**TSG-RAN Meeting #21** ***RP-030488***
**Frankfurt, Germany, 16-19 September 2003**

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

**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 | 045 | - | R99 | Guideline on introducing additional SIB types | F | 3.8.0 | 3.9.0 | R2-031844 | TEI |
| 25.921 | 046 | - | Rel-4 | Guideline on introducing additional SIB types | A | 4.5.0 | 4.6.0 | R2-031845 | TEI |
| 25.921 | 047 | - | Rel-5 | Guideline on introducing additional SIB types | A | 5.1.0 | 5.2.0 | R2-031846 | TEI |

**3GPP TSG-RAN2 Meeting #37**
**Budapest, Hungary, August 25th-29th, 2003**

***Tdoc ⌘R2-031844***

*CR-Form-v7*

# CHANGE REQUEST

⌘ **25.921** **CR** **045** ⌘ **rev** **-** ⌘ Current version: **3.8.0** ⌘

*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:*** | ⌘ | Guideline on introducing additional SIB types |
| ***Source:*** | ⌘ | RAN WG2 |
| ***Work item code:*** | ⌘ | TEI |
| ***Date:*** | ⌘ | August 2003 |
| ***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:*** | ⌘ | It is unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |
| ***Summary of change:*** | ⌘ | The proposal is to include the identification for SIB types beyond 31 only within the scheduling information<br><br>This CR concerns a guideline and hence it does not directly affect implementations. Changes that make use of the guideline are backwards compatible |
| ***Consequences if not approved:*** | ⌘ | It will remain unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 10.4.3.4.6 (New) |

| ***Other specs affected:*** | ⌘ | Y | N | | ⌘ |
|---|---|---|---|---|---|
| | | | X | Other core specifications | |
| | | | X | Test specifications | |
| | | | X | O&M Specifications | |

| | | |
|---|---|---|
| ***Other comments:*** | ⌘ | |

**How to create CRs using this form:**
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/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 Recommendations for extensions for further releases in RRC

### 10.4.3.1 General

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

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

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

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

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

**Example 1**

### 10.4.3.2 Critical Extensions

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

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

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

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

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

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

MessageA-r4-IEs ::=                 SEQUENCE {
```

```
    -- Here, the updated information elements used for MessageA in Release 4 are included.
}
```

**Example 2**

### 10.4.3.3 Non-critical Extensions

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

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**

## 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".

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

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

**Example 4**

### 10.4.3.4.2 Addition of an IE to a structured group

If the extension is the addition of an information element inside a CHOICE, SEQUENCE OF, etc. (meaning that the information element can be absent or present more than once, depending on some condition), the structure of the original message should be duplicated in MessageA-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 "choice1b" branch of "choice1", an additional information element is
-- defined in MessageA-v440ext-IEs ("element1a-3").
    choice1                             CHOICE {
        choice1a                            SEQUENCE {
            element1a-1                         Element1a-1
        },
        choice1b                            SEQUENCE {
            element1a-2                         Element1a-2
        }
    }
}

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

**Example 5**

### 10.4.3.4.3 Addition of a new CHOICE group

If the extension consists of moving some existing information elements inside a newly created CHOICE, the new branches of the created CHOICE should be included in MessageA-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
-- "choice1b" of the CHOICE "choice1" is used.
    element1                            Element1,
    element2                            Element2
}

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

**Example 6**

### 10.4.3.4.4 Extension of value range

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

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

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

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

**Example 7**

### 10.4.3.4.5 Replacement of a spare value with a new element

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

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

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

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

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

**Example 8**

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

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

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

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

**Example 9**

### 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 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,
    message                 DL-CCCH-MessageType
}

DL-CCCH-MessageType ::= CHOICE {
    cellUpdateConfirm                   CellUpdateConfirm-CCCH,
    rrcConnectionReject                 RRCConnectionReject,
    rrcConnectionRelease                RRCConnectionRelease-CCCH,
    rrcConnectionSetup                  RRCConnectionSetup,
    uraUpdateConfirm                    URAUpdateConfirm-CCCH,
    ext1                                Ext1Message-CCCH,
```

```
    ext2                                    Ext2Message-CCCH,
    extension                               DL-CCCH-MessageTypeExt
}

DL-CCCH-MessageTypeExt ::= CHOICE {
    Ext3                                    Ext3Message-CCCH,
    spare3                                  NULL,
    spare2                                  NULL,
    spare1                                  NULL
}
```

## 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).

```
FirstSegment ::=                    SEQUENCE {
    -- Other information elements
        sib-Type                            SIB-Type,
        seg-Count                           SegCount,
        sib-Data-fixed                      SIB-Data-fixed
}

SIB-Type ::=                                CHOICE {
        MasterInformationBlock                  NULL,
        systemInformationBlockType1             NULL,
        systemInformationBlockType2             NULL,
        systemInformationBlockType3             NULL,
        systemInformationBlockType4             NULL,
        systemInformationBlockType5             NULL,
        systemInformationBlockType6             NULL,
        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,
        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,
        spare7                                  NULL,
        spare6                                  NULL,
        spare5                                  NULL,
        spare4                                  NULL,
        spare3                                  NULL,
        spare2                                  NULL,
        spare1                                  NULL
```

```
}
```

## 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,
        vxy0NonCriticalExtensions           SEQUENCE {
            masterInformationBlock-vxy0ext      MasterInformationBlock-vxy0ext-IEs,
            nonCriticalExtensions               SEQUENCE {}                     OPTIONAL
        } OPTIONAL
}

SIBSb-ReferenceList ::=             SEQUENCE (SIZE (1..maxSIB)) OF
                                        SchedulingInformationSIBSb

SchedulingInformationSIBSb ::=          SEQUENCE {
    sibSb-Type                          SIBSb-TypeAndTag,
    scheduling                          SchedulingInformation
}

SIBSb-TypeAndTag ::=                    CHOICE {
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType2                        CellValueTag,
    sysInfoType3                        CellValueTag,
    sysInfoType4                        CellValueTag,
    sysInfoType5                        CellValueTag,
    sysInfoType6                        CellValueTag,
    sysInfoType7                        NULL,
    sysInfoType8                        CellValueTag,
    sysInfoType9                        NULL,
    sysInfoType10                       NULL,
    sysInfoType11                       CellValueTag,
    sysInfoType12                       CellValueTag,
    sysInfoType13                       CellValueTag,
    sysInfoType13-1                     CellValueTag,
    sysInfoType13-2                     CellValueTag,
    sysInfoType13-3                     CellValueTag,
    sysInfoType13-4                     CellValueTag,
    sysInfoType14                       NULL,
    sysInfoType15                       CellValueTag,
    sysInfoType16                       PredefinedConfigIdentityAndValueTag,
    sysInfoType17                       NULL,
    sysInfoTypeSB1                      CellValueTag,
    sysInfoTypeSB2                      CellValueTag,
    sysInfoType15-1                     CellValueTag,
    sysInfoType15-2                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-4                     CellValueTag,
    sysInfoType18                       CellValueTag,
    sysInfoType15-5                     CellValueTag,
    ext1                                NULL,
    ext2                                NULL,
    extension                           NULL
}

SIBSb-TypeAndTagExt ::=                 CHOICE {
    ext3                                NULL,
    spare7                              NULL,
    spare6                              NULL,
    spare5                              NULL,
    spare4                              NULL,
    spare3                              NULL,
    spare2                              NULL,
    spare1                              NULL
```

```
}

MasterInformationBlock-vxy0ext-IEs ::= SEQUENCE {
    extSIBTypeInfoSchedulingInfo-List       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 {
    schedulingInfoListIndex             INTEGER (1..maxSIB),
    valueTagInfo                        ValueTagInfo
}

ValueTagInfo ::= CHOICE {
    None                                NULL.
    sysInfoType2                        CellValueTag,
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag
}
```

**Example 12 – Recommended method**

**3GPP TSG-RAN2 Meeting #37**
**Budapest, Hungary, August 25th-29th, 2003**

***Tdoc ⌘R2-031845***

*CR-Form-v7*

# CHANGE REQUEST

⌘ **25.921** **CR** **046** ⌘ **rev** **-** ⌘ Current version: **4.5.0** ⌘

*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:*** | ⌘ | Guideline on introducing additional SIB types |
| ***Source:*** | ⌘ | RAN WG2 |
| ***Work item code:*** | ⌘ | TEI |
| ***Date:*** | ⌘ | August 2003 |
| ***Category:*** | ⌘ | **A** |
| ***Release:*** | ⌘ | REL-4 |

*Use one of the following categories:*
***F*** *(correction)*
***A*** *(corresponds to a correction in an earlier release)*
***B*** *(addition of feature),*
***C*** *(functional modification of feature)*
***D*** *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

*Use one of the following releases:*

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

| | | |
|---|---|---|
| ***Reason for change:*** | ⌘ | It is unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |
| ***Summary of change:*** | ⌘ | The proposal is to include the identification for SIB types beyond 31 only within the scheduling information<br><br>This CR concerns a guideline and hence it does not directly affect implementations. Changes that make use of the guideline are backwards compatible |
| ***Consequences if not approved:*** | ⌘ | It will remain unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 10.4.3.4.6 (New) |

| ***Other specs affected:*** | ⌘ | Y | N | | ⌘ |
|---|---|---|---|---|---|
| | | | X | Other core specifications | |
| | | | X | Test specifications | |
| | | | X | O&M Specifications | |

| | | |
|---|---|---|
| ***Other comments:*** | ⌘ | |

**How to create CRs using this form:**
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/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 Recommendations for extensions for further releases in RRC

### 10.4.3.1 General

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

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

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

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

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

**Example 1**

### 10.4.3.2 Critical Extensions

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

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

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

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

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

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

MessageA-r4-IEs ::=                 SEQUENCE {
```

```
    -- Here, the updated information elements used for MessageA in Release 4 are included.
}
```

**Example 2**

### 10.4.3.3 Non-critical Extensions

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

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**

## 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".

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

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

**Example 4**

### 10.4.3.4.2 Addition of an IE to a structured group

If the extension is the addition of an information element inside a CHOICE, SEQUENCE OF, etc. (meaning that the information element can be absent or present more than once, depending on some condition), the structure of the original message should be duplicated in MessageA-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 "choice1b" branch of "choice1", an additional information element is
-- defined in MessageA-v440ext-IEs ("element1a-3").
    choice1                             CHOICE {
        choice1a                            SEQUENCE {
            element1a-1                         Element1a-1
        },
        choice1b                            SEQUENCE {
            element1a-2                         Element1a-2
        }
    }
}

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

**Example 5**

### 10.4.3.4.3 Addition of a new CHOICE group

If the extension consists of moving some existing information elements inside a newly created CHOICE, the new branches of the created CHOICE should be included in MessageA-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
-- "choice1b" of the CHOICE "choice1" is used.
    element1                            Element1,
    element2                            Element2
}

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

**Example 6**

### 10.4.3.4.4 Extension of value range

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

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

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

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

**Example 7**

### 10.4.3.4.5 Replacement of a spare value with a new element

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

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

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

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

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

**Example 8**

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

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

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

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

**Example 9**

### 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 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,
    message                 DL-CCCH-MessageType
}

DL-CCCH-MessageType ::= CHOICE {
    cellUpdateConfirm                   CellUpdateConfirm-CCCH,
    rrcConnectionReject                 RRCConnectionReject,
    rrcConnectionRelease                RRCConnectionRelease-CCCH,
    rrcConnectionSetup                  RRCConnectionSetup,
    uraUpdateConfirm                    URAUpdateConfirm-CCCH,
    ext1                                Ext1Message-CCCH,
```

```
    ext2                                    Ext2Message-CCCH,
    extension                               DL-CCCH-MessageTypeExt
}

DL-CCCH-MessageTypeExt ::= CHOICE {
    Ext3                                    Ext3Message-CCCH,
    spare3                                  NULL,
    spare2                                  NULL,
    spare1                                  NULL
}
```

## 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).

```
FirstSegment ::=                    SEQUENCE {
    -- Other information elements
        sib-Type                            SIB-Type,
        seg-Count                           SegCount,
        sib-Data-fixed                      SIB-Data-fixed
}

SIB-Type ::=                                CHOICE {
        MasterInformationBlock                  NULL,
        systemInformationBlockType1             NULL,
        systemInformationBlockType2             NULL,
        systemInformationBlockType3             NULL,
        systemInformationBlockType4             NULL,
        systemInformationBlockType5             NULL,
        systemInformationBlockType6             NULL,
        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,
        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,
        spare7                                  NULL,
        spare6                                  NULL,
        spare5                                  NULL,
        spare4                                  NULL,
        spare3                                  NULL,
        spare2                                  NULL,
        spare1                                  NULL
```

```
}
```

## 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,
        vxy0NonCriticalExtensions           SEQUENCE {
            masterInformationBlock-vxy0ext      MasterInformationBlock-vxy0ext-IEs,
            nonCriticalExtensions               SEQUENCE {}                     OPTIONAL
        } OPTIONAL
}

SIBSb-ReferenceList ::=             SEQUENCE (SIZE (1..maxSIB)) OF
                                        SchedulingInformationSIBSb

SchedulingInformationSIBSb ::=          SEQUENCE {
    sibSb-Type                          SIBSb-TypeAndTag,
    scheduling                          SchedulingInformation
}

SIBSb-TypeAndTag ::=                    CHOICE {
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType2                        CellValueTag,
    sysInfoType3                        CellValueTag,
    sysInfoType4                        CellValueTag,
    sysInfoType5                        CellValueTag,
    sysInfoType6                        CellValueTag,
    sysInfoType7                        NULL,
    sysInfoType8                        CellValueTag,
    sysInfoType9                        NULL,
    sysInfoType10                       NULL,
    sysInfoType11                       CellValueTag,
    sysInfoType12                       CellValueTag,
    sysInfoType13                       CellValueTag,
    sysInfoType13-1                     CellValueTag,
    sysInfoType13-2                     CellValueTag,
    sysInfoType13-3                     CellValueTag,
    sysInfoType13-4                     CellValueTag,
    sysInfoType14                       NULL,
    sysInfoType15                       CellValueTag,
    sysInfoType16                       PredefinedConfigIdentityAndValueTag,
    sysInfoType17                       NULL,
    sysInfoTypeSB1                      CellValueTag,
    sysInfoTypeSB2                      CellValueTag,
    sysInfoType15-1                     CellValueTag,
    sysInfoType15-2                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-4                     CellValueTag,
    sysInfoType18                       CellValueTag,
    sysInfoType15-5                     CellValueTag,
    ext1                                NULL,
    ext2                                NULL,
    extension                           NULL
}

SIBSb-TypeAndTagExt ::=                 CHOICE {
    ext3                                NULL,
    spare7                              NULL,
    spare6                              NULL,
    spare5                              NULL,
    spare4                              NULL,
    spare3                              NULL,
    spare2                              NULL,
    spare1                              NULL
```

```
}

MasterInformationBlock-vxy0ext-IEs ::= SEQUENCE {
    extSIBTypeInfoSchedulingInfo-List       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 {
    schedulingInfoListIndex             INTEGER (1..maxSIB),
    valueTagInfo                        ValueTagInfo
}

ValueTagInfo ::= CHOICE {
    None                                NULL.
    sysInfoType2                        CellValueTag,
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag
}
```

**Example 12 – Recommended method**

**3GPP TSG-RAN2 Meeting #37**
**Budapest, Hungary, August 25th-29th, 2003**

***Tdoc ⌘R2-031846***

*CR-Form-v7*

# CHANGE REQUEST

⌘ **25.921** **CR** **047** ⌘ **rev** **-** ⌘ Current version: **5.1.0** ⌘

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

***Proposed change affects:*** UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☐

| | | |
|---|---|---|
| ***Title:*** | ⌘ | Guideline on introducing additional SIB types |
| ***Source:*** | ⌘ | RAN WG2 |
| ***Work item code:*** | ⌘ | TEI |
| ***Date:*** | ⌘ | August 2003 |
| ***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:*** | ⌘ | It is unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |
| ***Summary of change:*** | ⌘ | The proposal is to include the identification for SIB types beyond 31 only within the scheduling information<br><br>This CR concerns a guideline and hence it does not directly affect implementations. Changes that make use of the guideline are backwards compatible |
| ***Consequences if not approved:*** | ⌘ | It will remain unclear how additional SIB types should be introduced beyond the limit of 31, especially within segments |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 10.4.3.4.6 (New) |

| ***Other specs affected:*** | ⌘ | Y | N | | ⌘ |
|---|---|---|---|---|---|
| | | | X | Other core specifications | |
| | | | X | Test specifications | |
| | | | X | O&M Specifications | |

| | | |
|---|---|---|
| ***Other comments:*** | ⌘ | |

**How to create CRs using this form:**
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/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 Recommendations for extensions for further releases in RRC

### 10.4.3.1 General

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

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

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

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

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

**Example 1**

### 10.4.3.2 Critical Extensions

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

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

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

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

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

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

MessageA-r4-IEs ::=                 SEQUENCE {
```

```
    -- Here, the updated information elements used for MessageA in Release 4 are included.
}
```

**Example 2**

### 10.4.3.3 Non-critical Extensions

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

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**

## 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".

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

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

**Example 4**

### 10.4.3.4.2 Addition of an IE to a structured group

If the extension is the addition of an information element inside a CHOICE, SEQUENCE OF, etc. (meaning that the information element can be absent or present more than once, depending on some condition), the structure of the original message should be duplicated in MessageA-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 "choice1b" branch of "choice1", an additional information element is
-- defined in MessageA-v440ext-IEs ("element1a-3").
    choice1                             CHOICE {
        choice1a                            SEQUENCE {
            element1a-1                         Element1a-1
        },
        choice1b                            SEQUENCE {
            element1a-2                         Element1a-2
        }
    }
}

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

**Example 5**

### 10.4.3.4.3 Addition of a new CHOICE group

If the extension consists of moving some existing information elements inside a newly created CHOICE, the new branches of the created CHOICE should be included in MessageA-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
-- "choice1b" of the CHOICE "choice1" is used.
    element1                            Element1,
    element2                            Element2
}

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

**Example 6**

### 10.4.3.4.4 Extension of value range

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

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

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

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

**Example 7**

### 10.4.3.4.5 Replacement of a spare value with a new element

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

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

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

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

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

**Example 8**

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

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

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

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

**Example 9**

### 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 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,
    message                 DL-CCCH-MessageType
}

DL-CCCH-MessageType ::= CHOICE {
    cellUpdateConfirm                   CellUpdateConfirm-CCCH,
    rrcConnectionReject                 RRCConnectionReject,
    rrcConnectionRelease                RRCConnectionRelease-CCCH,
    rrcConnectionSetup                  RRCConnectionSetup,
    uraUpdateConfirm                    URAUpdateConfirm-CCCH,
    ext1                                Ext1Message-CCCH,
```

```
    ext2                                    Ext2Message-CCCH,
    extension                               DL-CCCH-MessageTypeExt
}

DL-CCCH-MessageTypeExt ::= CHOICE {
    Ext3                                    Ext3Message-CCCH,
    spare3                                  NULL,
    spare2                                  NULL,
    spare1                                  NULL
}
```

## 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).

```
FirstSegment ::=                    SEQUENCE {
    -- Other information elements
        sib-Type                            SIB-Type,
        seg-Count                           SegCount,
        sib-Data-fixed                      SIB-Data-fixed
}

SIB-Type ::=                                CHOICE {
        MasterInformationBlock                  NULL,
        systemInformationBlockType1             NULL,
        systemInformationBlockType2             NULL,
        systemInformationBlockType3             NULL,
        systemInformationBlockType4             NULL,
        systemInformationBlockType5             NULL,
        systemInformationBlockType6             NULL,
        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,
        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,
        spare7                                  NULL,
        spare6                                  NULL,
        spare5                                  NULL,
        spare4                                  NULL,
        spare3                                  NULL,
        spare2                                  NULL,
        spare1                                  NULL
```

```
}
```

## 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,
        vxy0NonCriticalExtensions           SEQUENCE {
            masterInformationBlock-vxy0ext      MasterInformationBlock-vxy0ext-IEs,
            nonCriticalExtensions               SEQUENCE {}                     OPTIONAL
        } OPTIONAL
}

SIBSb-ReferenceList ::=             SEQUENCE (SIZE (1..maxSIB)) OF
                                        SchedulingInformationSIBSb

SchedulingInformationSIBSb ::=          SEQUENCE {
    sibSb-Type                          SIBSb-TypeAndTag,
    scheduling                          SchedulingInformation
}

SIBSb-TypeAndTag ::=                    CHOICE {
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType2                        CellValueTag,
    sysInfoType3                        CellValueTag,
    sysInfoType4                        CellValueTag,
    sysInfoType5                        CellValueTag,
    sysInfoType6                        CellValueTag,
    sysInfoType7                        NULL,
    sysInfoType8                        CellValueTag,
    sysInfoType9                        NULL,
    sysInfoType10                       NULL,
    sysInfoType11                       CellValueTag,
    sysInfoType12                       CellValueTag,
    sysInfoType13                       CellValueTag,
    sysInfoType13-1                     CellValueTag,
    sysInfoType13-2                     CellValueTag,
    sysInfoType13-3                     CellValueTag,
    sysInfoType13-4                     CellValueTag,
    sysInfoType14                       NULL,
    sysInfoType15                       CellValueTag,
    sysInfoType16                       PredefinedConfigIdentityAndValueTag,
    sysInfoType17                       NULL,
    sysInfoTypeSB1                      CellValueTag,
    sysInfoTypeSB2                      CellValueTag,
    sysInfoType15-1                     CellValueTag,
    sysInfoType15-2                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag,
    sysInfoType15-4                     CellValueTag,
    sysInfoType18                       CellValueTag,
    sysInfoType15-5                     CellValueTag,
    ext1                                NULL,
    ext2                                NULL,
    extension                           NULL
}

SIBSb-TypeAndTagExt ::=                 CHOICE {
    ext3                                NULL,
    spare7                              NULL,
    spare6                              NULL,
    spare5                              NULL,
    spare4                              NULL,
    spare3                              NULL,
    spare2                              NULL,
    spare1                              NULL
```

```
}

MasterInformationBlock-vxy0ext-IEs ::= SEQUENCE {
    extSIBTypeInfoSchedulingInfo-List       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 {
    schedulingInfoListIndex             INTEGER (1..maxSIB),
    valueTagInfo                        ValueTagInfo
}

ValueTagInfo ::= CHOICE {
    None                                NULL.
    sysInfoType2                        CellValueTag,
    sysInfoType1                        PLMN-ValueTag,
    sysInfoType15-3                     SIBOccurrenceIdentityAndValueTag
}
```

**Example 12 – Recommended method**