

Standard Business Reporting

Australian Taxation Office

ATO ebMS3 Implementation Guide

Date: 7th November 2019

Status: Final – suitable for use

* This document and its attachments are **Unclassified.**

 For further information or questions, contact the SBR Service Desk at [SBRServiceDesk@sbr.gov.au](http://sbr.gov.au/agreement/Gateway/1.0/Push/PKI) or call 1300 488 231.

International callers may use +61-2-6216 5577

VERSION CONTROL

| **Version** | **Date** | **Description of changes** |
| --- | --- | --- |
| 1.4 | 7/11/2019 | Added new sub-section under Section 2.3 SBR ebMS3 Supported Service Invocation Types  **2.3.2 Pull Response Guidelines**  Included information on pulling a response.  Added 2 new sub-sections 5.3.1.1 and 5.3.1.2 under Section 5.3.1 Error Messages  **5.3.1.1 Business Rule Messages**  No change to the existing content.  **5.3.1.2 Business Processing Messages**  Added information about business processing messages. |

***Note:*** *Previous Version Control information is located in Section 6 of this document.*ENDORSEMENT

APPROVAL

Chief Solutions Architect

Standard Business Reporting

Michael Ferris Project Manager

Strategic Web Services

Australian Taxation Office

Copyright

© Commonwealth of Australia 2019 (see exceptions below).   
This work is copyright. Use of this Information and Material is subject to the terms and conditions in the “SBR Disclaimer and Conditions of Use” which is available at [http://www.sbr.gov.au](http://www.sbr.gov.au/software-developers/enabling-sbr-in-my-application/sbr-taxonomy/view-taxonomy). You must ensure that you comply with those terms and conditions. In particular, those terms and conditions include disclaimers and limitations on the liability of the Commonwealth and an indemnity from you to the Commonwealth and its personnel, the SBR Agencies and their personnel.

You must include this copyright notice in all copies of this Information and Material which you create.  If you modify, adapt or prepare derivative works of the Information and Material, the notice must still be included but you must add your own copyright statement to your modification, adaptation or derivative work which makes clear the nature of your modification, adaptation or derivative work and you must include an acknowledgement that the adaptation, modification or derivative work is based on Commonwealth or SBR Agency owned Information and Material. Copyright in SBR Agency specific aspects of the SBR Reporting Taxonomy is owned by the relevant SBR Agency.

Table of contents

[1 Introduction 5](#_Toc23333025)

[1.1 Purpose 5](#_Toc23333026)

[1.2 Audience 5](#_Toc23333027)

[1.3 Document context 5](#_Toc23333028)

[2 ATO SBR ebMS3 instructions 6](#_Toc23333029)

[2.1 Request Messages Types 6](#_Toc23333030)

[2.2 Supported Data Formats 8](#_Toc23333031)

[2.3 SBR ebMS3 Supported Service Invocation Types 8](#_Toc23333032)

[3 SBR ebMS3 Message Packaging 14](#_Toc23333033)

[3.1 Overview 14](#_Toc23333034)

[3.2 Single Request 15](#_Toc23333035)

[3.3 Single Receipt 17](#_Toc23333036)

[3.4 Single Pull Request 17](#_Toc23333037)

[3.5 Single Response (Non-Collect) 18](#_Toc23333038)

[3.6 Collect Response 20](#_Toc23333039)

[3.7 Batch/Bulk Request 22](#_Toc23333040)

[3.8 ELS Tag Batch Request 25](#_Toc23333041)

[3.9 Batch/Bulk Receipt 26](#_Toc23333042)

[3.10 Batch/Bulk Pull Request 26](#_Toc23333043)

[3.11 Batch/Bulk Response 26](#_Toc23333044)

[4 SBR ebMS3 Message Structure 29](#_Toc23333045)

[4.1 Security Header 29](#_Toc23333046)

[4.2 ebMS Header 29](#_Toc23333047)

[4.3 eb:USERMESSAGE – SBR ebMS3 Profile 29](#_Toc23333048)

[4.4 eb:SIGNALMESSAGE – ATO Profile 36](#_Toc23333049)

[5 General Instructions 38](#_Toc23333050)

[5.1 Anonymous Interactions 38](#_Toc23333051)

[5.2 Authorisation of online (cloud) service providers 38](#_Toc23333052)

[5.3 Response Messages 39](#_Toc23333053)

[6 Previous Version Control 47](#_Toc23333054)

# **Introduction**

Purpose

The purpose of this document is to provide information that will assist software developers in the implementation of calls to the web services offered by the Australian Taxation Office (ATO) through the Standard Business Reporting (SBR) ebMS3 platform.

Audience

The audience for this document is any organisation that will be building any ATO SBR services into their products. Typically this will be software application developers.

Document context

The first phase of the SBR program provided a platform that offers a collection of core services that are part of the implementation of the SBR initiative to simplify Business to Government reporting obligations.

Whilst that platform is currently still available for use, the next phase of the SBR program involves building and offering new services that are based upon the ebMS3/AS4 messaging standard.

The new SBR services differ from the previous SBR services mainly in the following ways:

* Messaging is based on the ebMS3 standard and AS4 Conformance Profile
* The addition of support for batch and bulk interactions
* The addition of support for asynchronous single interactions
* The addition of support for a wider range of reporting obligations

This document provides guidance for construction of request messages for SBR ebMS3.

Request messages that are targeted for SBR Core Services must be constructed using the Standard Business Document Format in accordance with the instructions below that are specified as being for SBR Core Services.

Request messages that are targeted for SBR ebMS3 must be constructed using the ebMS3 Message Format in accordance with the instructions that are specified as being for SBR ebMS3.

# **ATO SBR ebMS3 instructions**

As defined in SBR ebMS3 Web Services Implementation guide, “Interaction” is the combination of the Service and Action invoked by an (external client) BMS (“BMS Invokable Interaction”). For example: LodgeCTR.001.00, ListAS.001.00, AddClientRole.001.00.

Every SBR ebMS3 Interaction that can be invoked by an (external client) BMS is defined to have certain attributes that indicate how it is supported by the eCommerce Platform. These attributes, known as “Service Attributes” are:

* Request Message Type,
* Response Time Service Level, and
* Invocation Mode

## Request Messages Types

This section provides a high level overview of the compositions defined by the SBR ebMS3 Message Types.

In this document, a “Logical Record” is defined as the structured business request data that must be submitted for a single invocation of a particular BMS Invokable Interaction. Every Request Message sent to the ATO eCommerce Platform must contain at least one Logical Record.

For more information on the data, structure and order of message components, refer to Section 3 SBR ebMS3 Message Packaging. The SBR ebMS3 channel accepts three distinct request message types:

### Single Request

A Single Request Message must contain only one Logical Record for a specific Interaction e.g. ListAS, ValidateFBT, LodgeCTR.

A single request can either be submitted synchronously or asynchronously.

A single request can be sent containing either a “Service Request” (e.g. retrieve a list of accounts), a Standalone Form, or a Base Form with Optional Schedules:

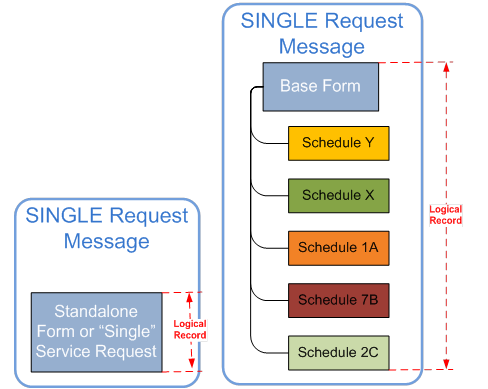


Figure : Single Request Message Composition

Additionally, the Single Message Type logically has a sub-type called “Collect” – which is a special type of Single Request Message that is used to collect information or communications made available by an Australian Government Agency such as the ATO.

### Bulk Request

A Bulk Request Message contains one Logical Record that is a multi-level construct comprised of a Parent (e.g.: Payer) and one or more Children (e.g.: Payees). The “Bulk” information is provided in the Children which are all of the same type and all relate to the Parent. In a Bulk Request Message each Child has a “business link” to the other children in that Request Message which is represented by the “Parent” e.g.: Private health insurance and member contribution statements. Like a batch, these requests can only be submitted in an asynchronous interaction pattern.

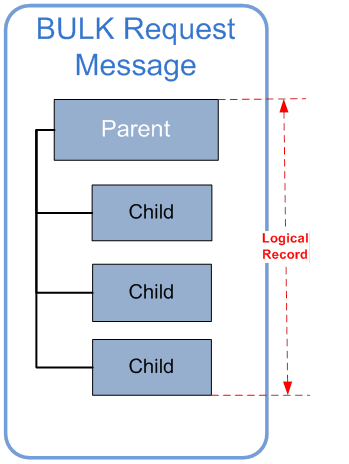


Figure : Bulk Request Message Composition

#### Batch Request

Batch messages serve as a container for multiple Logical Records. A Batch Request Message may contain multiple Logical Records of the same type (e.g.: Four LodgeCTRs) to be sent in one transmission, thereby facilitating what is effectively multiple invocations of an interaction using the one Request Message. A Batch Request Message can be one of 3 sub-Types:

* Batch of Standalone Forms or Service Requests
* Batch of Base Forms with Optional Schedules
* Batch of Bulk Requests

Note that the Logical Records in a Batch Request must all be of the same type. All batch requests are asynchronous.

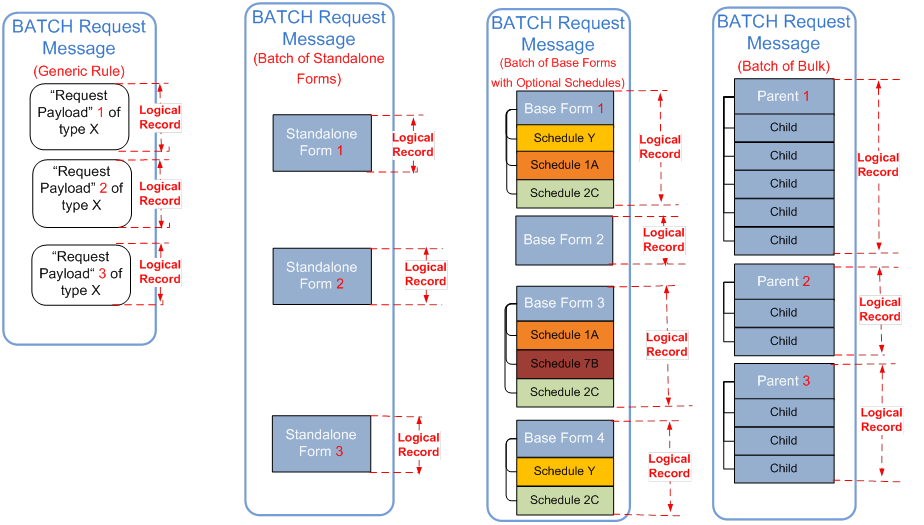


Figure : Batch Request Message Composition

Supported Data Formats

Each Logical Record may be expressed using one of the following data formats:

|  |  |
| --- | --- |
| **Data Format Abbreviation** | **Data Format** |
| XBRL | Extensible Business Reporting Language |
| JSON | JavaScript Object Notation |
| XML | Extensible Markup Language |

Table : Supported Data Formats

For information on the data format used for a particular interaction please refer to the ATO Service Registry.

SBR ebMS3 Supported Service Invocation Types

ATO supports all the invocation types as described in the SBR ebMS3 WIG. The ATO Service Registry describes the relevant invocation types for each ATO service implemented on the SBR ebMS3 platform.

### Polling Interval

The following sub-sections relate to the standard application of Polling intervals. Where a Service Action varies from the standards below, the intervals can be found in the ATO SBR Service Registry.

#### Pull-only Polling

The pull request messages are effectively used for polling for response messages.

For asynchronous requests the BMS shall poll for the response after a specific time interval. This section outlines the polling pattern for various asynchronous exchanges. The purpose for these directives is to police polling intervals via appropriate guidelines to ensure the service is not overloaded by requests. Polling intervals only applies to asynchronous interactions. As for synchronous interactions the BMS will halt the thread and wait for the response.

The time frames given in seconds and hours below are just indicative and not actual time frames.

##### Single

1. Initial attempt after 10 seconds.
2. Second attempt after 20 seconds. I.e. Add 10 seconds for the second attempt.
3. Third attempt after 40 seconds. I.e. double the second attempt.

* Continue to double the time frame for each subsequent poll.
* Final poll at 180 seconds, as there is a timeout after 3 minutes for single asynchronous requests.

**NOTE:** If the maximum timeframe is reached without receiving the response a timeout would occur and a user message response will be generated and returned stating that there was an unexpected error and if the problem persists, contact the agency support.

##### Intermediate Batch

1. Initial attempt after 30 seconds + 10 seconds for each document transmission.
2. Second attempt after the same time frame as step 1. I.e. if the attempt above is a total of 50 seconds, then the second poll can be repeated after another 50 seconds.
3. For the third attempt, double the time frame of the previous attempt. I.e. Poll after 100 seconds.
4. Continue to double the time frame for subsequent attempts.

**NOTE:** If a validation response is returned, reset the time and start again at step 1. Shouldn’t generally be required to poll after 1 hour since intermediate SLA is capped at 1 hour.

##### Delayed Batch/Bulk

1. Initial attempt after 1 hour + [10 seconds for each document in transmission].
2. Second attempt after the same time frame as step 1. For example : If there were two documents in transmission, the the first attempt would have occurred after 1 hour + 20 seconds. Therefore the second poll can be repeated after another 1 hour + 20 seconds.
3. For the third attempt, double the time frame of the previous attempt. I.e. For the above example, poll after 2 hours and 40 seconds.

* Continue to double the time frame for subsequent attempts until the polling interval reaches 24 hours, after which doubling can stop and polling can continue on a once per day interval.

**NOTE:** If a validation response is returned reset the time and start again at step 1.

#### Intermediate Collect (“Push-and-Pull”) Polling

Collect Polling differs from the above “Pull-only Polling” in that the “poll” actually uses a two-way/push-and-pull message exchange pattern rather than just a one-way/selective-pull message exchange pattern.

In the case of Collect Polling a BMS must:

1. send a Push request for which it should receive a receipt (signal) message.
2. send a Pull request for which it may receive either:
   1. an **ebMS3 User Message** that will contain a **business response** that may either:
      1. contain the item sought to be collected; or
      2. information advising that the item to be collected is not yet available for collection or that there are no items to collect.

**OR**

* 1. an **ebMS3 Signal Message** that is an **error** indicating that there is no business response eplaced at all at that point in time.

If the response to the Pull request contains an **ebMS3 User Message** that does not contain the item sought to be collected then upon delivery of that ebMS3 User Message the two-way/push-and-pull message exchange will be complete and in order to make another “Collect Polling” attempt the BMS must initiate another two-way/push-and-pull message exchange to “poll” for the business response.

If the response to the Pull request is an ebMS3 Signal Message then it will be necessary for the BMS to engage in “Pull-only” polling (as described above) until it receives an ebMS3 User Message in response.

Therefore, it becomes necessary to provide polling interval guidance at two different levels:

1) **Collect Polling** (i.e. how often should a two-way/push-and-pull be initiated in order to collect a specific item\*); and

2) **Pull-only Polling** (i.e. once the Push part of Collect Polling has been completed, how often should the Pull part of Collect Polling be attempted).

\*This guidance for the timing of Collect Polling shall not preclude polling for two different business responses within a polling interval. For example , if a BMS polls for report A it can also poll for report B within the polling interval if either:

1. report A is of a different type to that of report B;
2. report A or report B is an “on-demand”/requested report;

##### Collect Polling

1. Initial attempt as indicated in the product specific message implementation guide related to the item to be collected .
2. Second attempt after the same time frame as step 1.
3. For the third attempt, double the time frame of the previous attempt.
4. Continue to double the time frame for subsequent attempts ongoing.

##### Pull-only Polling within Collect Polling

1. Initial attempt 5 minutes after receipt of the corresponding Push request ebMS3 receipt signal message.
2. Second attempt after the same time frame as step 1.
3. For the third attempt, double the time frame of the previous attempt.
4. Continue to double the time frame for subsequent attempts ongoing.

When polling, the time frame occurrence is once a day. Doubling can stop and polling can continue on a once per day interval.

### Pull Response Guidelines

SBR ebMS3 channel is configured to allow pulling a response only once. This is applicable to both pull-only polling and intermediate collect polling patterns. Where there is more than one response (e.g. a batch of over 1000 requests will have a validation response and a second validation and business response), each response can be pulled only once.

### Submission Guidelines

**Note:** Software products should not multi-thread requests into SBR, unless each thread is being initiated via a direct end-user request. It is not allowable for software products if they have a stockpile of transactions to automatically within their software to spawn multiple threads to push them all through concurrently into SBR.

| **Message Type** | **When to Use** | **When not to Use** | **Consequence of Message Type** | **Example** |
| --- | --- | --- | --- | --- |
| Single Sync or Single Async | Where single requests are being initiated by end users and a timely response is critical | When requests have been stockpiled by BMS, in this instance this stockpile should be sent as a batch request | If unexpected errors are encountered then the client will be returned a message typically along the lines of unexpected error has occurred please retry and if problem exists contact ATO. This means that the ownership belongs on the client to retry and contact ATO. | BAS agent lodges an Activity Statement (AS) when the client is present and receives an immediate response. |
| Where single requests are being system initiated and a timely response is critical to the system behaviour | For requests which are large in size, in this instance these requests should be sent as a bulk requests |
| Batch | When single requests for different entities (same product and service action) have been stock piled for ‘end of day processing’ | For requests which are large in size, in this instance these requests should be sent as a bulk requests | As long as the push of the message has occurred successfully; because the Batch/Bulk capability has state then if any unexpected errors occur then the ownership of resolving any issues reside with ATO to ensure that the appropriate response is generated so that the pull request will get a business response.  Note: The client still has the responsibility to check the response to ensure that there were no validation errors etc. | A Tax Agent, as part of end of day processing, submits multiple Company Tax Return (CTR) with schedules for different clients and will review the result in the next morning. |
| Bulk | When single request with multiple data records belonging to the same entity that is large in size | For requests which are not large in size, in this instance these requests should be sent as a batch requests | Bulk – A supplier submits a Taxable Payments Annual Report (TPAR) for a single client (payer) and will review the result in the next morning.  Batch of bulk – A supplier as part of end of day processing, submits Taxable Payments Annual Reports (TPAR) for several clients and will review the result in the next day. |

Table : Submission Guidelines

# **SBR ebMS3 Message Packaging**

Overview

### General

All user messages sent to the ATO MUST conform to the minimum standards User Message structure specification in the SBR ebMS3 WIG.

Document metadata is captured differently between Single and Batch/Bulk messages due to differences in message structure.

Messages to and from SBR MUST be packaged using the SOAP messages with Attachments (SwA) standard. This section outlines the SwA message packaging format for various message exchange patterns supported by SBR.

Request Messages will basically comprised of:

* a Header; and
* a Payload.

Payloads are made up of one or more Logical Records.

For details on how this information applies to specific requests, refer to the Product Specific MIGs.

### Supported Attachments

The table below outlines supported attachment file types and MIME types where an ATO ebMS3 service allows attachments. If a service allows attachments yet only a subset, this will be called in the ATO Service Registry.

| **MIME Type** | **Filename extension** |
| --- | --- |
| application/msword | doc |
| application/pdf | pdf |
| application/rtf | rtf |
| application/vnd.ms-excel | xls |
| image/tiff | tif |
| image/jpeg | jpg |
| image/bmp | bmp |
| image/png | png |
| image/gif | gif |
| application/vnd.ms-project | mpp |
| application/vnd.ms-powerpoint | ppt |
| application/vnd.openxmlformats-officedocument.wordprocessingml.document | docx |
| application/vnd.openxmlformats-officedocument.wordprocessingml.template | dotx |
| application/vnd.openxmlformats-officedocument.spreadsheetml.sheet | xlsx |
| application/vnd.openxmlformats-officedocument.spreadsheetml.template | xltx |
| application/vnd.openxmlformats-officedocument.presentationml.presentation | pptx |
| application/vnd.openxmlformats-officedocument.presentationml.template | potx |
| application/vnd.openxmlformats-officedocument.presentationml.slideshow | ppsx |
| application/vnd.oasis.opendocument.text | odt |
| application/vnd.oasis.opendocument.spreadsheet | ods |

Table : Supported Attachments

Single Request

Both synchronous and asynchronous single requests are requests which contain a Payload of only one Logical Record i.e. a single invocation is being requested for a specific service action e.g. ListAS, ValidateFBT, LodgeCTR and etc.

A Logical Record may however, be comprised of one or more “Logical Documents”.

For a Single Request Payload the business management software SHALL create a separate physical document for each Logical Document that forms part of the Single Request Payload.

Each of those physical documents SHALL be packaged as a separate ebMS3 MIME part in the ebMS3 user message.

However, the first MIME part, which must contain the ebMS3 header and the SOAP body SHALL contain an empty SOAP body and an ebMS3 header as per the guidelines in section 4.2.

### Header – ebMS3 Header Content

ebMS3 header content is detailed in Section 4 SBR ebMS3 Message Structure.

**NOTE for base forms and schedules:** The base form and each schedule should have a corresponding PartInfo Node in the header section. If there are no schedules, only PartInfo Node 1 for the base form should be populated.

### Payload – ebMS3 Request MIME Part(s) Content

#### Single Non-Collect Request

In a Single Non-Collect Request, a “Logical Record” and a “Payload” are equivalent since there is only one Logical Record. The Payload for Single requests can contain any one of the following:

1. a Base ‘Form’ with a number of attached schedules. Each Document within the Logical Record must have a corresponding PartInfo Node in the ebMS3 header of the message. Each schedule should be placed in a subsequent separate MIME Part which is shown as “MIME Part 3” to “MIME Part X” in the figure below.
2. a standalone (Base) Form; or
3. a “Service Request” (e.g. a request to retrieve a list of accounts).



Figure : Single Non-Collect Request ebMS3 MIME Part Content

#### Collect Request

A Collect Request is a special type of Single Request. MIME Part 2 may contain either:

1. no payload i.e. empty MIME Part;

OR

1. one payload/logical record to request to collect one item (e.g. one report) only (note that if there is a Collect Request Payload it must be prefixed by a Delimiter).



Figure : Collect Request

Single Receipt

The SBR ebMS3 WIG provides the specification for the single receipt structure.

Single Pull Request

The SBR ebMS3 WIG provides the specification for the single pull request structure.

Single Response (Non-Collect)

The ebMS3 message containing the response to a single non-Collect request will be an ebMS user message comprised of multiple MIME parts.

The first MIME part will contain an empty SOAP body and an ebMS header structure as per the guidelines in section 5.3.

The second MIME part will contain an Overall Event Message Block comprising of

* Processing Steps completed
* System and Processing Errors

The event structure is further explained in the SBR ebMS3 WIG.

The third MIME part will contain a Business Event Message Block (following the event structure) comprising of either:

* Validation Report for the Logical Record in the request message[[1]](#footnote-2), or
* Errors/Warnings combined with Validation Report for the logical record[[2]](#footnote-3) in the request message.

This MIME part is not returned if the request fails prior to reaching the validation stage (e.g. unexpected system error).

Please refer to product-specific MIGs to find out the exact contents of this MIME part.

The fourth MIME part is optional and may contain a Business Response for the logical record[[3]](#footnote-4). Some Request Messages, for example those that are only validated (i.e. most ‘Validate’ service actions) or that are not expected to return any “business data” will not include a Business Response.



Figure : Single Response Message Packaging

## Collect Response

The ebMS3 message containing the response to a Collect request will be an ebMS user message comprising of two MIME parts as shown in the following diagram:



Figure : Collect Response Message Packaging

### MIME Part 1

The first MIME part (MIME Part 1 in the above diagram) shall contain an empty SOAP body and an ebMS3 header structured as per the guidelines in the SBR ebMS3 WIG.

### MIME Part 2

The first “block” in MIME Part 2 will be an Overall Event Message Block comprising of request level processing information, including:

* Processing Steps completed
* System and Processing Errors

The event structure is further explained in the SBR ebMS3 WIG.

For a Collect response the rest of MIME Part 2 will contain:

1. An optional Business Event Message Block (following the event structure) comprising of either:

* Validation Report for the Collect Request, or
* Errors/Warnings combined with Validation Report for the Collect Request.

1. The business response comprised of the document or communication item that is sought to be collected from the ATO. The business response will be preceded by a meta-data record (a.k.a ‘Delimiter’).

In some cases the business response may be split into “Business Response Parts” (in order to improve manageability and scalability) and in such cases each Business Response Part will be prefixed by a Delimiter. These Delimiters will contain the information to allow assembly of the Business Response Parts into the complete Business Response (document or communication item). So for example, a complete list of all of the Superannuation Products and each of their respective details (as held by the ATO at a point in time) might be broken into multiple documents where each document (Business Response Part) contains the details for one Superannuation Product.

Batch/Bulk Request

### Overview

For Batch and Bulk requests the business management system (BMS) shall create an ebMS message comprising of two MIME parts as shown in the following diagram:



Figure : Batch/Bulk Request Message Packaging

### ebMS3 MIME Part 1 Content

#### ebMS3 Header

The first MIME part (MIME Part 1 in the above diagram) shall contain an empty SOAP body and an ebMS3 header as per the guidelines in Section 4 SBR ebMS3 Message Structure.

**NOTE:** If multiple bulk records are contained within a message, a.k.a ‘a batch of bulks’, document type should be ‘BATCH’.

#### WS-Security Headers

Population of these headers should be done in accordance with the SBR ebMS3 Web Services Implementation Guide.

### ebMS3 MIME Part 2 Content

All Batch/Bulk requests (base forms including schedules) are to be placed in a single MIME Part which is shown as “MIME Part 2”.

MIME Part 2 shall contain all the documents for the Batch/Bulk request. The BMS shall order documents such that the Parent document must appear immediately before its related child documents (see the darker green boxes marked as “Logical Document 1, 2, 3 etc.” in MIME Part2 in the above diagram).

The diagram below shows an example for how the documents should be packaged in Batch and Bulk requests:



Figure : Payload MIME Part for Batch and Bulk Requests

A ‘record delimiter’ (a.k.a. ‘metadata tag’) shall be inserted before each document. The record delimiter contains information similar to the PartProperties sections within the ebMS3 Header. This information is used to correctly identify and process records within a message. Each document must be preceded with a record delimiter. This carriage return must be inserted between each document and the immediately following separator record. An example record delimiter is provided below:

<Record\_Delimiter DocumentID=”234356657659655” DocumentName=”CTR” DocumentType=”BASE” RelatedDocumentID=”” />

The Message specific parts of the Payload Info for each of the documents (the base form and the associated schedules) should be populated as follows:

#### Base/Parent Records

|  |  |
| --- | --- |
| **Field** | **Value** |
| **For Each Delimiter** |  |
| Record\_Delimiter/@Name=DocumentID | Unique identifier applied to the document (cannot exceed 50 characters).  e.g. 999356657659678 |
| Record\_Delimiter/@Name=DocumentName | {BASE | PARENT PRODUCT NAME} e.g. CTR |
| Record\_Delimiter/@Name =DocumentType | BASE | PARENT  e.g. BASE |
| Record\_Delimiter/@Name=RelatedDocumentID | “”{BLANK} |

Table : Base/Parent Form Record Delimiter Properties

#### Schedule/Child Records

| **Field** | **Value** |
| --- | --- |
| **For Each Delimiter** |  |
| Record\_Delimiter/@Name=DocumentID | Unique identifier applied to the document (cannot exceed 50 characters).  e.g. 234356657659655 |
| Record\_Delimiter/@Name=DocumentName | {SCHEDULE | CHILD PRODUCT NAME} e.g. IEE |
| Record\_Delimiter/@Name =DocumentType | SCHEDULE | CHILD | BINARY  e.g. SCHEDULE  use BINARY when the schedule/child are base 64 encoded |
| Record\_Delimiter/@Name=RelatedDocumentID | The ID of the base or parent form to which the schedule or child record relates:  e.g. 999356657659678 |
| Record\_Delimiter/@Name=ContentType | This attribute should be passed only if Record\_Delimiter/@Name=DocumentType is set to BINARY.  Content type of the document.  e.g. application/pdf |
| Record\_Delimiter/@Name=Filename | This attribute should be passed only if Record\_Delimiter/@Name=DocumentType is set to BINARY  Filename for this document along with the extension e.g. Schedule1.pdf |

Table : Schedule/Child Record Delimiter Properties

ELS Tag Batch Request

For prior year returns in ELS tag format, the business management system (BMS) shall create an ebMS message comprising of two MIME parts.

The first MIME part shall contain an empty SOAP body and an emMS3 header that shall contain the ELS Approval Number.

The ELS Approval number is 5 digits and shall be provided by the BMS in the payload info section of the ebMS3 header. Any given ELS Approval number must be linked to the Registered Agent Number, which will be associated with the supplied credential (refer to the [ATO SBR Physical End Points](https://www.sbr.gov.au/sites/default/files/ATO-SBR-Physical-End-Points.docx) document). The ELS Approval number is then passed into AS4 path property where the Batch and Bulk Request Processor can construct this as a Batch and Bulk request.

The second MIME part shall contain the ELS request as an attachment for the Batch/Bulk request and conform to the ELS transmission format (zip file with TXID and return files) as per the ELS user guide.



Figure : ELS Tag Request and Response Variants

The previous figure shows an example for how ELS Request messages should be packaged in Batch and Bulk requests.

Batch/Bulk Receipt

The SBR ebMS3 WIG provides the specification for the batch/bulk receipt structure.

Batch/Bulk Pull Request

The SBR ebMS3 WIG provides the specification for the batch/bulk pull request structure.

Batch/Bulk Response

The ebMS3 message containing the response to a Batch/Bulk request will be an ebMS user message comprising of two MIME parts as shown in the following diagram:



Figure : Batch/Bulk Response Message Packaging

### MIME Part 1

The first MIME part (MIME Part 1 in the above diagram) shall contain an empty SOAP body and an ebMS3 header structured as per the guidelines in the SBR ebMS3 WIG.

### MIME Part 2

The first “block” in MIME Part 2 will be an Overall Event Message Block comprising of request level processing information, including:

* Processing Steps completed
* System and Processing Errors

“Processing Steps completed” contains statistical information that provides an overview of the results of processing of the corresponding request message. Definitions and descriptions of the values returned in this statistical information are provided in the SBR eMS3 WIG.

The event structure is further explained in the SBR ebMS3 WIG.

For Bulk and Batch responses the rest of MIME Part 2 shall contain the individual responses to each of the Logical Records in the request message. Each of those responses is comprised of:

1. A Business Event Message Block (following the event structure) comprising of either:

* A Validation Report for the corresponding logical record in the request message[[4]](#footnote-5), or
* Errors/Warnings combined with a Validation Report for the corresponding logical record[[5]](#footnote-6) in the request message.

Please refer to product-specific MIGs to find out the exact contents of Business Event Message Block.

1. [Optionally] A Business Response

There will be no Business Response for a corresponding Logical Record[[6]](#footnote-7) if the Request Messages are only being validated (i.e. most ‘Validate’ service actions) or are not expected to return any business data but instead just a processing result (e.g. some lodgements).

Each response to a Logical Record from the request message (i.e. the Business Event Message Block and the Business Response) within MIME Part 2 is prefixed by a separating meta- data record (a.k.a. ‘Delimiter’). These delimiters will contain the information to correlate the Business Event Message Block and the Business Response to a specific Logical Record from the corresponding request message.

### Number of Response Messages

A Batch/Bulk Intermediate request will generate two responses. They are:

* Acknowledgement Receipt supplied by SBR (See WIG for further details)
* Final Response – conforming to the above message packaging. It will contain both the Business Event Message Blocks and the Business Responses (where applicable).

A Batch/Bulk Delayed request will generate 3 responses. They are:

* Acknowledgement Receipt supplied by SBR (See WIG for further details).
* Intermediate Response – conforming to the above message packaging, but does not include any Business Responses. Also the Business Event Message Blocks will only contain Validation Reports. [[7]](#footnote-8)
* Final Response – conforming to the above message packaging. It will contain both the Business Event Message Blocks and the Business Responses (where applicable).

# **SBR ebMS3 Message Structure**

The message structure presented here details the ATO specifics of the general information provided in the SBR ebMS3 WIG. Only details that differ from that presented in the WIG are detailed here.

Security Header

Refer to the SBR ebMS3 WIG for more information.

ebMS Header

Refer to the SBR ebMS3 WIG for more information.

eb:USERMESSAGE – SBR ebMS3 Profile

### Overview

The following tables outline ATO specific values that are required within the ebMS3 header of a user message for SBR ebMS3 interactions in addition to those defined in the SBR ebMS3 WIG.

The use of double quotes in the specification of values in the following table is for delimiting purposes only and those double quotes do not form part of the value that the field should be set to.

#### eb:PartyInfo/eb:From

**The table** below shows the ATO specific children elements require for eb:From, and their use within requests and responses.

| **Name** | **Description** | **Single Request**  **2 Way Sync**  **1 Way Push** | **Bulk/Batch**  **1 Way Push** | **Response**  **2 Way Sync**  **1 Way Pull** | **Optionality** |
| --- | --- | --- | --- | --- | --- |
| eb:PartyId | String value that identifies a party. | The value of this element must be set to the corresponding party identifier for the party submitting the request message:  a) **Tax Agent Number (TAN)** where the party submitting the request is a tax agent;  OR  b) **Registered Agent Number (RAN)** where the party submitting the request is a registered agent;  OR  c) **ABN of the submitting entity, represented as an 11 digit number with no internal separator characters**, where the party submitting the request is a Business or a Business Intermediary.  There MUST be one (and only one) “eb:From” eb:PartyId. | Same as single request. | Set to ATO’s ABN i.e. “51 824 753 556 “. | Required |
| eb:PartyId@type | The type attribute indicates the domain of names to which the string in the content of the PartyId element belongs. | Set to match the corresponding type of the value that eb:PartyId has been set to i.e.:   1. **http://ato.gov.au/PartyIdType/TAN** where the value is a tax agent number;   or   1. **http://ato.gov.au/PartyIdType/RAN** where the value is a registered agent number;   or   1. **http://abr.gov.au/PartyIdType/ABN** where the value is an Australian Business Number. | Same as single request. | Set by ATO e.g. when the PartyId is the ATO’s ABN this will be set to: “http://abr.gov.au/PartyIdType/ABN where the value is a unique Australian Business Number;” | Required |
| eb:Role | Identifies the authorized role of the Party sending the message. | Set to URI representing authorised roles:   1. [http://sbr.gov.au/ato/Role/Registered Agent](http://www.sbr.gov.au/software-developers/developer-tools/ato) where the party submitting the request is a registered agent (including tax agent);   or   1. [http://sbr.gov.au/ato/Role/Business Intermediary](http://sbr.gov.au/agreement/Gateway/1.0/SelectivePull/PKI) where the party submitting the request is a Business Intermediary;   or   1. [http://sbr.gov.au/ato/Role/Business](http://www.sbr.gov.au/software-developers/developer-tools/web-services) where the party submitting the request is a Business | Same as single request. | Set to URI representing the role of the ATO:  [http://sbr.gov.au/agency](http://sbr.gov.au/ato/Role/Business) | Required |

Table : eb:From Structure

#### eb:PartyInfo/eb:To

**The** table below shows the ATO specific children elements for eb:To, and their use within requests and responses.

| **Name** | **Description** | **Single Request**  **2 Way Sync**  **1 Way Push** | **Bulk/Batch**  **1 Way Push** | **Response**  **2 Way Sync**  **1 Way Pull** | **Optionality** |
| --- | --- | --- | --- | --- | --- |
| eb:PartyId | String value that identifies a party. | Set to ATO’s ABN i.e. “51824753556 “.  There MUST be one (and only one) “eb:To” eb:PartyId. | Same as single request. | Copy from PartyInfo.From.Party Id of related Request Message. | Required |
| eb:PartyId@type | The type attribute indicates the domain of names to which the string in the content of the PartyId element belongs. | Set to **http://abr.gov.au/PartyIdType/ABN** where the value is an Australian Business Number. | Same as single request. | Copy from PartyInfo.From.Party Id of related Request Message. | Required |
| eb:Role | Identifies the authorized role of the Party receiving the message. | Set to URI representing the role of the ATO:  “http://sbr.gov.au/agency” | Same as single request | Copy from PartyInfo.From.Role of related Request Message. | Required |

Table : eb:To Structure

### eb:UserMessage/eb:CollaborationInfo

The table below shows the children elements for eb:CollaborationInfo, and their use within requests and responses.

| **Name** | **Description** | **Single Request**  **2 Way Sync**  **1 Way Push** | **Bulk/Batch**  **1 Way Push** | **Response**  **2 Way Sync**  **1 Way Pull** | **Optionality** |
| --- | --- | --- | --- | --- | --- |
| eb: AgreementRef | String element that identifies the entity or artefact governing the exchange of messages between the parties | Set by BMS to URI for pMode file that relates to this MEP. pMode URI’s are specified in Appendix A. | Same as single request. | Set by agency to the corresponding value in the request. | Required |
| eb:Service | String identifying the service that acts on the message. | Set by BMS to the Service Name defined in the ATO product-specific MIG. | Same as single request. | Set by agency to the same value as in request. | Required |
| eb:Action | String element that identifies an operation or an activity within a Service. | Set by BMS to the Service Name defined in the ATO product-specific MIG. | Same as single request. | Set by agency to the same value as in request. | Required |
| eb:ConversationId | String element that identifies the set of related messages that make up a conversation between Parties. | Set to value defined in the SBR ebMS3 WIG. | Same as single request. | Set by agency to the same value as in request. | Required |

Table : eb:CollaborationInfo Structure

### eb:PayloadInfo/eb:PartInfo/eb:PartProperties

The table below shows the ATO specific eb:Property elements and associated attribute values to be used.

For single requests, a separate eb:PartInfo node is required for the base form and each schedule (e.g. A base form with 3 schedules will have 4 eb:PartInfo nodes)

| **Name** | **Description** | **Single Request**  **2 Way Sync**  **1 Way Push** | **Bulk/Batch**  **1 Way Push** | **Response**  **2 Way Sync**  **1 Way Pull** | **Optionality** |
| --- | --- | --- | --- | --- | --- |
| eb:Property@name | Property Name | Set by BMS to “DocumentName”. | Set by BMS to ”DocumentName” | Set by agency to ”DocumentName” | Required |
| eb:Property | DocumentName | Set by BMS to the “Business Name” of the type of document contained in the MIME Part.  e.g. CTR or NIPSS | Set by BMS to the “Business Name” of the type of document contained in the MIME Part.  e.g. CTR or NIPSS | Identifies the type of the document contained in the MIME Part  e.g. CTR or NIPSS etc. | Required |
| eb:Property@name | Property Name | Set by BMS to “DocumentType” | N/A | Set by agency “DocumentType” | Conditionally Mandatory for Single. |
| Eb:Property | DocumentType | Indicates the business type of the business document/s. Must be set to one of the following :  “BASE”  “SCHEDULE” | Must be set to one of the following:  “BATCH”  “BULK” | Describes the message packaging used to transport the business document/s. Must be set to one of the following:  “BATCH” “BULK”  “BASE” | Required |
| eb:Property@name | Property Name | N/A | Set to “ELS Approval Number” for ELS “prior year” request messages.  Only applicable for requests that are targeted from ELS. | N/A | Conditionally Required |
| eb:Property | ELS Approval Number | N/A | Set to [ELS Approval Number] “ for ELS “prior year” request messages.  Only applicable for requests that are targeted from ELS. |  | Conditionally Required |

Table : eb:PartProperties Structure

eb:SIGNALMESSAGE – ATO Profile

The following tables outline ATO specific values that are required to be inserted into the ebMS3 header of a signal message for various ELS2SBR interactions. The use of double quotes in the specification of values in the following table is for delimiting purposes only and those double quotes do not form part of the value that the field should be set to.

### eb:SignalMessage/eb:MessageInfo

The table below shows the children elements for eb:MessageInfo, and their use within different types of Signal requests

| **Name** | **Description** | **Pull Request from BMS** | **Receipt from SBR** | **Error from SBR** | **Optionality** |
| --- | --- | --- | --- | --- | --- |
| eb:Timestamp | Date at which the message header was created | Client MSH to set with DateTimestamp for the date:time that the pull request is sent to SBR ebMS3 | SBR ebMS3 to set with DateTimestamp for the date:time that the receipt message is sent to the BMS | SBR ebMS3 to set with DateTimestamp for the date:time that the receipt message is sent to the BMS | Required |
| eb:MessageId | Globally unique identifier conforming to MessageId [RFC2822] | Set by BMS | Set by SBR ebMS3 | Set by SBR ebMS3 | Required |
| eb:RefToMessageId | MessageId value of an ebMS Message to which this message relates, in a way that conforms to the MEP in use. | N/A | Copy of MessageInfo.MessageId field from related Request Message | Copy of MessageInfo.MessageId field from related Request Message | Required depending on the type of Signal Message i.e. Error, Pull or Receipt |

Table : ebMessageInfo Structure

### eb:SignalMessage/eb:PullRequest

| **Name** | **Description** | **Pull Request from BMS** | **Optionality** |
| --- | --- | --- | --- |
| eb: RefToMessageId | Selective pulling criteria as per ebMS3 advanced features, implemented by SBR ebMS3. | Set by BMS to the copy of MessageInfo.MessageId field from related Request Message i.e. the MessageId of the request message that this pull request is trying to pull the (business or validation) response to. | Required |

Table : eb:PullRequest Structure

### ebMS3 Signal-Receipt & Signal-Error Response Message

There are no ATO specific ebMS3 signal response message field values.

# **General Instructions**

Anonymous Interactions

ATO offers both authenticated and anonymous services. By default all ATO services are authenticated unless specified otherwise. If a service is anonymous this shall be specified in the ATO Service Registry.

Authorisation of online (cloud) service providers

For all SBR cloud request messages, ATO will check against its records that the sender (online (cloud) service provider) can submit the message to SBR on behalf of the entity (Reporting Party or Intermediary).

For these submissions Software ID must be provided otherwise authorisation will fail. If an online(cloud) service provider is lodging for themselves then a Software ID must not be provided.

If authorisation fails, then a response message with the authorisation error will be returned. Please refer to Appendix B for the list of error messages associated with authorisation.

Section 5.2.1 describes the implementation of the Software ID in each message. Please refer to the Cloud Services Authentication and Authorisation Implementation guide for more information.

### Software ID for SBR ebMS3

A new Message Property called “SoftwareSubscriptionId” should be added. To do that the API of the RequestUserMessage class setMessageProperty (String name, String value) of the embeddable client can be used. The method allows adding a new property with the specified value to the generated message.

The location of the new property is shown in the diagram below. It is located in the eb namespace ([http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/](http://sbr.gov.au/identifier/softwareSubscriptionId)).

Please refer to the SBR ebMS3 SDK developer guide for more information.



Figure : SoftwareSubscriptionId Message Property

Response Messages

A response message contains an Overall Event Message and a Business Event Message per Logical Record which may contain errors.

Each Business Event message contains a MaximumSeverity.Code which indicates the most severe level of error present in the associated EventItems.

Allowable values listed in descending order of importance (most important listed first) are:

1. Progressive
2. Partial
3. Error
4. Warning
5. Information

Where more than one code might apply the code with the higher importance MUST be selected.

The following table provides a description of each MaximumSeverity.Code value in a Business Event Message.

| **MaximumSeverity.Code** | **Description** |
| --- | --- |
| **PARTIAL** | One or more Logical Documents in a Logical Record were able to be successfully processed, but that errors were found in the remaining Logical Documents |
| **ERROR** | No Logical Documents in a Logical Record could be successfully processed. |
| **WARNING** | All Logical Documents have passed but some warnings were encountered during processing. |
| **INFORMATION** | Processing was successful for all Logical Documents in a Logical Record. |

Table : MaximumSeverity.Code Value Descriptions

The MaximumSeverity.Code in the Overall Event Message is based on highest MaximumSeverity.Code across all Business Event Messages. Therefore if only one Logical Record is contained in a batch, then the Overall Event Message will share the same MaximumSeverity.Code value as the Business Event Message.

The table below describes a matrix of MaximumSeverity.Code combinations in a batch scenario with two Logical Records.

| **MaximumSeverity.Code in Overall Event Message** | | **Logical Record 1** | | | |
| --- | --- | --- | --- | --- | --- |
|  | **MaximumSeverity.Code in Business Event Message** | **PARTIAL** | **ERROR** | **WARNING** | **INFORMATION** |
| **Logical Record 2** | **PARTIAL** | PARTIAL | PARTIAL | PARTIAL | PARTIAL |
| **ERROR** | PARTIAL | ERROR | ERROR | ERROR |
| **WARNING** | PARTIAL | ERROR | WARNING | WARNING |
| **INFORMATION** | PARTIAL | ERROR | WARNING | INFORMATION |

Table : MaximumSeverity.Code Combinations Matrix

| **Overall Event Message Request Level** | **Meaning** (in the context of the ebMS3 Request Payload) | **Action** |
| --- | --- | --- |
| **INFORMATION** | **Full Acceptance of Request**  All Logical Records are valid and fully accepted (meaning all Child Documents are valid and accepted.) | None required. |
| **WARNING** | **Full Acceptance of Request and some warnings were encountered during processing**  All Logical Records are valid and fully accepted (meaning all Child Documents are valid and accepted.) | Search the response for any **Warnings** and review |
| **ERROR** | **Full Rejection of Request**  One or more Logical Records in the received request had a validation result of Error. | Search the response looking for any **Errored** Logical Records and correct those records (**see below**) and resubmit\* only those corrected Logical Records. |
| **PARTIAL** | **Partial Rejection of Request**  One or more Logical Records had a validation result of Partial. | Search the response looking for any **Partially** Rejected Logical Records and correct (**see below**) only those records and resubmit\* them.  In addition, search the response looking for any **Errored** Logical Records and correct those records (**see below**) and resubmit\* only those corrected Logical Records. |

| **Business Event Message**  **Logical Record Level** | **Meaning** (in the context of a Logical Record) | **Action** |
| --- | --- | --- |
| **INFORMATION** | The Parent Logical Document and all of its Child Logical Documents are valid. | None required. |
| **WARNING** | The Parent Logical Document and all of its Child Logical Documents are valid however some warnings were encountered during processing. | Search the response for the Logical Record looking for any Warnings and review. |
| **ERROR** | In the Logical Record either:   1. All Child Documents are invalid.   AND/OR  The Parent Document is invalid. | Search the response for the Logical Record looking for all invalid Logical Document (which may include Parent and/or Child Logical Documents) results and correct those Logical Documents. |
| **PARTIAL** | The Parent Logical Document is valid and there is:   1. one or more Child Documents that are invalid;   AND   1. at least one Child Document that is valid. | Search the response for the Logical Record looking for any invalid Child Logical Document results and correct those Child Logical Documents. |

Table : Actions required for MaximumSeverity.Code

*\*Subject to specifics for re-submisssion – whether to set the final flag, submit as a separate update etc...*

### Error Messages

#### Business Rule Messages

Business rules that are expected to be implemented by software developers are described in each of the product-specific validation rules spreadsheets. Each rule is associated with a response message code. Where a submission does not comply with a given rule, the associated message code will be returned.

Message codes returned by the ATO have the following format:

**{Jurisdiction}.{Agency}.{Function}.{Id}**

where:

**Jurisdiction** = CMN (Commonwealth)

**Agency** = ATO

**Function** = GEN (General – can apply to many functions/forms); or  
Form specific, such as FBT, CTR, PTR, AS, SMSFAR, etc.

**Id** = function specific identifier.

For example: CMN.ATO.CTR.123456

#### Business Processing Messages

Interactive messages are returned for services capable of immediate processing and able to provide a business processing outcome response. There are two types of messages that can be returned:

* **Processing outcome** – A business processing outcome that can result in confirmation of successful lodgement or processing, or advice the warnings or errors which occurred during processing, and therefore advising the lodgement is unsuccessful.
* **Generic error** – A system/technical error is caused by a technical issue which prevents the business process from completing. This response will include a {code} parameter which can be quoted to technical support teams to assist with the investigation of the issue.

The possible business processing responses are provided in the product-specific validation rules spreadsheets, under Interactive Messages.

The message event contains a Error Code, Severity Code, Short Description and Long Description.

If an interactive message with severity code “Information”is returned, it is informational only. No corrective action required.

Example:

Error.Code: CMN.ATO.ABNREG.EM66044

Severity.Code: Information

Short.Description: Your ABN application has been submitted. You may be contacted by a staff member from the Tax Office regarding this application

What consumer can do:

No corrective action required.

If an interactive message with severity code “Warning” or “Error” is returned, consumer can check the error message description for the details of the error/warning and any instruction for corrective action. Once the corrective action is taken, consumer may re-submit it for processing. If the problem persists, contact ATO for assistance.

Example 1:

Error.Code: CMN.ATO.LDG.EM14073

Severity.Code: Error

Short.Description: Transaction identifier is incorrect.

Long Description: Check the Transaction identifier is correct and re-submit. If the problem persists, contact the ATO.

What consumer can do:

Check and correct the transaction identifier and re-submit.

Example 2:

Error.Code: CMN.ATO.GEN.EM0001

Severity.Code: Error

Short.Description: An unexpected error has occurred, try again. If problem persists, contact the

ATO. Error code {code}.

What consumer can do:

Re-submit. If problem persists, contact the ATO and provide the Error code {code} for investigation and assistance.

### Successful Requests

In the event of a successful request, for most (and all 2015 onwards) services, the following information message shall be returned (in addition to any warnings messages and or error messages):

| **Message Code** | **Severity Code** | **Short Description** |
| --- | --- | --- |
| CMN.ATO.GEN.OK | Information | Message Accepted |

Figure : Successful Request Message Code

### Partially Rejection

The Partial MaximumSeverity.Code is only applicable for bulk requests which support partial rejection. In this scenario, there will be at least one Child Logical Document with Severity Error and no EventItems related to a partially rejected Logical Record will contain an Information or Partial Severity.Code. This means no **EventItem** with Partial Severity.Code will be returned.

### Statistical Information

Each response message contain an Overall Event Message which will contain:

* 1. statistical information that provides an overview of the processing of the corresponding request message; and optionally
  2. system and processing errors

The statistical information (when available) will contain:

1. the summary information related to the number of transactions (Logical Records) received as part of one transmission;
2. the number of transactions which were authorised and validated successfully or failed; and
3. the number of successes and failures from the processing of the transaction.

More particularly, each of the Statistical Information fields are as follows:

*Text that is surrounded by curly braces (i.e. “ { }”) is dynamic information.*

*Dynamic Information is inserted as parameter values into the Short.Descriptions below (refer to the SBR ebMS3 Web Services Implementation Guide for details).*

| **Error.Code** | **Short.Description** | **Description** | **Rules** |
| --- | --- | --- | --- |
| **SBR.GEN.INFO.1** | *Transmission Unsuccessful*  *OR*  *Transmission Successful* | Indicates whether or not the related Request Message has been successfully processed. | ***If*** SBR.GEN.INFO.5, SBR.GEN.INFO.7, SBR.GEN.INFO.9 or SBR.GEN.INFO.10  have a count that is >  0 *(i.e. there are errors during processing)*  *this is s*et to *‘Transmission Unsuccessful’*  ***else***  this is set to *‘Transmission Successful’* |
| **SBR.GEN.INFO.2** | *Total number of transactions in the related Request Message is {number}* | Total number of transactions in the related Request Message | Where {*number*} is the total number of Logical Records in the related Request Message. |
| **SBR.GEN.INFO.3** | *Processing completion indicator is {indicator}*  Where {indicator} is TRUE or FALSE | Channel Processing completion indicator | Where *{indicator}* is set to TRUE this indicates channel processing has completed and where it is set to FALSE then channel processing has failed.  Applicable only for the service/actions which do not require backend processing. For other transactions, this field will not be available. |
| **SBR.GEN.INFO.4** | *Number of transactions passed authorisation check is {number}* | Number of transactions (logical records) passed authorisation check | Where *{indicator}* is the total number of Logical Records that have successfully passed authorisation. |
| **SBR.GEN.INFO.5** | *Number of transactions failed authorisation check is {number}* | Number of transactions (logical records) failed authorisation check | Where *{indicator}* is the total number of transactions (logical records) that have failed authorisation. |
| **SBR.GEN.INFO.6** | *Number of transactions passed channel validation check is {number}* | Number of transactions (logical records) passed channel validation | Where *{indicator}’* is the total number of Logical Records that have successfully passed channel validation.  Channel validation here refers to the form validation as discussed in section 6.5  Note. In a Bulk scenario the successful validation means that Parent and all associated Child records successfully passed their corresponding validations. The count corresponds to the number of parent level records. |
| **SBR.GEN.INFO.7** | *Number of transactions failed channel validation check is {number}* | Number of transactions (logical records) failed channel validation | Where *{indicator}’* is the total number of Logical Records that have failed channel validation. The count corresponds to the number of Parent level records.  Channel validation here refers to the form validation as discussed in section 6.5.  Note. In a Bulk scenario the failed validation means that Parent or any of the associated Child records failed their corresponding validations. |
| **SBR.GEN.INFO.8** | *Number of transactions successfully processed by the backend {number}* | Number of transactions (logical records) successfully processed by the backend | This field will keep a count of the number of Logical Records that were successfully processed in the backend for interactive\* forms/services. It will not be present for Service Actions involving non-interactive\* forms/services.  The count corresponds to the number of parent level records*.* |
| **SBR.GEN.INFO.9** | *Number of transactions failed the backend processing {number}* | Number of transactions (logical records) failed the backend processing | This field will keep a count of the number of Logical Records that failed backend processing for interactive\* forms/services. It will not be present for non-interactive\* forms/services.  The count corresponds to the number of parent level records*.* |
| **SBR.GEN.INFO.10** | *Number of unexpected errors is {number}* | Number of transactions (logical records) incurred unexpected errors | This field is used to keep a count of unexpected errors that occurred whilst attempting to process the logical records in the related Request Message (which means that the processing result of some records can’t be defined as success or failure). The errors will be tracked at a logical record level.  The “unexpected errors” category comprises of all errors other than authorisation, validation and backend system errors. These are the ATO eCommerce Platform’s technical system errors e.g. database failure.  The count for this statistical field will be set to X, where X is the number of logical records that fail due to a technical system error in the eCommerce platform. |

Table 13: Statistical Information

SBR.GEN.INFO.4-9 return separate “pass” and “fail” event counts for authorisation, validation and backend processing because for Batch transactions all logical records in a batch may not all pass and some may fail these events. The pass counts are used to communicate number of logical records in a batch that have passed these events and the fail counts are used to communicate the number of logical records that have failed these events. For ‘singleton’ transactions there is only a single logical record in the request message so the count will be 1 for either the pass or the fail event depending on whether the logical record has passed the corresponding (validation/authorisation/backend processing) event.

In case if any transmission level error occurs then it will be reported via additional last item in the overall message event block.

In case if this error occurred and record processing hasn’t completed then items SBR.GEN.INFO2, 4-10 might not be reported or reported accurately. This is due to validation error processing ceasing immediately in the case of some sever or catastrophic error and therefore the error number is reflective up to this point only.

\* interactive means that SBR services will get a response from the backend system while non-interactive is fire and forget.

# **Previous Version Control**

| **Version** | **Date** | **Description of changes** |
| --- | --- | --- |
| 1.3 | 30.05.2019 | **3 SBR ebMS3 Message Packaging**  **3.7.3 ebMS3 MIME Part 2 Content**  **3.7.3.1 Base/Parent Records**  For ‘Record\_Delimiter/@Name=DocumentID’ text added – (cannot exceed 50 characters)  ‘e.g. 234356657659655 updated to ‘e.g. 999356657659678’.  **3.7.3.2 Schedule/Child Records**  For ‘Record\_Delimiter/@Name=DocumentID’ text added – (cannot exceed 50 characters)  **3.8 ELS Tag Batch Request**  ‘AUSkey’ changed to ‘credential (refer to the ATO SBR Physical End Points document).’.  **5.3.4 Statistical Information**  Text added of ‘or reported accurately. This is due to validation error processing ceasing immediately in the case of some sever or catastrophic error and therefore the error number is reflective up to this point only.’ Added to paragraph three under Table 13. |
| 1.2 | 08.03.2018 | **2 ATO SBR ebMS3 Instructions**  **2.3 SBR ebMS3 Supported Service Invocation Types**  **2.3.1 Polling Intervals**  Text added – ‘The following sub-sections relate to the standard application of Polling intervals. Where a Service Action varies from the standards below, the intervals can be found in the ATO SBR Service Registry.’.  **2.3.1.3 Payroll Event Service Version 3 Polling**  Section deleted. Any variations to polling can now be found in the ATO SBR Service Registry. |
| 1.1 | 22.02.2018 | **3 SBR ebMS3 Message Packaging**  **3.1 Overview**  **3.1.2 Supported Attachments**  ‘odt’ and ‘ods’ extensions added.  **5 General Instructions**  **5.3.4 Statistical Information**  SBR.GEN.INFO.1 - Short description changed from ‘Request Unsuccesful’/‘Request Succesful’ to ‘Transmission Unsuccesful’/‘Transmission Succesful’ to relect the implementation on the Platform |
| 1.0 | 14.12.2017 | The information contained in this document was previously located in the ATO Common Message Implementation Guide (cMIG) version 3.0 dated 15th September 2016.  As this information is specific to the ATOs implementation of SBR ebMS3 it has been removed to this stand-alone document.  In addition to the content lifted from the cMIG, the following changes have been made:  Updated 2.3.1 Polling Intervals section with new polling interval guidance for Payroll Event service version 3.  Updated 3.11.3 Number of Response Messages section with a note that the intermediate response message is not applicable to the Payroll Event service version 3.  Updated 5.3 Response Messages section with details of Partial EventItems and added more details on error handling.  Added new section 5.3.3 to describe Partially Rejected Requests. |

1. If the logical record comprises of base form and one or more schedules, a single validation report is returned for the whole logical record. [↑](#footnote-ref-2)
2. If the logical record comprised of base form and one or more schedules, backend errors/warnings are returned for the base form only. [↑](#footnote-ref-3)
3. If the logical record comprised of base form and one or more schedules, the business response is returned for the base form only. [↑](#footnote-ref-4)
4. If the logical record comprises of either (a) base form and one or more schedules or (b) Parent and one or more Children, a single validation report is returned for whole logical record. [↑](#footnote-ref-5)
5. If the logical record comprises of either (a) base form and one or more schedules or (b) Parent and one or more Children, backend errors/warnings are returned for the base form or Parent only. [↑](#footnote-ref-6)
6. If the logical record comprises of either (a) base form and one or more schedules or (b) Parent and one or more Children, the business response is returned for the base form or Parent only. [↑](#footnote-ref-7)
7. Not applicable to Payroll Event service version 3 [↑](#footnote-ref-8)