AUG 29/19 All S/A'S

- D. McBRIDE

- O. TOWNSEND

- C. GIBBS

- B. ZADO

- M. STRAIN

- E. HANSON, H.O.

The following working
paper describes the
use of the Common

Less of the Common

Area feature available

It Center in IMS/90 Release 6.

SPERRY UNIVAC
Philadelphia Software Development Center

Working Paper Data Management Systems

	(*D=Database, U=Utilit	Revision # ies, A=Applications	
tiachet betrunt fo 2 <u></u> -	Title: Common Area		
	Abstract: This working paper d a method to implemen Area feature for IMS	t Common	
lio in the Bind. This working par romacture.	.vi naudscion insvis oni .vi naudscion insvis oni .vi naudscion (inspiral		
	្រុក 2 ក្រុំ និងស៊ីន ១៣០១ ជាង ១១៩៦១៣០ ក្រុក ខេត្ត		

Author: R. Vohra Date: 09/27/78

Distribution:

J. J. Sweeney C. Casals J. Merman
R. D. Pratt M. Lombardo R. Winkelman
M. N. Theodore K. Koenig P. Reichmuth

PROPRIETARY NOTICE — This document may contain information of a proprietary nature to the Sperry Univac Division of Sperry Rand Corporation. This document and the information thereon must not be used for other than internal company use unless written permission is obtained from Sperry Univac.



<u>.</u>:

1.0 General Description

Several installations have expressed a need for main storage resident common area. This common area can be used to keep vital information which can be accessed by all the terminals. Records in this area will be lockable and recoverable. Criteria for the support of common area is efficiency, flexibility, and ease of use. We are proposing, that common area be a main storage resident ISAM file. It is a multi-User can issue GET, GETUP or PÚT on any record record file. of this file as he currently does for an ISAM file. record locking procedure will hold for this file.

- 2.0 To implement the common area feature the following components of IMS 90 are affected:
 - Configurator Philippe and Somware Daveronment Center a.
 - ь. Startup
 - On-line c.
 - d. Shutdown

The first of the same of the s Dais Prenecembert Systems

a) Configurator

> Common area is main storage resident, variable blocked or fix blocked ISAM file. Any number of files can be main storage resident. Beside supporting the current options in the FILE section, two additional keyword parameters are supported for common area file.

CAFILE = / YES \ specifies that the ៊ីកាន់ខ មានត្រងកាត្ត ពុទ្ធពុទ្ធ។ ជាមិនសត្វកូចគ

a method to indiement – file is a common area file. Default is NO. If CAFILE=YES is specified, then configurator will set the bit X'80' in the flag byte ZF#FFLG2 of the File Control Table (FCT).

 CUPDATE = \ YES NO (

الطيعة المووار (والراز للمجور) في ال المحادث الموادر (المحدو الواد

Dan Dan Brain Andrews

This option will allow the user to specify whether updating of the disk images is necessary. Default is NO. If YES option is selected, a bit in the FCT (bit X'40' byte ZF#FFLG2) is set.

option will be valid for common area TRACE NO

files.

● If LOCK = UP is specified, then no before images Control ampendata (Control Substant Substant Control Segential Co

are written to the AUDIT file .- I.E. -- no on-line recovery takes place in case of any error in the user action program. preum - maine tan despresses de la dispresses de la dispresse de la dispresse

iggymys (voor soletti), goto yhtiegh, kirinigh elegh ugelyk, ok regelykeliks hi Gelessii (solkal

b)-Startup

During startup procedure of IMS 90, common data area file will be loaded in main storage. For error diagnostics, refer to technical documentation by P. Reichmuth.

ម្រើរ ខេត្ត មហៈ ២២២២១១៤១១% ខេត្ត ១១៣០៤៣

On-Line Processing

User is allowed to issue GET, GETUP, and PUT IMS 90 function calls for a common area file. All the applicable error codes are posted in the PIB.

- On-line and off-line recovery procedure is valid for common area files if user chooses recovery options.
- Common area file requests are handled by the common data area modules (ZF#CSAS for S/T and ZF#CSAM for M/T). The general request processor module will pass control to the appropriate common data area c., module.
- No changes are required to on-line recovery, off-line y or warm restart. The firm and the substant of the second recovery or warm restart.

Shutdown d)

£.

During shutdown, common data area files will be updated using the data in main storage.

Tight du e pagraph (expérifique) desfédit (de Mill) in cérille-vil

The asymptotic form is also before the first of the first

continues in Elifornia in particular particu

and the second of the content of the second of the second

report ရေး ၁၈၈ နေရေး အရေးခဲ့သည်။ နှုပ်ပြီးကို ကြည်သည် ပြီးပြီးသည်။ မေးသည့် ကျော့မသော ပြွေးပော်လူမှုနှင့် သူမေနကျောင်းသည်။ အစားလုံးရသည်။ အနိုင်သည်။ နေချိန်သည်။ ထိုက် လုံးသည် ရှိနေရှိနှင့် ခြုံစုံရှိနှင့် ကြွတ်စွေလုပ်

1. TITLE

ZF#CSAS. This module handles all requests to a CSA*-file in memory for single-thread IMS 90. The part of a property of the part of the par

*CSA = Common-Storage-Area Thistiph is held held held.

1.1 Purpose

Several installations have expressed a need for main storage resident common area. This common area can be used to keep vital information which can be accessed by all the terminals. Records in this area will be lockable and recoverable. Criteria for the support of common area is efficiency, flexibility, and ease of use. We are proposing, that common area be a main storage resident ISAM file. It is a multirecord file. User can issue GET, GETUP or PUT on any record of this file as he currently does for an ISAM file. IMS 90 record locking procedure will hold for this file.

As changer and required to on-line recevery. off-line. recovery of warm resized.

. ಶಿವರಿಸಲ್ಯ ಹಲಾರಾರದ . . ರಶಕಾಶಾರ್ (date) ಸಾಕ್ಷಣ (date ver) . ಶಿಕ್ಷ ಆಗ್ರತಕ ಅಕ್ಷ ಈ ಆತಿಕ್ಕಾ ಕೆಗಿತ್ ರತ್ತಿಕ ತಿರ್ಮಕ್ಷತ್ರ ಶಾರಾಶಾಶ್ವರ .

DESIGN DESCRIPTION FOR COMMON DATA AREA

General first factors

The following modules manipulate the requests to the CSA files which are in memory.

ZB#STRTN

during Startup processing

ZF#GEN/ZF#CSAS

during On-Line

ZB#SHUTN

during Shutdown Processing

Before the module ZF#GEN branches into ZF#CSAS, the FCTs are searched and bit \emptyset of ZF#FFLG2 is checked if it is on (X'8 \emptyset ').

If there is an occurrence, control is given to the start-label of ZF#CSAS with a BALR instruction. Following the BALR, a half-word value indicates from where control came.

Values of the Halfword

Value .	Set by/for
н'ø'	On-Line GETUP/PUT/GET
H'4'	On-Line Termination
H'8'	On-Line Cancel
	en de la companya de La companya de la co

Figure 2.1

2.2 STARTUP

2.2.1 Moment of Loading

Because the original CSA-files could be affected by a restart facility, loading of the records into memory starts immediately before Startup branches to Phase 2 of IMS 90.

Due to this reason, the high address of the storage pool has to be carried to this point (restoring of register 4 into ZB#MEHI at line 4490).

2.2.2 Memory Allocation for CSA

The item ZT#HPIBA of the THCB contains the first free location in the storage pool. This will be the Start Access of CSA. To allow IMS to load subsequent nonresident Action Programs, 8K are kept free at the top of storage pool. During the load of the CSA-records, this area is used as I/O-area for the CSA-files.

The CSA Start-Address (contents of ZT#HPIBA) will be stored into a new added fullword in the GEN-module (ZF#CSADR). ZT#HPIBA in the THCB will be updated after successful CDA-load.

. DESIGN DESCRIPTION FOR COMMON DATA AREA

2.1 General

The following modules manipulate the requests to the CSA files which are in memory.

ZB#STRTN

during Startup processing

ZF#GEN/ZF#CSAS

during On-Line

ZB#SHUTN

during Shutdown Processing

Before the module ZF#GEN branches into ZF#CSAS, the FCTs are searched and bit \emptyset of ZF#FFLG2 is checked if it is on (X'8 \emptyset ').

If there is an occurrence, control is given to the start-label of ZF#CSAS with a BALR instruction. Following the BALR, a half-word value indicates from where control came.

Values of the Halfword

Value .	Set by/for
H'Ø' H'4'	On-Line GETUP/PUT/GET On-Line Termination
н'8'	On-Line Cancel

Figure 2.1

2.2 STARTUP

2.2.1 Moment of Loading

Because the original CSA-files could be affected by a restart facility, loading of the records into memory starts immediately before Startup branches to Phase 2 of IMS 90.

Due to this reason, the high address of the storage pool has to be carried to this point (restoring of register 4 into ZB#MEHI at line 4490).

2.2.2 Memory Allocation for CSA

The item ZT#HPIBA of the THCB contains the first free location in the storage pool. This will be the Start Access of CSA. To allow IMS to load subsequent nonresident Action Programs, 8K are kept free at the top of storage pool. During the load of the CSA-records, this area is used as I/O-area for the CSA-files.

The CSA Start-Address (contents of ZT#HPIBA) will be stored into a new added fullword in the GEN-module (ZF#CSADR). ZT#HPIBA in the THCB will be updated after successful CDA-load.

2.2.3 Memory Layout before CSA-Loading

IMS 90

Root and Phases

2T#HPIBA points to

8K temporary storage

Input Staging Buffer.

Resident Action Prog.

free memory

_ZB#MEHI points to

Figure 2.2

2.2.4 Memory Layout after CSA-Loading

Remaining free storage

IMS 90

Root and Phases

CSA

ZT#HPIBA points to

Remaining free 8K temporary storage

Input Staging Buffer

Resident Action Prog.

during load-process used for I/O-area for the CSA-files

Figure 2.3

2.2.5 Loading Procedure

All FCT's are sequentially searched for the X'80' value in the ZF#FFLG2-byte.

If such an FCT exists, the corresponding file is read sequentially into memory. The format of the memory CSA-file is shown in Figure 2.4.

2.2.6 Format of the CSA in Memory

	Length	Name	Function	Values
4 1 .	1	ZF#DCSFT	Filetype	*
;	7	ZF#DCSFN	Filename	Copy of ZF#FNAME
	4	ZF#DCSFL	File length in bytes	
· [4	ZF#DCSIX	Index Table Address	**
	8 (2) V	•	Identification Record-length (if variable) Key and Data of Record	Copy of ZC#TTRID Nonzero = locked
	1		End of current file	X'FF'
	1		End of CSA	X'FF'

Figure 2.4

* See figure 2.5.

** See figure 2.6.

2.2.7 Filetype Byte in the CSA

	LOCK = TR	LOCK = UP	CUPDATE = YES	CUPDATE = NO	Filetype- Value
TRACE = YES TRACE = YES TRACE = YES TRACE = YES TRACE = NO TRACE = NO TRACE = NO RACE = NO	X X X	X X X	x x x x	x x x x	ØØ 2Ø 4Ø 6Ø 8Ø AØ CØ EØ

₹-

_ <u>Filetype-Byte</u> :	Bit Ø	.ø 1	TRACE = YES TRACE = NO
	Bit 1	ø 1	LOCK = TR (Recovery wanted) LOCK = UP (No recovery wanted)
	Bit 2	ø 1	CUPDATE = YES CUPDATE = NO
	Bits 3-6	alway	s Ø
	Bit 7	- <u>ø</u>	Fixed length records Variable length records

2.2.8 Failures

There are two possibilities a failure can occur during load:

- I/O Error on a specific CSA-file
- not enough memory available to hold all records of a file in memory

For both cases, the already used memory for that file is released and bit \emptyset of ZF#FFLG2 in the corresponding FCT is reset. (The CSA-file will be a normal file for the future processing).

A warning message is issued for the appropriate error and processing continues.

W	CSA	LOAD	I/O-ER	ROR	FOR	FILE	**			•
		· ·		en La			نىدى <u>ئارىمەسىمەر ئىللى</u> دەد ئات دەنىم ىلىكىدىن ئارى دىكا كارىدىن			-
W	CSA	LOAD	ERROR	FOR	FILE	3	**	MEMORY	EXHAUSTE)

2.2.9 Restriction on the I/O-Area-Size of CSA-Files

Due to the use of the "dummy-reserved" 8KB block as I/O-area for the CSA-files, blocks may not exceed this value.

<u>-Filetype-Byte</u> :	Bit Ø	.ø 1	TRACE = YES TRACE = NO	
• • • • • • • • • • • • • • • • • • •	Bit 1	ø 1 .	LOCK = TR (Recovery want	
	Bit 2	ø 1	CUPDATE = YES CUPDATE = NO	
	Bits 3-6	alway	s Ø	
	Bit 7	Ø	Fixed length records Variable length records	,

2.2.8 Failures

There are two possibilities a failure can occur during load:

- I/O Error on a specific CSA-file
- not enough memory available to hold all records of a file in memory

For both cases, the already used memory for that file is released and bit \emptyset of ZF#FFLG2 in the corresponding FCT is reset. (The CSA-file will be a normal file for the future processing).

A warning message is issued for the appropriate error and processing continues.

W	CSA	LOAD I/O-ER	ROR FO	R FILE	**		ن	•
	-	يعونها أرينا يرورانيسيسي أأأ ويحيد يجمعه	ورستون المع	and the second second second	ندن <u>از م</u> حم نها برند به درج انه درجانه دا ه مت		يعي برانوا بالماري والمجروري والمست	_
W	CSA	LOAD ERROR	FOR F	'ILE	**	MEMORY	EXHAUSTED	

2.2.9 Restriction on the I/O-Area-Size of CSA-Files

Due to the use of the "dummy-reserved" 8KB block as I/O-area for the CSA-files, blocks may not exceed this value.

3.1 General

2.2.33.3

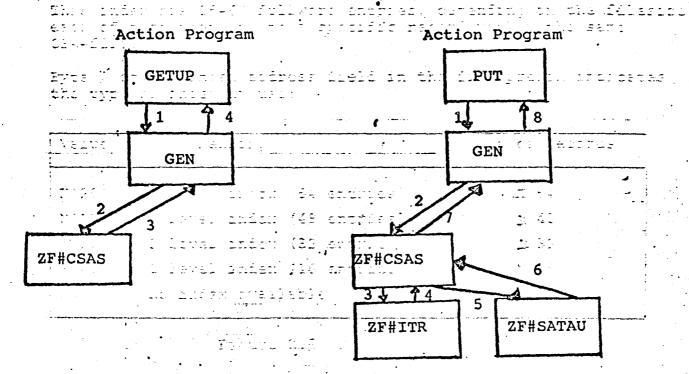
- ZF#CSAS is invoked by three kinds of requests:
 - normal requests like GET, GETUP or PUT from user is in a action programs in a line of fire dentities the lineary many
 - Rollback functions (GETUP and PUT) requested by the ZS#ROL action program
 - -jupon termination of an action/transaction is search for the specific sciences of the CSA the dileness and the fallelengue.

(LNG) [មិ ២៩២០១មិន (ខ ២៩២០២៦ ១៩៩២០២

2.3.2 Normal Requests for CSA

Record Rockey

a) Control flow for an update



GETUP

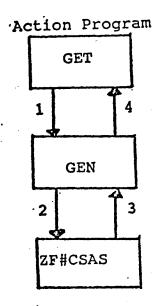
- 1. The action program issues a GETUP for a CSA file. Control is passed to the GEN module.
 terms requests like GET GETUP in BUT from vision
- 2. GEN passes control to ZF#CSAS, which locates and retrieves the specific record to the user's work area and manipulates the lock for the record.
- 3-4 Control returns to the action program.

PUT

- 1. The action program issues a PUT for a CSA file. Control is passed to the GEN module.
- 2. GEN passes control to ZF#CSAS, which locates and retrieves the specific record from the user's work area to the CSA.
- 3-4 Depending on the filetype-byte in the CSA, ZF#CSAS activates ZF#ITR to write before and after images to the trace file.
- 5-6 And/or before images to the audit file with the ZF#SATAU module.

When control returns to ZF#CSAS and the CUPDATE=YES option was specified, the specific record is written back to disk. Finally, the main-storage-record will be changed and, if lock=up is specified, the lock released.

7-8 Control returns to the action program.



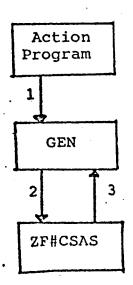
- The action program issues a GET for a CSA file. Control is passed to GEN.
- 2. GEN passes control to ZF#CSAS. The specific record is read out of the CSA and written into the specified user work area.
- 3-4 Control returns to the action program.

2.3.3 Rollback Functions from ZS#ROL

If bit 6 of ZC#TST1 in the terminal control table (TCT) $(x'\emptyset2' = Rollback in Process)$ is set, it can be determined if the request comes from ZU#ROL.

The same control flow applies as for a normal update request. (see 2.3.2), except that no before/after images are written to the audit/trace-file.

2.3.4 Termination of an Action/Transaction



- An action terminates. (e.g. PC, abnormal or normal termination). Control is passed to the GEN module.
- GEN activates ZF#CSAS to release all pending locks for this transaction.
- 3. Control is returned to GEN.

2.3.5 Record-Lock-Mechanism in Memory

As described in Figure 2.4, each record in memory is prefixed with an 8 byte identifier, which-tells from whom the record was locked. Every time a CSA-record is updated by a GETUP-Request, the specific record in memory is locked:

- the field ZC#TTRID of the TCT is copied into the identification field of the CSA-record.

During a GETUP, the record is not read from disk, although CUPDATE= YES is specified. If a user issues a GETUP on a locked record, the status and detailed status-codes in the PIB are set to 3, respectively 18.

The locks are released after a successful PUT or at Action/ Transaction Termination by clearing the identifier-field in the CSA to zeros.

Shutdown The ususes of mural to IT = ISA. The specific reconfus resi pur ĉi thi OSA and written Mnto the specifiej General warr work area.

When the system comes to Shutdown, all-records of every CSA memory file with CUPDATE=NO parameter specified are written back to disk sequentially. This is done by searching the FCT's for the ZZ#FFLG2 X'80' value.

Control in the name sale control (e)

ligon (lighters) as see, the cash in the carefrance

Failures

Because of a failure, two types of actions can occur:

- termination of processing for the current file
- termination of processing for all subsequent files
- a) Termination for current file
 - If an I/O-error occurs: W CSA UNLOAD I/O ERROR FOR FILE *
 - If a key in the CSA does not match with the key on disk: W CSA UNLOAD INVALID KEY FOR FILE *
 - If there are more records in memory than on disk: or vice versa W CSA UNLOAD INVALID # OF RECORDS FOR FILE *

b) Termination for all subsequent files

If a filename of a CSA-file in memory is inexistent in the FCT's

Control is returned to GEM.

- If the unload-processing cannot detect an end of CSA-indicator:

- the field posturid be the pot at identified in the OSA-

Puranç : GETTA (th) record (e) for record (b) so the record (e) and th

The locks are released after a sucre Franceouste lemmentuor by clearing OSA to demos.

Genéral

What whe spotem tongs to Shuthown, all fight who supplies to be paragroom species, sequency allys Thushis dend by square, green websel

禁事 もちゃっと

. **Be**odický i iso s isali one i hrví hrpývé isel.

INTERFACES

As described in 2.1., CSA functions are used in 3 different modules. The following entry and exit conditions apply:

.1 Entry Conditions

TO CEAR THE CART CARTON IN MENTING PERIOR

<u>ತಿದ್ದ ಕಿರ್ಧಿ ಆಯುತಿರದಲ್ಲಿ ಉಗರದಿಂದ ಅಲ್ಲಾಗೆ ಅವಡಿಗೆ</u>ದು. ಅ

Information	Startup	On-Line	Term/Cancel	Shutdown
Address of THCB	ZQ#THCB	RB\$	R8\$	RB\$
Address of TCT		RA\$	RA\$	
Address of FCT	ZB#SFCTI/ZF#FCTA	R3\$		
Address of DTF		R4\$.		
Function Code		rø\$		
Param-List-Address		R1\$		
Address of PIB		RC\$		
Address first free byte in storage	ZT#HPIBA			
Address last free byte in storage	ZB#MEHI (
Address I/O area CSA file	Contents ZB#MEHI +1		(<u>š</u>	
Address of CSA start		ZF#CSADR	ZF#CSADR	ZF#CSADR

4.2 Exit Conditions

After Startup, the fields ZT#HPIBA and ZF#CSADR are adjusted.

The registers remain unchanged except RE\$ and RF\$.

The return-address is picked up from a table in ZF#CSAS, depending on the displacement value following the BALR instruction in the calling module.

Coding is calling Program

Table in ZF#CSAS (Return-Address) ·

ON-LINE

BALR RE\$,RF\$
DC H'Ø'
BALR RE\$,RF\$
DC H'4'
BALR RE\$,RF\$
DC H'8'

DC V(ZF#GRTN) TERMINATION

DC V (ZF#GTCN1) CANCEL

DC V(ZF#GTT9)

4.3 <u>External References</u>

The following modules are activated during processing of ZF#CSAS:

___ZF#ITR

Write before/after images to Trace
Write before image to Audit

ZF#GSR (ZF#GEN)

ZF#SATAU

Allocate I/O area from Storage Pool

Entry/Exit Conditions for ZF#ITR	On-Line	Term Carcel	31.1.1
R3\$ Address of FCT user file	223	Les de la companya d	7.77
R4\$ Address user DTF_	F.J.S	FG.S	
RAS Address TCT CHESTOTA	i Tižž		
RB\$Address THCB	1.48		
RC\$ Address PIB	E.S.		•
DD\$KLE Must contain key length			-
DH\$SPB == Must contain record size	į nos 🦠		
DH\$KLC Must contain key location			
ZT#HRPLA * Must contain parameter list	address		:
ZT#HFWA Request-Packet (2 fullwords)			:
Byte 1: 83 ISAM before in 8B ISAM after im			
Byte 2: ØØ Update	ET CSITE	GFFCSADR (TACEALT
Byte 4-7: Address of key ar	gument		•

4.3.2 Entry/Exit Conditions for ZF#SATAU

R3\$ Address of FCT of the User File

R4\$ Address of the user DTF

RA\$ Address of the user TCT These registers can be restored from the GEN-S.A.

RC\$ Address of the user PIB

DH\$KARG must contain Address of the key argument ZF#MRN must contain record length of the user record ZF#REC must contain record address of user record ZT#HFWA must contain X'Ø3' to indicate ISAM request ZT#HFWA+4 must contain key argument-address DD\$KLE must contain key length

Entry/Exit Condition for ZF#GSR

R3\$: Address : FCT ac are activeled during proc

RB\$ Address THCB

ZF#FIGS must contain the size of the requested I/O area.

Gosta I Clarca Store Store & Poch

After exit of ZF#GSC: Whave belows image to Audio

R6\$ contains the address of the allocated I/O area

R7\$ is destroyed

.3.4 System Macros used for CSA Processing

Start sequential retrieval (Startup/Shutdown) SETL

ESETL End sequential retrieval (Startup/Shutdown)

GET Read record sequential (Startup/Shutdown)

PUT Update record sequential (Shutdown)

Read record randomly (On-Line) READ

Write record randomly (On-Line) WRITE "

WAITF Wait on I/O completion

hust conclus its 1971-111 255120

Muot contain parahéter Alisé address 2018 FF 5 12

Request-Recker MO ຢະນີ້ພື້ນດາຄົວ ZIGER.

\$3 'ISW' before intre

Isa kaiter anige

្សី ដូច្រីកស្ន Byte D:

Byve 42 : Address of Mey argument

DATA REFERENCES

This module does not need any new external data except a new added Flag-Byte in the File Control Table (ZF#FFLG2), which indicates whether a file is a CSA file or not.

ZF#FFLG2 - values: BIT \emptyset = normal file 1 = CSA file

Procentairs the address of modisk update

1 = disk update performed

The format of the generated comma-storage-area in main storage is seen in Figure 2.5.

Seri sequential retrieval.

Ladro sequential retrieval.

Ladro sed rational sequential (2)

Ladro sed rational sequential sed sequential (2)

STORAGE MAPPING

6.

The coding for the CSA processing is divided in three parts:

<u>Startup-Coding</u>. This coding is part of the ZB#STRTN module and for this only during Startup in memory.

On-Line-Coding. This coding is an own module (ZF#CSAS), which is part of the Root-Phase of IMS/90. The Start-Address of the CSA is stored in this module.

<u>Shutdown-Coding</u>. This coding is part of the ZB#SHUTN module, and for this only during shutdown in memory.

The Common-Storage-Area location is shown in Figure 2.3.

AVAILABILITY, RELIABILITY AND MAINTAINABILITY

Availability

As described in 2.2.8, a failure at the loading procedure does not force IMS to fail too. Processing can continue and it is the operator's decision to continue with that environment or not.

్ర కైట్లు కూడ్డు చేసిని క్రూడ్రందిందే.కూడి మేది చేదినాముకేంద్ర చేసి. కోటాక్ లో క్రామిక్ చేసి

7.2 Reliability

Access to CSA data is done in the very same way as a normal I/O request. (GET/PUT level). The user does not notice the existence of the CSA facility in his action programs.

Just so, the same error conditions occur as with normal requests.

For error messages, refer to 2.2.8 and 2.4.2.

7.3 Maintainability

Very few modules are touched by the CSA option. Beside the Startup and Shutdown processing, all requests are handled in an own module, which is called from ZF#GEN only. Thus, corrections and/or enhancements have to be done in four modules. (ZB#STRTN, ZB#SHUTN, ZF#GEN, ZF#CSAS).

IMPLEMENTATION CONSIDERATIONS

8 1 Type of Code

Serially reusable.

8.2 Source Language

Series 90 Assembler.

8.3 Coding Constraints

None.

8.4 System Generation

Will be supplied.

8.5 Conventions and Standards

All labels in the module start with the prefix ZF#CS followed by a 3 digit number.

egister usage:	R2\$	Cover
	R3\$	FCT
	R4\$	DTF
	R5\$-R9\$	free
•	RA\$	TCT
	RB\$	THCB
•	RC\$	PIB .
	RD\$	S.A.
•	RE\$+RF\$	branching register