**Maintenance Change Request**

**for the update of ISO 20022 financial repository items**

## Name of the request:

ISO 20022 Business Messsage Envelope - Maintenance 2024/2025

## Submitting organisation(s):

Swift

Standards and MICSS,

Avenue Adele, 1

1310 La Hulpe – Belgium

On behalf of the ISO 20022 Technical Support Group (TSG)

## Related messages:

Under this maintenance, below existing ISO 20022 message definitions will be maintained (resulting from the impact analysis performed on each CR).

**Business Message Envelope:**

|  |  |
| --- | --- |
| nvlp.001.001.01 | BusinessMessageEnvelopeV01 |

## Commitments of the submitting organisation:

The submitting organisations confirm that they can and will:

* undertake the development of the new version of the candidate ISO 20022 message models that it will submit to the RA for compliance review and evaluation. New valid Message Definition models will be made available to the RA by December 1.
* provide a new version of part 1 of the related Message Definition Reports (MDR) by December 1, and new examples of valid message instances of each candidate message (only when valid samples were published for current version) by May 1 at the latest.
* address any queries related to the description of the new models and messages as published by the RA on the ISO 20022 website.

Swift intends to implement the above new version on its SwiftNet network once the related documentation has been published by the RA.

The submitting organisations confirm their knowledge and acceptance of the ISO 20022 Intellectual Property Rights policy for contributing organisations, as follows.

*“Organisations that contribute information to be incorporated into the ISO 20022 Repository shall keep any Intellectual Property Rights (IPR) they have on this information. A contributing organisation warrants that it has sufficient rights on the contributed information to have it published in the ISO 20022 Repository through the ISO 20022 Registration Authority in accordance with the rules set in ISO 20022. To ascertain a widespread, public and uniform use of the ISO 20022 Repository information, the contributing organisation grants third parties a non-exclusive, royalty-free license to use the published information”.*

## Contact persons:

Jamie Osborne – Swift - MICSS - Standards Reusable Services, jamie.osborne@swift.com

Vincent Kuntz – Swift Standards, vincent.kuntz@swift.com

# Change request CR1407: Retype the References MessageElement

Sections A to G are copies of the of the change request approved by the TSG for further consideration as communicated to the submitting organization by the RA. The actual implementation as proposed by the submitting organization are defined in sections H and I.

## Origin of the request:

*A.1 Submitter*: Swift / ISO 20022 TSG.

*A.2 Contact person:* Jamie Osborne, Jamie.osborne@swift.com

*A.3 Sponsors: Swift / ISO 20022 TSG.*

## Related messages:

BusinessMessageEnvelopeV01 (nvlp.001.001.01)

## Description of the change request:

*Retype the References MessageElement with an ExternalSchema type.*



Figure 1: Standards Editor view of the change - the External Schema type is named “BusinessMessageMetadata”.

This remodelling supports backwards compatibility with nvlp.001.001.01 (by allowing users to enforce strict parsing against <https://www.iso20022.org/standardsrepository/type/Reference22>) while reducing the complexity of the resulting XML schema from 43 types to 6 as shown in the included schema. This schema is also five times smaller, and exponentially simpler to comprehend and use.

**Note:** This proposal removes the second Supplementary Data element in the Reference element (BizMsgEnvlp/Ref/SplmtryData), allowing deletion of the now redundant SupplementaryDataAndDocumentRule Textual Rule at the root element.

Note also that this proposal may be simplified even further by typing Reference with the same existing ISO 20022 LaxPayload generic external schema type as already used the Header and Document elements as shown here:



Figure 2: Alternative modelling using existing LaxPayload generic ExternalSchema type

Although that approach is certainly simpler, we propose to create a dedicated type based on LaxPayload because this allows us to add a further layer of definition and detail at the level of that type as shown.



Figure 3: Creating a dedicated ExternalSchema type is often done for ISO 20022 because it allows additional information to be provided to implementers.

This approach mimics what ISO does for the Supplementary Data Envelope, for example, which specialises an analogous lax ExternalSchema type with further detail to help implementers understand how to use that type, as shown.



Figure 4: The proposed change mimics how other ISO 20022 types like Supplementary Data are modelled.

This approach also makes it easier for users to restrict the BME-specific type-only with a more restrictive namespace-based restriction, as is done in the BAH Signature Element, which also adds more specific descriptive detail in the Documentation for the type:



Figure 5: The proposed approach simplifies and clarifies for communities who would like to further restrict the type to their specific use case.

The separate, explicit type also clarifies and encourages implementers to refine that specific type as desired, using common practices and tooling. Here is an example showing how TARGET has done this to the signature element in their RTGS Business Application Header. Here TARGET uses MyStandards to change it from lax to strict processing.



Figure 6: An example how a similar type is restricted by Target, using a tool like MyStandards.

## Purpose of the change:

The ISO 20022 Business Message Envelope (BME) closes an important gap in the ISO 20022 standard and opens many opportunities to improve end-to-end interoperability, transparency, and performance for users of the standard. It also offers an ISO 20022-compliant framework for communities to build valuable new adjacencies to their ISO 20022 messaging. It is therefore a welcome addition to ISO 20022.

However, because the BME is part of the business payload of an ISO 20022 Business Message, when present, it must be supported and transported end-to-end through the entire transaction. As the reader understands, an intermediary processing step may not drop information that is present in the Business Payload because it is unable to parse/process it.

Although it is part of the Business Payload, a technical processing envelope like the BME will be processed by far more technical, lower-level middleware than a typical ISO 20022 Business Message. While a typical ISO 20022-compliant Payments Application is expected to support a huge library of multiple versions of pacs, pain, camt, acmt, business standards, and to maintain these year on year, this is not the case for middleware and other systems that will interact with the BME.

For this reason, technical envelopes such as the BME must be flexible, while remaining as technically simple to process as possible. If data is present that an intermediary processing step does not understand or need to support its intermediary processing, then it must be as simple and efficient as possible to simply ignore and pass that data through.

Version 01 of the ISO 20022 Business Message Envelope (nvlp.001.001.01) is unnecessarily complex and difficult to support and maintain because it contains an explicitly modelled references element that is far too extensive and specific for a simple technical envelope. We know from more than 20 years of experience that ISO 20022 implementation and maintenance expense is directly related to the complexity of an ISO 20022 message – As the amount of formally modelled information in an ISO 20022 message grows, so does the effort and expense to support it.

This fact was highlighted when a CR was submitted against the BME before it was even one year old, and before Swift had even completed our review – the CR proposed an entirely new version of the BME simply because payments messages had begun to support a new version of the PostalAddress MessageComponent.

This CR highlighted the challenges with the current nvlp.001.001.01 Reference model – Not only does it impose a requirement to be able to process a complex structure on users of this element, it imposes a structure that contains data that is entirely irrelevant to a technical envelope (for example, postal addresses and the date and place of birth of individual persons) – This data can change for reasons that have nothing at all to do with any reasonable BME use-case.

Swift supports the original design intent of the references element, which is to provide a flexible mechanism for intermediaries to optionally introduce simple markup to the message to support various kinds of business tracking, auditing, etc, purposes. We do not feel, however, that ISO is ready to standardise these use cases, nor to restrict what users might want to put in this element. We can confidently say however, that we find no reasonable use case for the kind of data that is modelled in nvlp.001.001.01 [Reference22](https://www.iso20022.org/standardsrepository/type/Reference22) to be copied as-is into the BME from a business message such as a pacs.008. There can therefore be no reason to tightly couple these layers by sharing data elements.

After reviewing extensively with implementation experts across Swift and within our community, we felt that it was too risky to support the BME V01 because it imposes a requirement on Swift to be able to parse and honour the references end-to-end. The implementation cost throughout the Swift ecosystem is enormous and it cannot be justified, especially considering the potential for further changes like the aforementioned PostalAddress change.

Swift therefore requests that the formally modelled element is replaced with a simple, external schema that communities may formalise independently, where required, to support their use cases.

This design opens the possibility that communities may still submit models to formalise the contents of the references element to ISO as part of a less formal and restrictive submission process.

## Urgency of the request:

Swift recommends that this change is implemented as soon as possible – as mentioned above, the BME is a long overdue, welcome addition to the standard. Swift has identified BME use cases that we would like to explore, but nvlp.001.001.01 is not generally useful as-is for the reasons outline above.

Swift also believes that this change should be made before users begin to seriously consider implementing version 01. Although this proposal is backwards compatible with version 01, if we leave this update until after users have begun to implement, the early adopters will face unnecessary friction and maintenance burden when interoperating with other sub-systems that use version 02.

## Business examples:

Examples illustrating the change request provided separately to the TSG.

## SEG/TSG recommendation:

*This section is not to be taken care of by the submitter of the change request. It will be completed in due time by the SEG(s) in charge of the related ISO 20022 messages or the TSG for changes related to the BAH.*

|  |  |  |
| --- | --- | --- |
| **Consider** | X | **Timing** |
|  | - **Next yearly cycle: 2024/2025**(the change will be considered for implementation in the yearly maintenance cycle which starts in 2024 and completes with the publication of new message versions in the spring of 2025) | X |
|  | - **At the occasion of the next maintenance of the messages**(the change will be considered for implementation, but does not justify maintenance of the messages in its own right – will be pending until more critical change requests are received for the messages) |  |
|  | - **Urgent unscheduled**(the change justifies an urgent implementation outside of the normal yearly cycle) |  |  |
|  | - **Other timing:** |  |

Comments:

|  |  |
| --- | --- |
| **Reject** |  |

Reason for rejection:

## Impact analysis and type of impact:

The impact analysis confirms that only the BusinessMessageEnvelope (nvlp.001) is impacted by the Change Request.

## Proposed implementation:

Based on the requirements to simplify the Business Message Envelope, we propose following implementation:



Compared to previous version, following changes are applied:

* Removal of the unnecessary SupplementaryData element to allow for easier standardisation of the implementation – all implementation will have to use the new Reference element, which can cover also the SupplementaryData requirements
* Creation of a simplified ReferenceData extension element composed of an optional Type and a Data <any> component.

**Question to the TSG: should the Type be made mandatory or kept optional?**

Below example illustrates the implementation of a V01 element in the new structure for backward compatibility illustration:



## Proposed timing:

The submitting organisation confirms that it can implement the requested changes in the requested timing.

|  |  |
| --- | --- |
| Timing | Maintenance 2024/2025 |

## Final decision of the SEG(s):

|  |  |
| --- | --- |
| Approve | X |

**Comments from the 11 December 2024 TSG meeting:**

* Type definition is unclear and should be updated to cater for an external code set for the type.
* For backward compatibility purpose, it was however suggested to keep a proprietary element typed by Max350Text.
* Swift to make an updated proposal that includes those changes for the next TSG meeting.

**Response from Swift:**

* + We acknowledge the TSG proposal to update the reference element and have updated the proposal as requested. However, SWIFT does not support the use of an external code set for the Type/Code element in the BME.
	+ This could significantly impact existing implementations when the external code set is updated or extended with new codes, even if no changes are made to the current implementation.
	+ We recommend reverting to the original implementation proposal. If this is not acceptable to the TSG, we are open to considering the following alternative implementation proposal, which removes the need for an external code set:



**Comments from the 08 January 2025 TSG meeting:**

* Following assessment of the different implementation proposal, the TSG agreed with the original implementation proposal as suggested by Swift.
	+ The Type element should remain optional, and typed by the Max350Text
	+ The TSG also agreed that a list of valid codes that can/may be used and are “standardised” and the possibility to support a URL or an Xpath for backward compatibility reasons should be documented in the BME MUG to avoid proliferation of different implementations
* Final implementation is approved as following:



|  |  |
| --- | --- |
| Reject |  |

Reason for rejection: