

CA IDEAL and Time measurement

IDEALTIM – Version 2.1

Abstract

IDEALTIM gives you the possibility to measure and analyze your CA-IDEAL programs in hundred thousandth of seconds (1/100000 sec). Simply add a call to IDEALTIM in your application program at any point of your interest and you'll get all information about the resource consuming parts in the CICS Log and the IDEAL queue ADROUT.

Facts

Exact time measurement can't be done in an CA Ideal environment, because IDEAL doesn't give any subsecond information to the programmer. Tuning a specific program, a program part or an entire program tree is impossible.

Solution

By calling the program IDEALTIM the programmer has now the following possibilities:

- Getting the **system time in 1/100000** seconds.
- Getting the **time difference** between the first invocation and a later second invocation of the program IDEALTIM
- Getting the **time interval** between the following, subsequent calls to the program IDEALTIM

IDEALTIM Communication area

1	LNK-DISP-S-DATE	X 8	Start date in the form dd/mm/yy
2	FILLER	X 1	Space
2	LNK-DISP-S-TIME	X 14	Start time in the form hh:mm:ss.tttt
2	FILLER	X 1	Space
3	LNK-DISP-E-DATE	X 8	End date in the form dd/mm/yy
2	FILLER	X 1	Space
4	LNK-DISP-E-TIME	X 14	End time in the form hh:mm:ss.tttt
2	FILLER	X 1	Space
4	LNK-DISP-M-TIME	X 9	Interval time in seconds in the form sss.tttt
2	FILLER	X 1	Space
5	LNK-DISP-D-TIME	X 9	Difference between start date/time and end date/time in seconds in the form sss.tttt
2	FILLER	X 1	Space
	LNK-PGM	X 8	Can be used for the program name. Optional.
2	FILLER	X 1	Space
	LNK-TEXT	X 32	Can be used to transfer information to the CICS-Log
	FILLER	X 1	Space
	LNK-WORK-REQUEST	X 1	„S“ = Start gathering information for the first time. „D“ = Difference. Calculate the difference between the first and the current invocation.
	FILLER	X 1	Space
	LNK-DATE-AREA	N P 15	Internal use. Do not modify.
	LNK-DATE-AREA-M	N P 15	Internal use. Do not modify.

How to Call

Sample 1 (Measure specific parts of a program)

```

1      myarea      COPY DVW IDEALTIM

SET LNK-PGM = 'myprog'

SET LNK-TEXT = 'Start resource consuming part'
SET LNK-WORK-REQUEST = 'S'
CALL IDEALTIM USING UPDATE myarea
LIST myarea.LNK-LIST

LOOP
:      ← Program part 1 to measure
END-LOOP

SET LNK-WORK-REQUEST = 'D'
SET LNK-TEXT = 'End resource consuming part'

CALL IDEALTIM USING UPDATE myarea
LIST myarea.LNK-LIST

FOR FIRST .....
:      ← Program part 2 to measure
END-FOR

SET LNK-WORK-REQUEST = 'D'
SET LNK-TEXT = 'End of FOR construct'

CALL IDEALTIM USING UPDATE myarea
LIST myarea.LNK-LIST

```

Sample Results

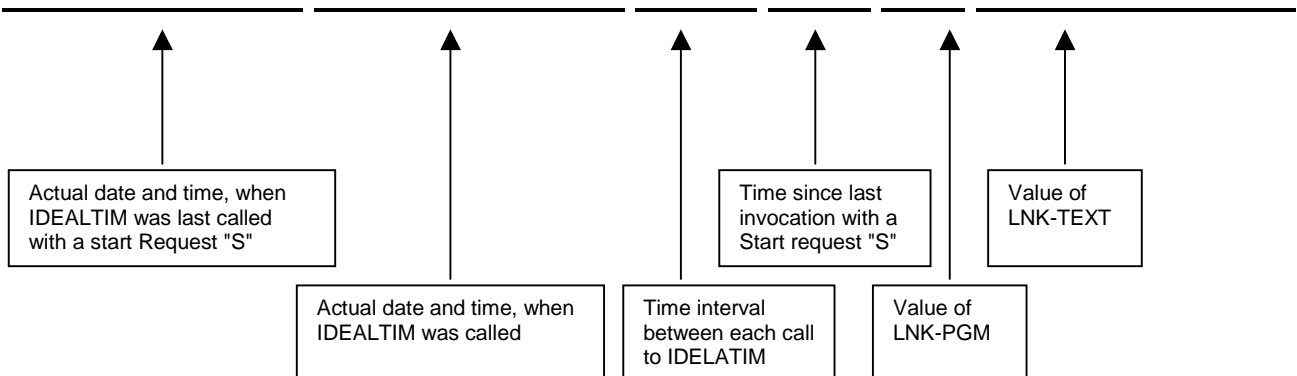
ADROUT

The sample code above will give you the following result in the IDEAL print queue ADROUT:

```

03.04.02 14:39:37.91572          0.00000  0.00000 MYPROG  START RESOURCE CONS...
03.04.02 14:39:37.91572 03.04.02 14:39:37.94370  0.02798  0.02798 MYPROG  END RESOURCE CONSUM ..
03.04.02 14:39:37.91572 03.04.02 14:39:37.95217  0.00847  0.03645 MYPROG  END OF FOR CONSTRUC...

```



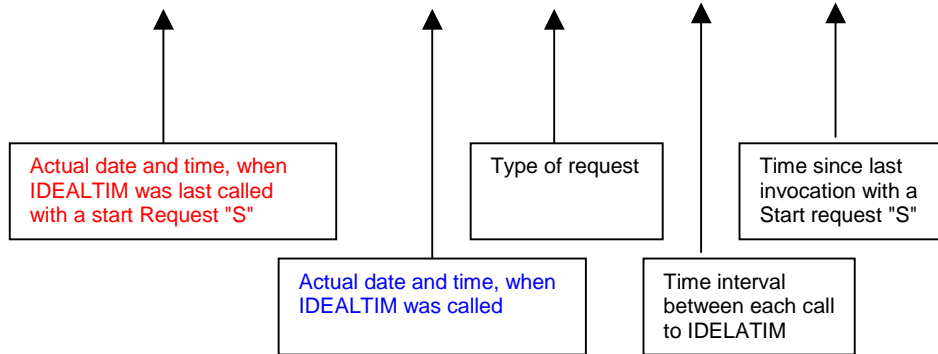
The fields in the dataview can be used for further processing by the programmer.

CICS-LOG

The sample code above will give you the following result in the CICS log (CADL):

```

TIM-01I 03.04.02 14:39:37.91572                0.00000  0.00000
TIM-02I MYPROG  START RESOURCE CONSUMING PART  S
TIM-21I 03.04.02 14:39:37.91572 03.04.02 14:39:37.94370  0.02798  0.02798
TIM-22I MYPROG  END RESOURCE CONSUMING PART    D
TIM-21I 03.04.02 14:39:37.91572 03.04.02 14:39:37.95217  0.00847  0.03645
TIM-22I MYPROG  END OF FOR CONSTRUCT          D
    
```



How it works

Time	Timestamp	Code	Time Interval	Elapsed time
↓		Program start LIST "IDEAL-TIME Controlled by IDEALTIM" ①		
	14:39:37.91572	SET PCOMMAREA.LNK-WORK-REQUEST = 'S' CALL IDEALTIM USING UPDATE PCOMMAREA	0.00000	0.00000
	14:39:37.94370	Here process part 1 (part of user-program) SET PCOMMAREA.LNK-WORK-REQUEST = 'D' CALL IDEALTIM USING UPDATE PCOMMAREA	0.02798	0.02798
	14:39:37.95217	Here process part 2 (part of user-program) SET PCOMMAREA.LNK-WORK-REQUEST = 'D' CALL IDEALTIM USING UPDATE PCOMMAREA	0.00847	0.03645

Notes:

- ① The first LIST in an IDEAL program is rather resource consuming. It is suggested to place it outside the measurement part of the program.

Hints an Tips

Multiple, different measurement start points	To handle multiple, different measurement start points different working storage definitions (using the same COPY DVW IDEALTIM) can be used.
Using the current time for multiple nested programs	For trace purpose, actualize always LNK-PGM.
Can the exact time be used in the application itself.	Yes. E.G. for Display User written accounting User written locking mechanism And so on
Migration to production	There is no problem migrating IDELTIM to a productive environment. Take care about the CICS-LOG size and especially about the maximum size of ADROUT (If a "fill up" of ADROUT occurs, a runtime abend will be the result. It's suggested not to migrate to production using your own LIST statements.

Installation

CICS System

The Program IDEALTIM and GETTIME must be copied in a PDS library which is allocated to CICS (DFHRPL). An appropriate PPT Entry for the programs must be defined if program-autoinstall is not used. IDEALTIM as a COBOL, GETTIME as an Assembler program.

IDEAL

The Dataview IDEALTIM must be defined as follows:

CATALOGED 03/27/02 09:47		SEQUENTIAL UPD=YES FILENAME IDEALTIM		
		RECSIZE= 0130		
1	1	IDEALTIM		
2	2	LNK-UBGRUPPE		
3	3	LNK-LIST		
4	4	LNK-DISP-S-DATE	X	8
5	4	LNK-DISP-FILL01	X	1
6	4	LNK-DISP-S-TIME	X	14
7	4	LNK-DISP-FILL02	X	1
8	4	LNK-DISP-E-DATE	X	8
9	4	LNK-DISP-FILL03	X	1
10	4	LNK-DISP-E-TIME	X	14
11	4	LNK-DISP-FILL04	X	1
12	4	LNK-DISP-M-TIME	X	9
13	4	LNK-DISP-FILL05	X	1
14	4	LNK-DISP-D-TIME	X	9
15	4	LNK-DISP-FILL06	X	1
16	4	LNK-PGM	X	8
17	4	LNK-DISP-FILL07	X	1
18	4	LNK-TEXT	X	32
19	4	LNK-DISP-FILL08	X	1
20	4	LNK-WORK-REQUEST	X	1
21	4	LNK-DISP-FILL09	X	1
22	3	LNK-DATE-AREA	N P	15
23	3	LNK-DATE-AREA-M	N P	15

IDEALTIM-Definition in CA-Ideal

Identification	
PROGRAM IDEALTIM	
Created	03/20/02 By EX00W
Last Modified	03/20/02 at 09:57 By EX00W
New Copy On Call? N	Access DB2? N
Short Description:	Ideal-Time in 1/100000 seconds
Language:	COBOL Target Date __ __ __ Actual Date __ __ __
Description:	_____ _____ _____ _____

Parameter Section							
Level	Field name	T	I	Ch/Dg	Occur	U	Comments/Dep on/Copy
===== TOP =====							
1	PCOMMAREA					U	
2	LNK-LIST						
3	LNK-DISP-S-DATE	X			8		
3	LNK-DISP-FILL01	X			1		
3	LNK-DISP-S-TIME	X			14		
3	LNK-DISP-FILL02	X			1		
3	LNK-DISP-E-DATE	X			8		
3	LNK-DISP-FILL03	X			1		
3	LNK-DISP-E-TIME	X			14		
3	LNK-DISP-FILL04	X			1		
3	LNK-DISP-M-TIME	X			9		
3	LNK-DISP-FILL05	X			1		
3	LNK-DISP-D-TIME	X			9		
3	LNK-DISP-FILL06	X			1		
3	LNK-PGM	X			8		
3	LNK-DISP-FILL07	X			1		
3	LNK-TEXT	X			32		
3	LNK-DISP-FILL08	X			1		
3	LNK-WORK-REQUEST	X			1		
3	LNK-DISP-FILL09	X			1		
2	LNK-DATE-AREA	N	P		15		
2	LNK-DATE-AREA-M	N	P		15		
===== BOTTOM =====							

Sample Ideal-Program which calls the Program IDEALTIM

Identification Section			
PROGRAM PRFTST01			
Created	03/20/02	By	EX00W
Last Modified	03/27/02 at 10:01	By	EX00W
Last Compiled	03/27/02 at 10:01		
Run Status:	PRIVATE		
Short Description:	Performance-Test-01		
Language:	IDEAL	Target Date	___ __ Actual Date ___ __
Description:	_____ _____ _____ _____		
Resources			
Type	Name of DVW/PGM/PNL or RPT	Version	System
===	===== T O P =====	====	===
DVW	IDEALTIM	0002	
PGM	IDEALTIM	002	\$ID
===	===== B O T T O M =====	====	===

Working Section

Level	Field Name	T	I	Ch/Dg	Occur	Value/Comments/Clauses
===== TOP =====						
1	PCOMMAREA					COPY DVW IDEALTIM
1	WLOOPCNT	U	Z		9	
1	WSTRUC1					
2	FIELD1	X			10	
2	FIELD2	X			10	
1	WSTRUC2					
2	FIELD1	X			10	
2	FIELD2	X			10	

Procedure Section

```

:-----
: MAIN PROCEDURE
:-----
<<MAINPROC>> PROC

DO PROGINIT

LOOP 100000 TIMES

    SET WSTRUC1 = WSTRUC2 BY POSITION

    SET WLOOPCNT = WLOOPCNT + 1

    IF $REMAINDER(WLOOPCNT,DIV=1000) = 0
        DO CALLDPGM
        DO PRNTDATA
    ENDIF

ENDLOOP

SET PCOMMAREA.LNK-TEXT = 'END OF PROGRAM'

DO CALLDPGM
DO PRNTDATA

ENDPROC
:-----
: INITIALIZE PROGRAM
:-----
<<PROGINIT>> PROC

    SET PCOMMAREA.LNK-TEXT = 'START PROGRAM'

DO CALLSPGM
DO PRNTDATA

SET WSTRUC1.FIELD1 = 'ABCDEFGHJIJ'
SET WSTRUC2.FIELD2 = '0123456789'

ENDPROC
:-----
: CALL SUBPROGRAM WITH PARAMETER 'D'
:-----
<<CALLDPGM>> PROC

SET PCOMMAREA.LNK-WORK-REQUEST = 'D'
CALL IDEALTIM USING UPDATE PCOMMAREA

```

```

ENDPROC
:-----
: CALL SUBPROGRAM WITH PARAMETER 'S'
:-----
<<CALLSPGM>> PROC

  SET PCOMMAREA.LNK-WORK-REQUEST = 'S'
  CALL IDEALTIM USING UPDATE PCOMMAREA

ENDPROC
:-----
: PRINT COMMAREA-DATA
:-----
<<PRNTDATA>> PROC

  LIST PCOMMAREA.LNK-LIST

ENDPROC

```

Version **2.10**

Date **April, 2002**

Contact / Additional Information

Marcel Irniger marcel.irniger@helsana.ch	Beat Merz beat.merz@helsana.ch
---	-----------------------------------