CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

06.93 CR **A022**	Current Version:	**7.3.0 (R98)**

GSM (AA.BB) or 3G (AA.BBB) specification number ↑   *↑ CR number as allocated by MCC support team*

For submission to:
List expected approval meeting # here
↑

for approval **X**
for information ☐

strategic ☐
non-strategic **X**

(for SMG use only)

Form: CR cover sheet, version 2 for 3GPP and SMG   The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM ☐   ME **X**   UTRAN / Radio ☐   Core Network **X**
(at least one should be marked with an X)

Source:	Siemens AG		**Date:**	28.02.2000

Subject: TX- and RX-Types Identifiers

Work item: AMR

Category:
(only one category Shall be marked With an X)

			Release:	
F	Correction	**X**	Phase 2	☐
A	Corresponds to a correction in an earlier release	☐	Release 96	☐
B	Addition of feature	☐	Release 97	☐
C	Functional modification of feature	☐	Release 98	**X**
D	Editorial modification	☐	Release 99	☐
			Release 00	☐

Reason for change: For purpose of clarification, it is felt as an advantage to have a full correspondance of the TX- and RX-Types Identifiers between the text of 06.93 and the C-code. This CR proposes a change in the C-code. The change has no impact on the functionality of the C-code.

Clauses affected: C code file

Other specs Affected:

Other 3G core specifications	☐	→ List of CRs:
Other GSM core specifications	☐	→ List of CRs:
MS test specifications	☐	→ List of CRs:
BSS test specifications	☐	→ List of CRs:
O&M specifications	☐	→ List of CRs:

Other comments:

help.doc

<---------- double-click here for help and instructions on how to create a CR.

1. Background

The version 7.3.0 of GSM 06.93 specifies the TX- and RX-Type Identifiers by the two following tables:

Table 1: TX TYPE identifiers

TX_TYPE Legend	Information Bits	Mode Indication
SPEECH_GOOD	speech frame, size 95..244 bits depending on codec mode; no errors known.	current code mode
SPEECH_DEGRADED (only in downlink in TFO)	Speech frame, size 95..244 bits, depending on codec mode; there might be errors in class 2 bits.	current codec mode
SPEECH_BAD (only in downlink in TFO)	Speech frame, size 95..244 bits, depending on codec mode; there are errors in class 1 bits.	current codec mode
SID_FIRST	marks the end of a talkspurt, respectively the beginning of a speech pause; does not contain information bits.	the codec mode that would have been used if TX_TYPE had been SPEECH
SID_UPDATE	comfort noise, 35 bits; no errors known	the codec mode that would have been used if TX_TYPE had been SPEECH
SID_BAD (only in downlink in TFO)	comfort noise, 35 bits; errors detected, parameters unusable	the codec mode that would have been used if TX_TYPE had been SPEECH
ONSET (only in downlink in TFO)	announces the beginning of a speech burst; does not contain information bits	the codec mode of the following speech frame
NO_DATA	no useful information	no useful information

Table 2: RX_TYPE identifiers

RX_TYPE Legend	Description
SPEECH_GOOD	Speech frame with CRC OK, Channel Decoder soft values also OK
SPEECH_DEGRADED	Speech frame with CRC OK, but 1B bits and class2 bits may be corrupted
SPEECH_BAD	(likely) speech frame, bad CRC (or very bad Channel Decoder measures)
SID_FIRST	first SID marks the beginning of a comfort noise period
SID_UPDATE	SID update frame (with correct CRC)
SID_BAD	Corrupt SID update frame (bad CRC; applicable only for SID_UPDATE frames)
ONSET	ONSET frames precede the first speech frame of a speech burst
NO_DATA	Nothing useable (for the speech decoder) was received. This applies for the cases of no received frames (DTX) or received FACCH or RATSCCH or SID_FILLER signalling frames.

In order to have full correspondance between the tables and the C-code, the CR proposes a change in the C-code. The change has no impact on the functionality of the C-code.

2. How the code is changed in file *frame.h*

2.1 Before the change (lines 26...42)

enum RXFrameType { RX_SPEECH_GOOD = 0,

```
                RX_SPEECH_PROBABLY_DEGRADED,
                RX_SPARE,
                RX_SPEECH_BAD,
                RX_SID_FIRST,
                RX_SID_UPDATE,
                RX_SID_BAD,
                RX_NO_DATA,
                RX_N_FRAMETYPES     /* number of frame types */
};

enum TXFrameType { TX_SPEECH = 0,
                TX_SID_FIRST,
                TX_SID_UPDATE,
                TX_NO_DATA,
                TX_N_FRAMETYPES     /* number of frame types */
};
```

2.2 After the change

```
enum RXFrameType { RX_SPEECH_GOOD = 0,
                RX_SPEECH_DEGRADED,
                RX_SPEECH_BAD,
                RX_SID_FIRST,
                RX_SID_UPDATE,
                RX_SID_BAD,
                RX_ONSET,
                RX_NO_DATA,
                RX_N_FRAMETYPES     /* number of frame types */
};

enum TXFrameType { TX_SPEECH_GOOD = 0,
                TX_SPEECH_DEGRADED,
                TX_SPEECH_BAD,
                TX_SID_FIRST,
                TX_SID_UPDATE,
                TX_SID_BAD,
                TX_ONSET,
                TX_NO_DATA,
                TX_N_FRAMETYPES     /* number of frame types */
};
```