TSG-RAN Meeting #23 Phoenix, 10-12 March 2004

Title: CRs on 25.921 R'99 (and linked CRs from later releases)

Source: TSG-RAN WG2

Agenda item: 7.3.3

Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Version-New	Doc-2nd-Level	Workitem
25.921	049	-	R99	Spare Extension in Data Frame	F	3.9.0	3.10.0	R2-040187	TEI
25.921	050	-	Rel-4	Spare Extension in Data Frame	Α	4.6.0	4.7.0	R2-040188	TEI
25.921	051	-	Rel-5	Spare Extension in Data Frame	Α	5.3.0	5.4.0	R2-040189	TEI
25.921	52	1	R99	Guideline on release independent ASN.1 updates	F	3.9.0	3.10.0	R2-040319	TEI
25.921	53	2	Rel-4	Guideline on release independent ASN.1 updates	Α	4.6.0	4.7.0	R2-040344	TEI
25.921	54	2	Rel-5	Guideline on release independent ASN.1 updates	Α	5.3.0	5.4.0	R2-040345	TEI
25.921	55	-	R99	Guideline on the use of variable length containers for late extensions	F	3.9.0	3.10.0	R2-040247	TEI
25.921	56	-	Rel-4	Guideline on the use of variable length containers for late extensions	Α	4.6.0	4.7.0	R2-040248	TEI
25.921	57	-	Rel-5	Guideline on the use of variable length containers for late extensions	Α	5.3.0	5.4.0	R2-040249	TEI
25.921	58	-	R99	Guideline for the naming of extensions to the RRC ASN.1	F	3.9.0	3.10.0	R2-040300	TEI
25.921	59	-	Rel-4	Guideline for the naming of extensions to the RRC ASN.1	Α	4.6.0	4.7.0	R2-040301	TEI
25.921	60	-	Rel-5	Guideline for the naming of extensions to the RRC ASN.1	Α	5.3.0	5.4.0	R2-040302	TEI

3GPP TSG-RAN2 Meeting #40 Sophia Antipolis, France, 12th – 16th January 2004

Tdoc #R2-040187

Sophia Antipolis, France, 12" – 16" January 2004														
			C	HAN	GE	REC	QUE	EST	-					CR-Form-v7
*	25	.921	CR	049		жrev	-	¥	Cur	rent ve	ersio	n:	3.9.0	#
For <u>HELP</u> on u	sing i	this for	m, see	bottom o	of this	page c	r lool	k at th	пе рој	p-up te	xt o	ver t	he Ж sy	mbols.
Proposed change affects: UICC apps ME Radio Access Network Core Network Title: Spare Extension in Data Frame														
Title: Ж	Spa	are Ex	tension	in Data	Frame	е								
Source: #	RA	N WG	2											
Work item code: ₩	TE									Date:	¥	12/0	1/2004	
Category: 第	Deta	F (con A (con release B (add C (fun D (edd iled exp	rrection) rrespond e) dition of nctional i itorial m olanatior	wing cate, ds to a co feature), modification dification s of the a	orrection ion of t n) above	n in an e feature)				lease: se <u>one</u> 2 R96 R97 R98 R99 Rel-4 Rel-5	of th (C (F (F (F (F (F	GSM Relea Relea Relea Relea Relea	lowing re Phase 2 ase 1996 ase 1997 ase 1998 ase 1999 ase 4) ase 5)))))
Reason for change	e: #	It is no	ot clear	how to u	use th	e Spare	Exte	ensior	n in D	ata Fra	ame	of F	rame P	rotocol.
Summary of chang	уе: Ж	- E	xtensio	n mecha	anism	in Data	Fran	ne of	Fram	e Proto	ocol	is in	troduce	d.
Consequences if not approved:	¥	Impact release This (the check Extended)	et Analy et asses se): CR has hange o sion ha	ssment to	oward impac ot on S n use oward	s the post of on the Spare E d in this	e pre xtens relea	us ver vious sion fu ase ye us rele	rsion versi unctic et, the	of the son of the onality.	spec he s But ct is	cifica speci sinc very	ation (sa fication te the Sp minor.	me because
Clauses affected:	¥	9a(n	ew)											
Other specs	¥	Y N X	Other	core spesificates	tions		¥							

Other comments:

How to create CRs using this form:

 \mathfrak{H}

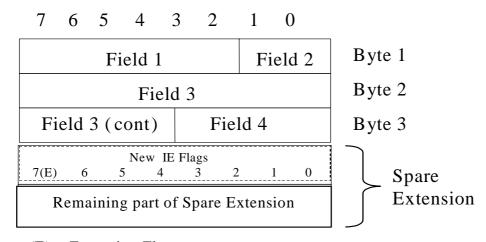
- 1) Fill out the above form. The symbols above marked \(\mathcal{H} \) contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "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.

9a Usage of lub/lur Frame Protocol

The following clauses contain guidelines for specification of frame protocols.

9a.1 Extensions for future releases in Data Frame

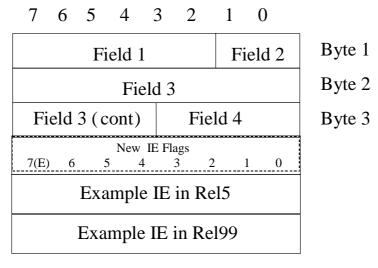
Spare Extension is used to define New IEs in the future. When the first IE is added in the Spare Extension in the Data Frame, *New IE Flags* IE shall be added in the first byte of the Spare Extension to indicate the validity of the value of the IEs in the Spare Extension. The last bit position of the *New IE Flags* IE is used as the Extension Flag to allow the extension of the *New IE Flags* IE in the future. The IEs in the Spare Extension will be added in the order in which the IEs are introduced regardless of the release.



(E) = Extension Flag

In the example below, it is assumed that after *Example IE in Rel5* IE was introduced, *Example IE in Rel99* IE is introduced.

In this example, New IE Flags (0) indicates the validity of *Example IE in Rel5* IE and the New IE Flags (1) indicates the validity of *Example IE in Rel99* IE. The IEs are added in the order of their introduction in the Spare Extension. For the Rel99 and Rel4 nodes, New IE Flags (0) and *Example IE in Rel5* IE will always be seen as Spare Bits while for Rel5 nodes, all the IEs (i.e., New IE Flags (0), New IE Flags(1)), *Example IE in Rel5* IE and *Example IE in Rel99* IE can be used.



Example of Spare Extension Usage

3GPP TSG-RAN2 Meeting #40 Sophia Antipolis, France, 12th – 16th January 2004

Tdoc #R2-040188

Sopnia Antipolis, France, 12" – 16" January 2004														
			(CHAN	GE	REC	UE	ST	•				C	R-Form-v7
*	25	.921	CR	050	8	⊭rev	-	æ	Curre	ent vers	sion:	4.6.	0	\mathfrak{H}
For <u>HELP</u> on u	sing	this fo	rm, see	bottom c	of this p	page or	look	at th	e pop-	up tex	t ove	r the ₩ :	sym	bols.
														
Proposed change	affec	ts:	UICC a	ıpps#]	ME	Ra	dio A	ccess	Netwo	ork <mark>X</mark>	Core	Net	work
Title: #	Spa	are Ex	tensior	n in Data F	Frame									
Source: #	RA	N WG	2											
			_									10.1.10.00		
Work item code: ₩	TE								E	Pate: #	12	/01/200	4	
Category: # A 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) Detailed explanations of the above categories can be found in 3GPP TR 21.900. Release: Rel-4 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) Rel-6 (Release 6)										ses:				
Reason for change	. H	It ic n	ot clos	r how to u	ico tho	Spara	Evto	ncion	in Do	to Eror	ma of	Eromo	Drot	ocol
Reason for change	;.	11 15 11	ot clea	i now to u	ise ille	Spare	EXIE	1151011	ШБа	la Fiai	ne oi	riaille	FIUL	ocoi.
Summary of chang	ge: ૠ	- E	extension	on mecha	nism i	n Data	Fram	e of F	rame	Protoc	col is	introduc	ed.	
Consequences if not approved:	**	In the future, adding new IEs in a backward compatible way is not guaranteed Impact Analysis: Impact assessment towards the previous version of the specification (same release): This CR has isolated impact on the previous version of the specification becathe change only affect on Spare Extension functionality. But since the Spare Extension hasn't been used in this release yet, the impact is very minor. Impact assessment towards the previous release of the specification: Not applicable since this is Rel99 correction.						e cause						
Clauses affected:	ąρ	02/2	ΟΜ/)											
Ciauses affected:	\mathfrak{H}	9a(n	ew)											
Other specs	ж	Y N X		r core spe		tions	¥							
aneoleu.		X		Specificat Specifica										

Other comments:

How to create CRs using this form:

 \mathfrak{H}

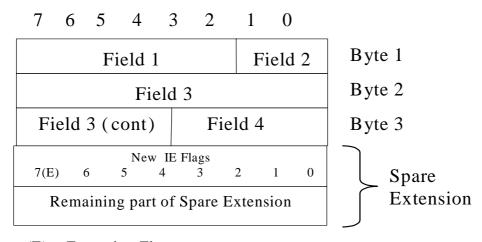
- 1) Fill out the above form. The symbols above marked \(\mathcal{H} \) contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "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.

9a Usage of lub/lur Frame Protocol

The following clauses contain guidelines for specification of frame protocols.

9a.1 Extensions for future releases in Data Frame

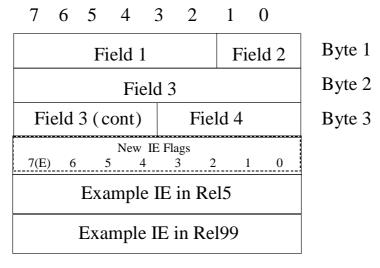
Spare Extension is used to define New IEs in the future. When the first IE is added in the Spare Extension in the Data Frame, *New IE Flags* IE shall be added in the first byte of the Spare Extension to indicate the validity of the value of the IEs in the Spare Extension. The last bit position of the *New IE Flags* IE is used as the Extension Flag to allow the extension of the *New IE Flags* IE in the future. The IEs in the Spare Extension will be added in the order in which the IEs are introduced regardless of the release.



(E) = Extension Flag

In the example below, it is assumed that after *Example IE in Rel5* IE was introduced, *Example IE in Rel99* IE is introduced.

In this example, New IE Flags (0) indicates the validity of *Example IE in Rel5* IE and the New IE Flags (1) indicates the validity of *Example IE in Rel99* IE. The IEs are added in the order of their introduction in the Spare Extension. For the Rel99 and Rel4 nodes, New IE Flags (0) and *Example IE in Rel5* IE will always be seen as Spare Bits while for Rel5 nodes, all the IEs (i.e., New IE Flags (0), New IE Flags(1)), *Example IE in Rel5* IE and *Example IE in Rel99* IE can be used.



Example of Spare Extension Usage

3GPP TSG-RAN2 Meeting #40 Sophia Antipolis, France, 12th – 16th January 2004

Tdoc #R2-040189

Sophia Antipoli	Sophia Antipolis, France, 12 th – 16 th January 2004													
			(CHAN	NGE	REC	UE	EST	-					CR-Form-v7
*	25	.921	CR	051		⊭rev	-	¥	Current	t vers	sion:	5.	3.0	¥
For <u>HELP</u> on u	ısing	this fo	rm, see	e bottom	of this	page o	r look	at th	е рор-ир	o text	over	the	₩ syr	mbols.
Proposed change	Proposed change affects: UICC apps# ME Radio Access Network X Core Network Title: Spare Extension in Data Frame													
Title: #	Sp	are Ex	tensio	n in Data	Frame	Э								
Source: # RAN WG2														
Work item code: ₩	TE	l							Da	te: ೫	12	/01/2	2004	
Category: #	Α								Releas	se: #	Re	l-5		
outogory.	<i>Use</i> Deta	F (co A (co release B (ac C (ful D (ec	rrection prespor e) Idition o nctional litorial ri planatio	owing cat) nds to a c of feature) I modifications of the TR 21.90	orrection in tion of fron above	n in an e eature)			Use <u>o</u> 2 R9 R9 R9 R9 Re	o <u>ne</u> of 96 97 98	the for (GSI (Rele (Rele (Rele	ollowi M Pha ease ease ease ease ease	ase 2) 1996) 1997) 1998) 1999) 4) 5)	
					<u> </u>						•			
Reason for change	e: #	It is n	ot clea	r how to	use the	e Spare	Exte	nsion	n in Data	Fram	ne of	Fran	me Pr	otocol.
Summary of chang	ge: ૠ	- E	xtensi	on mech	anism	in Data	Fram	ne of	Frame P	rotoc	ol is	intro	ducec	d.
Consequences if not approved: In the future, adding new IEs in a backward compatible way is not guarantee Impact Analysis: Impact assessment towards the previous version of the specification (same release): This CR has isolated impact on the previous version of the specification because the change only affect on Spare Extension functionality. But since the Spare Extension hasn't been used in this release yet, the impact is very minor. Impact assessment towards the previous release of the specification: Not applicable since this is Rel99 correction.									ne pecause					
Clauses offeets	مه	00/-	·O/W)											
Clauses affected:	\mathfrak{H}	9a(n	ew)											
Other specs	¥	Y N X		r core sp		ations	ж							
l		У		I Specific										

Other comments:

How to create CRs using this form:

 \mathfrak{H}

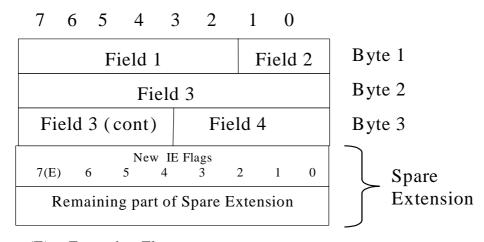
- 1) Fill out the above form. The symbols above marked \(\mathcal{H} \) contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "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.

9a Usage of lub/lur Frame Protocol

The following clauses contain guidelines for specification of frame protocols.

9a.1 Extensions for future releases in Data Frame

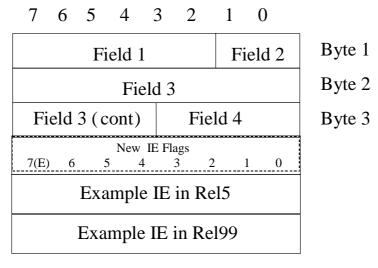
Spare Extension is used to define New IEs in the future. When the first IE is added in the Spare Extension in the Data Frame, *New IE Flags* IE shall be added in the first byte of the Spare Extension to indicate the validity of the value of the IEs in the Spare Extension. The last bit position of the *New IE Flags* IE is used as the Extension Flag to allow the extension of the *New IE Flags* IE in the future. The IEs in the Spare Extension will be added in the order in which the IEs are introduced regardless of the release.



(E) = Extension Flag

In the example below, it is assumed that after *Example IE in Rel5* IE was introduced, *Example IE in Rel99* IE is introduced.

In this example, New IE Flags (0) indicates the validity of *Example IE in Rel5* IE and the New IE Flags (1) indicates the validity of *Example IE in Rel9* IE. The IEs are added in the order of their introduction in the Spare Extension. For the Rel99 and Rel4 nodes, New IE Flags (0) and *Example IE in Rel5* IE will always be seen as Spare Bits while for Rel5 nodes, all the IEs (i.e., New IE Flags (0), New IE Flags(1)), *Example IE in Rel5* IE and *Example IE in Rel99* IE can be used.



3GPP TSG RAN WG2#40 Sophia Antipolis, France, 6th – 10th September 2003

			C	HANG	GE R	EQ	UES	ST				CR-Form-v7
*	25.	921	CR		<mark>52</mark>	ev	1	\mathbb{H}	Current ver	sion:	3.9.0	ж
For <u>HELP</u> on u	sing t	his for	rm, see b	ottom of	this pa	ge or i	look a	t the	pop-up tex	t over	the # syl	mbols.
Proposed change a	affect	ts: l	JICC ap _l	os# <mark> </mark>	N	ИЕ	Radi	o Ac	cess Netwo	ork	Core Ne	etwork
Title:	Gui	deline	on relea	se indep	endent	ASN.	1 upd	ates				
Source: #	RAI	N WG	2									
Work item code: 第	TEI								Date: 3	3 14/	/01/2004	
Category:	Use of	F (corr A (corr B (add C (fund D (edi led exp	the follow rection) responds dition of fe ctional mod torial mod blanations 3GPP TR	to a corre ature), odification lification) of the ab	ection in a	re)			2	f the for (GSN) (Rele (Rele (Rele (Rele (Rele	9 bllowing related Phase 2) pase 1996) pase 1997) pase 1998) pase 1999) pase 4) pase 5)	
Reason for change	e: X	curre lates wher may the (ently not it availab n extensi not be p	provided le versio ons for e ossible to ase in w	I. The ex on of the earlier no o use the chich the	kisting specion from e rele y wer	pract fication zen re ase in	tice is on. Ti eleas ndepe	nt features s to introduchis may res es are agre endent feau ed is frozen	ce suc ult in c ed. As iture u	ch featured decoding s a result until the A	s in the problems of this, it SN.1 of
Summary of chang	ye: ૠ	inder modi If this spec relea that a inform	pendent ifications s approa ifications ase for w are enco	features in the la ch is uns f groups hich the ding con In this ca	within the suitable suitable then the ASN.1 had ase, the	he AS ease for other he recenses be with t	N.1. Tor whither read on the comments of the c	The raich the ason ender a content of the content o	used when ecommend ne ASN.1 has s (eg. due t tion is to cro (by introduce nal release as are introd	ation in the state of a state of	is to introden frozen. nment witerovisions lummy parendent	luce h other in latest rameters
Consequences if not approved:	∺		y not be release						ndent featur s frozen	es un	til the ASN	N.1 of the
Clauses affected:	¥	10.4.	.3.6 (Nev	v)								
Other specs	¥	Y N X	Other o	ore spec	cification	ıs	ж					

affected:	X Test specifications O&M Specifications	
Other comments:	₩ Note that CR 55 introduces subclause 10.4.3.5	

How to create CRs using this form:

- 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.

10.4.3.6 Use of non critical extensions for release independent features

The objective of release independent features is that they do not require the UE to implement a specific release of the specification. Some release independent features may require the introduction of a non critical extension in one or more messages. In this case the recommendation is to introduce the extensions in the latest release of the specification for which the ASN.1 is frozen. This applies both to the tabular and the ASN.1.

In case, for some reasons eg. alignment with other specifications/ groups, the release independent feature is introduced in release X, which is later than the release recommended according to the above, the following recommendations apply:

- 1. Within the tabular notation, the non critical extension is introduced in release X.
- 2. Within the ASN.1, the information element is introduced as a non critical extension in release Y, which is the latest release of the specification for which the ASN.1 is frozen.
- 3. Within the ASN.1 of releases z, with $Y \le z < X$, the non critical extension is provisioned for by introducing a dummy information element that is encoding compatible with the actual information element introduced in release X

The reason for the provisioning of the extension in the ASN.1 of specifications earlier than release X is that for releases z, with $\underline{Y \le z < X}$, it is still possible to modify the ASN.1. As a result, if no provisions would be created in such releases, it would not be possible to use the release independent feature until release X is frozen.

An alternative approach would be to include the non critical extension within the variable length extension container (VLEC). However, considering the overhead associated with the introduction of such a container this approach is not recommended.

Note For release Y it is allowed to introduce non- critical extensions to release Y. Furthermore, backwards incompatible changes are allowed for later releases. If the release independent feature is introduced in release X, which is later than Y, it will be affected by such changes.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 12th – 16th January 2004

			CHANG	GE RE	QUE	ST				CR-Form-v7
*	25.9	<mark>921</mark> CR		<mark>53</mark> ⋇rev	/ 2	\mathbb{H}	Current vers	sion:	4.6.0	*
For <u>HELP</u> on us	sing th	nis form, se	e bottom of	this page	or look a	at the	pop-up text	over	the	nbols.
Proposed change a	affects	s: UICC	apps#	ME	Rad	io Ac	cess Netwo	rk	Core Ne	twork
Title: ♯	Guio	<mark>leline on re</mark>	lease indep	endent AS	N.1 upo	dates				
Source: #	RAN	I WG2								
Work item code: ₩	TEI						Date: ∺	14/	01/2004	
	F A B C D Detaile	(correction (correspor (addition of (functional)	ids to a corre f feature), modification nodification) ons of the ab	ection in an			Release:	the for (GSN (Relea (Relea (Relea (Relea (Relea (Relea	L-4 Dillowing rele M Phase 2) Pase 1996) Pase 1998) Pase 1999) Pase 4) Pase 5)	eases:
Reason for change	9: 米	currently r latest avai when exte may not be the (late) r	ot provided able versions asions for easions to the possible to the provided t	I. The exist on of the spearlier non o use the r hich they w	ing prace ecification frozen re elease i vere intr	tice is on. Ti eleas ndep	nt features v s to introduc his may resu es are agree endent feau ed is frozen.	e suc ult in d ed. As ture u	ch features decoding p s a result o until the AS	oroblems of this, it SN.1 of
Summary of chang	re: #	independe modification If this appropriate specification release for that are er information	nt features ons in the la oach is uns ons/ groups which the coding con	within the atest release suitable for the the ASN.1 has appared by the ace, the ace	ASN.1. e for whether recommed been from the decommed to the de	The raich the ason enda enda enda enda enda enda enda end	used when in recommendation ASN.1 has (eg. due to totion is to creation is to creation is to creation in the comment of the co	ation in section in a section i	s to introden frozen. nment with rovisions in ummy parendent	n other n latest ameters
Consequences if not approved:	*		be possible se in which				ndent feature s frozen	es unt	til the ASN	.1 of the
Clauses affected:	*	10.4.3.6 (1	lew)							
Other specs	¥.	Y N X Othe	r core spec	cifications	¥					

affected:	X Test specifications O&M Specifications
Other comments:	₩ Note that CR 56 introduces subclause 10.4.3.5

How to create CRs using this form:

- 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.

10.4.3.6 Use of non critical extensions for release independent features

The objective of release independent features is that they do not require the UE to implement a specific release of the specification. Some release independent features may require the introduction of a non critical extension in one or more messages. In this case the recommendation is to introduce the extensions in the latest release of the specification for which the ASN.1 is frozen. This applies both to the tabular and the ASN.1.

In case, for some reasons eg. alignment with other specifications/ groups, the release independent feature is introduced in release X, which is later than the release recommended according to the above, the following recommendations apply:

- 1. Within the tabular notation, the non critical extension is introduced in release X.
- 2. Within the ASN.1, the information element is introduced as a non critical extension in release Y, which is the latest release of the specification for which the ASN.1 is frozen.
- 3. Within the ASN.1 of releases z, with $Y \le z < X$, the non critical extension is provisioned for by introducing a dummy information element that is encoding compatible with the actual information element introduced in release X

The reason for the provisioning of the extension in the ASN.1 of specifications earlier than release X is that for releases z, with $Y \le z < X$, it is still possible to modify the ASN.1. As a result, if no provisions would be created in such releases, it would not be possible to use the release independent feature until release X is frozen.

An alternative approach would be to include the non critical extension within the variable length extension container (VLEC). However, considering the overhead associated with the introduction of such a container this approach is not recommended.

Note For release Y it is allowed to introduce non- critical extensions to release Y. Furthermore, backwards incompatible changes are allowed for later releases. If the release independent feature is introduced in release X, which is later than Y, it will be affected by such changes.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 12th – 16th January 2004

		СНА	NGE RE	QUES ⁻	Т		CR-Form-v7
X	25.92°	CR	<mark>54</mark> ⊭ re	2 ^೫	Current versi	5.3.0	*
For <u>HELP</u> on us	sing this f	orm, see botton	n of this page	or look at t	he pop-up text	over the	nbols.
Proposed change a	ffects:	UICC apps業[ME	Radio	Access Networ	k Core Ne	twork
Title:	Guidelin	e on release in	dependent AS	SN.1 update	es		
Source: #	RAN W	G2					
Work item code: ₩	TEI				Date: ₩	14/01/2004	
	Use <u>one</u> c F (cc A (cc B (ac C (fu D (ec Detailed e	of the following caperection) corresponds to a condition of feature, anctional modification and the conditions of the condition of the conditi	correction in an), ntion of feature) on) e above catego		Use <u>one</u> of a 2 se) R96 R97 R98 R99 Rel-4 Rel-5	REL-5 the following rele (GSM Phase 2) (Release 1996) (Release 1997) (Release 1998) (Release 1999) (Release 4) (Release 5) (Release 6)	eases:
Reason for change.	cur late who ma the	rently not proviest available velon extensions for your possible.	ded. The exist rsion of the spor earlier non le to use the ron which they was the ron which they was to be the the the was the the the was the the the was the	ing practice ecification. frozen rele elease inde vere introd	dent features we is to introduce. This may resu ases are agree ependent feautuced is frozen.	e such features It in decoding p d. As a result oure until the AS	oroblems of this, it SN.1 of
Summary of change	ind mo If th spe rele tha info	ependent featu difications in th his approach is ecifications/ gro ease for which t t are encoding	res within the e latest releas unsuitable for ups) then the he ASN.1 has compatible wis case, the ac	ASN.1. The for which other reas recommen been frozen the addited	be used when in the recommendary the ASN.1 has ons (eg. due to dation is to createn (by introducitional release introducitions are introducitions.	tion is to introd is been frozen. alignment with ate provisions in dummy paradependent	other n latest ameters
Consequences if not approved:		nay not be poss e) release in w			endent feature d is frozen	s until the ASN	.1 of the
Clauses affected:	策 10.	4.3.6 (New)					
Other specs	Y N		pecifications	¥			

affected:	X Test specifications O&M Specifications
Other comments:	₩ Note that CR 57 introduces subclause 10.4.3.5

How to create CRs using this form:

- 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.

10.4.3.6 Use of non critical extensions for release independent features

The objective of release independent features is that they do not require the UE to implement a specific release of the specification. Some release independent features may require the introduction of a non critical extension in one or more messages. In this case the recommendation is to introduce the extensions in the latest release of the specification for which the ASN.1 is frozen. This applies both to the tabular and the ASN.1.

In case, for some reasons eg. alignment with other specifications/ groups, the release independent feature is introduced in release X, which is later than the release recommended according to the above, the following recommendations apply:

- 1. Within the tabular notation, the non critical extension is introduced in release X.
- 2. Within the ASN.1, the information element is introduced as a non critical extension in release Y, which is the latest release of the specification for which the ASN.1 is frozen.
- 3. Within the ASN.1 of releases z, with $Y \le z < X$, the non critical extension is provisioned for by introducing a dummy information element that is encoding compatible with the actual information element introduced in release X

The reason for the provisioning of the extension in the ASN.1 of specifications earlier than release X is that for releases z, with $Y \le z < X$, it is still possible to modify the ASN.1. As a result, if no provisions would be created in such releases, it would not be possible to use the release independent feature until release X is frozen.

An alternative approach would be to include the non critical extension within the variable length extension container (VLEC). However, considering the overhead associated with the introduction of such a container this approach is not recommended.

Note For release Y it is allowed to introduce non- critical extensions to release Y. Furthermore, backwards incompatible changes are allowed for later releases. If the release independent feature is introduced in release X, which is later than Y, it will be affected by such changes.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 6th – 10th September 2003

CHANGE REQUEST											
*	25.921 CR	55	Current version: 3.9.0) #							
For <u>HELP</u> on us	sing this form, see bottom	n of this page or look at the	e pop-up text over the 光 s	ymbols.							
Proposed change affects: UICC apps # ME Radio Access Network Core Network Title: # Guideline on the use of variable length containers for late extensions											
Title: #	Guideline on the use of	variable length containers	s for late extensions								
Source: #	RAN WG2										
Work item code: ₩	TEI		<i>Date:</i>								
Category: 第	F Use one of the following car F (correction) A (corresponds to a combination of feature) C (functional modification of the determination of the determinati	orrection in an earlier release), tion of feature) on) e above categories can	Release: # R99 Use one of the following response to the following res	2) 6) 7) 8)							
Reason for change Summary of change	re: This CR introduces		rs are currently not provide used when performing lates in the RRC messages								
Consequences if not approved:	策 Extension containe additional signalling		optimal manner, resulting	in							
Clauses affected:	光 10.4.3.5										
Other specs affected:	Y N 米 X Other core sp X Test specifica X O&M Specifica	ations									
Other comments:											

How to create CRs using this form:

- 1) Fill out the above form. The symbols above marked \$\mathbb{X}\$ 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 $\underline{\text{ftp://ftp.3gpp.org/specs/}}$ 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.

10.4.3.3 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.

Before that Backward Compatibility is started for the following Release N+1, the non-critical extension information elements of the current Release N are added at the end of the message. At the point when Backward Compatibility is started for the following Release N+1, an optional BIT STRING container should be added before the information elements of the new release. In the case that further non-critical extension information elements need to be added to Release N they shall be placed within the BIT STRING container.

For example: As long as Backward Compatibility is not being enforced for Release 4, Release '99 extensions are added "normally" at the end of a message within a nonCriticalExtensions sequence. Once Backward Compatibility is started for Release 4, then new Release '99 specific extensions are introduced within an extension container. An extension container is a "normal" bit string field that encapsulates an extension structure. As a result:

- New extensions can be added both in Release '99 and Release 4 in a backward compatible way; and
- Release 4 systems are able to skip over unknown Release '99 extensions.

The extension container can be viewed as a specific type of non-critical extension and it is included in the same way. If the extension container is added to Release N before that Backward Compatibility has started for Release N+I, further non-critical extensions to Release N should not be included in the container, but should be placed after it, using the usual mechanism. In this way the extension container is not used until necessary, and therefore the corresponding length field overhead is not incurred unnecessarily. Additional guidelines concerning the use of extension containers are provided in 10.4.3.5.

The structure of the message of the example above is shown in Example 3 for Release '99 and 4 messages.

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

```
-- This shows the message structure in Release '99 (including one non-critical extension)
-- before backward compatibility is started for Release 4.
MessageA ::=
                            CHOICE {
                                    SEQUENCE {
    r3
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
            messageA-v380ext
                                                MessageA-v380ext-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-v380ext-IEs :: =
                                    SEQUENCE {
     - Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
-- This shows the Release '99 message structure once backward compatibility
-- has been started for Release 4.
                            CHOICE {
MessageA ::=
                                    SEQUENCE {
   r3
                                        MessageA-r3-IEs,
        messageA-r3
        v380 non Critical Extensions
                                             SEOUENCE {
            messageA-v380ext
                                                 MessageA-v380ext-IEs,
            laterNonCriticalExtensions
                                                 SEQUENCE {
                -- Container for additional Release '99 extensions
                messageA-r3-add-ext
                                                     BIT STRING
                    (CONTAINING MessageA-r3-add-ext-IEs)
                                                                     OPTIONAL,
                nonCriticalExtensions
                                                 SEQUENCE {} OPTIONAL
                OPTIONAL
```

```
OPTIONAL
                                   SEQUENCE {}
    criticalExtensions
}
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-v380ext-IEs :: =
                                  SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
MessageA-r3-add-ext-IEs :: =
                                  SEOUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
}
-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
                          CHOICE {
MessageA ::=
   r3
                                   SEQUENCE {
        messageA-r3
                                    MessageA-r3-IEs,
                                        SEQUENCE {
        v380nonCriticalExtensions
           messageA-v380ext
                                               MessageA-v380ext-IEs,
           laterNonCriticalExtensions
                                               SEQUENCE {
                -- Container for additional Release '99 extensions
                                                  BIT STRING
               messageA-r3-add-ext
                   (CONTAINING MessageA-r3-add-ext-IEs)
                                                                   OPTIONAL,
               v440nonCriticalExtensions
                                            SEQUENCE {
                                                     MessageA-v440ext-IEs,
                   messageA-v440ext
                   nonCriticalExtensions
                                                 SEQUENCE {} OPTIONAL
                   OPTIONAL
               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-v380ext-IEs :: =
                                   SEQUENCE {
   -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
MessageA-r3-add-ext-IEs :: =
                                   SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
MessageA-v440ext-IEs ::=
                                   SEOUENCE {
   -- Here are information elements added to Release 4 as extensions to the information
    -- contained in MessageA-r3-IEs and MessageA-v380ext-IEs.
```

Example 3

10.4.3.5 Additional guidelines on the use of variable length extension containers

"Variable length extension containers" (i.e. non critical extension containers that have their abstract syntax defined using the ASN.1 type "BIT STRING") have been defined to support the introduction of extensions to a release after the subsequent release is frozen (and UEs based on that subsequent release may appear).

Extension containers should be introduced in each message unless the size of the message is critical and the likelihood of late corrections is low. For downlink messages for which different versions have been defined, an extensions container should be introduced for each message version (branch).

In case a variable length extension container (VLEC) includes an extension, the PER encoder will include an additional length determinant. In case a separate container is introduced for each release, this would result in a significant signalling overhead. In order to avoid this signalling overhead, the extensions container should not be dedicated to late corrections of one specific release. If the extensions container is required to support the introduction of late corrections in an order of release, one or more release specific extensions container(s) may be nested within the original extension container.

The above guidelines are illustrated by means of an example. Suppose a message includes a single VLEC and one late R99 correction has been defined using this extension container, When the need for a late REL-5 correction arises, this correction may be added to the extensions container in two different ways:

- 1. Just by adding the extension after the late R99 correction
- 2. By introducing a REL-5 extensions container, that is nested within the existing extensions container

In case the first option is used, the addition of another late R99 correction (after the REL-5 correction) would require the R99 receiver to comprehend the transfer syntax of the REL-5 extension. This would require the introduction of the late REL-5 correction (or a type with an equivalent encoding) in the R99 transfer syntax. The second option avoids this problem, although this comes at the cost of additional signalling overhead.

Which option to use should be decided when introducing an extension for a release later than R99. This can be decided on a case by case basis eg. depending on whether for the concerned message further late R99 corrections need to be accommodated.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 6th – 10th September 2003

	CHANGE REQUEST											
*	25.9	21 CR	56 ⊯re\	- #	Current version:	4.6.0 **						
For <u>HELP</u> on us	sing th	is form, see bo	ttom of this page	or look at the	e pop-up text ove	er the % symbols.						
Proposed change a	affects	: UICC apps	sæ ME[Radio Ad	ccess Network	Core Network						
Title: ૠ	Guid	eline on the us	e of variable lengt	h containers	for late extension	ons						
Source: #	RAN	WG2										
Work item code: ₩	TEI				Date:	4/01/2004						
Category:	F A B C D	(addition of fea (functional mod (editorial modif	o a correction in an e ture), lification of feature) ication) of the above categor		2 (GS R96 (Re R97 (Re R98 (Re R99 (Re Rel-4 (Re Rel-5 (Re	EL-4 following releases: SM Phase 2) lease 1996) lease 1997) lease 1998) lease 1999) lease 4) lease 5)						
Reason for change	: X	Guidelines for	the use of extensi	on container	s are currently n	ot provided						
Summary of chang			uces a recommen oducing non critic									
Consequences if not approved:			ainers may be use alling overhead	ed in a sub-	optimal manner,	resulting in						
Clauses affected:	¥	10.4.3.3, 10.4.	3.5 (new)									
Other specs affected:	*	/ N Other co	re specifications cifications ecifications	¥								
Other comments:	\mathbb{H}											

How to create CRs using this form:

- 1) Fill out the above form. The symbols above marked \$\mathbb{X}\$ 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 $\underline{\text{ftp://ftp.3gpp.org/specs/}}$ 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.

10.4.3.3 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.

Before that Backward Compatibility is started for the following Release N+1, the non-critical extension information elements of the current Release N are added at the end of the message. At the point when Backward Compatibility is started for the following Release N+1, an optional BIT STRING container should be added before the information elements of the new release. In the case that further non-critical extension information elements need to be added to Release N they shall be placed within the BIT STRING container.

For example: As long as Backward Compatibility is not being enforced for Release 4, Release '99 extensions are added "normally" at the end of a message within a nonCriticalExtensions sequence. Once Backward Compatibility is started for Release 4, then new Release '99 specific extensions are introduced within an extension container. An extension container is a "normal" bit string field that encapsulates an extension structure. As a result:

- New extensions can be added both in Release '99 and Release 4 in a backward compatible way; and
- Release 4 systems are able to skip over unknown Release '99 extensions.

The extension container can be viewed as a specific type of non-critical extension and it is included in the same way. If the extension container is added to Release N before that Backward Compatibility has started for release N+1, further non-critical extensions to Release N should not be included in the container, but should be placed after it using the usual mechanism. In this way the extension container is not used until necessary, and therefore the corresponding length field overhead is not incurred unnecessarily. Additional guidelines concerning the use of extension containers are provided in 10.4.3.5.

The structure of the message of the example above is shown in Example 3 for Release '99 and 4 messages.

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

```
-- This shows the message structure in Release '99 (including one non-critical extension)
-- before backward compatibility is started for Release 4.
MessageA ::=
                            CHOICE {
                                     SEQUENCE {
    r3
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
            messageA-v380ext
                                                 MessageA-v380ext-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-v380ext-IEs :: =
                                     SEQUENCE {
     - Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
-- This shows the Release '99 message structure once backward compatibility
-- has been started for Release 4.
                            CHOICE {
MessageA ::=
                                     SEQUENCE {
   r3
                                        MessageA-r3-IEs,
        messageA-r3
        v380 non Critical Extensions
                                             SEOUENCE {
            messageA-v380ext
                                                 MessageA-v380ext-IEs,
            laterNonCriticalExtensions
                                                 SEQUENCE {
                 -- Container for additional Release '99 extensions
                messageA-r3-add-ext
                                                     BIT STRING
                    (CONTAINING MessageA-r3-add-ext-IEs)
                                                                      OPTIONAL,
                {\tt nonCriticalExtensions}
                                                 SEQUENCE {} OPTIONAL
                OPTIONAL
```

```
OPTIONAL
                                   SEQUENCE {}
    criticalExtensions
}
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-v380ext-IEs :: =
                                  SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
MessageA-r3-add-ext-IEs :: =
                                  SEOUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
}
-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
                          CHOICE {
MessageA ::=
   r3
                                   SEQUENCE {
        messageA-r3
                                    MessageA-r3-IEs,
                                        SEQUENCE {
        v380nonCriticalExtensions
           messageA-v380ext
                                               MessageA-v380ext-IEs,
           laterNonCriticalExtensions
                                               SEQUENCE {
                -- Container for additional Release '99 extensions
                                                  BIT STRING
               messageA-r3-add-ext
                   (CONTAINING MessageA-r3-add-ext-IEs)
                                                                   OPTIONAL,
               v440nonCriticalExtensions
                                            SEQUENCE {
                                                     MessageA-v440ext-IEs,
                   messageA-v440ext
                   nonCriticalExtensions
                                                 SEQUENCE {} OPTIONAL
                   OPTIONAL
               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-v380ext-IEs :: =
                                   SEQUENCE {
   -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
MessageA-r3-add-ext-IEs :: =
                                   SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
MessageA-v440ext-IEs ::=
                                   SEOUENCE {
   -- Here are information elements added to Release 4 as extensions to the information
    -- contained in MessageA-r3-IEs and MessageA-v380ext-IEs.
```

Example 3

10.4.3.5 Additional guidelines on the use of variable length extension containers

"Variable length extension containers" (i.e. non critical extension containers that have their abstract syntax defined using the ASN.1 type "BIT STRING") have been defined to support the introduction of extensions to a release after the subsequent release is frozen (and UEs based on that subsequent release may appear).

Extension containers should be introduced in each message unless the size of the message is critical and the likelihood of late corrections is low. For downlink messages for which different versions have been defined, an extensions container should be introduced for each message version (branch).

In case a variable length extension container (VLEC) includes an extension, the PER encoder will include an additional length determinant. In case a separate container is introduced for each release, this would result in a significant signalling overhead. In order to avoid this signalling overhead, the extensions container should not be dedicated to late corrections of one specific release. If the extensions container is required to support the introduction of late corrections in an order of release, one or more release specific extensions container(s) may be nested within the original extension container.

The above guidelines are illustrated by means of an example. Suppose a message includes a single VLEC and one late R99 correction has been defined using this extension container, When the need for a late REL-5 correction arises, this correction may be added to the extensions container in two different ways:

- 1. Just by adding the extension after the late R99 correction
- 2. By introducing a REL-5 extensions container, that is nested within the existing extensions container

In case the first option is used, the addition of another late R99 correction (after the REL-5 correction) would require the R99 receiver to comprehend the transfer syntax of the REL-5 extension. This would require the introduction of the late REL-5 correction (or a type with an equivalent encoding) in the R99 transfer syntax. The second option avoids this problem, although this comes at the cost of additional signalling overhead.

Which option to use should be decided when introducing an extension for a release later than R99. This can be decided on a case by case basis eg. depending on whether for the concerned message further late R99 corrections need to be accommodated.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 6th – 10th September 2003

CHANGE REQUEST											CR-Form-v/			
*		25.	921	CR		57	≋ rev	-	¥	Current v	ersior/	5.3	3.0	¥
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the % symbols.														
Proposed change affects: UICC apps# ME Radio Access Network Core Network														
Title: # Guideline on the use of variable length containers for late extensions														
Source: # RAN WG2														
Work iten	n code: ₩	TEI								Date	<i>:</i>	4/01/2	004	
Category	: #	E L Detail	(corr (corr (add (fund (edit ed exp	rection) responds lition of fe ctional m forial mod blanation	ving cates to a correction eature), odification s of the a	rection on of fe) above	n in an ea		eleas	Release Use ond 2 e) R96 R97 R98 R99 Rel-4 Rel-4	e of the (G (R) (R) (R) (R) (R) 4 (R)	REL-5 followir SM Pha elease elease elease elease elease elease	ase 2) 1996) 1997) 1998) 1999) 4)	eases:
Reason fo	or change	e: #	Guid	elines fo	or the us	e of e	extensio	n con	taine	rs are cur	rently i	not pro	vided	
Summary of change: ₩			This CR introduces a recommendation to be used when performing late corrections introducing non critical extensions in the RRC messages											
Conseque not appro		Ж			ntainers gnalling			d in a	sub-	optimal m	anner	, result	ing in	
Clauses a	iffected:	¥	10.4.	3.3, 10.	4.3.5 (ne	ew)								
Other spe affected:		*	Y N X X	Test sp	core spe pecificati Specifica	ions	tions	X						
Other con	nments:	\mathbf{x}												

How to create CRs using this form:

- 1) Fill out the above form. The symbols above marked \$\mathbb{X}\$ 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 $\underline{\text{ftp://ftp.3gpp.org/specs/}}$ 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.

10.4.3.3 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.

Before that Backward Compatibility is started for the following Release N+1, the non-critical extension information elements of the current Release N are added at the end of the message. At the point when Backward Compatibility is started for the following Release N+1, an optional BIT STRING container should be added before the information elements of the new release. In the case that further non-critical extension information elements need to be added to Release N they shall be placed within the BIT STRING container.

For example: As long as Backward Compatibility is not being enforced for Release 4, Release '99 extensions are added "normally" at the end of a message within a nonCriticalExtensions sequence. Once Backward Compatibility is started for Release 4, then new Release '99 specific extensions are introduced within an extension container. An extension container is a "normal" bit string field that encapsulates an extension structure. As a result:

- New extensions can be added both in Release '99 and Release 4 in a backward compatible way; and
- Release 4 systems are able to skip over unknown Release '99 extensions.

The extension container can be viewed as a specific type of non-critical extension and it is included in the same way. If the extension container is added to Release N before that Backward Compatibility has started for Release N+I, further non-critical extensions to Release N should not be included in the container, but should be placed after it using the usual mechanism. In this way the extension container is not used until necessary, and therefore the corresponding length field overhead is not incurred unnecessarily. Additional guidelines concerning the use of extension containers are provided in 10.4.3.5.

The structure of the message of the example above is shown in Example 3 for Release '99 and 4 messages.

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

```
-- This shows the message structure in Release '99 (including one non-critical extension)
-- before backwards compatibility is started for Release 4.
                            CHOICE {
MessageA ::=
                                     SEQUENCE {
    r3
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
            messageA-v380ext
                                                 MessageA-v380ext-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-v380ext-IEs :: =
                                     SEQUENCE {
     - Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
-- This shows the Release '99 message structure once backward campatibility
-- has been started for Release 4.
                            CHOICE {
MessageA ::=
                                     SEQUENCE {
   r3
                                        MessageA-r3-IEs,
        messageA-r3
        v380 non Critical Extensions
                                             SEOUENCE {
            messageA-v380ext
                                                 MessageA-v380ext-IEs,
            laterNonCriticalExtensions
                                                 SEQUENCE {
                 -- Container for additional Release '99 extensions
                messageA-r3-add-ext
                                                     BIT STRING
                    (CONTAINING MessageA-r3-add-ext-IEs)
                                                                      OPTIONAL,
                {\tt nonCriticalExtensions}
                                                 SEQUENCE {} OPTIONAL
                OPTIONAL
```

```
OPTIONAL
                                   SEQUENCE {}
    criticalExtensions
}
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-v380ext-IEs :: =
                                  SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
MessageA-r3-add-ext-IEs :: =
                                  SEOUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backwards compatibility was started for Release 4.
}
-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
                          CHOICE {
MessageA ::=
   r3
                                   SEQUENCE {
        messageA-r3
                                    MessageA-r3-IEs,
                                        SEQUENCE {
        v380nonCriticalExtensions
           messageA-v380ext
                                               MessageA-v380ext-IEs,
           laterNonCriticalExtensions
                                               SEQUENCE {
                -- Container for additional Release '99 extensions
                                                  BIT STRING
               messageA-r3-add-ext
                   (CONTAINING MessageA-r3-add-ext-IEs)
                                                                  OPTIONAL,
               v440nonCriticalExtensions
                                           SEQUENCE {
                                                     MessageA-v440ext-IEs,
                   messageA-v440ext
                   nonCriticalExtensions
                                                 SEQUENCE {} OPTIONAL
                   OPTIONAL
               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-v380ext-IEs :: =
                                   SEQUENCE {
   -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
MessageA-r3-add-ext-IEs :: =
                                   SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backwards compatibility was started for Release 4.
MessageA-v440ext-IEs ::=
                                   SEOUENCE {
   -- Here are information elements added to Release 4 as extensions to the information
    -- contained in MessageA-r3-IEs and MessageA-v380ext-IEs.
```

Example 3

10.4.3.5 Additional guidelines on the use of variable length extension containers

"Variable length extension containers" (i.e. non critical extension containers that have their abstract syntax defined using the ASN.1 type "BIT STRING") have been defined to support the introduction of extensions to a release after the subsequent release is frozen (and UEs based on that subsequent release may appear).

Extension containers should be introduced in each message unless the size of the message is critical and the likelihood of late corrections is low. For downlink messages for which different versions have been defined, an extensions container should be introduced for each message version (branch).

In case a variable length extension container (VLEC) includes an extension, the PER encoder will include an additional length determinant. In case a separate container is introduced for each release, this would result in a significant signalling overhead. In order to avoid this signalling overhead, the extensions container should not be dedicated to late corrections of one specific release. If the extensions container is required to support the introduction of late corrections in an order of release, one or more release specific extensions container(s) may be nested within the original extension container.

The above guidelines are illustrated by means of an example. Suppose a message includes a single VLEC and one late R99 correction has been defined using this extension container, When the need for a late REL-5 correction arises, this correction may be added to the extensions container in two different ways:

- 1. Just by adding the extension after the late R99 correction
- 2. By introducing a REL-5 extensions container, that is nested within the existing extensions container

In case the first option is used, the addition of another late R99 correction (after the REL-5 correction) would require the R99 receiver to comprehend the transfer syntax of the REL-5 extension. This would require the introduction of the late REL-5 correction (or a type with an equivalent encoding) in the R99 transfer syntax. The second option avoids this problem, although this comes at the cost of additional signalling overhead.

Which option to use should be decided when introducing an extension for a release later than R99. This can be decided on a case by case basis eg. depending on whether for the concerned message further late R99 corrections need to be accommodated.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 6th – 10th September 2003

CHANGE REQUEST												
*	25.921 CR	58 ⊭rev	- # (Current version	on: 3.9.0	¥						
For <u>HELP</u> on us	ng this form, see bot	tom of this page or	look at the	pop-up text o	over the	nbols.						
Proposed change affects: UICC apps% ME Radio Access Network Core Network												
Title:	Guideline for the nar	ning of extensions	to the RRC	ASN.1								
Source: #	RAN WG2											
Work item code: ₩	TEI			Date: ∺	15/01/2004							
Category: 第	lse <u>one</u> of the following F (correction)	a correction in an ear ure), fication of feature) cation) the above categories	rlier release)	2 (R96 (R97 (R98 (R99 (Rel-4 (Rel-5 (R99 he following rele (GSM Phase 2) (Release 1996) (Release 1997) (Release 1998) (Release 1999) (Release 4) (Release 5) (Release 6)	ases:						
Reason for change	manner. This is	EL-4 and REL-5 ex because the guide ently neither clear n	lines for the	e naming of e								
Summary of chang		duces a recommend oducing non critical				Э						
Consequences if not approved:	₩ There will be no ASN.1	way to achieve co	nsistent na	ming of exter	nsions to the R	RC						
Clauses affected:	第 10.4.2											
Other specs affected:	X Test spec	e specifications ifications cifications	*									
Other comments:	need not apply to track in which additional care it has already be	ntion means that so a specific suffix. The release/ version the when applying the been common praction suffix, this is not constant	is means the IE was in backwards ce to not al	nat it will be s ntroduced. Th compatibility ways use an	somewhat more his requires so rules. Howeve extension sho	e difficult me er, since						

How to create CRs using this form:

- 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 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. For this naming, "r3" is used for Release '99, "r4" for Release 4, "r5" for Release 5 and so on.

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-msq-r3
                                       Test-msg-r3-IEs,
       nonCriticalExtensions
                                       SEQUENCE {} OPTIONAL
                                  SEQUENCE {
    later-than-r3
       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-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                           SEOUENCE {
                                               Test-msg-v380ext-IEs,
           test-msg-v380ext
           nonCriticalExtensions
                                               SEQUENCE {} OPTIONAL
       } OPTIONAL
    later-than-r3
                                   SEOUENCE {
                                  RRC-TransactionIdentifier,
       rrc-TransactionIdentifier
                                       SEQUENCE {}
       criticalExtensions
    }
-- The Test-msg gets the following structure, if only a non-critical
-- extensions is introduced for Release 4 in v4.4.0 of the specification.
Test-msg ::= CHOICE {
                                   SEQUENCE {
   r3
        test-msg-r3
                                     Test-msg-r3-IEs,
       v380nonCriticalExtensions
                                           SEOUENCE {
                                               Test-msg-v380ext-IEs,
           test-msg-v380ext
           laterNonCriticalExtensions
                                               SEQUENCE
```

```
-- Container for additional Release '99 extensions
                                                    BIT STRING
                test-msg-r3-add-ext
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL.
                v440nonCriticalExtensions
                                                  SEQUENCE {
                    test-msg-v440ext
                                                        Test-msg-v440ext-IEs,
                    nonCriticalExtensions
                                                        SEQUENCE {} OPTIONAL
                    OPTIONAL
                OPTIONAL
        } OPTIONAL
    later-than-r3
                                    SEQUENCE {
        rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
        criticalExtensions
                                        SEQUENCE {}
}
-- In Release 5, the Test msg gets the following structure when a critical
-- extension is added
Test-msg ::= CHOICE {
                                    SEQUENCE {
   r3
                                       Test-msg-r3-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                         SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
            laterNonCriticalExtensions
                                               SEQUENCE {
                -- Container for additional Release '99 extensions
                test-msg-r3-add-ext
                                                   BIT STRING
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL.
                v440nonCriticalExtensions SEQUENCE {
                    test-msg-v440ext
                                                        Test-msg-v440ext-IEs,
                                                       SEQUENCE {} OPTIONAL
                    nonCriticalExtensions
                   OPTIONAL
                OPTIONAL
        } OPTIONAL
                                    SEQUENCE {
    later-than-r3
        rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
                                        CHOICE {
        criticalExtensions
                                           SEOUENCE {
           r5
                                                Test-msg-r5-IEs,
                test-msg-r5
                nonCriticalExtensions
                                                SEQUENCE {} OPTIONAL
            },
                                           SEQUENCE { }
            criticalExtensions
    }
}
```

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

If an abstract type is introduced in Release N when new elements are included in an extension, it should have a suffix "-rN". For Release '99 types, no such suffix is used. In case the type that is introduced in Release N includes one or more new (nested) types, the additional suffix need not be used for these nested types. In case the type that is introduced in Release N includes one or more revisions of exixting types, the suffix is needed to distinguish them from the earlier revisions. In case a revision of an abstract type that is introduced in Release N includes an IE for which the abstract type already existed in earlier releases, while that IE was not present in the previous revision(s) of the revised abstract type, the IE name should have a suffix "-rN".

If an abstract type is introduced in a release to extend an already existing type "TypeX", it should get the same name with a non-critical extension type suffix ("-vXYZext", e.g. "TypeX-v380ext") although in this case the final "- IEs" suffix is not added. In case the type that is introduced in Release N to extend an already existing type includes one or more new (nested) types that are extensions of an already existing type, the additional suffix should not be used for these nested types. In case the type that is introduced in Release N to extend an already existing type includes one or more new (nested) types, the abovely specified rules for new abstract types apply.

The above naming conventions are further illustrated in the example below:

```
Test-msg-v440ext-IEs::= SEQUENCE {
                                         NewIE-C
        newIE-C-r4
                                                                      OPTIONAL,
        existingIE-D-v440ext
                                         ExistingIE-D-v440ext
Test-msg-r5-IEs::= SEQUENCE {
        existingIE-E
                                         ExistingIE-E
                                                                      OPTIONAL,
        newUseOfexistingIE-E-rF
                                         ExistingIE-F
                                         NewIE-G
        newIE-G-r5
        revisionOfExistingIE-H-r5
                                         ExistingIE-H-r5
```

The abovely described naming convention means that some IEs introduced in a later release/version need not apply a specific suffix. This means that it will not allways be clear from the name of an IE whether or not backwards incompatible changes to it are allowed. Using the above naming rules, when changes are done in Release N, only changes in types with a suffix "-rN" or "-vXYZext" are allowed, in order to avoid conflicts with previous releases.

An exception is Tthe Message type itself is a special case, 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
                                     SEOUENCE {
                                         Test-msg-r3-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                            SEOUENCE {
                                             Test-msg-v380ext-IEs,
            test-msg-v380ext
            nonCriticalExtensions
                                             SEQUENCE {} OPTIONAL
        } OPTIONAL
    later-than-r3
                                     SEQUENCE {
        rrc-TransactionIdentifier
                                         RRC-TransactionIdentifier,
        criticalExtensions
                                         SEQUENCE {
            importantElements
                                             ImportantElements,
                                             CHOICE {
            rest-of-message
                r4
                                                 SEQUENCE {
                                                     Test-msg-r4-IEs,
                    test-msg-r4
                    nonCriticalExtensions
                                                     SEQUENCE {} OPTIONAL
                criticalExtensions
                                                 SEOUENCE { }
            }
        }
    }
}
```

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 1: The structure presented in this clause and the proposed naming rules are one possibility. Further possibilities are FFS.
- NOTE 2: When non-critical extensions are introduced in a message that does not have yet a critical Extension branch, they are introduced in the "Test-msg-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-rN-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-rN-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 critical Extension CHOICE branch, or the nonCritical Extension information element.

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

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

Before that Backward Compatibility is started for the following Release N+1, the non-critical extension information elements of the current Release N are added at the end of the message. At the point when Backward Compatibility is started for the following Release N+1, an optional BIT STRING container should be added before the information elements of the new release. In the case that further non-critical extension information elements need to be added to Release N they shall be placed within the BIT STRING container.

For example: As long as Backward Compatibility is not being enforced for Release 4, Release '99 extensions are added "normally" at the end of a message within a nonCriticalExtensions sequence. Once Backward Compatibility is started for Release 4, then new Release '99 specific extensions are introduced within an extension container. An extension container is a "normal" bit string field that encapsulates an extension structure. As a result:

- New extensions can be added both in Release '99 and Release 4 in a backward compatible way; and
- Release 4 systems are able to skip over unknown Release '99 extensions.

The extension container can be viewed as a specific type of non-critical extension and it is included in the same way. If the extension container is added to Release N before that Backward Compatibility has started for Release N+1, further non-critical extensions to Release N should not be included in the container, but should be placed after it, using the usual mechanism. In this way the extension container is not used until necessary, and therefore the corresponding length field overhead is not incurred unnecessarily.

The structure of the message of the example above is shown in Example 3 for Release '99 and 4 messages.

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

```
-- This shows the message structure in Release '99 (including one non-critical extension)
-- before backward compatibility is started for Release 4.
                            CHOICE {
MessageA ::=
    r3
                                    SEQUENCE {
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
                                                MessageA-v380ext-IEs.
            messageA-v380ext
            nonCriticalExtensions
                                            SEQUENCE {} OPTIONAL
            OPTIONAL
    },
    criticalExtensions
                                    SEOUENCE {}
}
MessageA-r3-IEs ::=
                                    SEOUENCE {
    -- This is not changed compared to the same IE in Release '99. It includes all information
    -- elements used in Release '99 for MessageA.
MessageA-v380ext-IEs :: =
                                    SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
-- This shows the Release '99 message structure once backward compatibility
-- has been started for Release 4.
MessageA ::=
                           CHOICE {
                                    SEQUENCE {
    r3
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
                                            SEOUENCE {
            messageA-v380ext
                                                MessageA-v380ext-IEs,
            laterNonCriticalExtensions
                                                SEQUENCE {
                -- Container for additional Release '99 extensions
                messageA-r3-add-ext
                                                     BIT STRING
```

```
(CONTAINING MessageA-r3-add-ext-IEs)
                                                                      OPTIONAL,
                nonCriticalExtensions
                                                 SEQUENCE {} OPTIONAL
                OPTIONAL
            OPTIONAL
                                     SEQUENCE {}
    criticalExtensions
}
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-v380ext-IEs :: =
                                    SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
MessageA-r3-add-ext-IEs :: =
                                   SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
                            CHOICE {
MessageA ::=
                                     SEQUENCE {
   r3
        messageA-r3
                                        MessageA-r3-IEs,
        v380nonCriticalExtensions
            | MessageA-v380ext-IEs, | laterNonCriticalExtensions | SECTIFMOR |
                                           SEQUENCE {
                -- Container for additional Release '99 extensions
                messageA-r3-add-ext
                                                    BIT STRING
                    (CONTAINING MessageA-r3-add-ext-IEs)
                {\tt v440nonCriticalExtensions} \hspace{1.5cm} {\tt SEQUENCE} \hspace{0.1cm} \{
                                                         MessageA-v440ext-IEs,
                    messageA-v440ext
                    nonCriticalExtensions
                                                    SEQUENCE {} OPTIONAL
                    OPTIONAL
                OPTIONAL
            OPTIONAL
        }
    criticalExtensions
                                     SEQUENCE {}
}
MessageA-r3-IEs ::=
                                    SEOUENCE {
    -- This is not changed compared to the same IE in Release '99. It includes all information
    -- elements used in Release '99 for MessageA.
}
MessageA-v380ext-IEs :: =
                                    SEOUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs.
}
MessageA-r3-add-ext-IEs :: =
                                    SEQUENCE {
    -- Here are information elements added to Release '99 as extensions to the information
    -- contained in MessageA-r3-IEs after backward compatibility was started for Release 4.
}
MessageA-v440ext-IEs ::=
                                     SEQUENCE {
    -- Here are information elements added to Release 4 as extensions to the information
    -- contained in MessageA-r3-IEs and MessageA-v380ext-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-v440ext-IEs.

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

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-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-v440ext-IEs, and "element1a-3" is included there in the appropriate branch.

```
MessageA-r3-IEs ::=
                                     SEQUENCE {
  For the "choicelb" branch of "choicel", an additional information element is
-- defined in MessageA-v440ext-IEs ("element1a-3").
    choice1
                                         CHOICE
        choicela
                                             SEQUENCE {
            element1a-1
                                                 Element1a-1
                                             SEQUENCE {
        choice1b
            element1a-2
                                                 Element1a-2
    }
}
MessageA-v440ext-IEs ::=
                                         SEQUENCE {
 - In the following CHOICE the same branch shall be used as in choicel in MessageA-r3-IEs.
    choice1
                                         CHOICE {
        choice1a
                                             NULL,
        choicelb
                                             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-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-v440ext-IEs" the branch
-- "choicelb" of the CHOICE "choicel" is used.
    element1
                                         Element1
    element2
                                         Element 2
}
                                         SEQUENCE {
MessageA-v440ext-IEs ::=
    choice1
                                          CHOICE {
                                              SEQUENCE { },
        choicela
                                              SEQUENCE {
        choice1b
            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-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).

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.

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

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.

```
-- In the previous version, MessageA-r3-IEs was defined:
MessageA-r3-IEs ::=
                                     SEQUENCE {
    element1
                                         CHOICE {
        e1
                                             E1,
        e2
                                             E2.
                                             NULL
        spare
}
-- 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
                                     SECUENCE {
MessageA-r4-ext-TEs ::=
-- the following shall be present, if element1 in MessageA-r3-IEs has the value e3.
                                                     OPTIONAL
    e3
                                         E3
```

Example 9

10.4.3.4.6 Introducing new System Information Block Types

In general new message types are introduced by replacing a spare value as described in subclause 10.4.3.4.5. That subclause also shows that in case there are insufficient spare values available, the last spare value can be replaced by a special extension value. If that value is used, an additional message type extension IE is included to distinguish between the additional message types, as shown in Example 10.

```
DL-CCCH-Message ::= SEQUENCE {
                            IntegrityCheckInfo
    integrityCheckInfo
                                                     OPTIONAL,
                            DL-CCCH-MessageType
    message
}
DL-CCCH-MessageType ::= CHOICE {
    cellUpdateConfirm
                                         CellUpdateConfirm-CCCH,
    rrcConnectionReject
                                         RRCConnectionReject,
                                         RRCConnectionRelease-CCCH,
    rrcConnectionRelease
    rrcConnectionSetup
                                         RRCConnectionSetup
    uraUpdateConfirm
                                         URAUpdateConfirm-CCCH,
    ext1
                                         Ext1Message-CCCH,
```

Example 10

For system information block types, the "SIB type" information element is also included in each of the segments. If in this case there are insufficient spare values, the last value can again be used to indicate "extension". If that value is used, an additional SIB type extension IE is included to distinguish between the additional SIB types. This additional IE is not included in the segments; it is only included in the scheduling information included in the MIB and/or the SBs.

NOTE: One could include this additional IE in the segments e.g. by changing the SIB-type into a choice as shown in example 11. This option should not be used since it involves additional overhead (more scarce BCH bits are needed to indicate the SIB type) and complicates the scheduling (more different SIB data sizes are to be considered).

```
SEQUENCE {
FirstSegment ::=
    -- Other information elements
        sib-Type
                                          SIB-Type,
        seq-Count
                                          SegCount
        sib-Data-fixed
                                          SIB-Data-fixed
}
SIB-Type ::=
                                          CHOICE {
        MasterInformationBlock
                                              NULL,
        systemInformationBlockType1
                                              NULL.
        systemInformationBlockType2
                                              NULL,
        systemInformationBlockType3
                                              NULL,
        systemInformationBlockType4
                                              NULL,
        systemInformationBlockType5
                                              NULL.
        systemInformationBlockType6
                                              NULTITAL.
        systemInformationBlockType7
                                              NULL.
        systemInformationBlockType8
                                              NULL,
        systemInformationBlockType9
                                              NULL,
        systemInformationBlockType10
                                              NULL.
        systemInformationBlockType11
                                              NULL,
        systemInformationBlockType12
                                              NULL,
        systemInformationBlockType13
                                              NULL,
        systemInformationBlockType13-1
                                              NULL,
        systemInformationBlockType13-2
                                              NULL,
        systemInformationBlockType13-3
                                              NULL,
        systemInformationBlockType13-4
                                              NULL,
        systemInformationBlockType14
                                              NULL.
        systemInformationBlockType15
                                              NULL.
        {\tt systemInformationBlockType15-1}
                                              NULL,
        systemInformationBlockType15-2
                                              NULL,
        systemInformationBlockType15-3
                                              NULL,
        systemInformationBlockType16
                                              NULL.
        systemInformationBlockType17
                                              NULL.
        systemInformationBlockType15-4
                                              NULL,
        systemInformationBlockType18
                                              NULL,
        schedulingBlock1
                                              NULL,
        schedulingBlock2
                                              NULL.
        systemInformationBlockType15-5
                                              NULL.
        ext1
                                              NULL.
        extension
                                              SIB-TypeExt
SIB-TypeExt ::=
                                      CHOICE {
        ext2
                                          NULL,
                                          NULL.
        spare7
        spare6
                                          NULL,
        spare5
                                          NULL,
        spare4
                                          NULL,
        spare3
                                          NULL,
        spare2
                                          NULT.
                                          NULL
        spare1
```

}

Example 11 - Not recommended

The addition of new SIB types to the scheduling information is illustrated by example 12. The example shows the extension of the choice. The example also shows that the information applicable for the extended choice values is appended at the end of the SIB (in this case the MIB), as a non critical extension.

NOTE: In this example only the number of SIB types is increased; the number of SIBs that can be scheduled (as reflected in the size of the list in the scheduling information) is not extended.

```
MasterInformationBlock ::=
                                     SEQUENCE {
        mib-ValueTag
                                         MIB-ValueTag,
        -- TABULAR: The PLMN identity and ANSI-41 core network information
        -- are included in PLMN-Type.
                                          PLMN-Type,
        plmn-Type
        sibSb-ReferenceList
                                         SIBSb-ReferenceList,
                                             SEQUENCE {
        vxy0NonCriticalExtensions
                                                 MasterInformationBlock-vxy0ext-IEs,
            {\tt masterInformationBlock-vxy0ext}
            nonCriticalExtensions
                                                  SEQUENCE {}
                                                                                    OPTIONAL
        } OPTIONAL
                                     SEQUENCE (SIZE (1..maxSIB)) OF
SIBSb-ReferenceList ::=
                                          SchedulingInformationSIBSb
SchedulingInformationSIBSb ::=
                                          SECUENCE {
    sibSb-Type
                                          SIBSb-TypeAndTag,
    scheduling
                                          SchedulingInformation
SIBSb-TypeAndTag ::=
                                          CHOICE {
    sysInfoType1
                                          PLMN-ValueTag,
    sysInfoType2
                                          CellValueTag,
    sysInfoType3
                                          CellValueTag,
                                          CellValueTag
    sysInfoType4
    sysInfoType5
                                          CellValueTag
                                          CellValueTag,
    sysInfoType6
                                          NULL,
    sysInfoType7
                                          CellValueTag,
    sysInfoType8
    sysInfoType9
                                          NULL,
    sysInfoType10
                                          NULL,
                                          CellValueTag,
    sysInfoType11
    sysInfoType12
                                          CellValueTag,
    sysInfoType13
                                          CellValueTag,
    sysInfoType13-1
                                          CellValueTag,
    sysInfoType13-2
                                          CellValueTag,
    sysInfoType13-3
                                          CellValueTag,
                                          CellValueTag,
    {\tt sysInfoType13-4}
    sysInfoType14
                                          NULL,
    sysInfoType15
                                          CellValueTag,
    sysInfoType16
                                          PredefinedConfigIdentityAndValueTag,
    sysInfoType17
                                          NULL,
    sysInfoTypeSB1
                                          CellValueTag
    sysInfoTypeSB2
                                          CellValueTag,
    sysInfoType15-1
                                          CellValueTag,
    sysInfoType15-2
                                          SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-3
                                          {\tt SIBOccurrenceIdentityAndValueTag},\\
    sysInfoType15-4
                                          CellValueTag,
    sysInfoType18
                                          CellValueTag,
    sysInfoType15-5
                                          CellValueTag,
    ext.1
                                          NULL.
    ext.2
                                          NULL
    extension
                                          NULL
}
SIBSb-TypeAndTagExt ::=
                                          CHOICE {
                                          NULL,
    spare7
                                          NULL,
    spare6
                                          NULL,
    spare5
                                          NULL,
    spare4
                                          NULL,
    spare3
                                          NULL,
    spare2
                                          NULL,
    spare1
                                          NULL
```

```
ExtSIBTypeInfoSchedulingInfo-List OPTIONAL
-- For each extended SIB type the value tag information is added at the end ExtSIBTypeInfoSchedulingInfo-List::= SEQUENCE (SIZE (1..maxSIB)) OF
                                             ExtSIBTypeInfoSchedulingInfo
ExtSIBTypeInfoSchedulingInfo-List::= SEQUENCE {
                                         INTEGER (1..maxSIB),
    schedulingInfoListIndex
    valueTagInfo
                                         ValueTagInfo
}
ValueTagInfo ::= CHOICE {
                                         NULL.
   None
    sysInfoType2
                                         CellValueTag,
    sysInfoType1
                                         PLMN-ValueTag,
    sysInfoType15-3
                                         SIBOccurrenceIdentityAndValueTag
}
```

Example 12 – Recommended method

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 12th – 16th January 2004

CHANGE REQUEST											
*	25.921 CR	59 ⊭rev	■ # Currer	nt version: 4.6.0)						
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the 策 symbols.											
Proposed change affects: UICC apps# ME Radio Access Network Core Network											
7.4	0 :1 !: (1										
Title:		aming of extensions	to the RRC ASN.	.1							
Source: #	RAN WG2										
Work item code: ₩	TEI		Da	ate: 第 <mark>15/01/2004</mark>							
Category: 第	B (addition of fe C (functional mo D (editorial mod	to a correction in an ea ature), odification of feature) lification) of the above categorie	zrlier release) 2 R: R: R: R: s can R:	<u>one</u> of the following re	2) 6) 7) 8)						
Reason for change	: 第 Currently the	REL-4 and REL-5 ex	tensions are not	named in a consist	ent						
		is because the guide rrently neither clear r		ing of extensions to	the RRC						
Summary of chang		roduces a recommen troducing non critical			ate						
Consequences if not approved:	There will be ASN.1	no way to achieve co	onsistent naming o	of extensions to the	RRC						
Clauses affected:	第 10.4.2										
Other specs affected:	X Test sp	ore specifications ecifications pecifications	*								
Other comments:	need not app to track in wh additional car it has already	vention means that soly a specific suffix. The ich release/version to the when applying the been common praction suffix, this is not consuffix, this is not consuffix.	nis means that it whe IE was introdunt backwards compaide to not always	will be somewhat maded. This requires attaction attacts attacts. Howe wise an extension s	ore difficult some ever, since						

How to create CRs using this form:

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

- 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.

10.4.2 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 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. For this naming, "r3" is used for Release '99, "r4" for Release 4, "r5" for Release 5 and so on.

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
                                    SEOUENCE {
        test-msg-r3
                                        Test-msg-r3-IEs,
       nonCriticalExtensions
                                        SEQUENCE {} OPTIONAL
                                   SEQUENCE {
    later-than-r3
       rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
                                        SEQUENCE {}
        criticalExtensions
-- 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-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                           SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
                                                SEQUENCE {} OPTIONAL
           nonCriticalExtensions
        } OPTIONAL
                                    SEQUENCE {
    later-than-r3
       rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
        criticalExtensions
                                       SEQUENCE {}
    }
-- The Test-msg gets the following structure, if only a non-critical
-- extensions is introduced for Release 4 in v4.4.0 of the specification.
Test-msg ::= CHOICE {
                                    SEQUENCE {
    r3
                                    Test-msg-r3-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                         SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
            laterNonCriticalExtensions
                                               SECUENCE {
                -- Container for additional Release '99 extensions
                test-msg-r3-add-ext
                                                   BIT STRING
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL,
                v440nonCriticalExtensions
                                                  SEQUENCE {
                    test-msq-v440ext
                                                        Test-msq-v440ext-IEs,
                    nonCriticalExtensions
                                                        SEQUENCE {} OPTIONAL
                    OPTIONAL
                OPTIONAL
        } OPTIONAL
```

```
later-than-r3
                                    SEOUENCE
        rrc-TransactionIdentifier
                                        RRC-TransactionIdentifier,
        criticalExtensions
                                        SEQUENCE {}
}
-- In Release 5, the Test msg gets the following structure when a critical
-- extension is added
Test-msg ::= CHOICE {
                                    SEQUENCE {
        test-msg-r3
                                       Test-msg-r3-IEs,
        v380nonCriticalExtensions
                                           SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
            laterNonCriticalExtensions
                                               SEQUENCE {
                 - Container for additional Release '99 extensions
                                                   BIT STRING
                test-msg-r3-add-ext
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL.
                                            SEQUENCE {
                v440nonCriticalExtensions
                    test-msg-v440ext
                                                        Test-msg-v440ext-IEs,
                    nonCriticalExtensions
                                                       SEQUENCE {} OPTIONAL
                   OPTIONAL
                OPTIONAL
        } OPTIONAL
    later-than-r3
                                    SECTIENCE {
        rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
        criticalExtensions
                                        CHOICE {
                                           SEQUENCE {
                test-msg-r5
                                                Test-msg-r5-IEs,
                                                SEQUENCE {} OPTIONAL
                nonCriticalExtensions
                                            SEQUENCE {}
            criticalExtensions
    }
}
```

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

If an abstract type is introduced in Release N when new elements are included in an extension, it should have a suffix "-rN". For Release '99 types, no such suffix is used. In case the type that is introduced in Release N includes one or more new (nested) types, the additional suffix need not be used for these nested types. In case the type that is introduced in Release N includes one or more revisions of exixting types, the suffix is needed to distinguish them from the earlier revisions. In case a revision of an abstract type that is introduced in Release N includes an IE for which the abstract type already existed in earlier releases, while that IE was not present in the previous revision(s) of the revised abstract type, the IE name should have a suffix "-rN".

If an abstract type is introduced in a release to extend an already existing type "TypeX", it should get the same name with a non-critical extension type suffix ("-vXYZext", e.g. "TypeX-v380ext") although in this case the final "- IEs" suffix is not added. In case the type that is introduced in Release *N* to extend an already existing type includes one or more new (nested) types that are extensions of an already existing type, the additional suffix should not be used for these nested types. In case the type that is introduced in Release *N* to extend an already existing type includes one or more new (nested) types, the abovely specified rules for new abstract types apply.

The above naming conventions are further illustrated in the example below:

```
Test-msg-v380ext-IEs ::= SEQUENCE {
        existingIE-A-v380ext
                                         ExistingIE-A-v380ext
                                                                      OPTIONAL,
        newIE-B
                                         NewIE-B
Test-msg-v440ext-IEs::= SEQUENCE
       newIE-C-r4
                                         NewIE-C
                                                                      OPTIONAL,
        existingIE-D-v440ext
                                         ExistingIE-D-v440ext
Test-msg-r5-IEs::= SEQUENCE {
                                         ExistingIE-E
       existingIE-E
                                                                      OPTIONAL,
        newUseOfexistingIE-E-rF
                                         ExistingIE-F
                                         NewIE-G
        newIE-G-r5
        revisionOfExistingIE-H-r5
                                         ExistingIE-H-r5
```

}

The abovely described naming convention means that some IEs introduced in a later release/ version need not apply a specific suffix. This means that it will not allways be clear from the name of an IE whether or not backwards incompatible changes to it are allowed. Using the above naming rules, when changes are done in Release N, only changes in types with a suffix "-rN" or "-vXYZext" are allowed, in order to avoid conflicts with previous releases. An exception is tThe Message type itself is a special case, 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 {
                                     SEQUENCE {
    r3
        test-msg-r3
                                         Test-msg-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
                                             Test-msg-v380ext-IEs,
            test-msg-v380ext
                                             SEQUENCE {} OPTIONAL
            nonCriticalExtensions
        } OPTIONAL
                                    SEQUENCE {
    later-than-r3
        rrc-TransactionIdentifier
                                         RRC-TransactionIdentifier,
        criticalExtensions
                                         SEQUENCE {
            importantElements
                                             ImportantElements,
            rest-of-message
                                             CHOICE {
                                                 SEQUENCE {
                r4
                    test-msg-r4
                                                     Test-msg-r4-IEs,
                    nonCriticalExtensions
                                                     SEQUENCE {} OPTIONAL
                                                 SEQUENCE {}
                criticalExtensions
            }
        }
    }
}
```

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 1: The structure presented in this clause and the proposed naming rules are one possibility. Further possibilities are FFS.
- NOTE 2: When non-critical extensions are introduced in a message that does not have yet a criticalExtension branch, they are introduced in the "Test-msg-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-rN-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.

3GPP TSG RAN WG2#40 Sophia Antipolis, France, 12th – 16th January 2004

CHANGE REQUEST CHANGE REQUEST										
	25.	. <mark>921</mark> CR		<mark>60</mark> ⊭re	v -	#	Current vers	sion:	5.3.0	*
For <u>HELP</u> on u	sing t	his form, se	ee bottom o	of this page	or look	at the	pop-up text	over	the	nbols.
Proposed change a	affec	ts: UICC	appsЖ	ME	. Rad	dio Ac	cess Netwo	rk	Core Ne	etwork
Title: ₩	Gui	deline for th	ne naming	of extension	ns to the	e RRC	ASN.1			
Source: #	RA	N WG2								
Work item code: 第	TEI						<i>Date:</i> ∺	15/	01/2004	
Category:	Deta	one of the fo. F (correction A (correspo B (addition of C (functional D (editorial led explanat und in 3GPF	n) nds to a cor of feature), I modification ions of the a	rrection in an on of feature) above catego)	elease)	Release: 光 Use <u>one</u> of 2) R96 R97 R98 R99 Rel-4 Rel-5 Rel-6	the for (GSN (Relea (Relea (Relea (Relea (Relea		eases:
Reason for change	e: X	manner. T	his is beca		ıidelines	for the	re not named e naming of e			
Summary of chang	ye: ₩						e used wher s in the RRC			е
Consequences if not approved:	¥	There will ASN.1	be no way	to achieve	consist	ent na	aming of exte	ensio	ns to the F	RRC
Clauses affected:	ж	10.4.2								
Other specs affected:	*	X Tes	er core spe t specificat M Specifica		器					
Other comments:	*	need not a to track in additional it has alre	apply a spe which rele care wher ady been o	ecific suffix ease/ version applying to common pr	This months The IE he back actice to	eans to was in wards onot a	roduced in a hat it will be ntroduced. To compatibility lways use auto be a prob	some This re y rule n exte	ewhat mor equires so s. Howeve	e difficult me er, since

How to create CRs using this form:

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

- 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.

10.4.2 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 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. For this naming, "r3" is used for Release '99, "r4" for Release 4, "r5" for Release 5 and so on.

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
                                    SEOUENCE {
        test-msg-r3
                                        Test-msg-r3-IEs,
       nonCriticalExtensions
                                        SEQUENCE {} OPTIONAL
                                   SEQUENCE {
    later-than-r3
       rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
                                        SEQUENCE {}
        criticalExtensions
-- 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-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                           SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
                                                SEQUENCE {} OPTIONAL
           nonCriticalExtensions
        } OPTIONAL
                                    SEQUENCE {
    later-than-r3
       rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
        criticalExtensions
                                       SEQUENCE {}
    }
-- The Test-msg gets the following structure, if only a non-critical
-- extensions is introduced for Release 4 in v4.4.0 of the specification.
Test-msg ::= CHOICE {
                                    SEQUENCE {
    r3
                                    Test-msg-r3-IEs,
        test-msg-r3
        v380nonCriticalExtensions
                                         SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
            laterNonCriticalExtensions
                                               SECUENCE {
                -- Container for additional Release '99 extensions
                test-msg-r3-add-ext
                                                   BIT STRING
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL,
                v440nonCriticalExtensions
                                                  SEQUENCE {
                    test-msq-v440ext
                                                        Test-msq-v440ext-IEs,
                    nonCriticalExtensions
                                                        SEQUENCE {} OPTIONAL
                    OPTIONAL
                OPTIONAL
        } OPTIONAL
```

```
later-than-r3
                                    SEOUENCE
        rrc-TransactionIdentifier
                                        RRC-TransactionIdentifier,
        criticalExtensions
                                        SEQUENCE {}
}
-- In Release 5, the Test msg gets the following structure when a critical
-- extension is added
Test-msg ::= CHOICE {
                                    SEQUENCE {
        test-msg-r3
                                       Test-msg-r3-IEs,
        v380nonCriticalExtensions
                                           SEQUENCE {
            test-msg-v380ext
                                                Test-msg-v380ext-IEs,
            laterNonCriticalExtensions
                                               SEQUENCE {
                 - Container for additional Release '99 extensions
                                                   BIT STRING
                test-msg-r3-add-ext
                    (CONTAINING Test-msg-r3-add-ext-IEs)
                                                                    OPTIONAL.
                                            SEQUENCE {
                v440nonCriticalExtensions
                    test-msg-v440ext
                                                        Test-msg-v440ext-IEs,
                    nonCriticalExtensions
                                                       SEQUENCE {} OPTIONAL
                   OPTIONAL
                OPTIONAL
        } OPTIONAL
    later-than-r3
                                    SECTIENCE {
        rrc-TransactionIdentifier
                                       RRC-TransactionIdentifier,
        criticalExtensions
                                        CHOICE {
                                           SEQUENCE {
                test-msg-r5
                                                Test-msg-r5-IEs,
                                                SEQUENCE {} OPTIONAL
                nonCriticalExtensions
                                            SEQUENCE {}
            criticalExtensions
    }
}
```

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

If an abstract type is introduced in Release N when new elements are included in an extension, it should have a suffix "-rN". For Release '99 types, no such suffix is used. In case the type that is introduced in Release N includes one or more new (nested) types, the additional suffix need not be used for these nested types. In case the type that is introduced in Release N includes one or more revisions of exixting types, the suffix is needed to distinguish them from the earlier revisions. In case a revision of an abstract type that is introduced in Release N includes an IE for which the abstract type already existed in earlier releases, while that IE was not present in the previous revision(s) of the revised abstract type, the IE name should have a suffix "-rN".

If an abstract type is introduced in a release to extend an already existing type "TypeX", it should get the same name with a non-critical extension type suffix ("-vXYZext", e.g. "TypeX-v380ext") although in this case the final "- IEs" suffix is not added. In case the type that is introduced in Release *N* to extend an already existing type includes one or more new (nested) types that are extensions of an already existing type, the additional suffix should not be used for these nested types. In case the type that is introduced in Release *N* to extend an already existing type includes one or more new (nested) types, the abovely specified rules for new abstract types apply.

The above naming conventions are further illustrated in the example below:

```
Test-msg-v380ext-IEs ::= SEQUENCE {
        existingIE-A-v380ext
                                         ExistingIE-A-v380ext
                                                                      OPTIONAL,
        newIE-B
                                         NewIE-B
Test-msg-v440ext-IEs::= SEQUENCE
       newIE-C-r4
                                         NewIE-C
                                                                      OPTIONAL,
        existingIE-D-v440ext
                                         ExistingIE-D-v440ext
Test-msg-r5-IEs::= SEQUENCE {
                                         ExistingIE-E
       existingIE-E
                                                                      OPTIONAL,
        newUseOfexistingIE-E-rF
                                         ExistingIE-F
                                         NewIE-G
        newIE-G-r5
        revisionOfExistingIE-H-r5
                                         ExistingIE-H-r5
```

}

The abovely described naming convention means that some IEs introduced in a later release/ version need not apply a specific suffix. This means that it will not allways be clear from the name of an IE whether or not backwards incompatible changes to it are allowed. Using the above naming rules, when changes are done in Release N, only changes in types with a suffix "-rN" or "-vXYZext" are allowed, in order to avoid conflicts with previous releases. An exception is tThe Message type itself is a special case, 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 {
                                     SEQUENCE {
    r3
        test-msg-r3
                                         Test-msg-r3-IEs,
        v380nonCriticalExtensions
                                            SEQUENCE {
                                             Test-msg-v380ext-IEs,
            test-msg-v380ext
                                             SEQUENCE {} OPTIONAL
            nonCriticalExtensions
        } OPTIONAL
                                    SEQUENCE {
    later-than-r3
        rrc-TransactionIdentifier
                                         RRC-TransactionIdentifier,
        criticalExtensions
                                         SEQUENCE {
            importantElements
                                             ImportantElements,
            rest-of-message
                                             CHOICE {
                                                 SEQUENCE {
                r4
                    test-msg-r4
                                                     Test-msg-r4-IEs,
                    nonCriticalExtensions
                                                     SEQUENCE {} OPTIONAL
                                                 SEQUENCE {}
                criticalExtensions
            }
        }
    }
}
```

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 1: The structure presented in this clause and the proposed naming rules are one possibility. Further possibilities are FFS.
- NOTE 2: When non-critical extensions are introduced in a message that does not have yet a criticalExtension branch, they are introduced in the "Test-msg-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-rN-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.