TSG-RAN Meeting #24 **RP-040204**
Seoul, Korea, 02-04 June 2004

Title: **CRs to 25.921 (R'99 and associated Rel-4/Rel-5)**

Source: **TSG-RAN WG2**

Agenda item: **7.3.3**

Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Version-New	Workitem	Doc-2nd-Level
25.921	061	-	R99	Empty non-critical extensions	F	3.10.0	3.11.0	TEI	R2-041080
25.921	062	-	Rel-4	Empty non-critical extensions	A	4.7.0	4.8.0	TEI	R2-041081
25.921	063	-	Rel-5	Empty non-critical extensions	A	5.4.0	5.5.0	TEI	R2-041082

3GPP TSG-RAN-WG2 Meeting #42
Montreal, Canada, 10th- 14st May 2004

Tdoc ⌘R2-041080

CR-Form-v7

CHANGE REQUEST

⌘	**25.921**	**CR**	**061**	⌘**rev**	**-**	⌘ Current version:	**3.10.0** ⌘

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

Proposed change affects: UICC apps⌘ □ ME ☒ Radio Access Network ☒ Core Network □

Title:	⌘ Empty non-critical extensions		
Source:	⌘ RAN WG2		
Work item code:	⌘ TEI	***Date:*** ⌘	May /2004
Category:	⌘ **F**	***Release:*** ⌘	R99
	Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.	*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *Rel-4 (Release 4)* *Rel-5 (Release 5)* *Rel-6 (Release 6)*	

Reason for change: ⌘	The specification currently covers the issues of receiving unknown non-critical extensions. However, it does not currently cover the issues of forward compatibility caused by the transmission of empty non-critical extensions – i.e. indicating the presence of “nonCriticalExtension” IEs.
Summary of change: ⌘	It is clarified that, for forward compatibility reasons, empty non-critical extension shall not be transmitted
Consequences if not approved: ⌘	Some UE/UTRAN implementations may transmit empty non-critical extensions and cause a ASN.1 violation decoding error, if the receiver supports a later version of ASN.1 (of the same of later release). Such decoding errors will impair the working of any functionality associated with the message where the error occurs and in the extreme cases camping on a cell or setting up a connection will not be possible. **Isolated Impact analysis**: The implementation of this CR is always required at the transmitter side, therefore it impacts UE/UTRAN implementations in the same way. Failure to do so may cause severe problems, as highlighted above. Although it is expected that most implementations already comply to this CR, any implementations that do transmit empty non-critical extensions will require modifications, when those non-critical extension are used in a future version of ASN.1 (same or later release).

Clauses affected:	⌘				
		Y	N		
Other specs	⌘		X	Other core specifications	⌘

affected:			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘	A related CR was recently approved in T1 and can be found in RP-040073			

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:

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

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

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

10.4.3.3 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. For forward compatibility reasons, the transmitter should never include fields reserved for future non-critical extensions. This principle is illustrated using the example defined below: a transmitter conforming to the v380 version of the standard including MessageA as defined in Example 3 (see below), should not include the IE nonCriticalExtensions.

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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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
    },
    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.
}



-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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,
                v440nonCriticalExtensions           SEQUENCE {
                    messageA-v440ext                    MessageA-v440ext-IEs,
                    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 ::=            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

3GPP TSG-RAN-WG2 Meeting #42
Montreal, Canada, 10th- 14st May 2004

Tdoc ⌘R2-041081

CR-Form-v7

CHANGE REQUEST

⌘	**25.921**	**CR**	**062**	⌘**rev**	**-**	⌘	Current version:	**4.7.0** ⌘

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

Proposed change affects: UICC apps⌘ □ ME ☒ Radio Access Network ☒ Core Network □

Title:	⌘	Empty non-critical extensions		
Source:	⌘	RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	May /2004
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)* ***D*** *(editorial modification)* 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)*

Reason for change:	⌘	The specification currently covers the issues of receiving unknown non-critical extensions. However, it does not currently cover the issues of forward compatibility caused by the transmission of empty non-critical extensions – i.e. indicating the presence of “nonCriticalExtension” IEs.
Summary of change:	⌘	It is clarified that, for forward compatibility reasons, empty non-critical extension shall not be transmitted
Consequences if not approved:	⌘	Some UE/UTRAN implementations may transmit empty non-critical extensions and cause a ASN.1 violation decoding error, if the receiver supports a later version of ASN.1 (of the same of later release). Such decoding errors will impair the working of any functionality associated with the message where the error occurs and in the extreme cases camping on a cell or setting up a connection will not be possible. **Isolated Impact analysis**: The implementation of this CR is always required at the transmitter side, therefore it impacts UE/UTRAN implementations in the same way. Failure to do so may cause severe problems, as highlighted above. Although it is expected that most implementations already comply to this CR, any implementations that do transmit empty non-critical extensions will require modifications, when those non-critical extension are used in a future version of ASN.1 (same or later release).

Clauses affected:	⌘					
		Y	**N**			
Other specs	⌘		**X**	Other core specifications	⌘	

affected:		X	Test specifications	
		X	O&M Specifications	
Other comments:	⌘	A related CR was recently approved in T1 and can be found in RP-040073		

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:

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

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

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

10.4.3.3 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. For forward compatibility reasons, the transmitter should never include fields reserved for future non-critical extensions. This principle is illustrated using the example defined below: a transmitter conforming to the v380 version of the standard including MessageA as defined in Example 3 (see below), should not include the IE nonCriticalExtensions.

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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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
    },
    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.
}



-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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,
                v440nonCriticalExtensions           SEQUENCE {
                    messageA-v440ext                    MessageA-v440ext-IEs,
                    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 ::=            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

3GPP TSG-RAN-WG2 Meeting #42
Montreal, Canada, 10th- 14st May 2004

Tdoc ⌘R2-041082

CR-Form-v7

CHANGE REQUEST

⌘	**25.921**	**CR**	**063**	⌘**rev**	**-**	⌘	Current version:	**5.4.0**	⌘

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

Proposed change affects: UICC apps⌘ □ ME ☒ Radio Access Network ☒ Core Network □

Title:	⌘	Empty non-critical extensions		
Source:	⌘	RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	May /2004
Category:	⌘	**A**	***Release:*** ⌘	Rel-5

Use one of the following categories:
- ***F** (correction)*
- ***A** (corresponds to a correction in an earlier release)*
- ***B** (addition of feature),*
- ***C** (functional modification of feature)*
- ***D** (editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change: ⌘ The specification currently covers the issues of receiving unknown non-critical extensions. However, it does not currently cover the issues of forward compatibility caused by the transmission of empty non-critical extensions – i.e. indicating the presence of "nonCriticalExtension" IEs.

Summary of change: ⌘ It is clarified that, for forward compatibility reasons, empty non-critical extension shall not be transmitted

Consequences if not approved: ⌘ Some UE/UTRAN implementations may transmit empty non-critical extensions and cause a ASN.1 violation decoding error, if the receiver supports a later version of ASN.1 (of the same of later release). Such decoding errors will impair the working of any functionality associated with the message where the error occurs and in the extreme cases camping on a cell or setting up a connection will not be possible.

Isolated Impact analysis:
The implementation of this CR is always required at the transmitter side, therefore it impacts UE/UTRAN implementations in the same way. Failure to do so may cause severe problems, as highlighted above.
Although it is expected that most implementations already comply to this CR, any implementations that do transmit empty non-critical extensions will require modifications, when those non-critical extension are used in a future version of ASN.1 (same or later release).

Clauses affected: ⌘

		Y	**N**		
Other specs	⌘		**X**	Other core specifications	⌘

affected:		X	Test specifications	
		X	O&M Specifications	
Other comments:	⌘	A related CR was recently approved in T1 and can be found in RP-040073		

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:

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

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

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

10.4.3.3 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. For forward compatibility reasons, the transmitter should never include fields reserved for future non-critical extensions. This principle is illustrated using the example defined below: a transmitter conforming to the v380 version of the standard including MessageA as defined in Example 3 (see below), should not include the IE nonCriticalExtensions.

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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        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.
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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
    },
    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.
}



-- This shows the structure of the Release 4 message
-- (including one Release 4 non-critical extension).
MessageA ::=                CHOICE {
    r3                              SEQUENCE {
        messageA-r3                     MessageA-r3-IEs,
        v380nonCriticalExtensions           SEQUENCE {
            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,
                v440nonCriticalExtensions           SEQUENCE {
                    messageA-v440ext                    MessageA-v440ext-IEs,
                    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 ::=            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