Highlights of the OSS NAS API

Role of the NAS message "originator"

- The encoder and the decoder take into account the **originator** of a message (either **UE** or **network**) specified by the user application. This originator determines the set of permitted values for certain fields and the interpretation of certain other field values, which affects the outcome of an encode or decode operation.

Highlights relative to the encoder

- The encoder checks all information elements (**IEs**) against their **minimum** and **maximum permitted lengths**, as specified by the NAS protocol.
- For fields that have a set of **forbidden values** specified by the NAS protocol, the encoder checks the value assigned by the user application against those forbidden values. If a violation is detected, the encoder generates an error message and terminates the operation.
- For fields that have a single **required value** (e.g., a "spare bits" field, which must be set to 0) specified by the NAS protocol, the encoder automatically sets the field to the required value. Therefore, the user application can simply ignore the existence of these fields.
- Each error message generated by the encoder includes the **complete path of the field** associated with the error, if applicable.

Highlights relative to the decoder

Generation of error messages during a decode operation

- The decoder maintains a **text buffer** (or a special **Diagnostics object** in the NAS/Java decoder) that may contain one or more **error messages** (concatenated). This buffer (or Diagnostics object) is emptied at the start of a decode operation.
- When the decoder encounters an information element (**IE**) prefixed by a **length** that is **outside the permitted range of lengths** for that IE, the decoder adds a **recoverable error message** to the error message buffer, skips the IE, and **continues** the operation.
- When the decoder encounters an **unknown IE** encoded as **"comprehension required"**, the decoder adds a **recoverable error message** to the error message buffer, skips the IE, and **continues** the operation.
- When the decoder encounters an **out-of-sequence IE** encoded as **"comprehension required"**, the decoder adds a **recoverable error message** to the error message buffer, skips the IE, and **continues** the operation.

  When the decoder encounters a field that has a set of **forbidden values** specified by the NAS protocol and is contained in a **mandatory IE**, the decoder checks the value against those forbidden values. If a violation is detected, the decoder adds a **recoverable error message** to the error message buffer and **continues** the operation.

- When the decoder encounters a field that has a single **required value** (with the exception of spare-bit fields) specified by the NAS protocol and is contained in a **mandatory IE**, the decoder checks the value against the required value. If a violation is detected, the decoder adds a **recoverable error message** to the error message buffer and **continues** the operation.
- When the decoder encounters a **mandatory IE** that is **syntactically incorrect**, the decoder adds a **critical error message** to the error message buffer and **aborts** the operation.
- At the end of a decode operation, if **one or more recoverable errors** have occurred during the operation, the decoder adds a **critical error message** to the error message buffer. Note that the NAS/Java decoder throws an exception instead of adding a critical error.
- At the end of a decode operation, the error message buffer (that contains all the error messages generated during the operation, if any) becomes available to the user application. If a decode operation returns a result code of zero (success), the error message buffer will be empty, otherwise it will contain one or more error messages. Note that the NAS/Java decoder throws an exception to report an error instead of returning a status code.
- Each error message generated by the decoder includes the **complete path of the field** associated with the error, if applicable.

Generation of warning messages during a decode operation

- The decoder maintains a **text buffer** (or a special **Diagnostics object** in the NAS/Java decoder) that may contain one or more **warning messages** (concatenated). This buffer (or Diagnostics object) is emptied at the start of a decode operation.
- When the decoder encounters an **unknown IE** that is not encoded as **"comprehension required"**, the decoder adds a **warning message** to the warning message buffer, skips the IE, and continues the operation.
- When the decoder encounters an **out-of-sequence IE** that is not encoded as **"comprehension required"**, the decoder adds a **warning message** to the warning message buffer, skips the IE, and continues the operation.
- If an **IE** with a format of T, TV, TLV, or TLV-E **occurs more than once** in a message, the decoder adds a **warning message** to the warning message buffer, skips the occurrence, and continues the operation when it encounters the second or any subsequent occurrence of that IE.
- When the decoder encounters a known **optional IE** that is **syntactically incorrect**, the decoder reports that IE as **absent**, adds a **warning message** to the warning message buffer, and continues the operation.
- When the decoder encounters a field that has a set of **forbidden values** specified by the NAS protocol and is contained in an **optional IE**, the decoder checks the value against the forbidden values. If a violation is detected, the decoder adds a **warning message** to the warning message buffer and continues the operation.
- When the decoder encounters a field that has a single **required value** (with the exception of spare-bit fields) specified by the NAS protocol and is contained in an **optional IE**, the decoder checks the value against the required value. If a violation is detected, the decoder adds a **warning message** to the warning message buffer and continues the operation.

- At the end of a decode operation, the warning message buffer (that contains all the warning messages generated during the operation, if any) becomes available to the user application. This applies to both successful and failed operations.
- **Most applications can ignore the warning messages.** This information is made available in case an application wants to know about any non-critical issues that were automatically solved by the decoder; for example, an invalid optional IE that was discarded in accordance with 3GPP technical specifications.
- Each warning message generated by the decoder includes the **complete path of the field** associated with the warning, if applicable.

Interpretation of the value of certain fields

- For certain fields, the NAS protocol specifies that some of their values must be interpreted as a certain other value. That is, when the decoder encounters a field for which the NAS protocol specifies a set of values **to be interpreted** as a certain **replacement value**, the decoder checks the value of the field against those values. When there is a match, the decoder **replaces** the value of the field with the specified **replacement value** and adds a **warning message** to the warning message buffer.

Features that support diagnostics and testing of NAS messages

Convert, compare, and deep copy messages

API functions are available that will

- **Convert** an **encoded** NAS message to **XML, JSON, or text.**
- **Convert** an **unencoded** message to **XML, JSON, or text.**
- **Compare** two **unencoded** NAS **messages** and report whether they are different.
- **Create** a **deep copy** of an unencoded NAS message.

Create, edit, and display NAS messages

- A **GUI-based application** is available as a separate product (NAS-1Step).