TSG-RAN Meeting #13 ***RP-010551***
Beijing, China, 18 - 21 September 2001

Title: **Agreed CRs (Release '99 and Rel-4 category A) to TR 25.921**

Source: **TSG-RAN WG2**

Agenda item: **8.2.3**

Doc-1st-	Status-	Spec	CR	Rev	Phase	Subject	Cat	Version	Versio
R2-011997	agreed	25.921	024	1	R99	Guidelines concerning conditions, spares, defaults and correction of inconsistencies	F	3.4.0	3.5.0
R2-012142	agreed	25.921	025		Rel-4	Guidelines concerning conditions, spares, defaults and correction of inconsistencies	A	4.1.0	4.2.0
R2-011849	agreed	25.921	026		R99	Naming convention for non-critical extensions	F	3.4.0	3.5.0
R2-012143	agreed	25.921	027		Rel-4	Naming convention for non-critical extensions	A	4.1.0	4.2.0
R2-012161	agreed	25.921	028	1	R99	Introduction of procedure specification guidelines specific to RLC	F	3.4.0	3.5.0
R2-012162	agreed	25.921	029		Rel-4	Introduction of procedure specification guidelines specific to RLC	A	4.1.0	4.2.0
R2-012145	agreed	25.921	030		R99	RAN3 usage of 1994 ASN.1 feature set	F	3.4.0	3.5.0
R2-012146	agreed	25.921	031		Rel-4	RAN3 usage of 1994 ASN.1 feature set	A	4.1.0	4.2.0

3GPP TSG-RAN WG2, Meeting #23
Helsinki, Finland, August 27th- 31st, 2001

Tdoc R2-011997

CR-Form-v4

CHANGE REQUEST

⌘ **25.921 CR 024** ⌘ ev **r1** ⌘ Current version: **3.4.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network ☒ Core Network □

Title:	⌘	Guidelines concerning conditions, spares, defaults and correction of inconsistencies		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	22-08-2001
Category:	⌘	**F**	***Release:*** ⌘	R99

Use one of the following categories:	*Use one of the following releases:*
F *(correction)*	*2 (GSM Phase 2)*
A *(corresponds to a correction in an earlier release)*	*R96 (Release 1996)*
B *(addition of feature),*	*R97 (Release 1997)*
C *(functional modification of feature)*	*R98 (Release 1998)*
D *(editorial modification)*	*R99 (Release 1999)*
Detailed explanations of the above categories can	*REL-4 (Release 4)*
be found in 3GPP TR 21.900.	*REL-5 (Release 5)*

Reason for change: ⌘ The guidelines for how to use conditional information elements, spares and defaults are not clear enough. Furthermore, guidelines are missing concerning how to handle inconsistencies when straightforward corrections are not feasible

Summary of change: ⌘ This CR proposes changes in the following areas:

Conditional IEs

The need for an IE based on a condition is clarified to be one of the following:

- Mandatorily present
- Mmandatory with default value
- Not needed
- Optional

Moreover, guidelines for how conditional expressions should be specified are added, for these four cases. A recommendation is also added that all cases should be explicitly covered by the conditional expression, e.g. by using the word "otherwise".

Spare values

- For IEs in included in UL messages spares are used also for IEs with need set to MP, CV or CH
- For spares that result after mapping an information element value the same guidelines/ handling applies
- Within the tabular description the number of spares needed should be indicated. Within the ASN.1 spares should be used only to increase the size of the encoded size of an IE, when needed. Furthermore, for choices at most one spare should be allowed

Default values

- Variables may be used to clarify the specification of default values that depend on information received in the past

Guidelines for handling inconsistencies when straightforward ASN.1 corrections

		are not feasible • The guidelines cover the changes needed to procedure specification, tabular description and ASN.1 to overcome the following inconsistencies: • inconsistencies between tabular description and ASN.1 regarding the “Need” of an IE • removal of information elements e.g. because the functionality is incomplete Note: The changes are only made for the tabular guidelines applicable for RRC, since a separate discussion in RAN3 is ongoing regarding conditional IEs applicable on the RAN3 set of specifications. Isolated impact • No impact. This CR concerns guidelines for how to specify protocols. Since it involves no direct changes of protocols, there is no impact on implementations
Consequences if not approved:	⌘	Inconsistent use of conditions, spare values, defaults and correction of inconsistencies may lead to ambiguities in future versions of the RRC. This may again result in inter-operability problems

Clauses affected:	⌘	4.5.1, 4.5.2, 9.1.1.2.1, 9.1.1.2.1.1, 9.1.1.2.1.3, 9.1.3.2, 10.4.3.4.5, 12 (new)		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	25.921 v4.1.0, CR 025
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

4.5.1 ~~Reserved~~ Spare values and spare fields

~~Reserved values shall be forbidden. Otherwise entity receiving such a value shall reject the message. This would create difficulties when provided on broadcast channel.~~

In case the protocol allows for the transfer of spare information element values, the behaviour of the receiving entity not comprehending these values shall be specified.

~~Spare field shall be forbidden. Otherwise entity receiving such a spare field shall not make any decoding on that field and shall not reject the message.~~

In case the protocol allows for the transfer of spare fields, the behaviour of the receiving entity not comprehending the spare fields shall be specified. Facilities may be needed to allow receivers that don't comprehend the spare fields to decode the other message parts.

4.5.2 Unspecified values

~~As far as possible default understanding shall be provided for unspecified values.~~

In case the protocol allows for the transfer of undefined information element values, the behaviour of the receiving entity not comprehending these values shall be specified.

9.1.1.2.1 Need and multiplicity (Multi) columns

These columns provide most of the information about the presence, absence and number of instances of the IE (in the message or in the group) or group of IEs. The different possibilities for these columns are described one by one.

The meaning of the 'need' column is summarised below:

MP Mandatorily present.

A value for that information is always needed, and no information is provided about a particular default value. If ever the transfer syntax allows absence (e.g., due to extension), then absence leads to an error diagnosis.

MD Mandatory with default value.

A value for that information is always needed, and a particular default value is mentioned (in the 'Semantical information' column). This opens the possibility for the transfer syntax to use absence or a special pattern to encode the default value.

CV Conditional on value.

The need for a value for that information depends on the value of some other IE or IEs, and/or on the message flow (e.g., channel, SAP). The need is specified by means of a condition, which result may be that the information is mandatorily present, mandatory with default value, not needed or optional.

~~A value for that information is needed (presence needed) or unacceptable (absence needed) when some conditions are met that can be evaluated on the sole basis of the content of the message.~~

~~If conditions for presence needed are specified, ~~If one of the results of the condition is that the information is mandatorily present, the transfer syntax must allow for the presence of the information. If in this case the information is absent ~~transfer syntax allows absence, absence when the conditions for presence are met leads to ~~an error is ~~diagnosis~~diagnosed.

If one of the results of the condition is that the information is mandatory with default value, and a particular default value is mentioned (in the 'Semantical information' column), the transfer syntax may use absence or a special pattern to encode the default value.

~~If conditions for absence needed are specified, ~~If one of the results of the condition is that the information is not needed, the transfer syntax must allow encoding the absence. If in this case the information is present~~ and the conditions for absence are met~~, it will be ignored. In specific cases however, an error ~~is ~~may be diagnosed instead.

If one of the results of the condition is that the information is optional, the transfer syntax must allow for the presence of the information. In this case, neither absence nor presence of the information leads to an error diagnosis.

~~When neither conditions for presence or absence are met, the information is treated as optional, as described for 'OP'.~~

CH Conditional on history.

~~A value for that information is needed (presence needed) or unacceptable (absence needed) when some conditions are met that must be evaluated on the basis of ~~The need for a value for that information depends on information obtained in the past (e.g., from messages received in the past from the other party). The need is specified by means of a condition, which result may be that the information is mandatorily present, mandatory with default value, not needed or optional.

The handling of the conditions is the same as described for CV.

~~If conditions for presence needed are specified, the transfer syntax must allow for the presence of the information. If the transfer syntax allows absence, absence when the conditions for presence are met leads to an error diagnosis.~~

- ~~If conditions for absence needed are specified, the transfer syntax must allow encoding the absence. If the information is present and the conditions for absence are met, an error is diagnosed.~~

- ~~When neither conditions for presence or absence are met, the information is treated as optional, as described for 'OP'.~~

OP Optional.

The presence or absence is significant and modifies the behaviour of the receiver. However whether the information is present or not does not lead to an error diagnosis.

9.1.1.2.1.1 Mandatory

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
Name	MP				
Name	MD			(default value is indicated)	

The multiplicity column may be left empty (see 9.1.1.2.1.5).

For mandatory IEs, the rules are as follows, applied on the number of instances given by the multiplicity column (leaving the multiplicity column empty means one and only one instance):

For an IE not belonging to a group MP indicates that the number of instances as given by the multiplicity column of 'Name IE' is necessary in the message.

For a group not belonging to another group, MP means that the number of instances as given by the multiplicity column of the 'Name group' is necessary in the message.

For an IE or a group belonging to another group, MP means that if the parent group is present, then the number of instances as given by the multiplicity column of the 'Name group' or 'Name IE' is necessary in the embedding group.

For an IE not belonging to a group MD indicates that the number of instances as given by the multiplicity column for information 'Name IE' is necessary in the message, and that a special value (the default value) exists, for all instances or individual instances, and is mentioned in the 'Semantics description' column.

For a group not belonging to another group, MD means that the number of instances as given by the multiplicity column for information structure 'Name group' is necessary in the message, and that a special value (the default value) exists, for all instances or individual instances, and is mentioned in the 'Semantics description' column.

For an IE or a group belonging to another group, MD means that if the parent group is present, then the number of instances as given by the multiplicity column for information structure 'Name group' or information 'Name IE' is necessary in the embedding group, and that a special value (the default value) exists and is mentioned in the 'Semantics description' column.

The default value might be fixed by the standard, or conditional to the value of some other IE or IEs, or conditional on information obtained in the past. In case the default value depends on information obtained in the past, variables may be used to specify it.

9.1.1.2.1.2 Optional

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
Name	OP				

The multiplicity column may be empty (see 9.1.1.2.1.5).

This indicates that the number of instances as given by the multiplicity column of the 'Name IE' or 'Name group' is not necessary in the message or the embedding group, and that the sender can choose not to include it.

9.1.1.2.1.3 Conditional

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
	CV *cond*				
	CH				

The multiplicity column may be empty (see 9.1.1.2.1.5).

CV indicates that the requirement for presence or absence of the number of instances as given by the multiplicity column of the IE or group of IE depends on the value of some other IE or IEs, and/or on the message flow (e.g., channel, SAP). In the CV case, the condition is to be described in a textual form in an explanatory clause. cond stands for a free text that is used as a reference in the title of the explanatory clause. In the CH case, the condition is described in the procedural section.

The expression *cond* should explicitely cover all cases. A typical *cond* expression is “The IE is *CS1* if *TE* and *CS2* otherwise”, where *CS1* and *CS1* are conditon statements and *TE* is an expression used in the if test. For example: “The IE is mandatory if the IE “XXX” has the value “YYY” and not needed otherwise”.

The result of evaluating the condition (if the condition is met or not) may mean that the IE is:

- Mandatorily present, when the condition statement says “mandatory”~~.~~

- Mandatory with default value, when the condition statement says “mandatory with default value”

- ~~Mandatorily absent~~Not needed, when the condition statement says “not needed”~~.~~

- Optional, when the condition statement says “optional”.~~.~~

- ~~Absent, but optional (this is meaningful only for extension).~~

The error handling shall be specified in the protocol for the cases when the requirement for presence or absence of an IE indicated by the condition is not followed.

9.1.3 Extension for further releases

9.1.3.1 Basic principle

Added elements or choice branches are included where they fit most naturally according to their semantics, independently from the version in which they were added.

9.1.3.2 Critical or non-critical

A case-by-case guideline is provided by informal text after the table itself.

Spare values shall not be defined for c~~C~~ritical~~ spare values are banned from the tabular~~ information elements. Extension of the value range of c~~C~~ritical ~~extensions~~ information elements ~~of enumerated~~ must be done by creating a critical extension.

~~Non-critical spare values shall not appear.~~

Spare values may be defined for non- critical information elements. For spares that result after mapping of the original information element value the same handling and/ or guidelines apply.

In case spares are defined, t~~T~~he tabular notation ~~shall~~ should indicate ~~only that 'at least one~~the number of ~~non critical~~ spare values that is needed~~'. The exact number of spare values is found in the transfer syntax~~. Within the ASN.1 spare values should only be used to increase the encoded size of an IE. This means that the ASN.1 should only include spares if the number of spare values that is needed exceeds the number of undefined values within the transfer syntax of the information element.

For downlink messages, spare values may be defined for n~~N~~on-critical information elements~~spare values~~ for which the need is specified to be ~~are only possible with~~ MD or OP ~~fields~~ (or CV case leading to MD or OP).~~,~~ In this case, ~~and an 'old'~~ receiver not comprehending the ~~receiving~~ received a ~~non-critical~~ spare value shall consider ~~it as~~ the information element to have ~~meaning respectively~~ the default value or consider it to be absent~~ce~~ respectively.

For uplink messages spares values may be defined for all information elements, including those for which the need is specified to be MP (or CV case leading to MP).

In all cases at most one spare should be defined for choices.

10.4.3.4.5 Replacement of a spare value with a new element

If a new value is to be included in an IE of type ENUMERATED, for which spare values were defined in the previous version, those spare values can be replaced with the new values.

If more new values are needed, than spare values included in the previous version, one spare value can be replaced by a special extension value (called e-new in example 8). If that value is used, a new element in the nonCriticalExtension part (element1-new) will define the new values, as shown in Example 8 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            ENUMERATED { e1, e2, spare1, spare2 }
}

-- Now three new values are needed for element1: e3, e4 and e5. MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If the following has the value e-new, the actual value of element1 is defined in
-- element1-new included in MessageA-r4-ext-IEs
    element1                            ENUMERATED { e1, e2, e3, e-new }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e-new.
    element1-new                        ENUMERATED { e4, e5, spare1, spare2 }   OPTIONAL
}
```

Example 8

If a spare value is included in a CHOICE, and that has to be replaced with a new information element and an appropriate type in the new version, the name of the element replaces the spare name in the CHOICE, but the type cannot be replaced, because that would lead to incompatibilities. Instead, the new type is included in the nonCriticalExtension part of the message, as shown in Example 9 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        spare~~1~~                              NULL~~,~~
~~        spare2                              NULL~~
    }
}

-- Now a new option is needed for the element1 CHOICE: e3 with type E3.
-- MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If element1 has the value e3, the value of e3 is specified in the element e3
-- included in MessageA-r4-ext-IEs.
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        e3                                  NULL~~,~~
~~        spare2                              NULL~~
    }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e3.
    e3                                  E3          OPTIONAL
}
```

Example 9

~~12~~12 Guidelines involving different specification parts

12.1 Correction of inconsistencies between tabular and ASN.1 in RRC

This section includes guidelines describing how to correct inconsistencies between the tabular description and the ASN.1 message specifications of a protocol. The guidelines apply when the inconsistency is not corrected in a straightforward manner.

In some cases clarification should be added to the procedure specification. If the error is procedure specific, this clarification should be added to the procedure specification. Otherwise it should be added to section specifying the general actions upon reception of the concerned IE In case the message is send from UE to UTRAN, the procedure specification part of the solution may not be needed because RRC specification focuses on UE requirements.

In case comments should be added to the tabular description, these are to be included in the semantics description column.

12.1.1 Correcting the “need” of an IE

12.1.1.1 IE is optional in ASN.1 while it is correctly specified as mandatory in the tabular

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification, that if the information is absent, it shall be considered as a protocol error.

~~□~~- Align the tabular description with the ASN.1 and add a comment that the IE is required but that the need is set to OP in order to align with the ASN.1

~~□~~- Add comments to ASN.1 that the IE should be mandatory in later versions of this message

~~□Add comments to ASN.1 that the IE should be optional in later versions of this message~~

12.1.1.2 IE is mandatory in ASN.1 while it is correctly specified as optional in tabular

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification concerning what information should be sent when the IE is not required from a functional point of view. Furthermore, clarify what shall be done upon reception of this information

~~□~~- Align the tabular description with the ASN.1 and add a comment that the IE is not required but that the need is set to MP in order to align with the ASN.1

~~□~~- Add comments to ASN.1 that the IE should be optional in later versions of this message

~~□Add comments to ASN.1 that the IE should be mandatory in later versions of this message~~

12.1.2 Removing an IE

12.1.2.1 IE is optional in ASN.1 while it should be absent

The guidelines provided in this section apply in case it is not expected that the IE is to be used in a later release.

The correction should involve the following elements:

~~□~~- Remove IE from procedure description (if applicable).

~~□~~- Remove IE from tabular description (if applicable)

~~□~~- In ASN.1, replace IE by dummy with same type. Add comment to ASN.1 that IE shall not be sent and shall be ignored, if received. Also add comment that the IE should be removed in later versions of this message

NOTE: There is no need for procedure specification additional to the general statement about ignoring of dummy IEs.

12.1.2.2 IE is optional in ASN.1 while the associated functionality should be removed from this release

The guidelines provided in this section apply in case the functionality is to be removed from the concerned release while it is likely that it is to be used in a later release.

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification that IE should not be sent and that if received, the configuration is considered either as invalid or as undefined is in this version

~~□~~- Add comment to tabular description that IE is not used in this version of the specification

~~□~~- Add comment to ASN.1 that, in this version of the specification, IE shall not be sent and shall be ignored, if received

3GPP TSG-RAN WG2, Meeting #23
Helsinki, Finland, August 27th- 31st, 2001

Tdoc R2-012142

CR-Form-v4

CHANGE REQUEST

⌘	**25.921**	**CR**	**025**	⌘	ev	**-**	⌘	Current version:	**4.1.0**	⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☒ Radio Access Network ☒ Core Network ☐

Title:	⌘	Guidelines concerning conditions, spares, defaults and correction of inconsistencies			
Source:	⌘	TSG-RAN WG2			
Work item code:	⌘	TEI	***Date:***	⌘	31-08-2001
Category:	⌘	**F**	***Release:***	⌘	REL-4

Use one of the following categories:	*Use one of the following releases:*	
F *(correction)*	*2*	*(GSM Phase 2)*
A *(corresponds to a correction in an earlier release)*	*R96*	*(Release 1996)*
B *(addition of feature),*	*R97*	*(Release 1997)*
C *(functional modification of feature)*	*R98*	*(Release 1998)*
D *(editorial modification)*	*R99*	*(Release 1999)*
Detailed explanations of the above categories can	*REL-4*	*(Release 4)*
be found in 3GPP TR 21.900.	*REL-5*	*(Release 5)*

Reason for change: ⌘ The guidelines for how to use conditional information elements, spares and defaults are not clear enough. Furthermore, guidelines are missing concerning how to handle inconsistencies when straightforward corrections are not feasible

Summary of change: ⌘ This CR proposes changes in the following areas:

Conditional IEs

The need for an IE based on a condition is clarified to be one of the following:

- Mandatorily present
- Mandatory with default value
- Not needed
- Optional

Moreover, guidelines for how conditional expressions should be specified are added, for these four cases. A recommendation is also added that all cases should be explicitly covered by the conditional expression, e.g. by using the word “otherwise”.

Spare values

- For IEs in included in UL messages spares are used also for IEs with need set to MP, CV or CH
- For spares that result after mapping an information element value the same guidelines/ handling applies
- Within the tabular description the number of spares needed should be indicated. Within the ASN.1 spares should be used only to increase the size of the encoded size of an IE, when needed. Furthermore, for choices at most one spare should be allowed

Default values

- Variables may be used to clarify the specification of default values that depend on information received in the past

Guidelines for handling inconsistencies when straightforward ASN.1 corrections

		are not feasible • The guidelines cover the changes needed to procedure specification, tabular description and ASN.1 to overcome the following inconsistencies: • inconsistencies between tabular description and ASN.1 regarding the “Need” of an IE • removal of information elements e.g. because the functionality is incomplete Note: The changes are only made for the tabular guidelines applicable for RRC, since a separate discussion in RAN3 is ongoing regarding conditional IEs applicable on the RAN3 set of specifications. Isolated impact • No impact. This CR concerns guidelines for how to specify protocols. Since it involves no direct changes of protocols, there is no impact on implementations
Consequences if not approved:	⌘	Inconsistent use of conditions, spare values, defaults and correction of inconsistencies may lead to ambiguities in future versions of the RRC. This may again result in inter-operability problems

Clauses affected:	⌘	4.5.1, 4.5.2, 9.1.1.2.1, 9.1.1.2.1.1, 9.1.1.2.1.3, 9.1.3.2, 10.4.3.4.5, 12 (new)			
Other specs affected:	⌘	□	Other core specifications	⌘	25.921 v3.4.0, CR 024r1
		□	Test specifications		
		□	O&M Specifications		
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

4.5.1 Spare ~~Reserved~~ values and spare fields

~~Reserved values shall be forbidden. Otherwise entity receiving such a value shall reject the message. This would create difficulties when provided on broadcast channel.~~

In case the protocol allows for the transfer of spare information element values, the behaviour of the receiving entity not comprehending these values shall be specified.

~~Spare field shall be forbidden. Otherwise entity receiving such a spare field shall not make any decoding on that field and shall not reject the message.~~

In case the protocol allows for the transfer of spare fields, the behaviour of the receiving entity not comprehending the spare fields shall be specified. Facilities may be needed to allow receivers that don’t comprehend the spare fields to decode the other message parts.

4.5.2 Unspecified values

~~As far as possible default understanding shall be provided for unspecified values.~~

In case the protocol allows for the transfer of undefined information element values, the behaviour of the receiving entity not comprehending these values shall be specified.

9.1.1.2.1 Need and multiplicity (Multi) columns

These columns provide most of the information about the presence, absence and number of instances of the IE (in the message or in the group) or group of IEs. The different possibilities for these columns are described one by one.

The meaning of the 'need' column is summarised below:

MP Mandatorily present.

A value for that information is always needed, and no information is provided about a particular default value. If ever the transfer syntax allows absence (e.g., due to extension), then absence leads to an error diagnosis.

MD Mandatory with default value.

A value for that information is always needed, and a particular default value is mentioned (in the 'Semantical information' column). This opens the possibility for the transfer syntax to use absence or a special pattern to encode the default value.

CV Conditional on value.

The need for a value for that information depends on the value of some other IE or IEs, and/or on the message flow (e.g., channel, SAP). The need is specified by means of a condition, which result may be that the information is mandatorily present, mandatory with default value, not needed or optional.

~~A value for that information is needed (presence needed) or unacceptable (absence needed) when some conditions are met that can be evaluated on the sole basis of the content of the message.~~

~~If conditions for presence needed are specified,~~ If one of the results of the condition is that the information is mandatorily present, the transfer syntax must allow for the presence of the information. If in this case the information is absent ~~transfer syntax allows absence, absence when the conditions for presence are met leads to~~ an error is diagnosed~~is~~.

If one of the results of the condition is that the information is mandatory with default value, and a particular default value is mentioned (in the 'Semantical information' column), the transfer syntax may use absence or a special pattern to encode the default value.

~~If conditions for absence needed are specified,~~ If one of the results of the condition is that the information is not needed, the transfer syntax must allow encoding the absence. If in this case the information is present ~~and the conditions for absence are met~~, it will be ignored. In specific cases however, an error ~~is~~ may be diagnosed instead.

If one of the results of the condition is that the information is optional, the transfer syntax must allow for the presence of the information. In this case, neither absence nor presence of the information leads to an error diagnosis.

~~When neither conditions for presence or absence are met, the information is treated as optional, as described for 'OP'.~~

CH Conditional on history.

~~A value for that information is needed (presence needed) or unacceptable (absence needed) when some conditions are met that must be evaluated on the basis of~~ The need for a value for that information depends on information obtained in the past (e.g., from messages received in the past from the other party). The need is specified by means of a condition, which result may be that the information is mandatorily present, mandatory with default value, not needed or optional.

The handling of the conditions is the same as described for CV.

~~If conditions for presence needed are specified, the transfer syntax must allow for the presence of the information. If the transfer syntax allows absence, absence when the conditions for presence are met leads to an error diagnosis.~~

~~If conditions for absence needed are specified, the transfer syntax must allow encoding the absence. If the information is present and the conditions for absence are met, an error is diagnosed.~~

~~When neither conditions for presence or absence are met, the information is treated as optional, as described for ’OP’.~~

OP Optional.

The presence or absence is significant and modifies the behaviour of the receiver. However whether the information is present or not does not lead to an error diagnosis.

9.1.1.2.1.1 Mandatory

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
Name	MP				
Name	MD			(default value is indicated)	

The multiplicity column may be left empty (see 9.1.1.2.1.5).

For mandatory IEs, the rules are as follows, applied on the number of instances given by the multiplicity column (leaving the multiplicity column empty means one and only one instance):

For an IE not belonging to a group MP indicates that the number of instances as given by the multiplicity column of ’Name IE’ is necessary in the message.

For a group not belonging to another group, MP means that the number of instances as given by the multiplicity column of the ’Name group’ is necessary in the message.

For an IE or a group belonging to another group, MP means that if the parent group is present, then the number of instances as given by the multiplicity column of the ’Name group’ or ’Name IE’ is necessary in the embedding group.

For an IE not belonging to a group MD indicates that the number of instances as given by the multiplicity column for information ’Name IE’ is necessary in the message, and that a special value (the default value) exists, for all instances or individual instances, and is mentioned in the ’Semantics description’ column.

For a group not belonging to another group, MD means that the number of instances as given by the multiplicity column for information structure ’Name group’ is necessary in the message, and that a special value (the default value) exists, for all instances or individual instances, and is mentioned in the ’Semantics description’ column.

For an IE or a group belonging to another group, MD means that if the parent group is present, then the number of instances as given by the multiplicity column for information structure ’Name group’ or information ’Name IE’ is necessary in the embedding group, and that a special value (the default value) exists and is mentioned in the ’Semantics description’ column.

The default value might be fixed by the standard, or conditional to the value of some other IE or IEs, or conditional on information obtained in the past. In case the default value depends on information obtained in the past, variables may be used to specify it.

9.1.1.2.1.2 Optional

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
Name	OP				

The multiplicity column may be empty (see 9.1.1.2.1.5).

This indicates that the number of instances as given by the multiplicity column of the ’Name IE’ or ’Name group’ is not necessary in the message or the embedding group, and that the sender can choose not to include it.

9.1.1.2.1.3 Conditional

IE/Group Name	Need	Multi	Type and reference	Semantics description	Version
	CV *cond*				
	CH				

The multiplicity column may be empty (see 9.1.1.2.1.5).

CV indicates that the requirement for presence or absence of the number of instances as given by the multiplicity column of the IE or group of IE depends on the value of some other IE or IEs, and/or on the message flow (e.g., channel, SAP). In the CV case, the condition is to be described in a textual form in an explanatory clause. *cond* stands for a free text that is used as a reference in the title of the explanatory clause. In the CH case, the condition is described in the procedural section.

The expression *cond* should explicitely cover all cases. A typical *cond* expression is “The IE is *CS1* if *TE* and *CS2* otherwise”, where *CS1* and *CS1* are conditon statements and *TE* is an expression used in the if test. For example: “The IE is mandatory if the IE “XXX” has the value “YYY” and not needed otherwise”.

The result of evaluating the condition (if the condition is met or not) may mean that the IE is:

- Mandatorily present, when the condition statement says “mandatory”.
- Mandatory with default value, when the condition statement says “mandatory with default value”
- ~~Mandatorily absent~~ Not needed, when the condition statement says “not needed”.
- Optional.

~~- Absent, but optional (this is meaningful only for extension).~~

The error handling shall be specified in the protocol for the cases when the requirement for presence or absence of an IE indicated by the condition is not followed.

9.1.3 Extension for further releases

9.1.3.1 Basic principle

Added elements or choice branches are included where they fit most naturally according to their semantics, independently from the version in which they were added.

9.1.3.2 Critical or non-critical

A case-by-case guideline is provided by informal text after the table itself.

Spare values shall not be defined for c~~C~~ritical~~ spare values are banned from the tabular~~ information elements. Extension of the value range of c~~C~~ritical information elements ~~extensions of enumerated ~~must be done by creating a critical extension.

~~Non-critical spare values shall not appear.~~

Spare values may be defined for non- critical information elements. For spares that result after mapping of the original information element value the same handling and/ or guidelines apply.

In case spares are defined, t~~T~~he tabular notation should ~~shall ~~indicate the number of ~~only that 'at least one ~~non critical spare values that is needed~~'~~. ~~The exact number of spare values is found in the transfer syntax.~~ Within the ASN.1 spare values should only be used to increase the encoded size of an IE. This means that the ASN.1 should only include spares if the number of spare values that is needed exceeds the number of undefined values within the transfer syntax of the information element.

For downlink messages, spare values may be defined for non-critical information elements for which the need is specified to be MD or OP (or CV case leading to MD or OP). In this case, a receiver not comprehending the received a spare value shall consider the information element to have the default value or consider it to be absent respectively.~~Non-critical spare values are only possible with MD or OP fields (or CV case leading to MD or OP), and an 'old' receiver receiving a non-critical spare value shall consider it as meaning respectively default value or absence.~~

For uplink messages spares values may be defined for all information elements, including those for which the need is specified to be MP (or CV case leading to MP).

In all cases at most one spare should be defined for choices.

10.4.3.4.5 Replacement of a spare value with a new element

If a new value is to be included in an IE of type ENUMERATED, for which spare values were defined in the previous version, those spare values can be replaced with the new values.

If more new values are needed, than spare values included in the previous version, one spare value can be replaced by a special extension value (called e-new in example 8). If that value is used, a new element in the nonCriticalExtension part (element1-new) will define the new values, as shown in Example 8 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            ENUMERATED { e1, e2, spare1, spare2 }
}

-- Now three new values are needed for element1: e3, e4 and e5. MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If the following has the value e-new, the actual value of element1 is defined in
-- element1-new included in MessageA-r4-ext-IEs
    element1                            ENUMERATED { e1, e2, e3, e-new }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e-new.
    element1-new                        ENUMERATED { e4, e5, spare1, spare2 }   OPTIONAL
}
```

Example 8

If a spare value is included in a CHOICE, and that has to be replaced with a new information element and an appropriate type in the new version, the name of the element replaces the spare name in the CHOICE, but the type cannot be replaced, because that would lead to incompatibilities. Instead, the new type is included in the nonCriticalExtension part of the message, as shown in Example 9 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        spare~~1~~                              NULL~~,~~
~~        spare2                              NULL~~
    }
}

-- Now a new option is needed for the element1 CHOICE: e3 with type E3.
-- MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If element1 has the value e3, the value of e3 is specified in the element e3
-- included in MessageA-r4-ext-IEs.
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        e3                                  NULL~~,~~
~~        spare2                              NULL~~
    }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e3.
    e3                                  E3          OPTIONAL
}
```

Example 9

~~12~~12 Guidelines involving different specification parts

12.1 Correction of inconsistencies between tabular and ASN.1 in RRC

This section includes guidelines describing how to correct inconsistencies between the tabular description and the ASN.1 message specifications of a protocol. The guidelines apply when the inconsistency is not corrected in a straightforward manner.

In some cases clarification should be added to the procedure specification. If the error is procedure specific, this clarification should be added to the procedure specification. Otherwise it should be added to section specifying the general actions upon reception of the concerned IE In case the message is send from UE to UTRAN, the procedure specification part of the solution may not be needed because RRC specification focuses on UE requirements.

In case comments should be added to the tabular description, these are to be included in the semantics description column.

12.1.1 Correcting the “need” of an IE

12.1.1.1 IE is optional in ASN.1 while it is correctly specified as mandatory in the tabular

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification, that if the information is absent, it shall be considered as a protocol error.

~~□~~- Align the tabular description with the ASN.1 and add a comment that the IE is required but that the need is set to OP in order to align with the ASN.1

~~□~~- Add comments to ASN.1 that the IE should be mandatory in later versions of this message

12.1.1.2 IE is mandatory in ASN.1 while it is correctly specified as optional in tabular

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification concerning what information should be sent when the IE is not required from a functional point of view. Furthermore, clarify what shall be done upon reception of this information

~~□~~- Align the tabular description with the ASN.1 and add a comment that the IE is not required but that the need is set to MP in order to align with the ASN.1

~~□~~- Add comments to ASN.1 that the IE should be optional in later versions of this message

12.1.2 Removing an IE

12.1.2.1 IE is optional in ASN.1 while it should be absent

The guidelines provided in this section apply in case it is not expected that the IE is to be used in a later release.

The correction should involve the following elements:

~~□~~- Remove IE from procedure description (if applicable).

~~□~~- Remove IE from tabular description (if applicable)

~~□~~- In ASN.1, replace IE by dummy with same type. Add comment to ASN.1 that IE shall not be sent and shall be ignored, if received. Also add comment that the IE should be removed in later versions of this message

NOTE: There is no need for procedure specification additional to the general statement about ignoring of dummy IEs.

12.1.2.2 IE is optional in ASN.1 while the associated functionality should be removed from this release

The guidelines provided in this section apply in case the functionality is to be removed from the concerned release while it is likely that it is to be used in a later release.

The correction should involve the following elements:

~~□~~- Add clarification to procedure specification that IE should not be sent and that if received, the configuration is considered either as invalid or as undefined is in this version

~~□~~- Add comment to tabular description that IE is not used in this version of the specification

~~□~~- Add comment to ASN.1 that, in this version of the specification, IE shall not be sent and shall be ignored, if received

3GPP TSG-RAN WG2 Meeting #23
Helsinki, Finland, 27-31.07.2001

Tdoc 011849

CR-Form-v4

CHANGE REQUEST

⌘	**25.921**	**CR**	**025**	⌘ ev	**-**	⌘ Current version:	**3.4.0**	⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network □ Core Network □

Title:	⌘	Naming convention for non-critical extensions		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	21 Aug 01
Category:	⌘	**F**	***Release:*** ⌘	R99
		Use one of the following categories: **F** *(correction)* **A** *(corresponds to a correction in an earlier release)* **B** *(addition of feature),* **C** *(functional modification of feature)* **D** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.	*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*	

Reason for change:	⌘	The guidelines specified in 25.921 do not correctly reflect the current best practice being utilised in RAN WG2.
Summary of change:	⌘	The sections relating to the naming of non-critical extensions have been updated to show the new format that is being used. All text and examples have been updated.
Consequences if not approved:	⌘	Out of date guidelines being used for the production of CRs, may lead to CRs being rejected for purely editorial reasons.

Clauses affected:	⌘	10.4.2, 10.4.3.2, 10.4.3.3, 10.4.3.4.1, 10.4.3.4.2, 10.4.3.4.3, 10.4.3.4.4		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	25.921 v4.1.0, CR 027
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

10.4 Extensions for future releases in RRC

10.4.1 Basic principles

All non-critical extensions are shown even if empty as it costs no bits.

10.4.2 Naming convention

The abstract type defining a message provides mechanisms to allow for extending the message in future releases:

- For critical extensions, this is done by defining the message as a CHOICE of two alternatives, one being the intended message structure, and the other being an empty SEQUENCE named "criticalExtensions".
- For non-critical extensions, this is done by defining an OPTIONAL element named "nonCriticalExtensions" of type "SEQUENCE {}" at the end of the message definition.

When extensions are introduced, this is done by replacing one of the empty SEQUENCEs by a new structure, that includes a new type containing the message extensions, and the same extension mechanism recursively for further extensions.

For critical extensions ~~T~~the new elements introduced to specify the extensions should be grouped together in an element with a name showing the release~~,~~ in which the extension was made, and this should be the same as for the new message root. ~~and the release of the message root, on which the extension was made (the second one applies only for non-critical extensions)~~. For this naming, "r3" is used for Release '99, "r4" for Release 4, "r5" for Release 5 and so on.~~ The suffix "ext" is used to indicate non-critical extensions.~~

For non-critical extensions the new elements introduced to specify the extensions should be grouped together in an element with a name showing the version of the specification where this extension will first be included, e.g. if the version of the specification being corrected is v3.7.0, then the suffix added to the name will be -v380ext (i.e. the next version).

If non-critical extensions for two different roots happen to be identical in contents, their types are still named differently, possibly with the second being declared as synonymous to the first.

An example is given below to illustrate these principles, on the message named "Test-msg"

```
-- In Release '99, the Test-msg is defined as following:
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        nonCriticalExtensions               SEQUENCE {} OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}
    }
}
-- A later correction to Release 99 adds a non-critical extension in v3.8.0
-- of the specification
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg-v380ext                        Test-msg-v380ext-IEs,
            nonCriticalExtensions                   SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}
    }
}
-- ~~In Release 4, t~~The Test-msg gets the following structure, if ~~critical and ~~only a non-critical
-- extensions ~~are ~~is introduced for Release 4 in v4.4.0 of the specification.
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg-~~r3-r4-~~v380ext                  Test-msg-~~r3-r4-~~v380ext-IEs,
            v440nonCriticalExtensions               SEQUENCE {
                test-msg-v440ext                        Test-msg-v440ext-IEs,
                nonCriticalExtensions                   SEQUENCE {} OPTIONAL
            }   OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}~~CHOICE {~~
~~        r4                                  SEQUENCE {~~
~~            test-msg-r4                         Test-msg-r4-IEs,~~
~~            nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
~~        },~~
~~        criticalExtensions                  SEQUENCE {}~~
    }
}

-- In Release 5, the Test msg gets the following structure when a critical
-- extension is added
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg- v380ext                   Test-msg-v380ext-IEs,
            v440nonCriticalExtensions               SEQUENCE {
                test-msg-v440ext                        Test-msg-v440ext-IEs,
                nonCriticalExtensions                   SEQUENCE {} OPTIONAL
            }   OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  CHOICE {
            r5                                  SEQUENCE {
                test-msg-r5                         Test-msg-r5-IEs,
                nonCriticalExtensions               SEQUENCE {} OPTIONAL
            },
            criticalExtensions                  SEQUENCE {}
        }
    }
}

~~-- In Release 5, the Test-msg gets the following structure, if more critical and non-critical~~
~~-- extensions are introduced.~~
~~-- Here, non-critical extensions are introduced in both the r3 and r4 root of Test-msg.~~
~~Test-msg ::= CHOICE {~~
~~    r3                                  SEQUENCE {~~
~~        test-msg-r3                         Test-msg-r3-IEs,~~
~~        nonCriticalExtensions               SEQUENCE {~~
~~            test-msg-r3-r4-ext                  Test-msg-r3-r4-ext-IEs,~~
~~            nonCriticalExtensions               SEQUENCE {~~
~~                test-msg-r3-r5-ext                  Test-msg-r3-r5-ext-IEs,~~
~~                nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
~~            }   OPTIONAL~~
~~        } OPTIONAL~~
~~    },~~
~~    criticalExtensions                  CHOICE {~~
~~        r4                                  SEQUENCE {~~
~~            test-msg-r4                         Test-msg-r4-IEs,~~
~~            nonCriticalExtensions               SEQUENCE {~~
~~                test-msg-r4-r5-ext                  Test-msg-r4-r5-ext-IEs,~~
~~                nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
~~            }   OPTIONAL~~
~~        },~~
~~        criticalExtensions                  CHOICE {~~
~~            r5                                  SEQUENCE {~~
~~                test-msg-r5                         Test-msg-r5-IEs,~~
~~                nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
~~            },~~
~~            criticalExtensions                  SEQUENCE {}~~
~~        }~~
~~    }~~
~~}~~
```

Critical extensions in Release *N* in message "Test-msg" should be included in the type "Test-msg-r*N*-IEs" (*N*=3 is used for Release '99).

~~Non-critical extensions in Release *N* included in the Release *M* branch of the top-level CHOICE should be included in a type "Test-msg-r*M*-r*N*-ext-IEs".~~

If an abstract type is introduced in Release *N* when new elements are included in an extension, it should have a suffix "-r*N*". For Release '99 types, no such suffix is used.

If an abstract type is introduced in a ~~R~~release ~~*N*~~ to extend an already existing type "TypeX", it should get the same name with a non-critical extension type suffix ("-~~r*N*-~~v*XYZ*ext", ~~i.e.~~ e.g. "TypeX~~-r*N*~~-v380ext")~~.~~ although in this case the final "–IEs" suffix is not added.

Using the above naming rules, when changes are done in Release *N*, only changes in types with a suffix "-r*N*" or "-~~r*N*-~~v*XYZ*ext" are allowed, in order to avoid conflicts with previous releases. An exception is the Message type itself, which can be changed by replacing the empty SEQUENCEs with extensions as shown above, and elements having spare values defined, where the spare value can be replaced with a newly introduced value.

An exception to the above structure can be needed, if there are some elements to be used in a message, which need to be comprehended even in case of critical extensions (e.g. for error handling procedures). In this case, the elements can be placed before one of the criticalExtensions CHOICEs, as shown in the example below:

```
Test-msg ::= CHOICE {
    r3                              SEQUENCE {
        test-msg-r3                     Test-msg-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            test-msg-~~r3-r4-~~v380ext              Test-msg-~~r3-r4-~~v380ext-IEs,
            nonCriticalExtensions           SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    later-than-r3                   SEQUENCE {
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
        criticalExtensions              SEQUENCE {
            importantElements               ImportantElements,
            rest-of-message                 CHOICE {
                r4                              SEQUENCE {
                    test-msg-r4                     Test-msg-r4-IEs,
                    nonCriticalExtensions           SEQUENCE {} OPTIONAL
                },
                criticalExtensions              SEQUENCE {}
            }
        }
    }
}
```

In the above example, the elements in "importantElements" can be comprehended from a UE implementing this structure, even if a future version of the message including critical extensions is transmitted (i.e. the criticalExtension branch of the second CHOICE is used).

NOTE: The structure presented in this clause and the proposed naming rules are one possibility. Further possibilities are FFS.

NOTE: When non-critical extensions are introduced in a message that does not have yet a criticalExtension branch, they are introduced in the "Test-msg-~~r*M*-r*N*-ext~~v380ext -IEs" type as described above. It is possible, that after this change, another change introduces a critical extension for the same message, thus defining a critical extension branch. In this case, the whole message is redefined in the type "Test-msg-r*N*-IEs", and care is to be taken to include in this new type also all non-critical extensions that were introduced previously, in a way that best fits the new structure of the message.

- To be prepared for such cases, it could be beneficial to define in advance the "Test-msg-r*N*-IEs" whenever a non-critical extension is introduced, which would be an unused type mirroring the actual structure of the message, as long as no critical extensions are introduced, and would be used as the basis of the message if a critical extension is introduced. It is FFS if this concept is feasible, and if it should be introduced in the future.

10.4.3 Recommendations for extensions for further releases in RRC

10.4.3.1 General

When in RRC an information element group is to be extended, the extension cannot be done directly in that IE, but only in the top level of the message, in the extension IEs of the message structure shown in Example 1. For implementing the extension, it has therefore to be investigated, in which messages the element to be extended is included.

Depending on criticality of the extension, this will be done by using the criticalExtension CHOICE branch, or the nonCriticalExtension information element.

The following subclauses provide some recommendations on how to use these elements.

```
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        nonCriticalExtensions           SEQUENCE {} OPTIONAL
    },
    criticalExtensions              SEQUENCE {}
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- All messageA related information elements are included here.
}
```

Example 1

10.4.3.2 Critical Extensions

When the extension is a critical one (i.e. the receiver has to reject the whole message, and handle according to the error procedures of the protocol), the criticalExtension branch of the top-level CHOICE in the message is used. In this case the message information elements can be updated similar to the tabular, providing a message structure for the new release's information elements, similar to the updated structure in the tabular description.

Example 2 shows the structure of MessageA presented above, how it would become after a critical extension in Release 4.

In this example, in the criticalExtensions branch a new information element is defined (MessageA-r4-IEs) which will contain all messageA specific elements for Release 4, including the extensions in the place they fit naturally according to the semantics.

Note that in the new structure additional nonCriticalExtensions and criticalExtensions information elements are defined to allow for further extensions in future releases.

```
MessageA ::= CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        nonCriticalExtensions           SEQUENCE {} OPTIONAL
    },
    later-than-r3                   SEQUENCE {
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
____    criticalExtensions              CHOICE {
        ____r4                              SEQUENCE {
            ____messageA-r4                     MessageA-r4-IEs,
            ____nonCriticalExtensions           SEQUENCE {} OPTIONAL
        ____},
        ____criticalExtensions          SEQUENCE {}
        }
    }
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- This is not changed compared to the above example. It includes all information
    -- elements used in Release '99 for messageA.
}

MessageA-r4-IEs ::=                 SEQUENCE {
    -- Here, the updated information elements used for MessageA in Release 4 are included.
}
```

Example 2

10.4.3.3 Non-critical Extensions

For non-critical extensions (i.e. the receiver shall just ignore the extensions, and use the rest of the message as if the extensions were not present), the approach is to use the nonCriticalExtensions information element, which is encoded at the end of the message, allowing backward compatibility.

The structure of the message of the example above is shown in Example 3 for the Release 4 message

Examples for special non-critical extensions and MessageA~~-r3-r4-~~v440ext-IEs are given in the following subclauses.

```
MessageA-r4 ::=                 CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            messageA-~~r3-r4-~~v440ext                  MessageA-~~r3-r4-~~v440ext-IEs,
            nonCriticalExtensions           SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    criticalExtensions              SEQUENCE {}
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- This is not changed compared to the same IE in Release ’99. It includes all information
    -- elements used in Release ’99 for messageA.
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    -- Here are additional information elements needed to describe the extensions compared to
    -- the information included in MessageA-r3-IEs.
}
```

Example 3

10.4.3.4 Examples of non-critical extensions

10.4.3.4.1 Addition of a separate IE

If the extension is the addition of an information element (not inside a CHOICE, SEQUENCE OF, SET OF etc.), this new element can be directly included in MessageA~~-r3-r4-~~v440ext-IEs.

Example4 shows how the MessageA is extended to include a new element, "element3".

```
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            Element1,
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    element3                            Element3-r4
}
```

Example 4

10.4.3.4.2 Addition of an IE to a structured group

If the extension is the addition of an information element inside a CHOICE, SEQUENCE OF, etc. (meaning that the information element can be absent or present more than once, depending on some condition), the structure of the original message should be duplicated in MessageA~~-r3-r4-~~v440ext-IEs using only the elements relevant to the extension (usually the CHOICEs, SEQUENCE OFs, etc.), and a comment should be included to indicate that the two structures should be used consistently (e.g. when a CHOICE is duplicated, the same branch should be followed in both places, when a SEQUENCE OF is duplicated, the number of occurrences should be the same etc.).

This is illustrated in Example5, where a new element, "element1a-3", has to be included inside the "choice1b" branch of the "choice1" CHOICE. Here "choice1" is included again in MessageA~~-r3-r4-~~v440ext-IEs, and "element1a-3" is included there in the appropriate branch.

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- For the "choice1b" branch of "choice1", an additional information element is
-- defined in MessageA-~~r3-r4-~~v440ext-IEs ("element1a-3").
    choice1                             CHOICE {
        choice1a                            SEQUENCE {
            element1a-1                         Element1a-1
        },
        choice1b                            SEQUENCE {
            element1a-2                         Element1a-2
        }
    }
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
-- In the following CHOICE the same branch shall be used as in choice1 in MessageA-r3-IEs.
    choice1                             CHOICE {
        choice1a                            NULL,
        choice1b                            SEQUENCE {
            element1a-3                         Element1a-3-r4
        }
    }
}
```

Example 5

10.4.3.4.3 Addition of a new CHOICE group

If the extension consists of moving some existing information elements inside a newly created CHOICE, the new branches of the created CHOICE should be included in MessageA~~-r3-r4-~~v440ext-IEs, and the CHOICE marked OPTIONAL, where absence means that the old elements are used. If the CHOICE is present, the old elements should be set to some default values, in order for older equipment to be understood, and new equipment should ignore the information therein.

This is illustrated in Example 6, where "element1" is to be moved inside the branch "choice1a" of a new CHOICE ("choice1").

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- The contents of "element1" shall be ignored, if in "MessageA-~~r3-r4-~~v440ext-IEs" the branch
-- "choice1b" of the CHOICE "choice1" is used.
    element1                            Element1,
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    choice1                             CHOICE {
        choice1a                            SEQUENCE {},
        choice1b                            SEQUENCE {
            element3                            Element3-r4
        }
    }
}
```

Example 6

10.4.3.4.4 Extension of value range

If the value range of an element is to be extended, an element including the new values should be defined in MessageA-~~r3-r4-~~v440ext-IEs. If one of the new values is to be used, the already existing element from Release '99 should be set to some defined value (or be absent if it was OPTIONAL), in order for older equipment to work properly, and the new value should be signalled in the new information element.

In Example 7, "element1" is extended to have a range (0..15).

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- "element1" shall be ignored if "element1" in MessageA-~~r3-r4-~~v440ext-IEs is present, and the
-- value of that element used instead.
    element1                            INTEGER (0..7)
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    element1                            INTEGER (0..15)         OPTIONAL
}
```

Example 7

10.4.3.4.5 Replacement of a spare value with a new element

If a new value is to be included in an IE of type ENUMERATED, for which spare values were defined in the previous version, those spare values can be replaced with the new values.

If more new values are needed, than spare values included in the previous version, one spare value can be replaced by a special extension value (called e-new in example 8). If that value is used, a new element in the nonCriticalExtension part (element1-new) will define the new values, as shown in Example 8 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            ENUMERATED { e1, e2, spare1, spare2 }
}

-- Now three new values are needed for element1: e3, e4 and e5. MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If the following has the value e-new, the actual value of element1 is defined in
-- element1-new included in MessageA-r4-ext-IEs
    element1                            ENUMERATED { e1, e2, e3, e-new }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e-new.
    element1-new                        ENUMERATED { e4, e5, spare1, spare2 }   OPTIONAL
}
```

Example 8

If a spare value is included in a CHOICE, and that has to be replaced with a new information element and an appropriate type in the new version, the name of the element replaces the spare name in the CHOICE, but the type cannot be replaced, because that would lead to incompatibilities. Instead, the new type is included in the nonCriticalExtension part of the message, as shown in Example 9 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        spare1                              NULL,
        spare2                              NULL
    }
}

-- Now a new option is needed for the element1 CHOICE: e3 with type E3.
-- MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If element1 has the value e3, the value of e3 is specified in the element e3
-- included in MessageA-r4-ext-IEs.
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        e3                                  NULL,
        spare2                              NULL
    }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e3.
```

```
    e3                                  E3          OPTIONAL
}
```

Example 9

3GPP TSG-RAN WG2 Meeting #23
Helsinki, Finland, 27-31.07.2001

Tdoc 012143

CR-Form-v4

CHANGE REQUEST

⌘	**25.921**	**CR**	**027**	⌘	ev	**-**	⌘	Current version:	**4.1.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network □ Core Network □

Title:	⌘	Naming convention for non-critical extensions		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	30 Aug 01
Category:	⌘	**F**	***Release:*** ⌘	REL-4
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	The guidelines specified in 25.921 do not correctly reflect the current best practice being utilised in RAN WG2.
Summary of change:	⌘	The sections relating to the naming of non-critical extensions have been updated to show the new format that is being used. All text and examples have been updated.
Consequences if not approved:	⌘	Out of date guidelines being used for the production of CRs, may lead to CRs being rejected for purely editorial reasons.

Clauses affected:	⌘	10.4.2, 10.4.3.2, 10.4.3.3, 10.4.3.4.1, 10.4.3.4.2, 10.4.3.4.3, 10.4.3.4.4		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	25.921 v3.4.0, CR 026
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

10.4 Extensions for future releases in RRC

10.4.1 Basic principles

All non-critical extensions are shown even if empty as it costs no bits.

10.4.2 Naming convention

The abstract type defining a message provides mechanisms to allow for extending the message in future releases:

- For critical extensions, this is done by defining the message as a CHOICE of two alternatives, one being the intended message structure, and the other being an empty SEQUENCE named "criticalExtensions".

- For non-critical extensions, this is done by defining an OPTIONAL element named "nonCriticalExtensions" of type "SEQUENCE {}" at the end of the message definition.

When extensions are introduced, this is done by replacing one of the empty SEQUENCEs by a new structure, that includes a new type containing the message extensions, and the same extension mechanism recursively for further extensions.

For critical extensions ~~T~~the new elements introduced to specify the extensions should be grouped together in an element with a name showing the release~~,~~ in which the extension was made, and this should be the same as for the new message root.~~ and the release of the message root, on which the extension was made (the second one applies only for non-critical extensions)~~. For this naming, "r3" is used for Release '99, "r4" for Release 4, "r5" for Release 5 and so on.~~ The suffix "ext" is used to indicate non-critical extensions.~~

For non-critical extensions the new elements introduced to specify the extensions should be grouped together in an element with a name showing the version of the specification where this extension will first be included, e.g. if the version of the specification being corrected is v3.7.0, then the suffix added to the name will be -v380ext (i.e. the next version).

If non-critical extensions for two different roots happen to be identical in contents, their types are still named differently, possibly with the second being declared as synonymous to the first.

An example is given below to illustrate these principles, on the message named "Test-msg"

```
-- In Release ’99, the Test-msg is defined as following:
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        nonCriticalExtensions               SEQUENCE {} OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}
    }
}
-- A later correction to Release 99 adds a non-critical extension in v3.8.0
-- of the specification
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg-v380ext                        Test-msg-v380ext-IEs,
            nonCriticalExtensions                   SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}
    }
}
-- ~~In Release 4, t~~The Test-msg gets the following structure, if ~~critical and ~~only a non-critical
-- extensions ~~are ~~is introduced for Release 4 in v4.4.0 of the specification.
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg-~~r3-r4-~~v380ext               Test-msg-~~r3-r4-~~v380ext-IEs,
            v440nonCriticalExtensions               SEQUENCE {
                test-msg-v440ext                        Test-msg-v440ext-IEs,
                nonCriticalExtensions                   SEQUENCE {} OPTIONAL
            }   OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  SEQUENCE {}~~CHOICE {~~
        ~~r4                                  SEQUENCE {~~
            ~~test-msg-r4                         Test-msg-r4-IEs,~~
            ~~nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
        ~~},~~
        ~~criticalExtensions                  SEQUENCE {}~~
    }
}

-- In Release 5, the Test msg gets the following structure when a critical
-- extension is added
Test-msg ::= CHOICE {
    r3                                  SEQUENCE {
        test-msg-r3                         Test-msg-r3-IEs,
        v380nonCriticalExtensions               SEQUENCE {
            test-msg- v380ext                   Test-msg-v380ext-IEs,
            v440nonCriticalExtensions               SEQUENCE {
                test-msg-v440ext                        Test-msg-v440ext-IEs,
                nonCriticalExtensions                   SEQUENCE {} OPTIONAL
            }   OPTIONAL
        } OPTIONAL
    },
    later-than-r3                       SEQUENCE {
        rrc-TransactionIdentifier           RRC-TransactionIdentifier,
        criticalExtensions                  CHOICE {
            r5                                  SEQUENCE {
                test-msg-r5                         Test-msg-r5-IEs,
                nonCriticalExtensions               SEQUENCE {} OPTIONAL
            },
            criticalExtensions                  SEQUENCE {}
        }
    }
}

~~-- In Release 5, the Test-msg gets the following structure, if more critical and non-critical~~
~~-- extensions are introduced.~~
~~-- Here, non-critical extensions are introduced in both the r3 and r4 root of Test-msg.~~
~~Test-msg ::= CHOICE {~~
    ~~r3                                  SEQUENCE {~~
        ~~test-msg-r3                         Test-msg-r3-IEs,~~
        ~~nonCriticalExtensions               SEQUENCE {~~
            ~~test-msg-r3-r4-ext                  Test-msg-r3-r4-ext-IEs,~~
            ~~nonCriticalExtensions               SEQUENCE {~~
                ~~test-msg-r3-r5-ext                  Test-msg-r3-r5-ext-IEs,~~
                ~~nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
            ~~}   OPTIONAL~~
        ~~} OPTIONAL~~
    ~~},~~
    ~~criticalExtensions                  CHOICE {~~
        ~~r4                                  SEQUENCE {~~
            ~~test-msg-r4                         Test-msg-r4-IEs,~~
            ~~nonCriticalExtensions               SEQUENCE {~~
                ~~test-msg-r4-r5-ext                  Test-msg-r4-r5-ext-IEs,~~
                ~~nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
            ~~}   OPTIONAL~~
        ~~},~~
        ~~criticalExtensions                  CHOICE {~~
            ~~r5                                  SEQUENCE {~~
                ~~test-msg-r5                         Test-msg-r5-IEs,~~
                ~~nonCriticalExtensions               SEQUENCE {} OPTIONAL~~
            ~~},~~
            ~~criticalExtensions                  SEQUENCE {}~~
        ~~}~~
    ~~}~~
~~}~~
```

Critical extensions in Release *N* in message "`Test-msg`" should be included in the type "`Test-msg-r`*`N`*`-IEs`" (*N*=3 is used for Release '99).

~~Non-critical extensions in Release *N* included in the Release *M* branch of the top-level CHOICE should be included in a type "`Test-msg-r`*`M`*`-r`*`N`*`-ext-IEs`".~~

If an abstract type is introduced in Release *N* when new elements are included in an extension, it should have a suffix "`-r`*`N`*". For Release '99 types, no such suffix is used.

If an abstract type is introduced in a ~~R~~release ~~*N*~~ to extend an already existing type "`TypeX`", it should get the same name with a non-critical extension type suffix ("`-` ~~`r`*`N`*`-`~~`v`*`XYZ`*`ext`", ~~i.e.~~ e.g. "`TypeX`~~`-r`*`N`*~~`-v380ext`")~~.~~ although in this case the final "–IEs" suffix is not added.

Using the above naming rules, when changes are done in Release *N*, only changes in types with a suffix "`-r`*`N`*" or "`-`~~`r`*`N`*`-`~~`v`*`XYZ`*`ext`" are allowed, in order to avoid conflicts with previous releases. An exception is the Message type itself, which can be changed by replacing the empty SEQUENCEs with extensions as shown above, and elements having spare values defined, where the spare value can be replaced with a newly introduced value.

An exception to the above structure can be needed, if there are some elements to be used in a message, which need to be comprehended even in case of critical extensions (e.g. for error handling procedures). In this case, the elements can be placed before one of the criticalExtensions CHOICEs, as shown in the example below:

```
Test-msg ::= CHOICE {
    r3                              SEQUENCE {
        test-msg-r3                     Test-msg-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            test-msg-~~r3-r4-~~v380ext              Test-msg-~~r3-r4-~~v380ext-IEs,
            nonCriticalExtensions           SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    later-than-r3                   SEQUENCE {
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
        criticalExtensions              SEQUENCE {
            importantElements               ImportantElements,
            rest-of-message                 CHOICE {
                r4                              SEQUENCE {
                    test-msg-r4                     Test-msg-r4-IEs,
                    nonCriticalExtensions           SEQUENCE {} OPTIONAL
                },
                criticalExtensions              SEQUENCE {}
            }
        }
    }
}
```

In the above example, the elements in "importantElements" can be comprehended from a UE implementing this structure, even if a future version of the message including critical extensions is transmitted (i.e. the criticalExtension branch of the second CHOICE is used).

NOTE: The structure presented in this clause and the proposed naming rules are one possibility. Further possibilities are FFS.

NOTE: When non-critical extensions are introduced in a message that does not have yet a criticalExtension branch, they are introduced in the "Test-msg-~~r*M*-r*N*-ext~~v380ext -IEs" type as described above. It is possible, that after this change, another change introduces a critical extension for the same message, thus defining a critical extension branch. In this case, the whole message is redefined in the type "Test-msg-r*N*-IEs", and care is to be taken to include in this new type also all non-critical extensions that were introduced previously, in a way that best fits the new structure of the message.

- To be prepared for such cases, it could be beneficial to define in advance the "Test-msg-r*N*-IEs" whenever a non-critical extension is introduced, which would be an unused type mirroring the actual structure of the message, as long as no critical extensions are introduced, and would be used as the basis of the message if a critical extension is introduced. It is FFS if this concept is feasible, and if it should be introduced in the future.

10.4.3 Recommendations for extensions for further releases in RRC

10.4.3.1 General

When in RRC an information element group is to be extended, the extension cannot be done directly in that IE, but only in the top level of the message, in the extension IEs of the message structure shown in Example 1. For implementing the extension, it has therefore to be investigated, in which messages the element to be extended is included.

Depending on criticality of the extension, this will be done by using the criticalExtension CHOICE branch, or the nonCriticalExtension information element.

The following subclauses provide some recommendations on how to use these elements.

```
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        nonCriticalExtensions           SEQUENCE {} OPTIONAL
    },
    criticalExtensions              SEQUENCE {}
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- All messageA related information elements are included here.
}
```

Example 1

10.4.3.2 Critical Extensions

When the extension is a critical one (i.e. the receiver has to reject the whole message, and handle according to the error procedures of the protocol), the criticalExtension branch of the top-level CHOICE in the message is used. In this case the message information elements can be updated similar to the tabular, providing a message structure for the new release's information elements, similar to the updated structure in the tabular description.

Example 2 shows the structure of MessageA presented above, how it would become after a critical extension in Release 4.

In this example, in the criticalExtensions branch a new information element is defined (MessageA-r4-IEs) which will contain all messageA specific elements for Release 4, including the extensions in the place they fit naturally according to the semantics.

Note that in the new structure additional nonCriticalExtensions and criticalExtensions information elements are defined to allow for further extensions in future releases.

```
MessageA ::= CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        nonCriticalExtensions           SEQUENCE {} OPTIONAL
    },
    later-than-r3                   SEQUENCE {
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
        criticalExtensions              CHOICE {
            r4                              SEQUENCE {
                messageA-r4                     MessageA-r4-IEs,
                nonCriticalExtensions           SEQUENCE {} OPTIONAL
            },
            criticalExtensions          SEQUENCE {}
        }
    }
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- This is not changed compared to the above example. It includes all information
    -- elements used in Release ’99 for messageA.
}

MessageA-r4-IEs ::=                 SEQUENCE {
    -- Here, the updated information elements used for MessageA in Release 4 are included.
}
```

Example 2

10.4.3.3 Non-critical Extensions

For non-critical extensions (i.e. the receiver shall just ignore the extensions, and use the rest of the message as if the extensions were not present), the approach is to use the nonCriticalExtensions information element, which is encoded at the end of the message, allowing backward compatibility.

The structure of the message of the example above is shown in Example 3 for the Release 4 message

Examples for special non-critical extensions and MessageA-~~r3-r4-~~v440ext-IEs are given in the following subclauses.

```
MessageA-r4 ::=                 CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            messageA-~~r3-r4-~~v440ext                  MessageA-~~r3-r4-~~v440ext-IEs,
            nonCriticalExtensions           SEQUENCE {} OPTIONAL
        } OPTIONAL
    },
    criticalExtensions              SEQUENCE {}
}

MessageA-r3-IEs ::=                 SEQUENCE {
    -- This is not changed compared to the same IE in Release '99. It includes all information
    -- elements used in Release '99 for messageA.
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    -- Here are additional information elements needed to describe the extensions compared to
    -- the information included in MessageA-r3-IEs.
}
```

Example 3

10.4.3.4 Examples of non-critical extensions

10.4.3.4.1 Addition of a separate IE

If the extension is the addition of an information element (not inside a CHOICE, SEQUENCE OF, SET OF etc.), this new element can be directly included in MessageA-~~r3-r4-~~v440ext-IEs.

Example4 shows how the MessageA is extended to include a new element, "element3".

```
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            Element1,
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    element3                            Element3-r4
}
```

Example 4

10.4.3.4.2 Addition of an IE to a structured group

If the extension is the addition of an information element inside a CHOICE, SEQUENCE OF, etc. (meaning that the information element can be absent or present more than once, depending on some condition), the structure of the original message should be duplicated in MessageA-~~r3-r4-~~v440ext-IEs using only the elements relevant to the extension (usually the CHOICEs, SEQUENCE OFs, etc.), and a comment should be included to indicate that the two structures should be used consistently (e.g. when a CHOICE is duplicated, the same branch should be followed in both places, when a SEQUENCE OF is duplicated, the number of occurrences should be the same etc.).

This is illustrated in Example5, where a new element, "element1a-3", has to be included inside the "choice1b" branch of the "choice1" CHOICE. Here "choice1" is included again in MessageA~~-r3-r4-~~v440ext-IEs, and "element1a-3" is included there in the appropriate branch.

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- For the "choice1b" branch of "choice1", an additional information element is
-- defined in MessageA-~~r3-r4-~~v440ext-IEs ("element1a-3").
    choice1                             CHOICE {
        choice1a                            SEQUENCE {
            element1a-1                         Element1a-1
        },
        choice1b                            SEQUENCE {
            element1a-2                         Element1a-2
        }
    }
}

MessageA-~~r3-r4-~~v440ext-IEs ::=            SEQUENCE {
-- In the following CHOICE the same branch shall be used as in choice1 in MessageA-r3-IEs.
    choice1                             CHOICE {
        choice1a                            NULL,
        choice1b                            SEQUENCE {
            element1a-3                         Element1a-3-r4
        }
    }
}
```

Example 5

10.4.3.4.3 Addition of a new CHOICE group

If the extension consists of moving some existing information elements inside a newly created CHOICE, the new branches of the created CHOICE should be included in MessageA~~-r3-r4-~~v440ext-IEs, and the CHOICE marked OPTIONAL, where absence means that the old elements are used. If the CHOICE is present, the old elements should be set to some default values, in order for older equipment to be understood, and new equipment should ignore the information therein.

This is illustrated in Example 6, where "element1" is to be moved inside the branch "choice1a" of a new CHOICE ("choice1").

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- The contents of "element1" shall be ignored, if in "MessageA-~~r3-r4-~~v440ext-IEs" the branch
-- "choice1b" of the CHOICE "choice1" is used.
    element1                            Element1,
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=            SEQUENCE {
    choice1                             CHOICE {
        choice1a                            SEQUENCE {},
        choice1b                            SEQUENCE {
            element3                            Element3-r4
        }
    }
}
```

Example 6

10.4.3.4.4 Extension of value range

If the value range of an element is to be extended, an element including the new values should be defined in MessageA-~~r3-r4-~~v440ext-IEs. If one of the new values is to be used, the already existing element from Release '99 should be set to some defined value (or be absent if it was OPTIONAL), in order for older equipment to work properly, and the new value should be signalled in the new information element.

In Example 7, "element1" is extended to have a range (0..15).

```
MessageA-r3-IEs ::=                 SEQUENCE {
-- "element1" shall be ignored if "element1" in MessageA-~~r3-r4-~~v440ext-IEs is present, and the
-- value of that element used instead.
    element1                            INTEGER (0..7)
    element2                            Element2
}

MessageA-~~r3-r4-~~v440ext-IEs ::=              SEQUENCE {
    element1                            INTEGER (0..15)         OPTIONAL
}
```

Example 7

10.4.3.4.5 Replacement of a spare value with a new element

If a new value is to be included in an IE of type ENUMERATED, for which spare values were defined in the previous version, those spare values can be replaced with the new values.

If more new values are needed, than spare values included in the previous version, one spare value can be replaced by a special extension value (called e-new in example 8). If that value is used, a new element in the nonCriticalExtension part (element1-new) will define the new values, as shown in Example 8 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            ENUMERATED { e1, e2, spare1, spare2 }
}

-- Now three new values are needed for element1: e3, e4 and e5. MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If the following has the value e-new, the actual value of element1 is defined in
-- element1-new included in MessageA-r4-ext-IEs
    element1                            ENUMERATED { e1, e2, e3, e-new }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e-new.
    element1-new                        ENUMERATED { e4, e5, spare1, spare2 }   OPTIONAL
}
```

Example 8

If a spare value is included in a CHOICE, and that has to be replaced with a new information element and an appropriate type in the new version, the name of the element replaces the spare name in the CHOICE, but the type cannot be replaced, because that would lead to incompatibilities. Instead, the new type is included in the nonCriticalExtension part of the message, as shown in Example 9 below:

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=                 SEQUENCE {
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        spare1                              NULL,
        spare2                              NULL
    }
}

-- Now a new option is needed for the element1 CHOICE: e3 with type E3.
-- MessageA-r3-IEs is redefined:
MessageA-r3-IEs ::=                 SEQUENCE {
-- If element1 has the value e3, the value of e3 is specified in the element e3
-- included in MessageA-r4-ext-IEs.
    element1                            CHOICE {
        e1                                  E1,
        e2                                  E2,
        e3                                  NULL,
        spare2                              NULL
    }
}

MessageA-r4-ext-IEs ::=             SEQUENCE {
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e3.
```

```
    e3                                  E3          OPTIONAL
}
```

Example 9

3GPP TSG-RAN WG2, Meeting #23
Helsinki, Finland, August 27th- 31st, 2001

Tdoc R2-012161

CR-Form-v4

CHANGE REQUEST

⌘	**25.921**	**CR**	**028**	⌘	ev	**r1**	⌘	Current version:	**3.4.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network ☒ Core Network □

Title: ⌘	Introduction of procedure specification guidelines specific to RLC			
Source: ⌘	TSG-RAN WG2			
Work item code: ⌘	TEI		***Date:*** ⌘	04-09-2001
Category: ⌘	**F**		***Release:*** ⌘	R99
	Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*	

Reason for change: ⌘	RLC specific procedure specification guidelines are missing
Summary of change: ⌘	RLC specific procedure specification guidelines are introduced The scope of 25.921 was modified to include user plane protocols. Isolated impact • This CR concerns all RLC procedures/ functions. However, the CR has isolated impact since the CR has no direct impact because it only introduces guidelines in a technical report. The CRs introducing the corresponding changes in the RLC specification only concerns clarification and hence it should have no impact on implementations that have assumed the proposed behaviour
Consequences if not approved: ⌘	The clarity, ambiguity and consistency of the RLC specification is not ensured

Clauses affected: ⌘	1, 7.3 (new)		
Other specs affected: ⌘	□ Other core specifications	⌘	25.921 v4.1.0, CR 029
	□ Test specifications		
	□ O&M Specifications		
Other comments: ⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

1 Scope

The present document provides a guideline for protocol specification of UMTS stage 2 and 3 including the usage of formal languages and rules for error handling. This document covers control-plane and user-plane protocols specified in TSG-RAN such as RRC, RLC, RANAP, RNSAP, NBAP and SABP

7 Protocol procedure specification rules

7.1 General

- A protocol specification shall contain a ’Procedures’ clause, which specifies the functional behaviour, using "procedures". A procedure is typically a sequence of events, with a start and an end, which can be observed in the protocol and/or in the interfaces to other layers (upper and/or lower layers).

- The procedure specification shall be made using text and verbal forms.

- The verbal forms, such as "shall", "should" and "may" are used in conformance with [11] Annex E.

- The procedures should be specified in an asymmetric way, by concentrating on the behaviour on one side of the interface. As guidance, the "controlled" side, rather than the "controlling" side of the interface should be specified.

- The procedures should be specified using the externally observable behaviour, to ease writing of test specifications.

- All normal cases shall be covered. Normal cases are straightforward cases, branches of procedures and combinations of procedures.

- All error cases shall be specified, either explicitly or implicitly. The error cases are all cases that are not considered as normal cases. The error handling should be divided between error handling global to the protocol layer and procedure specific error handling. The procedure specific error cases should be put after the normal cases in each procedure.

- Redundancy/duplication shall be avoided, in order to avoid problems with later CR, even if this makes the specification initially less readable.

- States and state variables should be used when it provides unambiguity, a way to describe nested procedures and colliding cases.

- Timers, variables and constants and usage of them must be specified.

- Explicit explanation when the action shall be performed is specified in the procedure itself.

- When there are procedural differences between the FDD and TDD modes these should be clearly pointed out using a consequent notation, e.g. "FDD only", "In TDD, …".

- When optional IEs are possible in a given message, the meaning of the presence (i.e.: which «function» are activated with the given IE) shall be specified in the procedure for the receiving entity. The requirements on when to include a given IE shall be specified for the transmitting entity. An exception for this rule is when the requirements on the entity is not specified by the protocol.

- Requirements on the content of a message at the sending entity are put before analysis of the message at the receiving entity.

- References to IEs that are parts of another IE is allowed.

- When referring to an IE a formal notation shall be used.

- When referring to a message, a formal notation shall be used.

7.1a Specification of algorithms and formulas

When algorithms or formulas are used in the specifications, a formal notation shall be used and mathematical expressions should be used to reduce ambiguity.

- The notation "*OP1* div *OP2*" shall be interpreted as the signed integer result after integer division (truncating any fractional part) of the operand *OP1* with operand *OP2*.

- The notation "*OP1* mod *OP2*" shall be interpreted as the signed remainder after integer division of the operand *OP1* with operand *OP2*.

7.2 RRC specific rules

- The specification shall focus on the UE behaviour.

- Only UE timers are normative (when UTRAN timers are present, it is for information).

- The procedure specification text shall specify how the UE shall handle the IEs.

- As much as possible of the UE behaviour shall be tied to reception and non-reception of IEs and included in the subclause "Generic actions upon receipt and absence of an information element", to avoid duplication of text.

- "UTRAN shall" shall be only used when UTRAN behaviour is normative.

- It shall be specified whether timers shall be started when RRC sends the message to lower layers or when the message is effectively sent at the radio interface.

- When referring to messages in the procedure text, the notation "EXAMPLE message" is used (excluding the quotation marks). For example: "The UE shall transmit an RRC CONNECTION REQUEST message".

- When referring to IEs in the procedure text, the tabular description of IEs should be used as basis. The notation "IE "Example"" is used (including the inner but not the outer quotation marks). Values of IEs are put within quotation marks. For example: "The UE set the IE "Protocol error indicator" to "FALSE" in the RRC CONNECTION REQUEST message".

- UE performance requirements are considered to be TSG RAN WG2 work. These must be specified only if they are testable.

7.3 RLC specific rules

- The behaviour of an RLC entity is specified by means of one or more elementary procedures. Each elementary procedure involves a Sender and a Receiver. In this respect the Sender is the RLC protocol entity that initiates the procedure, while the Receiver is the peer entity of the Sender. For each elementary procedure there are two possible configurations: one in which the UE is the Sender and UTRAN is the Receiver and another one in which UTRAN is the Sender and the UE is the Receiver.

- An RLC entity is normally configured to be either Sender or Receiver. In some cases however e.g. the AM RLC entity, the RLC entity consists of a transmitting and a receiving side. Such RLC entities can be Sender for some elementary procedures as well Receiver for other elementary procedure(s).

- The procedure specifications shall be specified independently as elementary procedures. Inter- relations between elementary procedures may be described in a general section describing the initiation of the procedure.

- The procedure specification text shall specify both how the Sender shall set the PDU parameters and how the Receiver shall handle them.

- It shall be specified when the timers are started, for example whether a timer shall be started when RLC submits a PDU to lower layers or when the successful or unsuccessful transmission of the PDU on the radio interface is indicated by lower layer.

- When referring to messages in the procedure text, the notation "EXAMPLE PDU" is used (excluding the quotation marks). For example: "The receiver shall transmit a RESET ACK PDU".

- When referring to PDU parameters in the procedure text, double quotation marks are used. The notation "PDU parameter "Example"" illustrates this usage (including the inner but not the outer quotation marks). Values of PDU parameters are also put within double quotation marks. For example: "If the PDU parameter "Length indicator" equals "0000000" the Receiver shall...".

- When referring to configuration parameters in the procedure text, double quotation marks are used. The notation "parameter "Example" is configured as" illustrates this usage (including the inner but not the outer quotation marks). Values of configuration parameters are also put within double quotation marks. For example: "If the Sender has the parameter "Delivery of erroneous PDUs" configured for "No Detect" …".

- When referring to state variables in the procedure text, the notation "VT(EXAMPLE)" and "VR(EXAMPLE)" is used (without the quotation marks) for state variables used in the RLC transmitter and RLC receiver respectively. For example: "If the value of VT(DAT) is equal to..."

- When referring to timers in the procedure text, the notation "Timer_Example" is used (without the quotation marks). For example: "If the timer Timer_Poll_Prohibit has expired the sender shall..."

- When referring to RLC protocol states in the procedure text, the notation "EXAMPLE_NAME state" is used (without the quotation marks). For example: "If the sender is in RESET_PENDING state, the sender shall..."

- Cross-references to other sections are referred to as "clauses", subsections are referred to as "subclauses". For example: "see subclause 9.3".

- Procedures which include conditional behaviour use the convention "if" … "else if" … "otherwise" to specify each condition.

3GPP TSG-RAN WG2, Meeting #23
Helsinki, Finland, August 27th- 31st, 2001

Tdoc R2-012162

CR-Form-v4

CHANGE REQUEST

⌘ **25.921 CR 029** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network ☒ Core Network □

Title:	⌘	Introduction of procedure specification guidelines specific to RLC		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	04-09-2001
Category:	⌘	**F** *Use one of the following categories:* ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.	***Release:*** ⌘	R4 *Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	RLC specific procedure specification guidelines are missing
Summary of change:	⌘	RLC specific procedure specification guidelines are introduced The scope of 25.921 was modified to include user plane protocols. Isolated impact • This CR concerns all RLC procedures/ functions. However, the CR has isolated impact since the CR has no direct impact because it only introduces guidelines in a technical report. The CRs introducing the corresponding changes in the RLC specification only concerns clarification and hence it should have no impact on implementations that have assumed the proposed behaviour
Consequences if not approved:	⌘	The clarity, ambiguity and consistency of the RLC specification is not ensured

Clauses affected:	⌘	1, 7.3 (new)		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	25.921 v3.4.0, CR 028
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

1 Scope

The present document provides a guideline for protocol specification of UMTS stage 2 and 3 including the usage of formal languages and rules for error handling. This document covers control-plane and user-plane protocols specified in TSG-RAN such as RRC, RLC, RANAP, RNSAP, NBAP and SABP

7 Protocol procedure specification rules

7.1 General

- A protocol specification shall contain a ’Procedures’ clause, which specifies the functional behaviour, using "procedures". A procedure is typically a sequence of events, with a start and an end, which can be observed in the protocol and/or in the interfaces to other layers (upper and/or lower layers).
- The procedure specification shall be made using text and verbal forms.
- The verbal forms, such as "shall", "should" and "may" are used in conformance with [11] Annex E.
- The procedures should be specified in an asymmetric way, by concentrating on the behaviour on one side of the interface. As guidance, the "controlled" side, rather than the "controlling" side of the interface should be specified.
- The procedures should be specified using the externally observable behaviour, to ease writing of test specifications.
- All normal cases shall be covered. Normal cases are straightforward cases, branches of procedures and combinations of procedures.
- All error cases shall be specified, either explicitly or implicitly. The error cases are all cases that are not considered as normal cases. The error handling should be divided between error handling global to the protocol layer and procedure specific error handling. The procedure specific error cases should be put after the normal cases in each procedure.
- Redundancy/duplication shall be avoided, in order to avoid problems with later CR, even if this makes the specification initially less readable.
- States and state variables should be used when it provides unambiguity, a way to describe nested procedures and colliding cases.
- Timers, variables and constants and usage of them must be specified.
- Explicit explanation when the action shall be performed is specified in the procedure itself.
- When there are procedural differences between the FDD and TDD modes these should be clearly pointed out using a consequent notation, e.g. "FDD only", "In TDD, …".
- When optional IEs are possible in a given message, the meaning of the presence (i.e.: which «function» are activated with the given IE) shall be specified in the procedure for the receiving entity. The requirements on when to include a given IE shall be specified for the transmitting entity. An exception for this rule is when the requirements on the entity is not specified by the protocol.
- Requirements on the content of a message at the sending entity are put before analysis of the message at the receiving entity.
- References to IEs that are parts of another IE is allowed.
- When referring to an IE a formal notation shall be used.
- When referring to a message, a formal notation shall be used.

7.1a Specification of algorithms and formulas

When algorithms or formulas are used in the specifications, a formal notation shall be used and mathematical expressions should be used to reduce ambiguity.

- The notation "*OP1* div *OP2*" shall be interpreted as the signed integer result after integer division (truncating any fractional part) of the operand *OP1* with operand *OP2*.

- The notation "*OP1* mod *OP2*" shall be interpreted as the signed remainder after integer division of the operand *OP1* with operand *OP2*.

7.2 RRC specific rules

- The specification shall focus on the UE behaviour.

- Only UE timers are normative (when UTRAN timers are present, it is for information).

- The procedure specification text shall specify how the UE shall handle the IEs.

- As much as possible of the UE behaviour shall be tied to reception and non-reception of IEs and included in the subclause "Generic actions upon receipt and absence of an information element", to avoid duplication of text.

- "UTRAN shall" shall be only used when UTRAN behaviour is normative.

- It shall be specified whether timers shall be started when RRC sends the message to lower layers or when the message is effectively sent at the radio interface.

- When referring to messages in the procedure text, the notation "EXAMPLE message" is used (excluding the quotation marks). For example: "The UE shall transmit an RRC CONNECTION REQUEST message".

- When referring to IEs in the procedure text, the tabular description of IEs should be used as basis. The notation "IE "Example"" is used (including the inner but not the outer quotation marks). Values of IEs are put within quotation marks. For example: "The UE set the IE "Protocol error indicator" to "FALSE" in the RRC CONNECTION REQUEST message".

- UE performance requirements are considered to be TSG RAN WG2 work. These must be specified only if they are testable.

7.3 RLC specific rules

- The behaviour of an RLC entity is specified by means of one or more elementary procedures. Each elementary procedure involves a Sender and a Receiver. In this respect the Sender is the RLC protocol entity that initiates the procedure, while the Receiver is the peer entity of the Sender. For each elementary procedure there are two possible configurations: one in which the UE is the Sender and UTRAN is the Receiver and another one in which UTRAN is the Sender and the UE is the Receiver.

- An RLC entity is normally configured to be either Sender or Receiver. In some cases however e.g. the AM RLC entity, the RLC entity consists of a transmitting and a receiving side. Such RLC entities can be Sender for some elementary procedures as well Receiver for other elementary procedure(s).

- The procedure specifications shall be specified independently as elementary procedures. Inter- relations between elementary procedures may be described in a general section describing the initiation of the procedure.

- The procedure specification text shall specify both how the Sender shall set the PDU parameters and how the Receiver shall handle them.

- It shall be specified when the timers are started, for example whether a timer shall be started when RLC submits a PDU to lower layers or when the successful or unsuccessful transmission of the PDU on the radio interface is indicated by lower layer.

- When referring to messages in the procedure text, the notation "EXAMPLE PDU" is used (excluding the quotation marks). For example: "The receiver shall transmit a RESET ACK PDU".

- When referring to PDU parameters in the procedure text, double quotation marks are used. The notation "PDU parameter "Example"" illustrates this usage (including the inner but not the outer quotation marks). Values of PDU parameters are also put within double quotation marks. For example: "If the PDU parameter "Length indicator" equals "0000000" the Receiver shall...".

- When referring to configuration parameters in the procedure text, double quotation marks are used. The notation "parameter "Example" is configured as" illustrates this usage (including the inner but not the outer quotation marks). Values of configuration parameters are also put within double quotation marks. For example: "If the Sender has the parameter "Delivery of erroneous PDUs" configured for "No Detect" …".

- When referring to state variables in the procedure text, the notation "VT(EXAMPLE)" and "VR(EXAMPLE)" is used (without the quotation marks) for state variables used in the RLC transmitter and RLC receiver respectively. For example: "If the value of VT(DAT) is equal to..."

- When referring to timers in the procedure text, the notation "Timer_Example" is used (without the quotation marks). For example: "If the timer Timer_Poll_Prohibit has expired the sender shall..."

- When referring to RLC protocol states in the procedure text, the notation "EXAMPLE_NAME state" is used (without the quotation marks). For example: "If the sender is in RESET_PENDING state, the sender shall..."

- Cross-references to other sections are referred to as "clauses", subsections are referred to as "subclauses". For example: "see subclause 9.3".

- Procedures which include conditional behaviour use the convention "if" … "else if" … "otherwise" to specify each condition.

3GPP TSG-RAN WG2 Meeting #23
Helsinki, Finland, 27-31 August, 2001

R2-012145

CR-Form-v3

CHANGE REQUEST

⌘ **25.921** **CR** **030** ⌘ rev **-** ⌘ Current version: **3.4.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network ☒ Core Network ☒

Title: ⌘	RAN3 usage of 1994 ASN.1 feature set		
Source: ⌘	TSG-RAN WG3		
Work item code: ⌘	TEI	***Date:*** ⌘	2001-08-21
Category: ⌘	**F**	***Release:*** ⌘	R99
	Use one of the following categories: ***F*** *(essential correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(Addition of feature),* ***C*** *(Functional modification of feature)* ***D*** *(Editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change: ⌘	All RAN3 protocol specifications has been changed to refer to the 1997 versions of X.680, X.681 and X.691 since the 1994 versions has been superseded. The RAN3 protocol specifications shall however as agreed at RAN3#8 only make use of the feature set defined in the 1994 versions and this shall be documented in 25.921.
Summary of change: ⌘	Add a statement in 25.921 that even though version 1997 of ASN.1 is referenced by the RAN3 protocol specifications, the protocol specifications will only make use of the feature set defined in the 1994 versions.
Consequences if not approved: ⌘	The agreement from RAN3#8 that RAN3 will only make use of the feature set defined in the 1994 versions of ASN.1 will not be documented anywhere. The proposed changes are backwards compatible.

Clauses affected: ⌘	10	
Other specs affected: ⌘	☒ Other core specifications ⌘ □ Test specifications □ O&M Specifications	25.921 v4.1.0, CR031
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

10 Usage of ASN.1

The following clauses contain guidelines for specification of protocol messages with ASN.1.The purpose of ASN.1 is to make it possible to specify message contents description of a message (i.e. what is the contents of a message) separately from its transfer syntax (i.e. how a message is encoded for transmission).

The clause 11 specifies how message transfer syntax is specified. It should be noted that importance of some transfer syntax properties must be determined early during specification because of their effect on message contents description specification possibilities. The properties are **compactness** and **extensibility.** If extreme compactness is required then extensibility must be restricted. If good extensibility is required then compromises must be done regarding compactness. The sections concerning these issues are marked in the following clauses as **COMPACTNESS** and **EXTENSIBILITY**.

Identifiers that could be keywords of some language (e.g.: SDL, C, ASN.1, JAVA, C++, …) should be avoided.

In the current version of the ASN.1 specifications, user-defined constraints are not used.

RANAP, SABP, RNSAP and NBAP specifications refers to the 1997 versions of X.680, X.681 and X.691, but the protocols shall only make use of the feature set available in the 1994 versions of X.680, X.681 and X.691.

10.1 Message level

10.2 Information element level

3GPP TSG-RAN WG2 Meeting #23
Helsinki, Finland, 27-31 August, 2001

R2-012146

CR-Form-v3

CHANGE REQUEST

⌘	**25.921**	**CR**	**031**	⌘	rev	**-**	⌘	Current version:	**4.1.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network ☒ Core Network ☒

Title: ⌘	RAN3 usage of 1994 ASN.1 feature set			
Source: ⌘	TSG-RAN WG3			
Work item code: ⌘	TEI		***Date:*** ⌘	2001-08-21
Category: ⌘	**A**		***Release:*** ⌘	REL-4
	Use one of the following categories: ***F*** *(essential correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(Addition of feature),* ***C*** *(Functional modification of feature)* ***D*** *(Editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*	

Reason for change: ⌘	All RAN3 protocol specifications has been changed to refer to the 1997 versions of X.680, X.681 and X.691 since the 1994 versions has been superseded. The RAN3 protocol specifications shall however as agreed at RAN3#8 only make use of the feature set defined in the 1994 versions and this shall be documented in 25.921.
Summary of change: ⌘	Add a statement in 25.921 that even though version 1997 of ASN.1 is referenced by the RAN3 protocol specifications, the protocol specifications will only make use of the feature set defined in the 1994 versions.
Consequences if not approved: ⌘	The agreement from RAN3#8 that RAN3 will only make use of the feature set defined in the 1994 versions of ASN.1 will not be documented anywhere. The proposed changes are backwards compatible.

Clauses affected: ⌘	10	
Other specs affected: ⌘	☒ Other core specifications ⌘ □ Test specifications □ O&M Specifications	25.921 v3.4.0, CR030
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

10 Usage of ASN.1

The following clauses contain guidelines for specification of protocol messages with ASN.1.The purpose of ASN.1 is to make it possible to specify message contents description of a message (i.e. what is the contents of a message) separately from its transfer syntax (i.e. how a message is encoded for transmission).

The clause 11 specifies how message transfer syntax is specified. It should be noted that importance of some transfer syntax properties must be determined early during specification because of their effect on message contents description specification possibilities. The properties are **compactness** and **extensibility.** If extreme compactness is required then extensibility must be restricted. If good extensibility is required then compromises must be done regarding compactness. The sections concerning these issues are marked in the following clauses as **COMPACTNESS** and **EXTENSIBILITY**.

Identifiers that could be keywords of some language (e.g.: SDL, C, ASN.1, JAVA, C++, …) should be avoided.

In the current version of the ASN.1 specifications, user-defined constraints are not used.

RANAP, SABP, RNSAP and NBAP specifications refers to the 1997 versions of X.680, X.681 and X.691, but the protocols shall only make use of the feature set available in the 1994 versions of X.680, X.681 and X.691.

10.1 Message level

10.2 Information element level