



---

## LLD Automatic Call Repeat

---

Project	G23-GSM Protocol Stack
Document Type	AT Command Interface Description
Title	LLD Automatic Call Repeat
Author	TI Employee
Creation Date	6 November, 2003
Last Modified	6 November, 2003
ID and Version	8462.719.03.001
Status	Being Processed

Copyright © 2001 Texas Instruments, Inc. All rights reserved.

**Texas Instruments Proprietary Information – Strictly Private**

## Document Control

© Copyright Texas Instruments, Inc. 2001  
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
8462.719.03.001	RM	6 November, 2003	Being Processed

### 0.2 References, Abbreviations, Terms

- [1] 7010.801, References and Vocabulary, Texas Instruments.
- [2] GSM 2.07, Mobile Stations (MS) features

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>1 General requirements in ACI.....</b>	<b>4</b>
<b>2 Effort in ACI.....</b>	<b>5</b>
<b>3 Error indication vs. call repeat attempts.....</b>	<b>6</b>
<b>4 AT-command syntax.....</b>	<b>7</b>
4.1 AT%RDL .....	7
4.2 AT%RDLB.....	7
<b>5 MSC's for ACI support of automatic calling repeat.....</b>	<b>9</b>
5.1 MSC's of AT command AT%RDL .....	9
5.1.1 Set automatic calling repeat mode .....	9
5.1.2 Query automatic calling repeat mode.....	9
5.2 MSC's of AT command AT%RDLB .....	10
5.2.1 Deleting of black list .....	10
5.2.2 Query black list entries .....	10
5.3 Further MSC's .....	11
5.3.1 Check Causes of unsuccessful Call Attempts .....	11
5.3.2 Elapsing of Redial Timer .....	12
5.3.3 Check Redial State for MOC, MTC, ATH and AT+CHUP .....	12
<b>6 Modifications in ACI .....</b>	<b>14</b>
6.1 Extension of Existing Data Types.....	14
6.1.1 Module ati_cmd.c .....	14
6.1.1.1 Table cmds[] .....	14
6.1.2 Module aci_cmh.h .....	14
6.1.2.1 Type T_ACI_AT_CMD .....	14
6.1.3 Module cmh.h .....	14
6.1.3.1 Type RAT_ID.....	14
6.1.4 Module psa_cc.h.....	14
6.1.4.1 Type T_CC_CALL_TBL.....	14
6.1.4.2 Type T_CC_CLTP .....	15
6.2 Modification of Existing Functions .....	15
6.2.1 Module cmh_ccf.c .....	15
6.2.1.1 cmhCC_SndDiscRsn ().....	15
6.2.2 Module cmh_ccr.c .....	15
6.2.2.1 cmhCC_CallReleased () .....	15
6.2.3 Module cmh_ccq.c.....	15
6.2.3.1 qAT_PlusCLCC() .....	15
6.2.4 Module cmh_ccs.c.....	15
6.2.4.1 cmhCC_Dial () .....	15
6.2.4.2 sAT_H ().....	15
6.2.4.3 sAT_PlusCHUP () .....	16
6.2.5 Module cmh_fc .....	16
6.2.5.1 cmh_timeout () .....	16
6.2.6 Module cmh_mmis.c .....	16
6.2.6.1 cmhMM_handleAudioTone ().....	16
6.2.7 Module aci_pei.c.....	16
6.2.7.1 pei_init () .....	16
6.2.7.2 pei_exit () .....	16
6.2.8 Module psa_ccf.c.....	16
6.2.8.1 psaCC_InitCtbNtry ().....	16
6.2.9 Module psa_ccp.c.....	16
6.2.9.1 psa_mncc_disconnect_ind ().....	16
6.2.9.2 psa_mncc_reject_ind ().....	16
6.2.9.3 psa_mncc_release_cnf () .....	16

6.2.9.4	psa_mncc_release_ind () .....	16
6.2.9.5	psa_mncc_setup_cnf () .....	16
6.2.9.6	psa_mncc_setup_cnf () .....	17
6.2.9.7	psa_mncc_setup_req () .....	17
<b>7</b>	<b>New Functions and Data Structures.....</b>	<b>18</b>
7.1	Predefined Values .....	18
7.1.1	Module l4_tim.h .....	18
7.1.1.1	Timer.....	18
7.1.2	Module aci_cmh.h .....	18
7.1.2.1	Constants.....	18
7.2	New Data Types .....	18
7.2.1	Module aci_cmh.h .....	18
7.2.1.1	T_ACI_CC_BLACKL_ENTRY.....	18
7.2.1.2	T_ACI_CC_RDL_BLACKL_STATE.....	18
7.2.1.3	T_ACI_CC_RDL_FFS .....	19
7.2.1.4	T_ACI_CC_REDIAL_BLACKL .....	19
7.2.1.5	T_ACI_CC_REDIAL_BLMODE .....	19
7.2.1.6	T_ACI_CC_REDIAL_MODE.....	19
7.2.1.7	T_ACI_CC_REDIAL_NOTIF .....	19
7.2.1.8	T_ACI_CC_REDIAL_STATE .....	19
7.2.1.9	T_ACI_CC_REDIAL_PAR.....	20
7.2.2	Extending of FFS .....	20
7.2.2.1	RedialMode.....	20
7.3	New Functions.....	20
7.3.1	Module ati_cc.c .....	20
7.3.1.1	setatPercentRDL .....	20
7.3.1.2	qeataPercentRDL .....	21
7.3.1.3	setatPercentRDLB .....	21
7.3.1.4	qeataPercentRDLB .....	21
7.3.2	Module ati_ret.c .....	21
7.3.2.1	rCI_PercentRDL .....	21
7.3.2.2	rCI_PercentRDLB .....	22
7.3.3	Module cmh_ccf.c .....	22
7.3.3.1	cmhCC_redialCheck .....	22
7.3.3.2	cmhCC_redialChkBlackl.....	22
7.3.3.3	cmhCC_rd_mode_FFS .....	22
7.3.4	Module cmh_ccr.c .....	23
7.3.4.1	cmhCC_redialTimeout.....	23
7.3.5	Module cmh_ccs.c .....	23
7.3.5.1	rdlPrm_exit .....	23
7.3.5.2	rdlPrm_init .....	23
7.3.5.3	sAT_PercentRDL.....	23
7.3.5.4	sAT_PercentRDLB.....	24
7.3.6	Module cmh_ccq.c.....	24
7.3.6.1	qAT_PercentRDL.....	24
7.3.6.2	qAT_PercentRDLB .....	24
7.3.7	Module smi_dmy.c .....	25
7.3.7.1	rAT_PercentRDL.....	25
7.3.7.2	rAT_PercentRDLB .....	25
<b>8</b>	<b>Test Strategy .....</b>	<b>26</b>
8.1	Tests under Windows .....	26
8.2	Tests on target.....	26

## **Introduction**

This document contains the needed effort for the ACI support of the feature “Automatic Calling Repeat”. The behaviour for “Automatic calling repeat call attempt restrictions” is described in GSM 02.07, Annex A. Up to now this feature was implemented in MFW.

## **1 General requirements in ACI**

For support of automatic calling repeat of voice calls two new AT commands are necessary:

1. Setting redial mode ON/OFF of automatic call repeat: AT%RDL
2. Show /delete of blacklist of phone numbers: AT% RDLB

The redialling of the last unsuccessful outgoing phone number depends on error indications described in [2]. These indications requests different redialling procedures.

The ACI has to manage a "black list" of outgoing phone numbers (at least 8). Their use is forbidden until the user deletes the list. If the black list is full the automatic calling repeat procedure is stopped until deleting of entries.

The entry in the black list depends on both parameters redial counter and call attempt error indication. The counter is reset after successful connection and after phone reset. The black list is deleted after phone switch on.

For the handling of different redialling procedures ACI needs some different redial timer described in [2].

Each outgoing call is checked against the black list entries.

The redialling procedure is stopped for:

1. Each outgoing call (if not black listed) activated by user
2. Each incoming call
3. User set redial mode OFF
4. User switches off redialling via ATH and via AT+CHUP

The user can decide whether he want to get redialling state indications via the both new AT commands.

ACI has to store/restore the redial mode in FFS.

Note: SAT related redialling procedures are not focussed.

## 2 Effort in ACI

Description	New Commands/Functions	Comments
Implementation AT commands Redial mode ON/OFF, unsolicited indication on/off	AT%RDL setatPercentRDL sAT_PercentRDL qetatPercentRDL qAT_PercentRDL	Set/Test/Query functions ATI, CMH Testcases
Implementation AT commands Show / Delete black list entries, unsolicited indication on/off	AT%RDLB setatPercentRDLB sAT_PercentRDLB qetatPercentRDLB qAT_PercentRDLB	Set/Test/Query functions ATI, CMH Testcases
Evaluation of unsuccessful call attempts, add phone number in black list, start redial timer	cmhCC_redialCheck	Check a lot of error indications considering outgoing calls in PSA
Implementation of redial timer administration	cmhCC_redialTimeout	New timer handles and function in CMH
Check black list against outgoing phone number; abort redialling procedure in case of an outgoing call		Check is to be implemented for each mobile outgoing call in CMH
Send if requested redial state indication to TE		
Check redial timer for received mobile terminated calls		Check is to be implemented for each mobile incoming call in PSA
User aborts redialling mobile outgoing call via AT%RDL		Check call type CT_MOC_RDL in CMH
Store of redial mode in FFS	cmhCC_rd_mode_FFS	Test only on target possible

### 3 Error indication vs. call repeat attempts

The relevant error indications of unsuccessful call attempts are[2]:

1. "Busy destination":

Cause number	17	User busy
--------------	----	-----------

2. "Unobtainable destination - temporary":

Cause number	18	No user responding
	19	User alerting, no answer
	27	Destination out of order
	34	No circuit/channel available
	41	Temporary failure
	42	Switching Equipment congestion
	44	Requested circuit/channel not available
	47	Resources unavailable, unspecified

3. "Unobtainable destination - permanent/long term":

Cause number	1	Unassigned (unallocated) number
	3	No route to destination
	22	Number changed
	28	Invalid number format (uncomplete number)
	38	Network out of order.

This table describes the different redialling procedures:

Note: For the categories 1 and 2 above, n shall be 10; for category 3, n shall be 1.

call attempts	Minimum duration between Call attempt
Initial call attempt	-
1st repeat attempt	5 sec
2nd repeat attempt	1 min
3rd repeat attempt	1 min
4th repeat attempt	1 min
5th repeat attempt	3 min
	.
	.
nth repeat attempt	3 min

The conditions for the entry of a number in the black list are the following:

1. **Error indications of category 1 + 2:** the number of 10 unsuccessful call attempts is achieved and the black list is not full yet
2. **Error indications of category 3:** the first call repeat was unsuccessful and the black list is not full yet



## 4 AT-command syntax

### 4.1 AT%RDL

Command	Possible response(s)
%RDL=[<mode>,[<n>]]	+CME ERROR: <err>
%RDL?	%RDL: <mode>, <n>
%RDL=?	%RDL: (supported <mode >, supported <n>s)

#### Description

This function allows the user to set and reset the automatic calling repeat of unsuccessful outgoing call attempts. When <n>=1 the unsolicited result %RDL: <state> is sent to TE.

#### Defined values:

<mode> :  
0      Disable automatic calling repeat  
1      Enable automatic calling repeat  
<n>:  
0      No redial state indication to TE  
1      Redial state indication to TE  
<state>:  
2      Redialling timer is started  
3      Automatic call repeat is started  
4      Call attempt fails  
5      Call attempt successful  
6      Automatic call repeat is stopped

### 4.2 AT%RDLB

Command	Possible response(s)
%RDLB=[<mode>,[<n>]]	+CME ERROR: <err>
%RDLB?	%RDLB: <n> %RDLB: <counter>, <number>[[...] <CR><LF>%RDLB: <number>]] +CME ERROR: <err>
%RDLB=?	%RDLB: (list of supported <max counter>, supported <n>s)

#### Description

This function allows the user to show and delete the entries of the black list. When <n>=1 the unsolicited result %RDLB: <state> is sent to TE.

#### Defined values:

<mode> :  
0      No delete black list  
1      Delete black list

<n>:

- 0 No black list state indication to TE
- 1 Black list state indication to TE

<counter> :

integer type values : number of black list entries

<number>:

string type: phone number

<max counter >

integer type values : maximal number of black list entries

<state>:

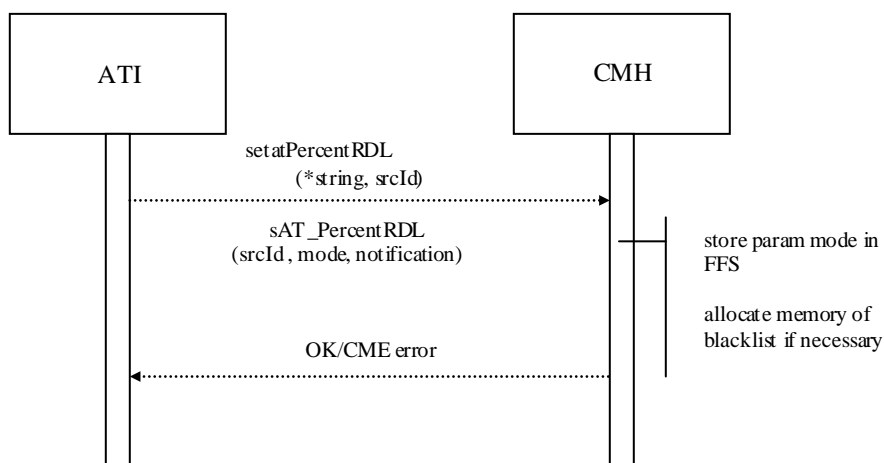
- 2 Black list is full
- 3 Phone number entry is set in black list
- 4 Phone number is an black list entry

## 5 MSC's for ACI support of automatic calling repeat

### 5.1 MSC's of AT command AT%RDL

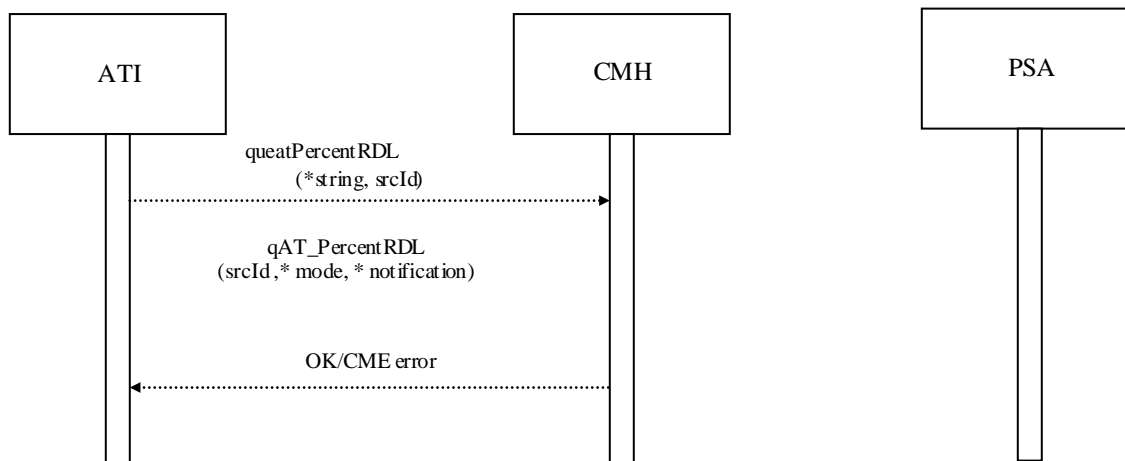
#### 5.1.1 Set automatic calling repeat mode

With the set AT command %RDL the chosen mode and notification state will be stored into the CC redial parameter and in the Flash File System (FFS). Furthermore memory of blacklist is allocated if it does not exist yet.



#### 5.1.2 Query automatic calling repeat mode

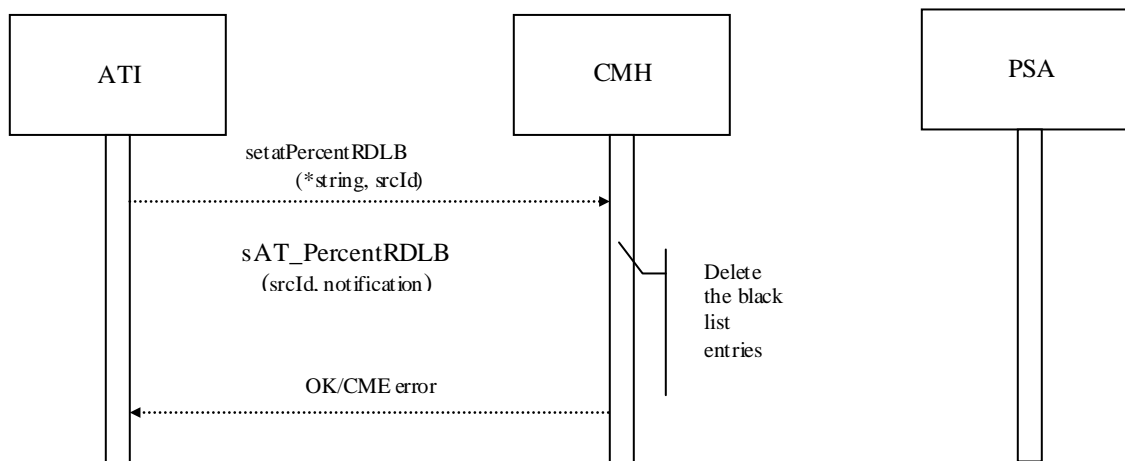
With the query AT command %RDL the chosen mode and notification status will be interrogated.



## 5.2 MSC's of AT command AT%RDLB

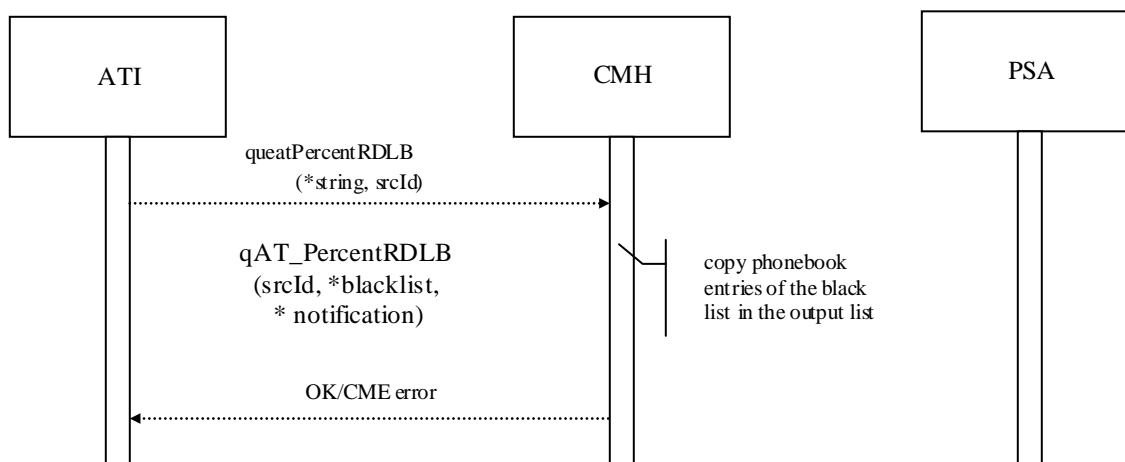
### 5.2.1 Deleting of black list

With the AT command %RDLB the user can delete all of phone number entries in the black list and or set the notification status of black list



### 5.2.2 Query black list entries

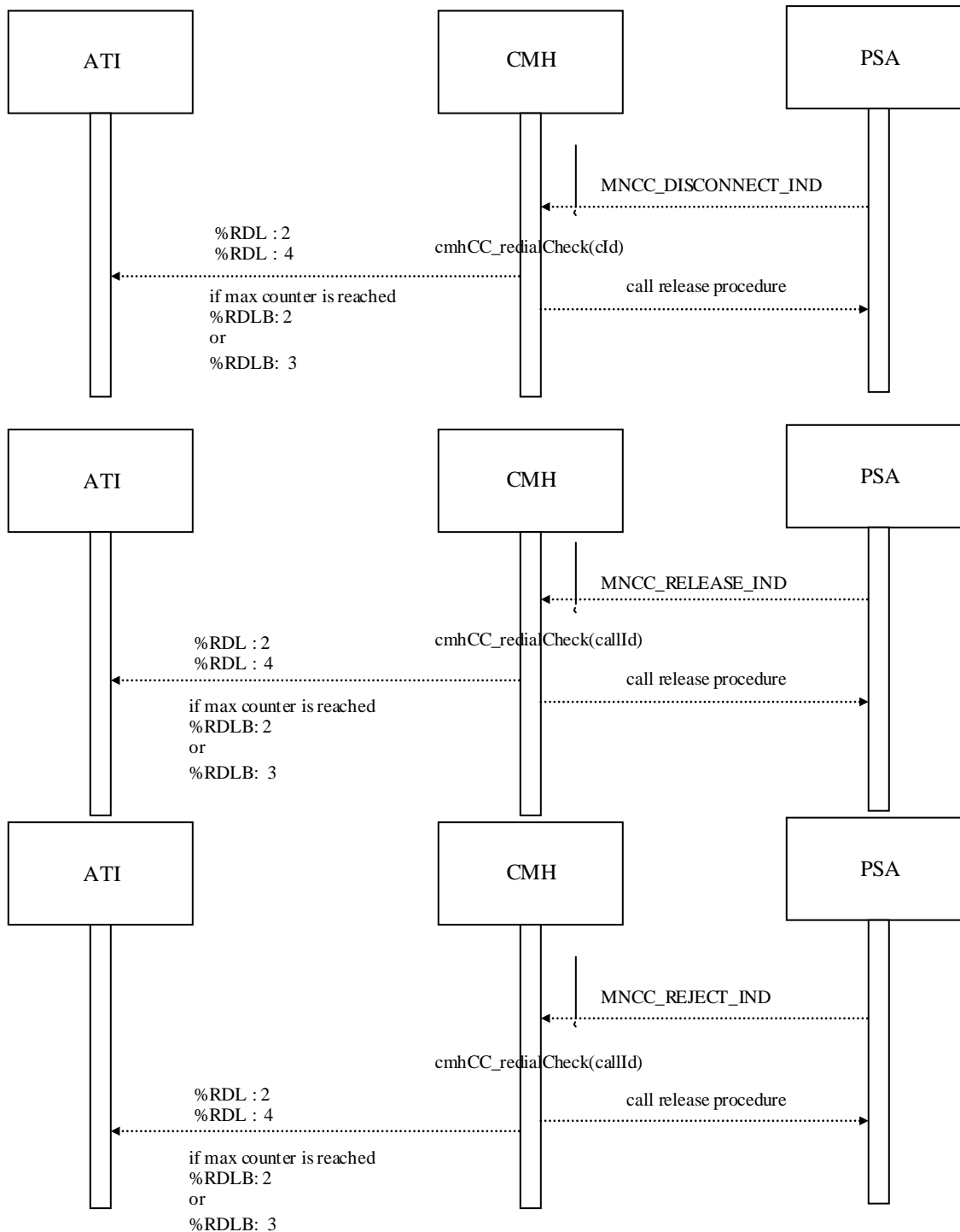
With the query AT command %RDLB the user gets all of phone number entries of the black list and the notification status.



## 5.3 Further MSC's

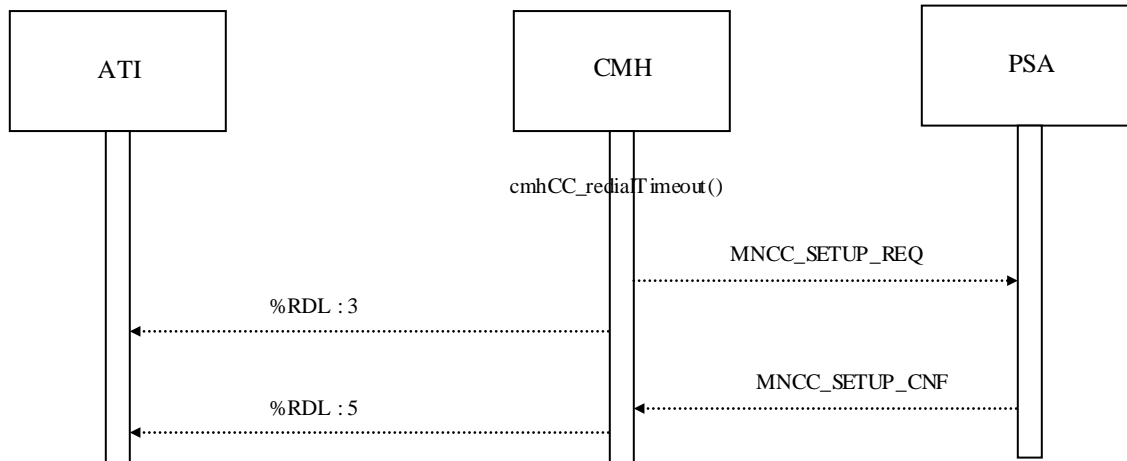
### 5.3.1 Check Causes of unsuccessful Call Attempts

For the redial procedure the **normal and reject causes** of unsuccessful outgoing call attempts are checked if the following primitives are received in ACI.



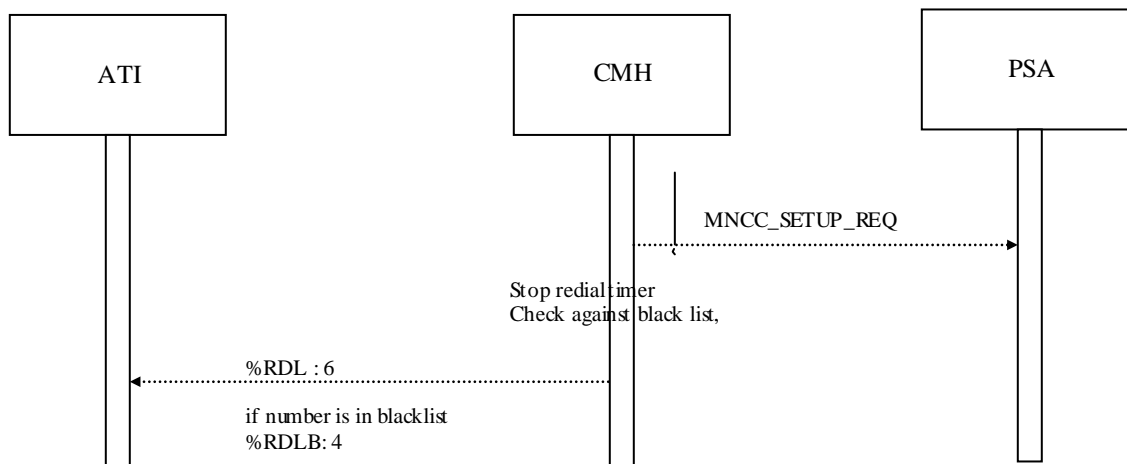
### 5.3.2 Elapsing of Redial Timer

If the redial timer is elapsed ACI starts a new outgoing call. It is successful.

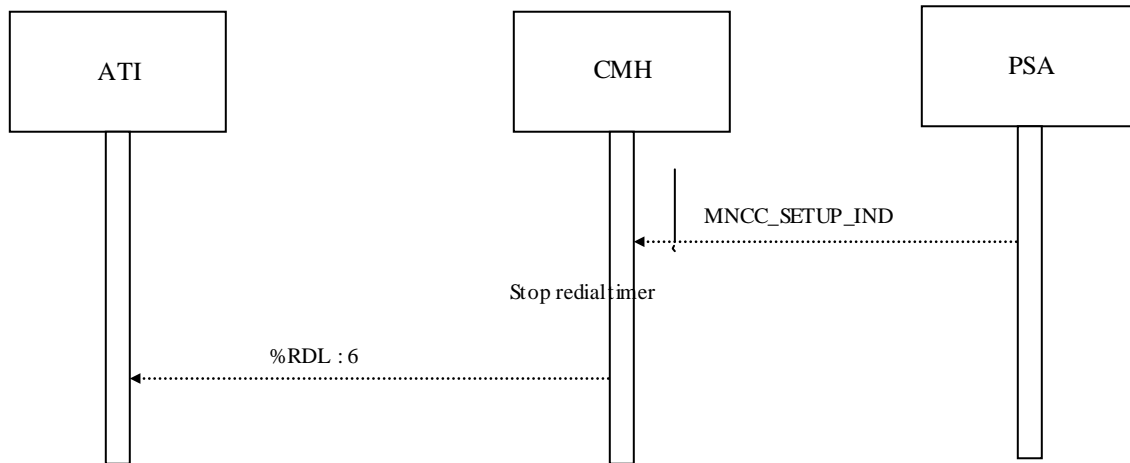


### 5.3.3 Check Redial State for MOC, MTC, ATH and AT+CHUP

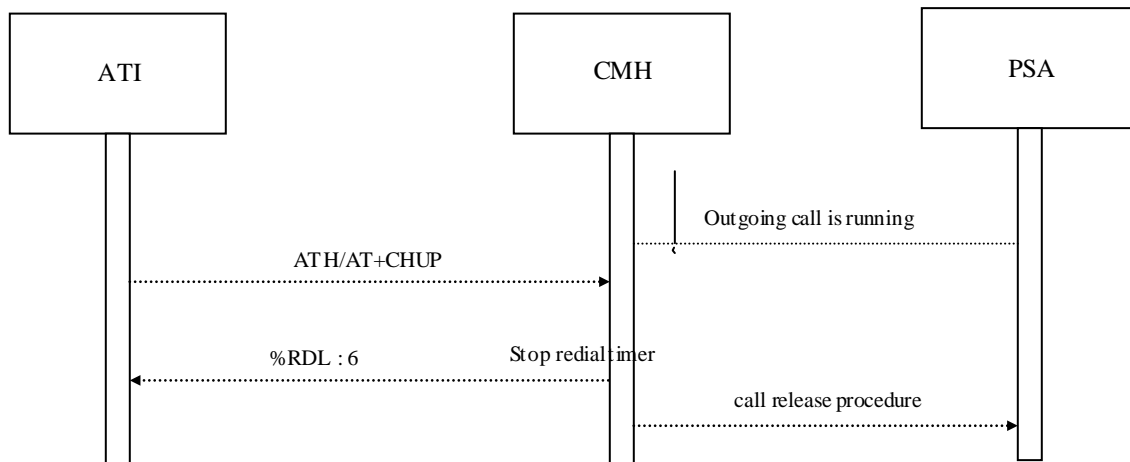
During redial timer is running user starts a new outgoing call. The timer is stopped immediately. In case the new call number is already set in the black list, the new outgoing call is canceled.



During redial timer is running ACI receives an incoming call indication. The timer is stopped immediately.



User want to disconnect an outgoing call.



## 6 Modifications in ACI

### 6.1 Extension of Existing Data Types

#### 6.1.1 Module `ati_cmd.c`

##### 6.1.1.1 Table `cmds[]`

- Add new entries: `setatPercentRDL`, `qeataPercentRDL`, `setatPercentRDLB`, `qeataPercentRDLB`

#### 6.1.2 Module `aci_cmh.h`

##### 6.1.2.1 Type `T_ACI_AT_CMD`

```
typedef enum {  
    ...  
    AT_CMD_RDL,                /* %% RDL command, process redial mode */  
    AT_CMD_RDLB,              /* %% RDLB command, process black list in redial mode */  
    AT_CMD_MAX                 /* maximum command id */  
} T_ACI_AT_CMD;
```

#### 6.1.3 Module `cmh.h`

##### 6.1.3.1 Type `RAT_ID`

```
typedef enum {  
    ...  
    RAT_RDL,                   /* %% RDL command, process black list in redial mode */  
    RAT_RDLB                   /* %% RDLB command, process black list in redial mode */  
} RAT_ID;
```

#### 6.1.4 Module `psa_cc.h`

##### 6.1.4.1 Type `T_CC_CALL_TBL`

```
typedef struct CCCallTbl {  
    ...  
    UBYTE rdCnt;               /* redial counter */  
    UBYTE rdTimIndex;          /* redial timer index for 5th and more attempts */  
} T_CC_CALL_TBL;
```

#### Description:

For the state of redial counter referring to the callId the new parameter `rdCnt` is to be introduced.



#### 6.1.4.2 Type T\_CC\_CLTP

```
typedef enum{  
    NO_VLD_CT = 0,                /* not a valid call type */  
    CT_MOC,                       /* mobile originated call */  
    CT_MTC,                       /* mobile terminated call */  
    CT_NI_MOC,                   /* network initiated mobile originated call */  
    CT_MOC_RDL                   /* redialling mobile originated call */  
} T_CC_CLTP;
```

#### Description:

For distinction from usual mobile originated call the new call type CT\_ MOC\_RDL is to be introduced.

## 6.2 Modification of Existing Functions

### 6.2.1 Module cmh\_ccf.c

#### 6.2.1.1 cmhCC\_SndDiscRsn ()

- If redialling is running caltable entries are not changed

### 6.2.2 Module cmh\_ccr.c

#### 6.2.2.1 cmhCC\_CallReleased ()

- If redialling is running caltable redial call Id entry is not changed

### 6.2.3 Module cmh\_ccq.c

#### 6.2.3.1 qAT\_PlusCLCC()

- Add check of call type CT\_MOC\_RDL for call table output
- Enhance for state CS\_ACT\_REQ call type CT\_MOC\_RDL output during redial attempt

### 6.2.4 Module cmh\_ccs.c

#### 6.2.4.1 cmhCC\_Dial ()

- Add comparison of black list entries against outgoing phone number
- Add stop of redial timer and reset of redial parameters if necessary

#### 6.2.4.2 sAT\_H ()

- Add check of call type CT\_MOC\_RDL for stop of redial procedure and reset redial parameter if user aborts redialling mobile originated call

#### **6.2.4.3 sAT\_PlusCHUP ()**

- Add check of call type CT\_MOC\_RDL in state CS\_ACT\_REQ for stop of redial procedure and reset redial parameter if user aborts redialling mobile originated call; inform user

#### **6.2.5 Module cmh\_f.c**

##### **6.2.5.1 cmh\_timeout ()**

- Call new function cmhCC\_redialTimeout()

#### **6.2.6 Module cmh\_mmis.c**

##### **6.2.6.1 cmhMM\_handleAudioTone ()**

- for redialling BUSY tone is started several times

#### **6.2.7 Module aci\_pei.c**

##### **6.2.7.1 pei\_init ()**

- Add function rdIPrm\_init() for redial parameter
- Add function cmhCC\_rd\_mode\_FFS() reading of redial mode from FFS

##### **6.2.7.2 pei\_exit ()**

- Add function rdIPrm\_exit() for redial parameter , freeing of black list

#### **6.2.8 Module psa\_ccf.c**

##### **6.2.8.1 psaCC\_InitCtbNtry ()**

- Add initializing redial parameter

#### **6.2.9 Module psa\_ccp.c**

##### **6.2.9.1 psa\_mncc\_disconnect\_ind ()**

- Add call new function cmhCC\_redialCheck ()

##### **6.2.9.2 psa\_mncc\_reject\_ind ()**

- Add call new function cmhCC\_redialCheck ()

##### **6.2.9.3 psa\_mncc\_release\_cnf ()**

- Add call new function cmhCC\_redialCheck () ; no change calltable entry if redialling time is running

##### **6.2.9.4 psa\_mncc\_release\_ind ()**

- Add call new function cmhCC\_redialCheck (); no change calltable entry if redialling time is running

##### **6.2.9.5 psa\_mncc\_setup\_cnf ()**

- For successful mobile originated calls add reset redial parameters; change call type CT\_MOC\_RDL to CT\_MOC; inform user, reset redial parameter

**6.2.9.6 psa\_mncc\_setup\_cnf ()**

- For successful mobile originated calls add reset redial parameters; change call type CT\_MOC\_RDL to CT\_MOC; inform user, reset redial parameter

**6.2.9.7 psa\_mncc\_setup\_req ()**

- add check call type CT\_MOC\_RDL

## 7 New Functions and Data Structures

### 7.1 Predefined Values

#### 7.1.1 Module `l4_tim.h`

##### 7.1.1.1 Timer

`/* time in milliseconds */`

`ACI_REPEAT            5000            /* 5 sec for waiting for 1th redialing    */`

`ACI_REPEAT_2_4       60000          /* 1 min for waiting for 2th to 4st redialing */`

`/* timer index */`

`ACI_REPEAT_HND`

#### 7.1.2 Module `aci_cmh.h`

##### 7.1.2.1 Constants

`MAX_NUM_REPEAT_ATT 10            /* maximal number of call repeat attempts */`

`MAX_NUM_ENTR_BLACK_LIST 10      /* maximal number of black list phone numbers */`

## 7.2 New Data Types

#### 7.2.1 Module `aci_cmh.h`

##### 7.2.1.1 `T_ACI_CC_BLACKL_ENTRY`

`typedef struct{`

`UBYTE        numb_len;`

`CHAR        number[MAX_CC_CALLED_NUMBER];`

`T_ACI_TOA   type;`

`}T_ACI_CC_BLACKL_ENTRY;`

##### 7.2.1.2 `T_ACI_CC_RDL_BLACKL_STATE`

`typedef enum{`

`BLACKLIST_FULL = 2,            /* black list is full */`

`ENTRY_BLACKLISTED,            /* phone number set in black list */`

`ENTRY_IN_BLACKLIST            /* phone number is in black list */`

`}T_ACI_CC_RDL_BLACKL_STATE`

### 7.2.1.3 T\_ACI\_CC\_RDL\_FFS

typedef enum

```
{  
    READ_RDLmode = 0,          /* read redial mode from FFS */  
    WRITE_RDLmode             /* write redial mode in FFS */  
}T_ACI_CC_RDL_FFS;
```

### 7.2.1.4 T\_ACI\_CC\_REDIAL\_BLACKL

typedef struct CCblacklist{

```
    UBYTE bCount;              /* counter of black list entries */  
    T_ACI_CC_BLACKL_ENTRY bNum[MAX_NUM_ENTR_BLACK_LIST];  
                                /* table with blacklisted phone number */
```

}T\_ACI\_CC\_REDIAL\_BLACKL;

#### Description:

For blacklist of the outgoing phone numbers is to be introduced. It is possible to save maximal 10 numbers.

### 7.2.1.5 T\_ACI\_CC\_REDIAL\_BLMODE

typedef enum{

```
    BLMODE_NO_PRESENT = -1,    /* parameter is not given */  
    BL_NO_DELETE,  
    BL_DELETE                  /* blacklist is deleted */  
}T_ACI_CC_REDIAL_BLMODE;
```

### 7.2.1.6 T\_ACI\_CC\_REDIAL\_MODE

typedef enum{

```
    AUTOM_REP_NOT_PRESENT = -1, /* parameter is not given */  
    AUTOM_REPEAT_OFF = 0,       /* automatic calling repeat is switched off */  
    AUTOM_REPEAT_ON             /* automatic calling repeat is switched on */  
} T_ACI_CC_REDIAL_MODE;
```

### 7.2.1.7 T\_ACI\_CC\_REDIAL\_NOTIF

typedef enum{

```
    NOTIF_NO_PRESENT = -1,      /* parameter is not given */  
    NO_NOTIF_USER,             /* no notification to user */  
    NOTIF_USER                  /* notification to user */  
} T_ACI_CC_REDIAL_NOTIF;
```

### 7.2.1.8 T\_ACI\_CC\_REDIAL\_STATE

typedef enum{

```
    REDIAL_TIM_START = 2,       /* redial timer starts */  
    REDIAL_ATT_START,           /* start redialling attempt */
```

```
CALL_ATTEMPT_FAILED,  
CALL_ATTEMPT_SUCCESSFUL,  
REDIAL_STOP                                /* redialling finished */  
}T_ACI_CC_REDIAL_STATE;
```

### 7.2.1.9 T\_ACI\_CC\_REDIAL\_PAR

```
typedef struct {  
    T_ACI_CC_REDIAL_MODE rdMod;           /* redial mode */  
    T_ACI_CC_REDIAL_NOTIF rdModN;         /* redial state notification */  
    SHORT rdId;                           /* call id for redialling */  
    BOOL rdIState;                        /* internal redial state */  
    T_ACI_CC_REDIAL_NOTIF rdIBIN;         /* black list state notification */  
    T_ACI_CC_REDIAL_BLMODE rdIBMod;       /* black list mode */  
}T_ACI_CC_REDIAL_PAR;
```

#### Description:

The global structure contains all needed redial parameter.

## 7.2.2 Extending of FFS

### 7.2.2.1 RedialMode

Pathname: /GSM/COM/; fixed object

BYTE	redialMode
0	/* default value: automatic calling repeat switch off */
1	/* automatic calling repeat switch on */

## 7.3 New Functions

### 7.3.1 Module ati\_cc.c

#### 7.3.1.1 setatPercentRDL

##### Prototype:

```
GLOBAL T_ATI_RSLT setatPercentRDL (char *cl, UBYTE srcId)
```

##### Parameters:

cl	command string
srcId	source of AT command

##### Return:

ATI_FAIL	wrong parameter or set redial mode fails
ATI_CMPL	set redial mode successful

##### Description:

This function checks the AT command string and calls sAT\_PercentRDL.

### 7.3.1.2 queatPercentRDL

**Prototype:**

GLOBAL T\_ATI\_RSLT queatPercentRDL (char \*cl, UBYTE srcId)

**Parameters:**

cl	command string
srcId	source of AT command

**Return:**

ATI_FAIL	wrong parameter or query redial mode fails
ATI_CMPL	query redial mode successful

**Description:**

This function checks the AT command string and calls qAT\_PercentRDL.

### 7.3.1.3 setatPercentRDLB

**Prototype:**

GLOBAL T\_ATI\_RSLT setatPercentRDLB (char \*cl, UBYTE srcId)

**Parameters:**

cl	command string
srcId	source of AT command

**Return:**

ATI_FAIL	wrong parameter or delete black list fails
ATI_CMPL	delete black list successful

**Description:**

This function checks the AT command string and calls sAT\_PercentRDLB.

### 7.3.1.4 queatPercentRDLB

**Prototype:**

GLOBAL T\_ATI\_RSLT queatPercentRDLB (char \*cl, UBYTE srcId)

**Parameters:**

cl	command string
srcId	source of AT command

**Return:**

ATI_FAIL	wrong parameter or empty black list
ATI_CMPL	show black list successful

**Description:**

This function checks the AT command string and calls qAT\_PercentRDLB.

## 7.3.2 Module ati\_ret.c

### 7.3.2.1 rCI\_PercentRDL

**Prototype:**

GLOBAL void rCI\_PercentRDL ( T\_ACI\_CC\_REDIAL\_STATE state )

**Parameters:**

state	WAITING / REDIAL_START / CALL_ATTEMPT_FAILED CALL_ATTEMPT_SUCCESSFUL / REDIAL_STOPPED
-------	--

This call back function sends the response string with user notification about redial state in ATI if the user notification is requested by user.

#### 7.3.2.2 rCI\_PercentRDLB

**Prototype:**

GLOBAL void rCI\_PercentRDLB ( T\_ACI\_CC\_REDIAL\_STATE state )

**Parameters:**

state	BLACKLIST_FULL/ ENTRY_BLACKLISTED
-------	-----------------------------------

This call back function sends the response string with user notification about black list state in ATI if the user notification is requested by user.

### 7.3.3 Module cmh\_ccf.c

#### 7.3.3.1 cmhCC\_reidialCheck

**Prototype:**

GLOBAL void cmhCC\_redialCheck (SHORT cId)

**Parameters:**

cId	call Id
-----	---------

**Return:**

—

If an outgoing call attempt was not successful this function evaluates the received error indications. It checks the redial counter with respect to the error indication and starts the appropriate redial timer. If necessary the function adds entries in the black list. If requested the function informs the user about the redial and/or black list state.

### 7.3.3.2 cmhCC\_redialChkBlackl

### Prototype:

GLOBALvoid cmhCC\_redialChkBlackl (SHORT cId)

### Parameters:

cId	call Id
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

**Return:**

—

The function checks the desired MOC number against the black list entries.

### 7.3.3.3 cmhCC rd mode FFS

**Prototype:**

[illegible]

### Parameters:

```
redial_mode      AUTOM_REPEAT_ON/ AUTOM_REPEAT_OFF
ffs_mode        READ RDLmode / WRITE RDLmode
```



**Return:**

ATI_FAIL	set redial mode fails
ATI_CMPL	set redial mode successful

**Description:**

The function writes the redial mode in FFS.

**7.3.4 Module cmh\_ccr.c**

**7.3.4.1 cmhCC\_redialTimeout**

**Prototype:**

GLOBAL void cmhCC\_redialTimeout ( void )

**Parameters:**

-

**Return:**

-

**Description:**

This function checks the current redial mode and starts a call repeat. It increases the redial counter of the related call Id.

**7.3.5 Module cmh\_ccs.c**

**7.3.5.1 rdIPrm\_exit**

**Prototype:**

GLOBAL void rdIPrm\_exit(void)

**Description:**

This function resets all redial parameter of structure rdIPrm and give the black list memory free.

**7.3.5.2 rdIPrm\_init**

**Prototype:**

GLOBAL void rdIPrm\_init(void)

**Description:**

This function initializes all redial parameter of structure rdIPrm.

**7.3.5.3 sAT\_PercentRDL**

**Prototype:**

GLOBAL T\_ACI\_RETURN sAT\_PercentRDL ( T\_ACI\_CMD\_SRC srcId,  
T\_ACI\_CC\_REDIAL\_MODE redial\_mode,  
T\_ACI\_CC\_REDIAL\_NOTIF notification)

**Parameters:**

srcId	source of AT command
redial_mode	AUTOM_REPEAT_ON/ AUTOM_REPEAT_OFF
notification	NO_NOTIF_USER /NOTIF_USER

**Return:**

ATI_FAIL	set redial mode fails
ATI_CMPL	set redial mode successful

**Description:**

This function saves the redial mode and user notification state in the structure T\_CC\_REDIAL\_PAR rdIPrm. Furthermore it stores the mode in the FFS. If the chosen redial mode is AUTOM\_REPEAT\_ON the function allocates the black list if it does not exist. In case AUTOM\_REPEAT\_OFF the function resets the redial parameter, stops the redial timer and frees the call table entry (if necessary).

### 7.3.5.4 sAT\_PercentRDLB

**Prototype:**

```
GLOBAL T_ACI_RETURN sAT_PercentRDLB ( T_ACI_CMD_SRC srcId,
                                         T_ACI_CC_REDIAL_BLMODE blacklist_mode,
                                         T_ACI_CC_REDIAL_NOTIF notification)
```

**Parameters:**

srcId	source of AT command
blacklist_mode	BL_NO_DELETE/BL_DELETE
notification	NO_NOTIF_USER/NOTIF_USER

**Return:**

ATI_FAIL	delete black list fails
ATI_CMPL	delete black list successful

**Description:**

This function clears the black list if blackl\_clear is TRUE and stores the state of user notification.

### 7.3.6 Module cmh\_ccq.c

#### 7.3.6.1 qAT\_PercentRDL

**Prototype:**

```
GLOBAL T_ACI_RETURN qAT_PercentRDL ( T_ACI_CMD_SRC srcId,
                                         T_ACI_CC_REDIAL_MODE * redial_mode,
                                         T_ACI_CC_REDIAL_NOTIF *notification)
```

**Parameters:**

srcId	source of AT command
redial_mode	AUTOM_REPEAT_ON/ AUTOM_REPEAT_OFF
notification	NO_NOTIF_USER/NOTIF_USER

**Return:**

ATI_FAIL	query redial mode fails
ATI_CMPL	query redial mode successful

**Description:**

This function informs about the state of redial mode and user notification.

#### 7.3.6.2 qAT\_PercentRDLB

**Prototype:**

```
GLOBAL T_ACI_RETURN qAT_PercentRDLB ( T_ACI_CMD_SRC srcId,
                                         T_ACI_CC_REDIAL_BLACKL *blackl,
                                         T_ACI_CC_REDIAL_NOTIF *notification )
```

**Parameters:**

srcId	source of AT command
blackl	list of blacklisted phone numbers
notification	NO_NOTIF_USER / NOTIF_USER

**Return:**

ATI_FAIL	show black list fails
ATI_CMPL	show black list successful

**Description:**

This function returns the black list entries and the state of user notification to the user.

### 7.3.7 Module smi\_dmy.c

#### 7.3.7.1 rAT\_PercentRDL

**Prototype:**

GLOBAL void rAT\_PercentRDL (T\_ACI\_CC\_REDIAL\_STATE state)

**Parameters:**

state	WAITING / REDIAL_START / CALL_ATTEMPT_FAILED CALL_ATTEMPT_SUCCESSFUL / REDIAL_STOPPED
-------	--

**Description:**

This is the callback dummy function in SMI.

#### 7.3.7.2 rAT\_PercentRDLB

**Prototype:**

GLOBAL void rAT\_PercentRDLB (T\_ACI\_CC\_REDIAL\_STATE state)

**Parameters:**

state	BLACKLIST_FULL / ENTRY_BLACKLISTED
-------	------------------------------------

**Description:**

This is the callback dummy function in SMI.

## **8 Test Strategy**

### **8.1 Tests under Windows**

Add new testcases:

- for AT% RDL with correct and incorrect parameters
- for AT% RDLB with correct and incorrect parameters; delete black list entries
- Evaluation of all kind of related error indications, inform user
- Set phone numbers in black list until the list is full, inform user about black list
- Check blacklist entries against outgoing phone number; inform user
- Check redialling status for incoming calls; inform user
- If redialling is successful user is informed
- Redialling if timer is elapsed; inform user
- Test ATH and AT+CHUP for redialling ; Test AT+CLCC
- Abort redialling procedure by user with AT% RDL = 0
- Test with four window stacks

### **8.2 Tests on target**

- Store/Restore redial mode in FFS: target test
- Test the both new AT Commands: set correct and incorrect parameters
- Abort redialling procedure by user with AT% RDL = 0, x
- Evaluation of error indication USER BUSY
- Set entries in blacklist, delete entries in black list
- Check blacklist entries against outgoing phone number
- Check redialling status for incoming call
- Redialling if timer is elapsed
- Control responses via ATI
- Test with some images