,TRC,DSP,PCI,SRM
06:08:05.23 STC00159 00000090 *0088 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
06:08:10.65 JEDVL01 00000290 R 88,U
06:08:10.66 STC00159 00000090 IEE600I REPLY TO 0088 IS;U
06:08:10.66 STC00159 00000090 U
06:08:10.69 STC00159 00000090 AHL906I THE OUTPUT BLOCK SIZE OF          27920 WILL BE USED FOR OUTPUT 938
                   938 00000090          DATA SETS:
                   938 00000090          SYS1.TRACE
06:08:10.69 STC00159 00000090 AHL080I GTF STORAGE USED FOR GTF DATA: 939
                   939 00000090          GTFBLOCK STORAGE          81K BYTES (BLOK=          40K)
                   939 00000090          PRIVATE STORAGE          1036K BYTES (SIZE=          1024K)
                   939 00000090          SADMP HISTORY          54K BYTES (SADMP=          40K)
                   939 00000090          SDUMP HISTORY          54K BYTES (SDUMP=          40K)
                   939 00000090          ABEND DUMP DATA          0K BYTES (ABDUMP=          0K)
06:08:10.69 STC00159 00000090 AHL031I GTF INITIALIZATION COMPLETE

```

# Starting GTF – considerations

- When starting GTF, **AHL125A** message is normally issued to allow GTF trace options to be changed before GTF is started (unless the NOPROMPT parameter was specified)
- If no change is desired, just reply **U**
- If you want to change trace option, respond with  
**R xx,TRACE={new\_gtf\_option1,new\_gtf\_option2,...}**
- Some trace options will prompt you to specify additional parameters

# Starting GTF – example 2

```

03:21:37.73 JEDVL01 00000290 S GTF.EXAMPLE2
...
03:21:37.81 STC00183 00000090 AHL121I TRACE OPTION INPUT INDICATED FROM MEMBER GTFPARM OF PDS
SYS1.PARMLIB
03:21:37.82 STC00183 00000090 TRACE=SYSM,USR,TRC,DSP,PCI,SRM
03:21:37.82 STC00183 00000090 AHL103I TRACE OPTIONS SELECTED --SYSM,USR,TRC,DSP,PCI,SRM
03:21:37.82 STC00183 00000090 *0104 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
03:21:47.52 JEDVL01 00000290 R 104,TRACE=IO,SVCP
03:21:47.52 STC00183 00000090 IEE600I REPLY TO 0104 IS;TRACE=IO,SVCP
03:21:47.52 STC00183 00000090 TRACE=IO,SVCP
03:21:47.52 STC00183 00000090 *0105 AHL101A SPECIFY TRACE EVENT KEYWORDS --SVC=
03:22:09.66 JEDVL01 00000290 R 105,SVC=(18,21)
03:22:09.67 STC00183 00000090 IEE600I REPLY TO 0105 IS;SVC=(18,21)
03:22:09.67 STC00183 00000090 SVC=(18,21)
03:22:09.67 STC00183 00000090 *0106 AHL102A CONTINUE TRACE DEFINITION OR REPLY END
03:22:21.99 JEDVL01 00000290 R 106,END
03:22:21.99 STC00183 00000090 IEE600I REPLY TO 0106 IS;END
03:22:22.00 STC00183 00000090 END
03:22:22.00 STC00183 00000090 AHL103I TRACE OPTIONS SELECTED --IO,SVC=(18,21)
03:22:22.00 STC00183 00000090 *0107 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
03:22:26.35 JEDVL01 00000290 R 107,U
03:22:26.35 STC00183 00000090 IEE600I REPLY TO 0107 IS;U
03:22:26.36 STC00183 00000090 U
...
03:22:26.37 STC00183 00000090 AHL031I GTF INITIALIZATION COMPLETE

```

# Stopping GTF

```
06:13:36.74 JEDVL01 00000290 P GTF.EXAMPLE2
06:13:36.74 STC00159 00000090 AHL006I GTF ACKNOWLEDGES STOP COMMAND
06:13:36.74 STC00159 00000090 AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA : 982
                                982 00000090                                SYS1.TRACE
06:13:36.75 STC00159 00000090 IEF404I GTF - ENDED - TIME=06.13.36
06:13:36.75 STC00159 00000285 $HASP395 GTF          ENDED
```

# Starting and stopping GTF – identifier

- If GTF identifier is omitted, and the trace MODE parameter is set to MODE=EXT or MODE=DEFER, the identifier is created automatically as device address.
- You can find the identifier from output of the **D A,L** operator command.
- The identifier is needed to stop GTF.

# Starting and stopping GTF – identifier

```

06:52:09.72 JEDVL01 00000290 S GTF
...
06:52:11.36 STC00164 00000090 TRACE=SYSM,USR,TRC,DSP,PCI,SRM
06:52:11.36 STC00164 00000090 AHL103I TRACE OPTIONS SELECTED --SYSM,USR,TRC,DSP,PCI,SRM
06:52:11.36 STC00164 00000090 *0091 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
06:52:15.12 JEDVL01 00000290 R 91,U
...
06:52:15.15 STC00164 00000090 AHL031I GTF INITIALIZATION COMPLETE
06:52:20.45 JEDVL01 00000290 D A,L
06:52:20.45 JEDVL01 00000090 IEE114I 06.52.20 2014.142 ACTIVITY 938
          938 00000090 JOBS      M/S      TS USERS      SYSAS      INITS      ACTIVE/MAX VTAM      OAS
          938 00000090 00004      00037      00001      00045      00015      00001/00050      00026
          938 00000090 LSERV      LSERV      LSERV      NSW S      HZSPROC      HZSPROC      HZSSTEP      NSW SO
          ...
          938 00000090 OPERAVL    OPERAVL    CAIWCASS    NSW S      SYSVQA      SYSVQA      SYSVIEW      NSW SO
          938 00000090 SYSVQAUR   SYSVQAUR   SYSVIEW     NSW SO     PMO         PMO         PMO44        NSW S
          938 00000090 GTF        299B      IEFPROC     NSW S
          938 00000090 JEDVL01 IN
06:52:28.42 JEDVL01 00000290 P GTF.299B
06:52:28.42 STC00164 00000090 AHL006I GTF ACKNOWLEDGES STOP COMMAND
06:52:28.42 STC00164 00000090 AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA : 941
          941 00000090 SYS1.TRACE
06:52:28.43 STC00164 00000090 IEF404I GTF - ENDED - TIME=06.52.28
06:52:28.43 STC00164 00000285 $HASP395 GTF ENDED

```

# Processing GTF output in IPCS

- GTF can be formatted using IPCS
- Data set with GTF data is accessed in IPCS the same way as dumps
- It is important to format GTF output in the matching version of IPCS to prevent confusing formatting problems!
- Unlike dumps, IPCS does not warn you if GTF data was obtained on a system with a different z/OS release!

# Processing GTF output in IPCS – z/OS release

- To determine the z/OS release of a system where GTF output was obtained:

```

DSLISL - Data Sets Matching JEDVL01.GTF*                               Row 1 of 3
Command ===>                                                            Scroll ===> HALF

Command - Enter "/" to select action                                Message                                Volume
-----
 B   JEDVL01.GTFTRACE                                               PIT008
     JEDVL01.GTFTRAC2                                               PIT008
     JEDVL01.GTFTRAC3                                               MVSD22
***** End of Data Set list *****

```

- Do BROWSE on a data set containing GTF output



# Processing GTF output in IPCS – z/OS release

- To determine the z/OS release of a system where GTF output was obtained:

```

BROWSE      JEDVL01.GTFTRACE
Command ==>
***** Top of Data *****
....$ò./Uö¥./.....GTS .. SP7.1.3 HBB7780 XE23 ..(...ø.....$.
..ò.Â.ð]/& ..Ü....FDR4CPY EXP4. FDRCOPY .'l..ñ...-..FDRXPASS.....N.â..½ N/A      N/A      N/A
..ò.Â.ð]"& .....DDD.....D#PIT005PIT005          PIT.RCS233.LOAD
..ò.Â.ð]a& .....
..ò.Â.ð]µ& .....
..ò.Â.ð]~& .....s.w.á..áµMì.~...á^ôqáµMì.~...áµMì.~...K.s.tEá..áµMì.~...K.s.tøá..áµM
..ò.Â.ð]t& .....&.o...â.Kµ.ôî...j.-!j.-$ø..Àø..Áø..Ãø..Àø..>ø
..ò.Â.ð]x& .....
..ò.Â.ð´yÀ ..Ü....FDR4CPY EXP4. FDRCOPY .'l..ñ...-..FDRXPASS...½.....NH°ö}. N/A      N/A      N/A
..ò.Â.ð´§À .....DDD.....D#PIT005PIT005          PIT.RCS233.LOAD
..ò.Â.ð´¼À .....
..ò.Â.ð´MÀ .....

```

- Find character string in the beginning starting with “SP7.”

# Processing GTF output in IPCS – z/OS release

- To determine the z/OS release of a system where GTF output was obtained:

```

BROWSE      JEDVL01.GTFTRACE
Command ==>
***** Top of Data *****
....$ò./Uö¥./.....GTS .. .SP7.1.3 HBB7780 XE23 ..(...Ø.....$.
..ò.Â.ð]/& ..Ü....FDR4CPY EXP4. FDRCOPY .'1..ñ...FDRXPASS.....N.â..½ N/A      N/A      N/A
..ò.Â.ð]"& .....DDD.....D#PIT005PIT005          PIT.RCS233.LOAD
  
```

z/OS release

- For z/OS 2.1 the value is SP7.2.1
- For z/OS 1.13 the value is SP7.1.3
- For z/OS 1.12 the value is SP7.1.2
- For z/OS 1.11 the value is SP7.1.1

System name

**!** GTF output needs to be formatted in matching versions of IPCS, **!**  
 ● otherwise confusing formatting problems may (and probably will) occur.

# Processing GTF output in IPCS

- To assess GTF data specify option '**0 DEFAULTS**' from the IPCS primary menu:

```

----- z/OS 01.13.00 IPCS PRIMARY OPTION MENU -----
OPTION  ===>

0  DEFAULTS  - Specify default dump and options
1  BROWSE    - Browse dump data set
2  ANALYSIS  - Analyze dump contents
3  UTILITY   - Perform utility functions
4  INVENTORY - Inventory of problem data
5  SUBMIT    - Submit problem analysis job to batch
6  COMMAND   - Enter subcommand, CLIST or REXX exec
T  TUTORIAL  - Learn how to use the IPCS dialog
X  EXIT      - Terminate using log and list defaults

*****
* USERID   - JEDVL01
* DATE     - 14/06/07
* JULIAN    - 14.158
* TIME     - 16:05
* PREFIX   - JEDVL01
* TERMINAL - 3278
* PF KEYS  - 24
*****

Enter END command to terminate IPCS dialog

```

# Processing GTF output in IPCS

- Specify the name of the data set with GTF data in **Source** field:

```
----- IPCS Default Values -----  
Command ==>  
  
You may change any of the defaults listed below.  The defaults shown before  
any changes are LOCAL.  Change scope to GLOBAL to display global defaults.  
  
Scope ==> LOCAL  (LOCAL, GLOBAL, or BOTH)  
  
If you change the Source default, IPCS will display the current default  
Address Space for the new source and will ignore any data entered in  
the Address Space field.  
  
Source ==> DSNAME('JEDVL01.GTFTRACE')  
Address Space ==>  
Message Routing ==> NOPRINT TERMINAL NOPDS  
Message Control ==> NOCONFIRM VERIFY FLAG(WARNING)  
Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL  
  
Press ENTER to update defaults.  
  
Use the END command to exit without an update.
```

# Processing GTF output in IPCS

- There are 4 IPCS commands related to processing GTF output:
  - **IPCS GTFTRACE**
    - formats GTF trace records
  - **IPCS COPYTRC**
    - copies trace entries or records
  - **IPCS MERGE** and **MERGEEND**
    - merges multiple traces
- IPCS subcommands are documented in the [z/OS MVS IPCS Commands](#) manual

# Processing GTF output in IPCS - GTFTRACE

- **IPCS GTFTRACE**
  - The single most important command to work with GTF in IPCS
  - Formats GTF trace records in dump or trace data set
  - By default, formats all the trace records collected in the trace
  - You can use parameters to further specify which records should be formatted

# Processing GTF output in IPCS - GTFTRACE

- Sample output of the 'IPCS GTFTRACE' command:

```

IPCS OUTPUT STREAM -----
Command ==>
***** TOP OF DATA *****
    **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
SSCH=ALL  IO=ALL  CCW=SI
SVC=ALL  PI=ALL  IOX=ALL
EXT  DSP  SLIP  RNIO  SRM  RR

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
Minimum tracing for IO, SSCH, SVC, PI, EXT, and FRR events
All GTRACE events requested
All events associated with the execution should be traced
All DISPATCHER events traced
PCI events are to be traced
System resource manager events traced

        **** GTF TRACING ENVIRONMENT ****
Release: SP7.1.3   FMID: HBB7780   System name: XE23
CPU Model: 2827   Version: FF   Serial no. 074D07

DSP   ASCB.... 00FDD100 CPU..... 0001   PSW..... 07060000 0000002F 00000000 00000000
          ****   R0..... ****   R1..... ****
  
```

# Processing GTF output in IPCS - GTFTRACE

- Information displayed on GTF trace records:

```

IPCS OUTPUT STREAM -----
Command ===>
***** TOP OF DATA *****
    *** GTFTRACE DISPLAY OPTIONS IN EFFECT ***
    SSCH=ALL  IO=ALL  CCW=SI
    SVC=ALL  PI=ALL  IOX=ALL
    EXT  DSP  SLIP  RNIO  SRM  RR

    **** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
    Minimum tracing for IO, SSCH, SVC, PI, EXT, and FRR events
    All GTRACE events requested
    All events associated with the execution should be traced
    All DISPATCHER events traced
    PCI events are to be traced
    System resource manager events traced

        **** GTF TRACING ENVIRONMENT ****
        Release: SP7.1.3  FMID: HBB7780  System name: XE23
        CPU Model: 2827  Version: FF  Serial no. 074D07

    DSP  ASCB.... 00FDD100 CPU..... 0001  PSW..... 07060000 0000002F 00000000 00000000
           ****  R0..... ****  R1..... ****
  
```



# Processing GTF output in IPCS - GTFTRACE

- Information on types of records that were collected by GTF:

```

IPCS OUTPUT STREAM -----
Command ==>
***** TOP OF DATA *****
    **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
SSCH=ALL IO=ALL CCW=SI
SVC=ALL PI=ALL IOX=ALL
EXT DSP SLIP RNIO SRM RR

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
Minimum tracing for IO, SSCH, SVC, PI, EXT, and FRR events
All GTRACE events requested
All events associated with the execution should be traced
All DISPATCHER events traced
PCI events are to be traced
System resource manager events traced

    **** GTF TRACING ENVIRONMENT ****
Release: SP7.1.3   FMID: HBB7780   System name: XE23
CPU Model: 2827   Version: FF   Serial no. 074D07

DSP   ASCB.... 00FDD100 CPU..... 0001   PSW..... 07060000 0000002F 00000000 00000000
      ***** R0..... ***** R1..... *****
  
```

# Processing GTF output in IPCS - GTFTRACE

- Information on the environment where the GTF trace was collected:

```

IPCS OUTPUT STREAM -----
Command ==>
***** TOP OF DATA *****
    **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
SSCH=ALL IO=ALL CCW=SI
SVC=ALL PI=ALL IOX=ALL
EXT DSP SLIP RNIO SRM RR

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
Minimum tracing for IO, SSCH, SVC, PI, EXT, and FRR events
All GTRACE events requested
All events associated with the execution should be traced
All DISPATCHER events traced
PCI events are to be traced
System resource manager events traced

**** GTF TRACING ENVIRONMENT ****
Release: SP7.1.3   FMID: HBB7780   System name: XE23
CPU Model: 2827   Version: FF   Serial no. 074D07

DSP   ASCB.... 00FDD100 CPU..... 0001   PSW..... 07060000 0000002F 00000000 00000000
      ***** RO..... ****   R1..... ****
  
```

# Processing GTF output in IPCS - GTFTRACE

- Beginning of the trace data:

```

IPCS OUTPUT STREAM -----
Command ==>
***** TOP OF DATA *****
    **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
SSCH=ALL IO=ALL CCW=SI
SVC=ALL PI=ALL IOX=ALL
EXT DSP SLIP RNIO SRM RR

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
Minimum tracing for IO, SSCH, SVC, PI, EXT, and FRR events
All GTRACE events requested
All events associated with the execution should be traced
All DISPATCHER events traced
PCI events are to be traced
System resource manager events traced

        **** GTF TRACING ENVIRONMENT ****
Release: SP7.1.3   FMID: HBB7780   System name: XE23
CPU Model: 2827   Version: FF   Serial no. 074D07

DSP   ASCB.... 00FDD100 CPU..... 0001   PSW..... 07060000 0000002F 00000000 00000000
          ****   R0..... ****   R1..... ****
  
```

# Processing GTF output in IPCS - GTFTRACE

- Entries displayed in chronological order
- Oldest entry at the top, newest at the bottom

```

...
EXT  CODE.... 1004      ASCB.... 00FDD100 CPU..... 0000      PSW..... 07060000 00001004 00000000
      00000000 TQE-TCB. N/A
      GMT-05/22/2014 10:10:34.502240 LOC-05/22/2014 06:10:34.502240

DSP  ASCB.... 00FB9400 CPU..... 0000      PSW..... 07040000 80000001 00000000 0B2A0178
      807E38C8 R0..... 00000001 R1..... 4N0%
      GMT-05/22/2014 10:10:34.502304 LOC-05/22/2014 06:10:34.502304

SVC  CODE.... 002      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00000000 R0..... 00000000 R1..... 0B105024
      GMT-05/22/2014 10:10:34.502321 LOC-05/22/2014 06:10:34.502321

SVCR CODE.... 002      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00FE00FA R0..... 00000002 R1..... 00FE00FF
      GMT-05/22/2014 10:10:34.502325 LOC-05/22/2014 06:10:34.502325
...

```

# Processing GTF output in IPCS - GTFTRACE

- Every trace record is time-stamped
- Useful when combining GTF records with other data

```

...
EXT  CODE.... 1004      ASCB.... 00FDD100 CPU..... 0000      PSW..... 07060000 00001004 00000000
      00000000 TQE-TCB. N/A
      GMT-05/22/2014 10:10:34.502240 LOC-05/22/2014 06:10:34.502240

DSP  ASCB.... 00FB9400 CPU..... 0000      PSW..... 07040000 80000001 00000000 0B2A0178
      807E38C8 R0..... 00000001 R1..... 4N0%
      GMT-05/22/2014 10:10:34.502304 LOC-05/22/2014 06:10:34.502304

SVC  CODE.... 002      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00000000 R0..... 00000000 R1..... 0B105024
      GMT-05/22/2014 10:10:34.502321 LOC-05/22/2014 06:10:34.502321

SVCR CODE.... 002      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00FE00FA R0..... 00000002 R1..... 00FE00FF
      GMT-05/22/2014 10:10:34.502325 LOC-05/22/2014 06:10:34.502325
...

```

# Processing GTF output in IPCS - GTFTRACE

- The contents of record varies depending on its type
- The records are described in [MVS Diagnosis: Tools and Service Aids](#)

```

...
EXT  CODE.... 1004      ASCB.... 00FDD100 CPU..... 0000      PSW..... 07060000 00001004 00000000
      00000000 TQE-TCB. N/A
      GMT-05/22/2014 10:10:34.502240 LOC-05/22/2014 06:10:34.502240

DSP  ASCB.... 00FB9400 CPU..... 0000      PSW..... 07040000 80000001 00000000 0B2A0178
      807E38C8 R0..... 00000001 R1..... 4N0%
      GMT-05/22/2014 10:10:34.502304 LOC-05/22/2014 06:10:34.502304

SVC  CODE.... 012      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00000000 R0..... 00000000 R1..... 0B105024
      GMT-05/22/2014 10:10:34.502321 LOC-05/22/2014 06:10:34.502321

SVCR CODE.... 012      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000
      007E35A0 R15..... 00FE00FA R0..... 00000002 R1..... 00FE00FF
      GMT-05/22/2014 10:10:34.502325 LOC-05/22/2014 06:10:34.502325
...

```

# EXAMPLE: SVC and SVCR records

- If SVC tracing is active, SVC call will generate SVC and SVCR records

```

SVC  CODE.... 012      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000 80C4
      007E35A0 R15..... 00000000 R0..... 00000000 R1..... 0B105024
      GMT-05/22/2014 10:10:34.502321  LOC-05/22/2014 06:10:34.502321

SVCR CODE.... 012      ASCB.... 00FB9400 CPU..... 0000      PSW..... 07041000 80000002 00000000 80C4
      007E35A0 R15..... 00FE00FA R0..... 00000002 R1..... 00FE00FF
      GMT-05/22/2014 10:10:34.502325  LOC-05/22/2014 06:10:34.502325
  
```

- The following information is provided:
  - CODE - SVC interruption code (AKA SVC number)
  - ASCB - Address of the ASCB for AS where the interruption occurred
  - CPU - Address of the processor on which the interruption occurred
  - PSW - Pointing +2 bytes after SVC instructions (**truncated above**)
  - R15, R0, R1 - Contents of respective registers

# EXAMPLE: IO record

- If IO tracing is active, IO interruptions will generate IO record

```

IO..... 05167   ASCB.... 00FB9400 CPUID... 0000   JOBN.... GTV   PSW..... 07060000 00000000
                IRB..... 00C04007 1AC4FA10 0C000000 00010002 00000000 TCB..... 007E3950 SENSE...
                OPT..... 00           DVRID... 02           IOSLVL.. 01           UCBLVL.. 01           UCBWGT..

                I/O Statistics:
                Connect. 20000000 Pending. 01FC0000 Discon.. 01280000 CUQ..... 00000000 DAO.....
                ICMR.... 00000000 StartCt. 00000000 SampCt. 79A00000
                GMT-05/22/2014 10:10:34.502927  LOC-05/22/2014 06:10:34.502927

```

- The following information is provided:
  - IO - Device number (from UCB), can be translated to VOLSER
  - JOBN - Name of the job associated with the task that requested the I/O
  - IRB - First five words of interruption response block (IRB) operand
  - TCB - TCB for the task that requested the I/O
  - I/O stats - times for specific parts of I/O request processing in units of 0.5 microsecond



# IPCS GTFTRACE examples

- **IP GTF SLIP**  
Formats SLIP trace records
- **IP GTF SVC**  
Formats SVC trace records
- **IP GTF SVC(18,20,21,34,35)**  
Formats SVC trace records for specified SVC numbers
- **IP GTF JOBNAME('XCFAS')**  
Formats trace records for specified jobname
- **IP GTF ASCB(00FB9400,00FDD100)**  
Formats trace records related to specific ASCBs
- **IP GTF IO(05167)**  
Formats IO trace records for specified device number(s)

# GTF and SLIP

- **SLIP** - a diagnostic aid (**s**erviceability **l**evel **i**ndication **p**rocessing), intercepts certain system events and takes action for them.
- May be used with **GTF** to trace the program execution
- SLIP command is documented in the [z/OS MVS System Commands](#) manual

# GTF and SLIP

- Procedure for obtaining **GTF** trace with **SLIP** records:
  1. Start **GTF** with **TRACE=SLIP** option
  2. Set **SLIP** with **SLIP SET** command
  3. Recreate the problem or the event to be recorded in the trace
  4. Delete **SLIP** with **SLIP DEL** command
  5. Stop **GTF**

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**  
  
**IF** - specifies the event as an instruction fetch  
**SBT** - specifies the event as a successful branch trace caused by branch instruction into the specified range, or a branch within the specified range  
  
**A=TRACE** - when trap matches, it creates GTF trace record  
**A=TRDUMP** - same as A=TRACE, but also schedules an SVC dump when trap is disabled or deleted

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored

**LPAEP=(name[, start[, end]])**

**LPAMOD=(name[, start[, end]])**

- **LPAEP / LPAMOD** monitors link pack area (LPA) entry points / modules
- **Name** – specifies entry point or module name or alias
- **Start / end** – (optional) the offsets from the start of the module / EP, indicates the range to be monitored. If only start is specified, range is a single address

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored

**NUCEP=(name[, start[, end]])**

**NUCMOD=(name[, start[, end]])**

**PVTEP=(name[, start[, end]])**

**PVTMOD=(name[, start[, end]])**

- **NUCEP / NUCMOD** monitors modules in nucleus
- **PVTEP / PVTMOD** monitors entry points / modules in the private area

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored

- Examples:

LPAEP=IEECB907

NUCMOD=(IEAVTRTS, C4)

PVTMOD=(MYMOD, 100, 120)

PVTMOD=(MYMODB)

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored  
**RANGE=(start[, end])**

- **RANGE** specifies the starting and ending address of virtual storage to be monitored
- If only **start** is specified, range consists of that 1 byte
- If **start** > **end**, the addresses wrap around
- **RANGE** option supports both *direct* and *indirect addressing*



# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored  
**RANGE=(start[, end])**

- Indirect address is a location that contains a pointer
- Indirect address is specified by percent sign (%) for 24bit pointer, a question mark (?) for 31bit pointer, or an exclamation point (!) for 64bit pointer. Pointers can be chained together.
- Addresses can include displacement(s) of up to x'7FFF' bytes

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored  
**RANGE=(start[, end])**

- Examples:

**RANGE=00BA7868**

**RANGE=(00B9DEC4?, +200)**

**RANGE=(10?+C8?+84?+90?+40, +100)**

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored

**TRDATA=(STD[, REGS][, list])**

**TRDATA=(list)**

- **STD** writes a SLIP standard trace record
- **REGS** specifies SLIP records to contain the contents of 16 GPRs & ARs
- **List** specifies one or more sets of addresses for storage areas to dump
- It supports indirect addressing and allows pointers in GPRs to be specified as part of the address

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – specifying module/entry point/storage to be monitored

**TRDATA=(STD[, REGS][, list])**

**TRDATA=(list)**

- Examples:

**TRDATA=(STD, REGS)**

**TRDATA=(STD, REGS, 0R?, +40, 1R?+60?, +80)**

**TRDATA=(STD, REGS, 00B9DEB0, 00B9DEBF)**

# Processing GTF output in IPCS - GTFTRACE

- TRDATA=(**STD,REGS**)

```

SLIP S+U      ASCB.... 00F95580 CPU..... 0000      JOB..... ZGFVPRT1 TID..... ASDF      ASID.... 0041
              TCB..... 007FF3A0 MFLG.... 0131      EFLG.... 0000      SFLG.... 60        DAUN.... 0000
              OFFS.... 000000F8 IADR.... 00000000 03A68960      INS..... 0BE8      EXSIAD.. N/A
              EXSINS.. N/A      BRNGH... N/A      BRNGA... N/A      BRNGD... N/A
              OPSW.... 47041000 80000000 00000000 03A68962      PIC/ILC. 00020080 PERC.... 40
              PKM..... 84C0      SASID... 0041      AX..... 0000      PASID... 0041      ASC..... 0
  
```

## GENERAL PURPOSE REGISTER VALUES

```

0-3..... FF46EDA0 007E6E10 D7C4E2D4 00FDA438
4-7..... 007FF3A0 007FD548 83A68868 00F95580
8-11.... 83A68962 00FBF470 00000002 00FBDC70
12-15... 00000001 7F429148 80FF3C00 FF46ED98
  
```

## GPR HIGH HALF VALUES

```

0-3..... 00000000 00000000 FFFFFFFF FFFFFFFF
4-7..... FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
8-11.... 00000000 FFFFFFFF FFFFFFFF FFFFFFFF
12-15... FFFFFFFF 00000000 00000000 00000000
  
```

## ACCESS REGISTER VALUES

```

0-3..... 00000028 00000000 FFFFFFFF FFFFFFFF
4-7..... FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
8-11.... FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
12-15... FFFFFFFF FFFFFFFF 00000000 00000000
  
```

GMT-06/06/2014 13:34:22.939857 LOC-06/06/2014 09:34:22.939857

# Processing GTF output in IPCS - GTFTRACE

- TRDATA=(STD,REGS,**0R?**,+40)

```

...
SLIP USR          CPU..... 0001      EXT..... 0001      CNTLN... 00
0041 0001004C E2E5E3C8 D4F1F4F5 00000000 00000000 00000000 00000000 00000000 | ...<SVTHM145..... |
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 | ..... |
00                                          | . |
GMT-06/06/2014 14:19:24.241214 LOC-06/06/2014 10:19:24.241214

```

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options:

- ID=slip\_ID**
- specifies a 1 to 4 character trap identifier
  - if not specified, it is provided by the system
  - needed when deleting the SLIP

# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
**SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END**

Options – restricting SLIP to certain job(s) or address space(s)

**JOBNAME=jobname** - specifies the name of the job  
or STC to be monitored

- may include wildcard (\*,?) characters

- Examples:

**JOBNAME=ENF**

**JOBNAME=WEBSSR\***

**JOBNAME=BATCH??R**



# GTF and SLIP – setting the SLIP

- SLIP is set using MVS SLIP SET command:  
`SLIP SET, {IF | SBT}, A={TRACE | TRDUMP}, options, END`

Options:

- `MATCHLIM=m`
- generally specifies that the SLIP is to be disabled after **m** matches ( $1 \leq m \leq 65535$ )
  - for `A=TRACE` and `A=SBT` it also specifies the number of SLIP records written by GTF before the trap is disabled

- Example:

`MATCHLIM=50`

# GTF and SLIP – setting the SLIP

- Examples of SLIP SET command:

```
SL SET, IF, ID=EX01, RANGE=(00F81868, +620),  
A=SBT, TRDATA=(STD, REGS), END
```

```
SL SET, IF, PVTEP=(FDRXPASS), A=TRACE,  
TRDATA=(STD, REGS, 1R?, +FF, 6R?, 3F, 10R?, +124, 15R?, 3F),  
ID=FDRC, JOBNAME=FDR4CPY, E
```

```
SL SET, IF, RANGE=10?+C8?+84?+90?, A=TRACE,  
TRDATA=(STD, REGS, 0R?, +3F, 1R?, +57), JOBNAME=DSS*, E
```

# GTF and SLIP – removing the SLIP

- SLIP is removed using MVS SLIP DEL command:  
**SLIP DEL, ID=slip\_ID**
- Example:  
**SLIP DEL, ID=0005**
- The list of SLIPs on the system may be displayed by MVS DISPLAY command:  
**DISPLAY SLIP**

# Performance considerations

- There is some CPU and I/O overhead associated with GTF
- In order to limit the overhead:
  - Use MODE=INT to store the data in internal trace table
  - Restrict the tracing only to specific record types you're interested in.
- There may be cases when more trace data is generated that can be written to a data set.
- To alleviate this problem you may modify the GTF procedure to specify multiple trace data sets for parallel I/O operations to store the trace data.

# Summary

- GTF is used to collect historical data. It can generate trace records for a variety of system events including SVCs, IO, interruptions, subchannel programs, recovery operations.
- With SLIP, GTF can be used to trace the program execution, including the contents of registers and storage for executed instruction.
- You can use GTF to gain insights into what the programs are doing and how they work.
- GTF is part of the system – it is available on every z/OS system.

# Practical laboratory GTF session

- If you are interested in getting hands-on experience with GTF tracing, here's your chance:

## **16161: GTF Trace Hands-on Lab**

Friday, August 8, 8:30 AM-9:30 AM

Room 301 (David L. Lawrence Convention Center)

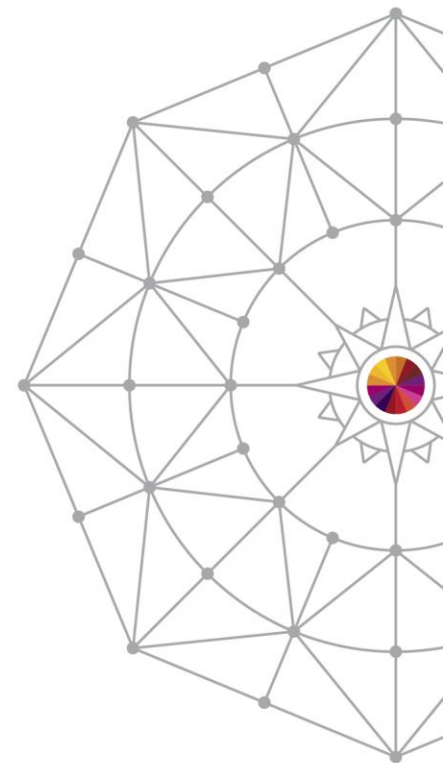
- During the session, we will set up a procedure to obtain the GTF trace of a test program
- We will be tracing SVC calls, I/O operations, and the program execution
- We will review the resulting trace in IPCS

# thank you

***Vlastimil Jedek***

*Sustaining engineering  
CA Technologies*

[Vlastimil.Jedek@ca.com](mailto:Vlastimil.Jedek@ca.com)



#SHAREorg



# Reference information (1/2)

- [z/OS MVS Diagnosis: Tools and Service Aids](#)  
(The Generalized Trace Facility - GTF)
- [z/OS MVS System Commands](#)  
(START GTF command)
- [z/OS MVS Initialization and Tuning Reference](#)  
(GTFPARM member in SYS1.PARMLIB)
- [z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG](#)  
(GTRACE macro description)



## Reference information (2/2)

- [z/OS MVS IPCS Commands](#)  
(IPCS GTFTRACE, COPYTRC, and MERGE subcommands)
- [z/OS MVS IPCS Customization](#)  
[z/OS MVS IPCS User's Guide](#)  
(General IPCS documentation)
- [z/OS MVS System Commands](#)  
(SLIP command)