



%CSCN Network service change notify

Project	G23-GSM Protocol Stack
Document Type	Technical Documentation
Title	% CSCN Network service change notify
Author	Thomas Schott
Creation Date	15.10.2003 17:27
Last Modified	3/8/2015 8:45:00 PM
ID and Version	8462.734.01.001
Status	Planned

Copyright © 2003 Texas Instruments, Inc. All rights reserved.

Texas Instruments Proprietary Information – Strictly Private

0 Document Control

© Copyright Texas Instruments, Inc. 2003
All rights reserved.

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Texas Instruments reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Texas Instruments. Texas Instruments accepts no liability for any loss or damage arising from the use of any information contained in this document.

The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Texas Instruments.

0.1 Document History

ID	Author	Date	Status
8462.734.01.001	Thomas Schott	15.10.2003	Planned

0.2 References, Abbreviations, Terms

[TI 7010.801] 7010.801, References and Vocabulary, Texas Instruments.

Table of Contents

1	Introduction	2
1.1	AT-command syntax	2
1.2	Internal ACI behaviour (MSC's).....	4
1.2.1	Calling the %CSCN from terminal located outside.....	4
1.2.2	Calling the %CSCN? from terminal located outside.....	4
1.2.3	Raising of unsolicited message %CCCN	5
1.2.4	Raising of unsolicited message %CSSN.....	5
2	Modifications in ACI	6
2.1	Modifications in ATI	6
2.1.1	Module ati_cmd.c	6
2.1.2	Module ati_ss.c.....	6
2.1.3	Module ati_ret.c	6
2.2	Modifications in CMH	7
2.2.1	Module aci_cmh.h	7
2.2.2	Module cmh.h.....	10
2.2.3	Module cmh_sss.c.....	11
2.2.4	Module cmh_ssq.c.....	11
2.2.5	Module cmh_ccr.h	11
2.2.6	Module cmh_ccr.c	12
2.2.7	Module cmh_ssr.h.....	13
2.2.8	Module cmh_ssr.c.....	13
2.3	Modifications in PSA	14
2.3.1	Module psa_ss,h.....	14
2.3.2	Module psa_cc,h.....	14
2.3.3	Module psa_ccf.c	15
2.3.4	Module psa_ssf.c.....	15
2.3.5	Module psa_ccp.c	15
2.3.6	Module psa_ccs.c.....	15
2.3.7	Module psa_ssp.c.....	16
2.3.8	Module psa_sss.c	16
3	Test.....	17
3.1	Target Test.....	17
3.1.1	Test during call setup.....	17
3.1.2	Test while using call forwarding feature	17
3.2	Simulation Test.....	18
3.2.1	Test of %CSCN settings.....	18
3.2.2	Test of %CCCN and %CSSN output	18

1 Introduction

This document describes the modifications to be done to provide the %CSCN network service change notify message.

The unsolicited messages %CCCN and %CSSN will give you an overview about network service changes. With the %CSCN AT-command this messages will be switched on or off.

1.1 AT-command syntax

Command	Possible response(s)
%CSCN=<ss>, <ss_dir>, <cc>, <cc_dir>	OK +CME ERROR: <err>
%CSCN?	%CSCN: <ss>, <ss_dir>, <cc>, <cc_dir>
%CSCN=?	%CSCN: (0,1), (0,1,2), (0,1) (0,1,2)

Description

This function allows the user to be notified when a network service is changed.

Some can be mapped with the command +CSSN(+CSSI/+CSSU) but some notifications required new AT Commands such as the ECT status. May be create a new unsolicited message for all these notifications or check one by one how these notifications can be found?

The %CSCN switched the trace output to the AT-command interface, separate by supplementary service and call control service.

If a service is switched on, the unsolicited commands will be raised if a network service change occurs:

- %CCCN displays the NW service change for call control service
- %CSSN displays the NW service change for supplementary service

Command	Possible response(s)
-	%CCCN: <direction>, <cId>, <FacilityIEString>
-	%CSSN: <direction>, <trans_part>, <FacilityIEString>

Defined values

<ss> :

0	Disable
1	Enable

<ss_dir> :

0	in
1	out
2	both

<cc> :

0	Disable
1	Enable

<cc_dir> :

0	in
1	out
2	both

<direction> :

0	in
1	out
2	both

<trans_part>

0	BeginTrans
1	FacTrans
2	EndTrans

<cld> :

integer type; call identification number as described in GSM 02.30 subclause 4.5.5.1

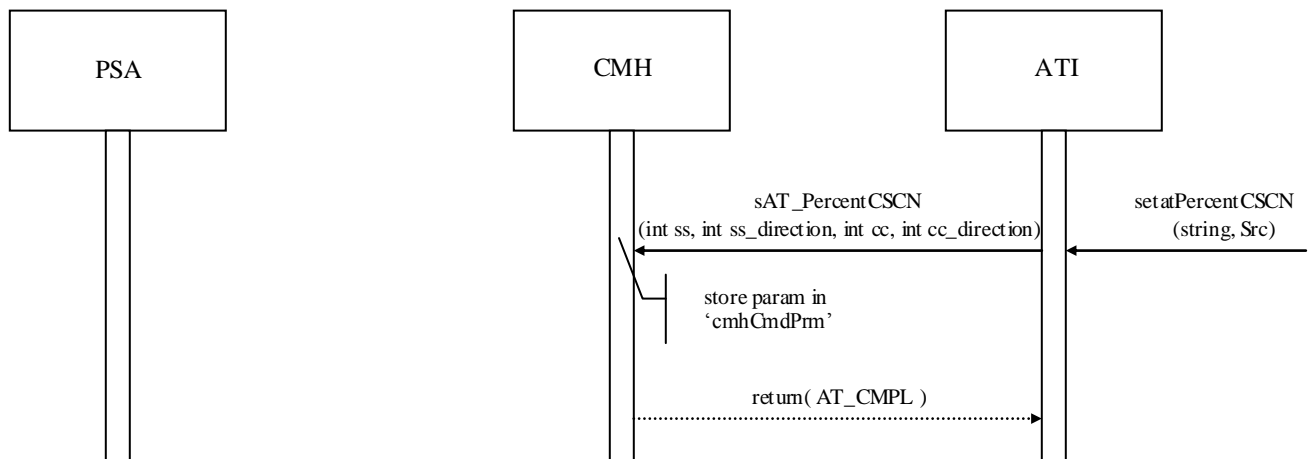
<FacilityIEString> :

“string” of information elements as described in ‘fac.doc’ (Air Interface Message Specification)

1.2 Internal ACI behaviour (MSC's)

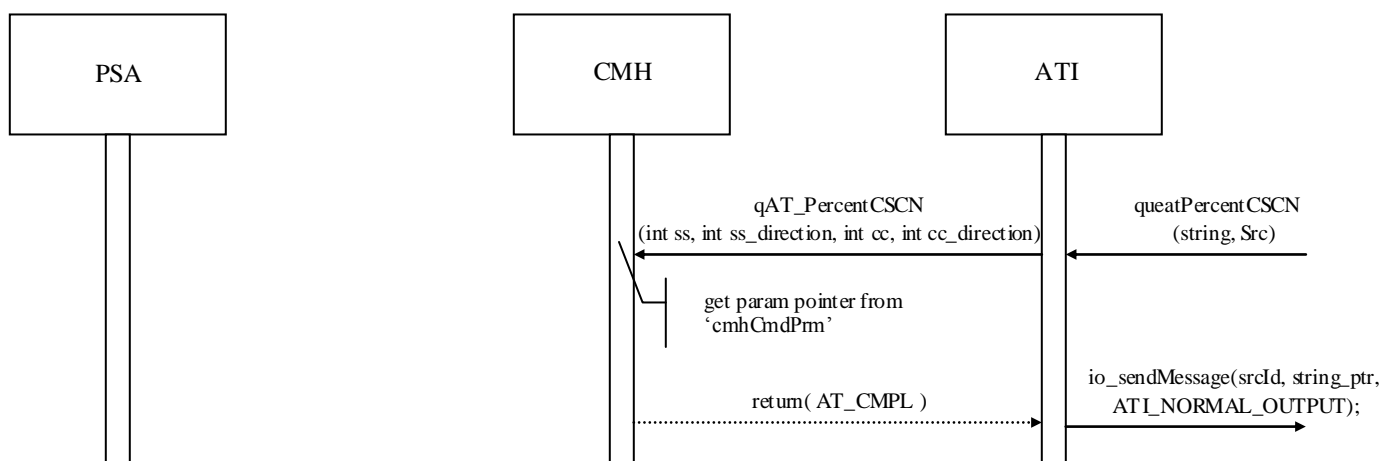
1.2.1 Calling the % CSCN from terminal located outside

After enter the %CSCN command at AT-Interface, for example by using the terminal program, the four parameter for switching on/off and direction will be stored into the shared parameter inside the command handler.



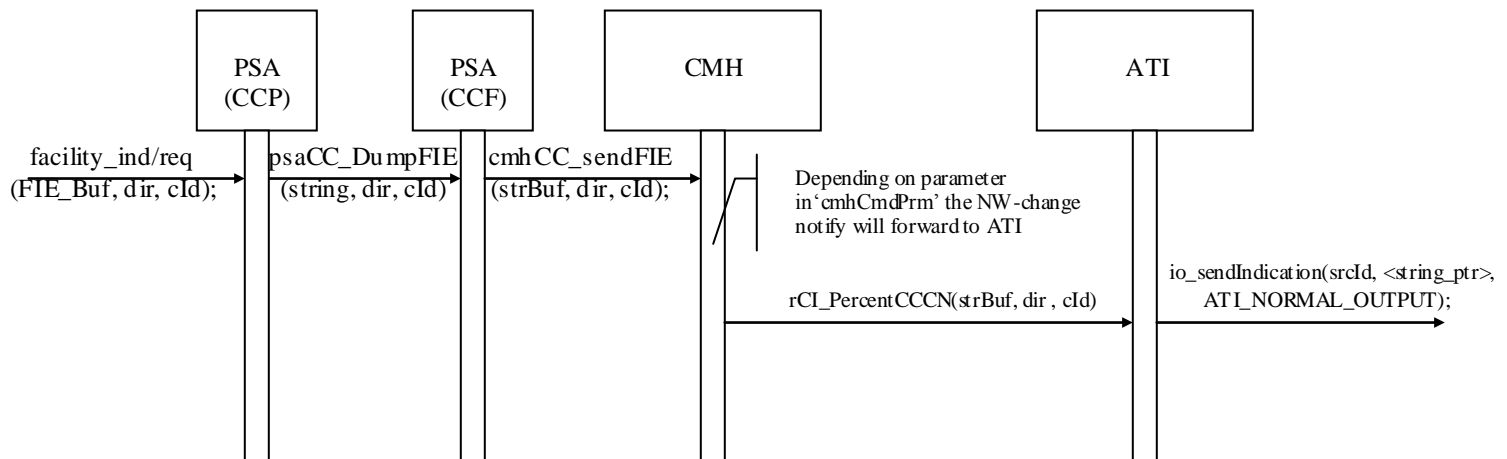
1.2.2 Calling the % CSCN? from terminal located outside

After enter the %CSCN? command at AT-Interface, for example by using the terminal program, the four parameter for switching on/off and direction will be get from command handler and display on the terminal screen.



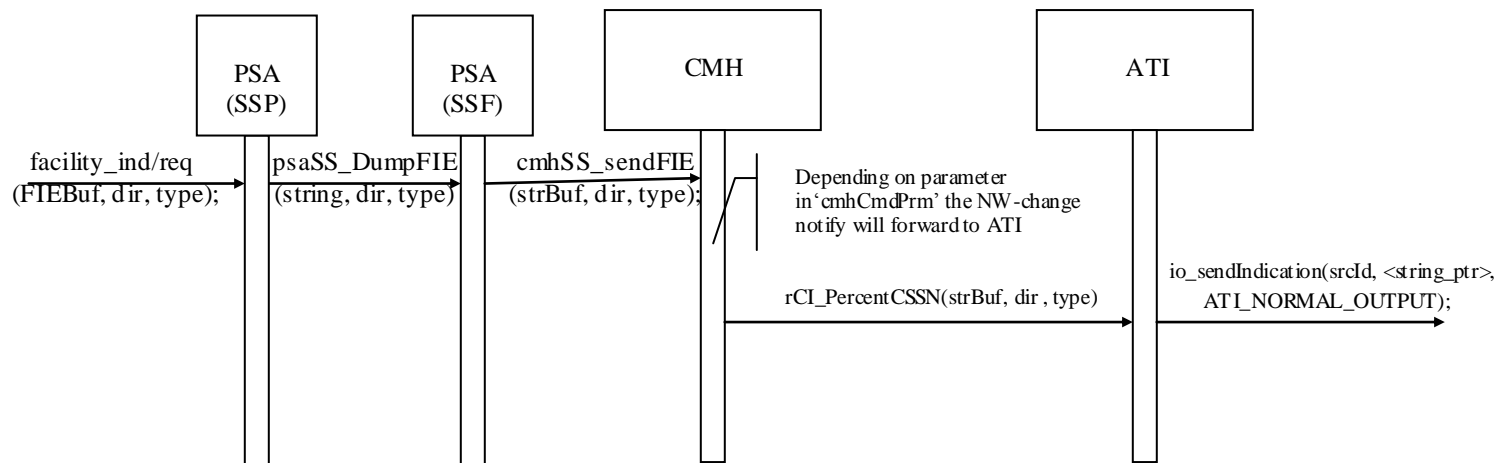
1.2.3 Raising of unsolicited message % CCCN

For each network service change a unsolicited message will be raised. Depending on settings in 'cmhCmdPrm' a string will be send to all connected sources to inform about this network change.



1.2.4 Raising of unsolicited message % CSSN

For each network service change a unsolicited message will be raised. Depending on settings in 'cmhCmdPrm' a string will be send to all connected sources to inform about this network change.



2 Modifications in ACI

2.1 Modifications in ATI

2.1.1 Module ati_cmd.c

- Additional prototypes
 - `EXTERN T_ATI_RSLT setatPercentCSCN (char *cl, UBYTE srcId);`
 - `EXTERN T_ATI_RSLT queatPercentCSCN (char *cl, UBYTE srcId);`

- Additional row in

```
LOCAL const ATCommand cmds [] = {  
:  
{"% CSCN", AT_CMD_CSCN, setatPercentCSCN, test_gen, queatPercentCSCN, "%s: (0,1),(0,1),(0,1),(0,1)"},  
:  
}
```

2.1.2 Module ati_ss.c

- Implement new functions
 - `T_ATI_RSLT setatPercentCSCN (char *cl, UBYTE srcId) {
:
sAT_PercentCSCN(int ss, int ss_direction, int cc, int cc_direction);
:
}`
 - `T_ATI_RSLT queatPercentCSCN (char *cl, UBYTE srcId) {
:
qAT_PercentCSCN(&ss, &ss_direction, &cc, &cc_direction);
:
ci_remTrailCom(string_ptr, pos);
io_sendMessage(srcId, string_ptr, ATI_NORMAL_OUTPUT);
}`

2.1.3 Module ati_ret.c

- Implement new functions to format output string and put it out to terminal
 - `GLOBAL void rCI_PercentCCCN (char *string, T_FAC_DIR dir, short cId)
{
 UBYTE srcId = srcId_cb;
 sprintf(/* here format string */);
 io_sendIndication(srcId, <string_ptr>, ATI_NORMAL_OUTPUT);
}`

2.2 Modifications in CMH

2.2.1 Module aci_cmh.h

- Additional row before AT_CMD_MAX → define new AT-command
typedef struct {
:
AT_CMD_CSCN,
AT_CMD_MAX
} T_ACI_AT_CMD
- New enum type to extending psa<XX> DumpFIE –functions → facility type for %CSSN
typedef enum
{
CSCN_FACILITY_TRANS_TYPE_BEGIN = 0,
CSCN_FACILITY_TRANS_TYPE,
CSCN_FACILITY_TRANS_TYPE_END,
CSCN_FACILITY_TRANS_TYPE_MAX
} T_ACI_FAC_TRANS_TYPE;
- New enum type to extending psa<XX> DumpFIE –functions → direction for %CSSN and &CCCN
typedef enum
{
CSCN_FACILITY_DIRECTION_IN = 0,
CSCN_FACILITY_DIRECTION_OUT,
CSCN_FACILITY_DIRECTION_MAX
} T_ACI_FAC_DIR;
- new prototypes to format terminal output string and put it out to terminal
 - EXTERN void rCI_PercentCCCN (char *string, T_ACI_FAC_DIR dir, short cId);
 - EXTERN void rCI_PercentCSSN (char *string, T_FAC_DIR uiDir,
T_ACI_FAC_TRANS_TYPE uiFacTransType);

- new CC command handler shared-parameter types

typedef enum

```
{  
    CC_CSCN_MOD_STATE_OFF = 0,  
    CC_CSCN_MOD_STATE_ON,  
    CC_CSCN_MOD_STATE_MAX  
} T_ACI_CC_CSCN_MOD_STATE;
```

typedef enum

```
{  
    CC_CSCN_MOD_DIR_IN = 0,  
    CC_CSCN_MOD_DIR_OUT,  
    CC_CSCN_MOD_DIR_MAX  
} T_ACI_CC_CSCN_MOD_DIRECTION;
```

typedef struct

```
{  
    T_ACI_CC_CSCN_MOD_STATE CcCSCNModeState;  
    T_ACI_CC_CSCN_MOD_DIRECTION CcCSCNModeDirection;  
} T_ACI_CC_CSCN_MOD;
```

- new prototypes for command handler control functions

```
EXTERN T_ACI_RETURN sAT_PercentCSCN ( T_ACI_CMD_SRC srcId,  
    short ss_switch,  
    short ss_direction,  
    short cc_switch,  
    short cc_direction );
```

```
EXTERN T_ACI_RETURN sAT_PercentCSCN ( T_ACI_CMD_SRC srcId,  
    short *ss_switch,  
    short *ss_direction,  
    short *cc_switch,  
    short *cc_direction );
```

- new SS command handler shared-parameter types

typedef enum

```
{  
    SS_CSCN_MOD_STATE_OFF = 0,  
    SS_CSCN_MOD_STATE_ON,  
    SS_CSCN_MOD_STATE_MAX  
} T_ACI_SS_CSCN_MOD_STATE;
```

typedef enum

```
{  
    SS_CSCN_MOD_DIR_IN= 0,  
    SS_CSCN_MOD_DIR_OUT,  
    SS_CSCN_MOD_DIR_MAX  
} T_ACI_SS_CSCN_MOD_DIRECTION;
```

typedef struct

```
{  
    T_ACI_SS_CSCN_MOD_STATE CcCSCNModeState;  
    T_ACI_SS_CSCN_MOD_STATE CcCSCNModeDirection;  
} T_ACI_SS_CSCN_MOD;
```

2.2.2 Module cmh.h

- Additional command-handler parameter in cmh-shared parameter

- additional CC command handler parameter

```
typedef struct ccCmdPrm      /* command parameters related to CC */
{
    :
    T_ACI_CC_CSCN_MOD CSCNcc_mode;
} T_CC_CMD_PRM;
```

- additional SS command handler parameter

```
typedef struct ssCmdPrm      /* command parameters related to SS */
{
    :
    T_ACI_SS_CSCN_MOD CSCNss_mode;
} T_SS_CMD_PRM;
```

2.2.3 Module cmh_sss.c

- New function must be implement for storing the parameter in command-handler shared parameter

```
sAT_PercentCSCN(int ss, int ss_direction, int cc, int cc_direction)
{
    /* store the data into the 'cmhPrm'-shared parameter */
}
```

2.2.4 Module cmh_ssqc

- New function must be implement to get the parameter pointer in command-handler shared parameter

```
qAT_PercentCSCN(int *ss, int *ss_direction, int *cc, int *cc_direction)
{
    /* get the pointer to the 'cmhPrm'-shared parameter */
}
```

2.2.5 Module cmh_ccr.h

- New function prototype to forwarding the facilities to ATI against settings in 'cmhCmdPrm'

```
GLOBAL SHORT cmhCC_sendFIE(strBuf, dir , cId);
```

2.2.6 Module cmh_ccr.c

- New function must be implement to forwarding the facilities to ATI against settings in 'cmhCmdPrm'

```
GLOBAL SHORT cmhCC_sendFIE(strBuf, dir , cId)
{
    /* check settings in 'cmhCmdPrm' here against 'dir' and 'cId' */
    for (idx = 0; idx < CMD_SRC_MAX; idx++)
    {
        if (IS_SRC_USED (idx))
        {
            if (idx EQ CMD_SRC_LCL) /* local source == always MFW, sometimes SMI */
            {
                #if defined SMI OR defined MFW OR defined FF_MMI_RIV
                    rAT_PercentCCCN(strBuf, dir , cId);
                #endif
            }
            else
            {
                #ifdef AT_INTERPRETER
                    srcId_cb = (UBYTE)idx;
                    rCI_PercentCCCN(strBuf, dir , cId);
                #endif
            }
            return( 0 );
        }
    }
}
```

2.2.7 Module cmh_ssr.h

- New function prototype to forwarding the facilities to ATI against settings in 'cmhCmdPrm'

GLOBAL SHORT cmhSS_sendFIE(strBuf, dir , type);

2.2.8 Module cmh_ssr.c

- New function must be implement to forwarding the facilities to ATI against settings in 'cmhCmdPrm'

GLOBAL SHORT cmhSS_sendFIE(strBuf, dir , type)

```
{
    /* check settings in 'cmhCmdPrm' here against 'dir' and 'type' */
    for (idx = 0; idx < CMD_SRC_MAX; idx++)
    {
        if (IS_SRC_USED (idx))
        {
            if (idx EQ CMD_SRC_LCL) /* local source == always MFW, sometimes SMI */
            {
                #if defined SMI OR defined MFW OR defined FF_MMI_RIV
                    rAT_PercentCSSN(strBuf, dir , type);
                #endif
            }
            else
            {
                #ifdef AT_INTERPRETER
                    srcId_cb = (UBYTE)idx;
                    rCI_PercentCSSN(strBuf, dir , type);
                #endif
            }
            return( 0 );
        }
    }
}
```

2.3 Modifications in PSA

2.3.1 Module **psa_ss,h**

- modify prototype of 'GLOBAL void psaSS_DumpFIE (T_fac_inf * fie,)'
GLOBAL void psaCC_DumpFIE (T_fac_inf * fie,
T_PSA_FACILITY_DIRECTION_ENUM tFacDir,
T_PSA_FAC_TRANS_ENUM tFacTrans);

2.3.2 Module **psa_cc,h**

- modify prototype of 'GLOBAL void psaCC_DumpFIE (T_fac_inf * fie,)'
GLOBAL void psaCC_DumpFIE (T_fac_inf * fie,
T_PSA_FACILITY_DIRECTION_ENUM tFacDir,
short cId);

2.3.3 Module **psa_ccf.c**

- modify the function 'GLOBAL void psaCC_DumpFIE (...)'
GLOBAL void psaCC_DumpFIE (...)
{
:
TRACE_EVENT_P1("%s", strBuf);
cmhCC_sendFIE(strBuf, dir, cId); /* trying to send the facility informations to terminal */
}

2.3.4 Module **psa_ss.c**

- modify the function 'GLOBAL void psaSS_DumpFIE (...)'
GLOBAL void psaSS_DumpFIE (...)
{
:
TRACE_EVENT_P1("%s", strBuf);
cmhSS_sendFIE(strBuf, dir, type); /* trying to send the facility informations to terminal */
}

2.3.5 Module **psa_ccp.c**

- Modify function psa_mncc_facility_ind(T MNCC FACILITY_IND *mncc_facility_ind)
→ modify call of psaCC_DumpFIE (&mncc_facility_ind -> fac_inf); here
psaCC_DumpFIE (&mncc_facility_ind -> fac_inf, FACILITY_DIRECTION_IN, cId);

2.3.6 Module **psa_ccs.c**

- Modify function LOCAL SHORT psaCC_SendSS (SHORT cId)
→ modify call of psaCC_DumpFIE (&mncc_facility_req -> fac_inf); here
psaCC_DumpFIE (&mncc_facility_ind -> fac_inf, FACILITY_DIRECTION_OUT, cId);

2.3.7 Module **psa_ssp.c**

- Modify function GLOBAL const void psa_mnss_begin_ind(T MNSS_BEGIN_IND *mnss_begin_ind)
→ modify call of psaSS_DumpFIE (&mnss_begin_ind->fac_inf); here
psaSS_DumpFIE(&mnss_begin_ind->fac_inf, **FACILITY_DIRECTION_IN, BEGIN_FAC_TRANS**);
- Modify function GLOBAL const void psa_mnss_end_ind(T MNSS_END_IND *mnss_end_ind)
→ modify call of psaSS_DumpFIE (&mnss_end_ind->fac_inf); here
psaSS_DumpFIE(&mnss_end_ind->fac_inf, **FACILITY_DIRECTION_IN, END_FAC_TRANS**);
- Modify funct. GLOBAL const void psa_mnss_facility_ind(T MNSS_FACILITY_IND *mnss_facility_ind)
→ modify call of psaSS_DumpFIE (&mnss_facility_ind->fac_inf); here
psaSS_DumpFIE(&mnss_facility_ind->fac_inf, **FACILITY_DIRECTION_IN, FAC_TRANS**);

2.3.8 Module **psa_sss.c**

- Modify function GLOBAL SHORT psaSS_NewTrns (SHORT sId)
→ modify call of psaSS_DumpFIE (&mnss_begin_req->fac_inf); here
psaSS_DumpFIE(&mnss_begin_req->fac_inf, **FACILITY_DIRECTION_OUT, BEGIN_FAC_TRANS**);
- Modify function GLOBAL SHORT psaSS_EndTrns (SHORT sId)
→ modify call of psaSS_DumpFIE (&mnss_end_req->fac_inf); here
psaSS_DumpFIE(&mnss_end_req->fac_inf, **FACILITY_DIRECTION_OUT, END_FAC_TRANS**);
- Modify function GLOBAL SHORT psaSS_CntTrns (SHORT sId)
→ modify call of psaSS_DumpFIE (&mnss_facility_req->fac_inf); here
psaSS_DumpFIE(&mnss_facility_req->fac_inf, **FACILITY_DIRECTION_OUT, FAC_TRANS**);

3 Test

3.1 Target Test

Using the terminal application will do the test at the target. In order to receive the FIE-Strings a call can be established and the display of the FIE-strings must be switched on by using the AT+CSCN command.

Furthermore additional FIE strings can be raised by using the call forwarding feature with the AT+CLCC command during an established call.

3.1.1 Test during call setup

```
AT-Command Interpreter ready
at+cfun=1
OK
at+cops=0
OK
at+cmee=2
OK
at%cscn=1,2,1,2
OK

OK
atd03039831546;
OK
%CCCN: 0,0,"A10E0201010201103006810128840107"
ath
OK
```

3.1.2 Test while using call forwarding feature

```
at+ccfc=5,0
%CSSN: 1,0,"A10B02010002010D3003040128"
%CSSN: 0,2,"A214020100300F02010DA00A04012830053003840104"
at+ccfc=5,1,"03039831546"
%CSSN: 1,0,"A10B02010402010C3003040128"
%CSSN: 0,2,"A21F020104301A02010CA015040128301030068301108401073006830160840107"
```

3.2 Simulation Test

The simulated test will be done by four Windows simulation test cases.

3.2.1 Test of % CSCN settings

The ACISS090 test case are the responsible to test all possible settings to control this feature.

3.2.2 Test of % CCCN and % CSSN output

Test's ACISS091, ACISS092 and ACISS093 are the responsible to test the correct output of the unsolicited responses %CCCN and %CSSN which display the FIE strings.