



Strategy for maintenance and
release of NES deliverables



Strategy for maintenance and release

| | | |
|-----|--|---|
| 1 | The overall strategy | 2 |
| 2 | Versioning..... | 2 |
| 2.1 | Subject to versioning..... | 2 |
| 3 | Responsibilities | 3 |
| 3.1 | The role of the working group..... | 3 |
| 3.2 | The role of the Steering Committee..... | 3 |
| 4 | Co-existence of different releases | 3 |

Strategy for maintenance and release

1 The overall strategy

The challenge is to balance the need for stability vs. new functionality. Stability is needed to support implementation and practical use. New functionality is needed to support requirements of current and future participants.

New functionality should be released on a scheduled basis. No more frequent than every 6 months. There should be a 3 month advanced notice, including a public review.

Error corrections need to be released as soon as possible.

The aim is to track changes and signal compatibility to previous releases. The changes to previous releases will be announced as “release notes”.

2 Versioning

NES will adopt a straight forward version number is used consisting of
<major>.<minor>[.<revision>]

- Major is used to denote a major change, i.e. not backwards compatible
- Minor is used to denote a minor change, i.e. a change that is backwards compatible
- Revision may be used for changes that does not effect compatibility, i.e. for changes to supporting documentation.

“Backwards compatible” implies that what was allowed in the previous release is still valid in the new release (e.g. can not add new mandatory features, an existing mandatory feature can be made optional). Compatibility is measured against the immediate preceding publication

An example of applying this regime may look as follows:

- 1.0 Original publication
- 1.0.1 Changes to documentation in order to improve clarity
- 1.1 Adding a new optional feature
- 1.1.1 Changes to documentation in order to improve clarity
- 2.0 Adding a new mandatory feature
- 2.1 Adding a new optional feature
- 2.1.1 Documentation change

2.1 Subject to versioning

The XML instances exchanged contains the following mandatory elements used to determine the version:

- UBLVersionID - 2.0
- CustomizationID - NES
- ProfileID - urn:www:nesubl:eu:profiles:profile1:ver1.0

Strategy for maintenance and release

The implication is that a NES conformant instance is linked to a particular version of a NES profile.

Furthermore guidelines and code lists are defined as “a normative part of NES”. NES guidelines and code lists are applicable across all profiles, but cannot be separately versioned in an XML instance.

Therefore NES deliverables will be versioned as a complete package. A NES version = NES Profiles + NES guidelines + NES code lists

3 Responsibilities

3.1 The role of the working group

It is the responsibility of the NES Working group to recommend new releases when it is viewed necessary. The reason for a new release can be the need for new functionality or error correction.

The working group should make the necessary descriptions to the Steering group of the suggested changes and explain the reason for making them. The group should also make recommendations to the steering group on what kind of versioning it should be (major, minor or revision). The working group shall after the public review send the suggested changes to the steering group with necessary explanations at least two week in advance before the steering group takes the decision.

3.2 The role of the Steering Committee

It is the responsibility of the NES Steering group on the basis of the recommendation from the working group to decide on the need for a new release and to determine what kind of versioning it should be. After the steering group has decided a new release it shall be published on the NES web site.

4 Co-existence of different releases

There can be no more than two releases of NES in play at the same point in time. This can be illustrated as in the figure below:

