Tertiary Education Commission
Te Amorangi Mātauranga Matua

# Industry Training Register

# Guide to integration for ITOs

Version 5.0

Objective id A823307

Published 15 January 2013

# Contents

# 1 Introduction

## 1.1 About this document

This document is for Industry Training Organisations (ITOs), Modern Apprenticeship Coordinators (MACs), TMS vendors, and other parties wishing to integrate with the Industry Training Register (ITR) using the Message Queuing System (MQS). It is written with a technical audience in mind and assumes an understanding of XML schemas, WSDL and SOAP concepts.

ITR's message architecture is one of two layers: an outer layer that manages the control of the message and an inner layer that contains the content of the message. This document describes the outer layer (which is used to integrate with the MQS system).

The document does not cover the following:

- the operational policy that defines the validation rules;
- business impacts of the ITR (which is covered found in the "ITR business impacts guide");
- schema payload definitions and data validation rules for programme and learner event messages (which are covered in the "ITR schema definition").

IMPORTANT NOTE: The ITR will not be used to enforce all the rules and conditions set out in the Industry Training Operational and Modern Apprenticeships policies. ITOs will be required to make an annual attestation independently from the ITR and it will be the responsibility of the ITOs to submit information (including through the ITR) in accordance with the TEC's terms and conditions and policies.

## 1.2 Status of this document

The document is evolving as the details of the project are decided. It will be revised regularly, so please check www.tec.govt.nz for the most up-to-date version.

## 1.3 Other resources

The following resources are also available:

| Resource | Available from |
|---|---|
| Industry Training Act, 1992 | www.legislation.govt.nz |
| ITR Business Impacts Guide | www.tec.govt.nz/Resource-Centre |
| ITR Schema Definition | www.tec.govt.nz/Resource-Centre |
| NSI website | MoE website at www.minedu.govt.nz/NZEducation/EducationPolicies/TertiaryEducation/ForTertiaryEducationInstitutions/NationalStudentIndex.aspxhttp://www.minedu.govt.nz/NZEducation/EducationPolicies/TertiaryEducation/ForTertiaryEducationInstitutions/NationalStudentIndex.aspx |
| Statistical standards | www.stats.govt.nz/methods_and_services/surveys-and-methods/classifications-and-standards.aspx |
| TEC funding rules and conditions | www.tec.govt.nz/Resource-Centre |
| TEC information feeds | www.tec.govt.nz/Site-information/rss-feeds |
| TEC ITO rules and conditions | www.tec.govt.nz/Resource-Centre |
| TEC website | www.tec.govt.nz. |

## 1.4 Feedback and getting help

All questions and comments can be sent to the TEC Service Centre or emailed to ITRfeedback@tec.govt.nz. Information is also available in the ITO section of www.tec.govt.nz.

## 1.5 Terminology and conventions

As this document describes data to be supplied in an XML format, the terminology used is consistent with the XML standard. Some basic terminology and conventions used in this document are:

- **Element** – an item of data to be supplied is known as an “element” e.g. Surname and FirstName are examples of elements;
- **Cardinality** – Describes how many instances of an element must be supplied:

  0 – Optional, can only supply 1;

  1 – Mandatory, must supply a single value;

  0..n – Optional may supply many instances[1];

  1..n – Mandatory, must supply at least one value but could supply multiple.
- **Data Types** – The document lists the type or format of data required for a given element (e.g. string, integer etc), and uses XML types to describe this. Where the element contains sub elements, the data type will then be listed as “complex”.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",

"SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are standard terms, the meaning of which is published in an international standard RFC2199

## 1.6 Namespaces

| Namespace | Definition |
|---|---|
| MQS ITR Broker Service | urn:nzl:govt:educating:integration:tec:itr:1 |

## 1.7 Acronyms and Abbreviations

| Acronym/Abbreviation | Definition |
|---|---|
| MQS | Message Queuing System |
| ESIS | Education Section Integration Services – this system has been replaced by MQS |
| ESAA | Education Sector Authentication and Authorisation |
| URL | Uniform Resource Locator |
| XML | Extensible Mark-up Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| TMS | Training Management System |
| SOAP | Simple Object Access Protocol |

[1] The schema may specify a maximum number of instances that may be supplied.

# 2 Getting Started

Interaction with the ITR system is coordinated through the Message Queuing System (MQS). MQS is a set of integration services which provides authorisation and authentication, messaging and queuing.

MQS is essentially an indirect messaging broker for the ITR web services and all communication control with the ITR web services will be managed through MQS.

## 2.1 TMS Integration Overview

The following diagram illustrates the sequence of messages between the TMS, MQS and the ITR in a typical transaction. This document will not discuss the details of the integration between MQS and the ITR, but will only focus on TMS integration interfaces with ITR using the MQS gateway. We do acknowledge the benefit to TMS vendors in having a complete view of the ITR transaction life cycle.

The diagram describes the following three main domain systems:

- TMS System

  The TMS system is responsible for communicating learner event data to MQS using the Upload Learner Data interface. See section 5.1 for a more detailed description.

  The TMS systems will poll (ask for) results for previously submitted learner event data and receive either the results or a response indicating if the previously submitted data is pending process at the ITR. All communication with MQS will be SOAP over HTTPS.

  ITOs will be required to have ESAA machine accounts provisioned for their TMS in order to interface with the ITR system. Please see section 2.2 for more information on obtaining these accounts.

  **Note:** ITO and MAC staff will need personal ESAA user accounts in order to view the data held within the ITR, via a secure ITR website. Personal ESAA accounts are different compared to the machine accounts describe in this section. Please refer to the ITR Business Impacts Guide document for information on how users obtain these ESAA accounts.

- Message Queuing System

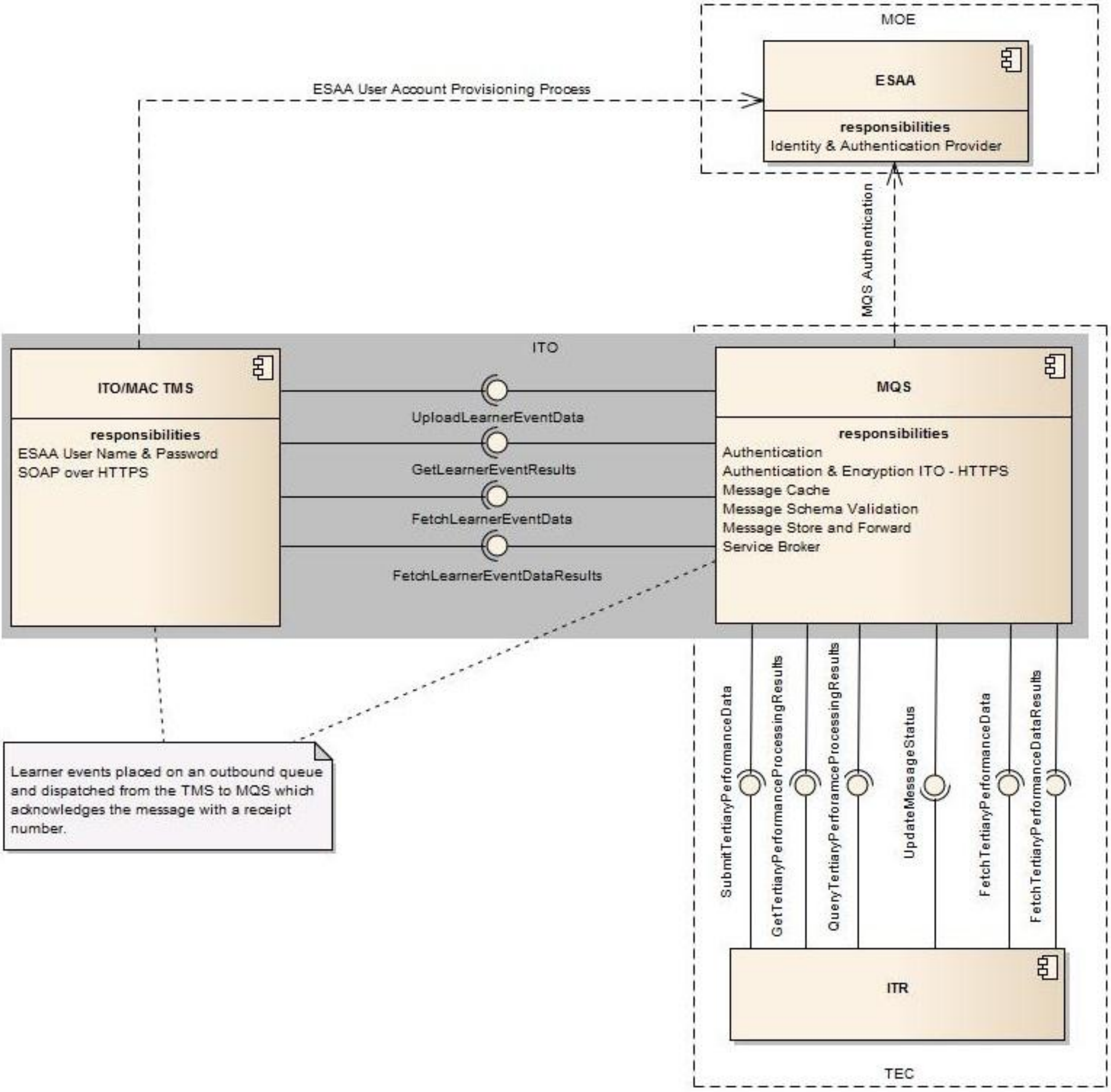
  The Message Queuing system provides web services and reliable message delivery capacity. TMS systems will use MQS as the gateway service to interface with the ITR system. This will be accomplished by consuming SOAP based web services made available by MQS and is discussed in more details in section 5.

- MOE – ESAA System

  The MOE system provides authentication. MQS will use ESAA to validate the user names, passwords and EDUMIS numbers included in messages.

- ITR System

  The ITR system will interface with MQS and will receive and process all the data that was uploaded by ITOs. MQS will directly contact the ITR system when the ITR has indicated that the data has been processed with results to be collected.


MOE
ESAA
responsibilities
Identity & Authentication Provider
ESAA User Account Provisioning Process
MQS Authentication
ITO
ITO/MAC TMS
responsibilities
ESAA User Name & Password
SOAP over HTTPS
UploadLearnerEventData
GetLearnerEventResults
FetchLearnerEventData
FetchLearnerEventDataResults
MQS
responsibilities
Authentication
Authentication & Encryption ITO - HTTPS
Message Cache
Message Schema Validation
Message Store and Forward
Service Broker
Learner events placed on an outbound queue and dispatched from the TMS to MQS which acknowledges the message with a receipt number.
SubmitTertiaryPerformanceData
GetTertiaryPerformanceProcessingResults
QueryTertiaryPerforamceProcessingResults
UpdateMessageStatus
FetchTertiaryPerformanceData
FetchTertiaryPerformanceDataResults
ITR
TEC

## 2.2 Education Sector Authentication and Authorisation (ESAA)

The Ministry of Education’s Education Sector Authentication and Authorisation (ESAA)[2] is a mechanism that provides users with access to several online Education Sector Services.

Every instance (per ITO) of a TMS is required to have an ESAA machine or service account, provisioned for use with the ITR system. Please see section 3.2.1 for information on how to obtain and ESAA machine account for testing purposes, and section 3.2.2 for information on how to obtain to provision the ITO TMS with a Production ESAA machine account.

### 2.2.1 How to get an ESAA machine account for testing

To obtain a test ESAA machine account, please provide the ITR Project Manager (via ITRFeedback@tec.govt.nz) with the following information:

- End points (i.e. IP address) for the development/test services - Mandatory
- Email address (for confirmation of user setup, password resets, or any profile modification, etc) - Mandatory
- Organisation (EDUMIS) number(s) for testing - Mandatory
- Organisation Name(s) so MoE can map to the EDUMIS number(s) - Optional

### 2.2.2 How to provision an ITO’s TMS for a Production ESAA account

Questions and comments can be directed through ITRFeedback@tec.govt.nz. The TEC will contact the Vendor with the relevant forms, process and instructions on how to obtain an ESAA machine account.

## 2.3 What TMS vendors need before they build

- Understand this document, WSDL and XSD file contents
- Understand how MQS will be used as the ITR service broker (See section 3)
- An ESAA machine user account (see section 3.2.1)
- MQS End-Points

The following table provides two end-points. The production endpoint should only be used by the ITO once ITR is available to the sector. The User Acceptance endpoint is used to help Vendors and ITO testing their TMS systems. Separate ESAA accounts are required for both these end-points.

| Production for ITOs | User Acceptance (for TMS) |
|---|---|
| https://itrmq.tec.govt.nz/ESIS_TecItrLearnerEventServices_v1.svc<br>https://integration.education.govt.nz<br>Public IP: 202.37.32.174<br>This is using VeriSign cert.<br>Authenticated by ESAA Production | https://itrmq-uat.tec.govt.nz/ESIS_TecItrLearnerEventServices_v1.svc<br>https://ppintegration.education.govt.nz<br>Public IP: 202.37.37.174<br>This is using internal issued cert.<br>Authenticated by ESAA PreProduction |

[1] WikiPedia – Definition of a web service web site http://en.wikipedia.org/wiki/Web_service

[2] ESAA – Ministry of Education web site (http://cms.steo.govt.nz/News+and+Info/ESAA.htm)

## 2.4 What TMS vendors need to build

Integration with the ITR system requires that the TMS can consume and use soap web services.

Web services are typically application programming interfaces (API) or web APIs that are accessed via Hypertext Transfer Protocol (HTTP) and executed on a remote system hosting the requested services. Web services tend to fall into one of two categories: Big Web Services[1] and RESTful Web Services. ITR will be using the “Big Web Services” definition. ITR uses the “Big Web Services” category.

# 3 Technology Standards Used by ITR

In order to ensure interoperability and provide clear guidelines for integration, the TEC ITR web service complies with a number of technology standards. Systems SHOULD be designed using these standards in order to be fully compatible with the TEC ITR web service.

## 3.1 Simple Object Access Protocol (SOAP) 1.1

The TEC ITR web service uses the Simple Object Access Protocol (SOAP) specification for exchanging structured "tertiary event" messages between the TMS and ITR systems. Messages sent to the TEC ITR web service MUST adhere to the specification for SOAP 1.1 as specified at 08 May 2000[3].

### 3.1.1 WS-I Basic Profile 1.1

The TEC ITR web service complies with the Web Services Interoperability (WS-I) Basic Profile 1.1 specification, which defines a basic subset of the full WSDL specification. All messages sent to the TEC ITR web service MUST comply with the WS-I Basic Profile 1.1[4].

## 3.2 XML Dates

Dates (both *date* and *datetime* XML types) MUST adhere to the set of Gregorian calendar dates as defined in section 5.2.1 of ISO 8601[5] document (i.e. [YYYY]-[MM]-[DD]T[hh]:[mm]).

## 3.3 Macron Encoding

Messages to the TEC ITR web service MUST use UTF-8[6] encoding, as required by the WS-I Basic Profile 1.1.

## 3.4 Transport Security (SSL)

Communication between the TMS and the ITR system using MQS is through one-way SSL 3.0 transport level security (HTTPS)[7].

## 3.5 Basic Security Profile - Version 1.0

Security and the authentication process with the ITR system using MQS conforms with the WS-I Basic Security Profile 1.0 specification[8].

[3] Simple Object Access Protocol (SOAP) 1.1 (http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)

[4] WS-I Basic Profile 1.1 Specification (http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html)

[5] Data elements and interchange formats -- Information interchange -- Representation of dates and times (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26780)

[6] The Unicode Standard, Version 5.0, §3.9 D92, §3.10 D95 (2007)

[7] The SSL Protocol Version 3.0 (http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt)

[8] WS-I Basic Security Profile (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2007-03-30.html)

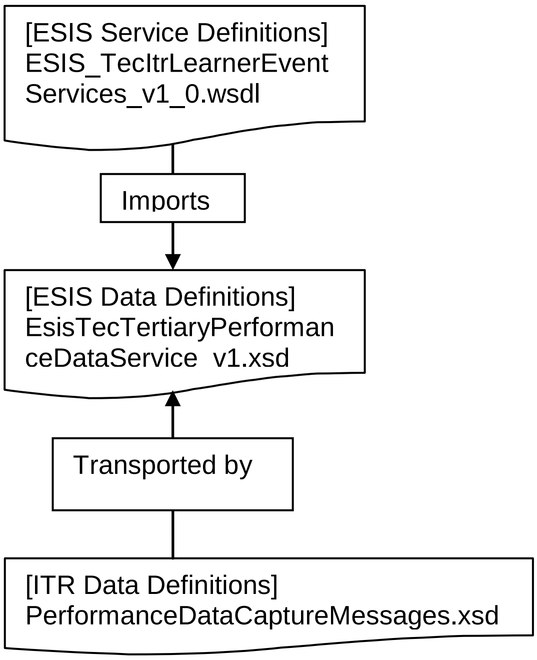
# 4 Common message components

## 4.1 Overview

There are a number of data items that are contained in multiple different ITR transactions.  The definitions for these items are described in this section rather than repeating them across the document.

## 4.2 Schema organisation

ITR messages are defined using an XML schema, which specifies the elements required and structural rules such as cardinality, data types etc.  The XML schema is a technical representation of the structures described in this document. TMS vendors will need to validate and use ITR system messages described in the following schema files:



The purpose of each schema file is as follows:

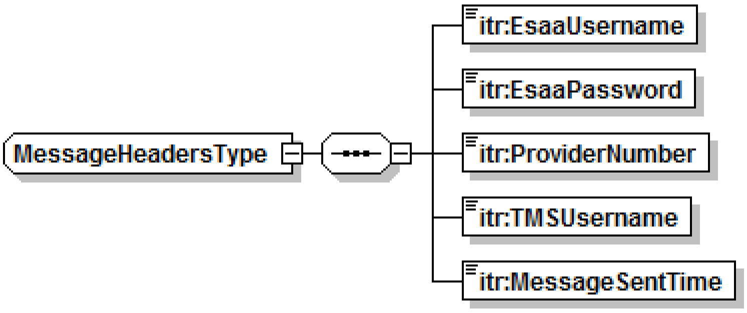
**ESIS_TecItrLearnerEventServices_v1_0.wsdl** – Provides MQS service definitions used to help construct the soap service envelope that will be used by the TMS.

**EsisTecTertiaryPerformanceDataService_v1.xsd** – Provides all of the complex data types used and referenced in the WSDL file above.

**PerformanceDataCaptureMessages.xsd** – Provides all of the complex data types used for the purposes of capturing industry training related data e.g. learner details, training agreements, industry training programmes and enrolments etc. Although not explicitly described in this document we will be referencing these types and these are well documented in the ITR Schema Definition Guide (see References).

## 4.3 Message Headers

Almost all transactions sent to ITR using MQS will supply a common set of attributes required in each message. These attributes are captured as part of each transaction to allow TEC and MQS to audit the transactions and ensure that suitable service levels are met. All data will be transmitted over a secure HTTP channel using SSL (HTTPS) to ensure that data is protected between the TMS and ITR systems.



| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| ESAAUsername | 1 | String | The ESAA provisioned machine account name used authenticate and access ITR services.<br>We recommend making this data value configurable in your TMS system. |
| ESAAPassword | 1 | String | The ESAA provisioned machine account password associated to the ESAAUsername. This field will contain the clear text password, but will only be transmitted over HTTPS, ensuring the proper encryption and security during transmission of this data. It is the responsibility of the TMS systems to store passwords or messages containing passwords in a secure manner.<br>We recommend making this data value configurable in your TMS system. |
| ProviderNumber | 1 | String | A unique number that identifies an education/training organisation, this must match the provider number of the ESAAUsername.<br>We recommend making this data value configurable in your TMS system. |
| TMSUserName | 1 | String | The TMS user who initiated the learner transaction request to ITR. If no TMS user account is applicable, then the username MUST be the same name as used in ESAAUserName element.<br>We recommend making this data value configurable in your TMS system. |
| MessageSentTime | 1 | Datetime | The current date and time the message was sent to the ITR. This will be in the same format as described in section 3.2 of this document. |

# 5 Web Service Operations & Message Structures

The following sections detail each of the ITR web service operations available to TMS systems using MQS, the expected input, output and all possible return codes supplied with the output.

The following operations are available to consume:

- UploadLearnerEventData

  This operation is used to upload learner event data to ITR using MQS.

- GetLearnerEventResults

  This operation is used to check any ITR feedback results for the learner event data uploaded.

- FetchLearnerEventData

  This operation is used to request details about learner event data from ITR using MQS.

- FetchLearnerEventDataResults

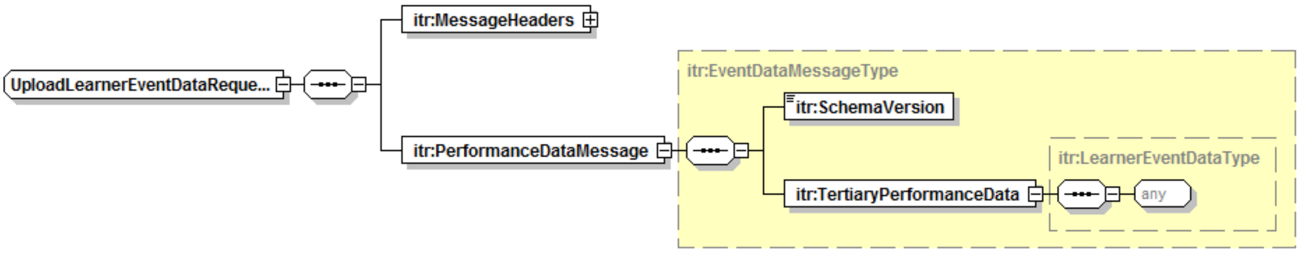
  This operation is used to check and retrieve any results for a request for learner event data made using the FetchLearnerEventData operation.

## 5.1 UploadLearnerEventData Operation

### 5.1.1 Request Message

The requesting message is used to transfer learner event data from your TMS system to the ITR system using MQS. The learner event data request will allow a number of messages to be uploaded, including creating programs, enrolments, training agreements etc. More information on the message data that can be used within this requesting message can be found in the ITR Schema Definition Guide.

#### 5.1.1.1 XML Request Message Structure



| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageHeaders | 1 | Complex | See Section 4.3 for detailed description. |
| SchemaVersion | 1 | String | *Legacy field which is no longer used.* |
| TertiaryPerformanceData | 1 | Complex | This element will contain the actual learner event data XML as described in the ITR schema definition guide and should be validated against the PeformanceDataManagement.xsd schema definition before creating the TertiaryPerformanceData element. |

#### 5.1.1.2 Soap Request Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically.

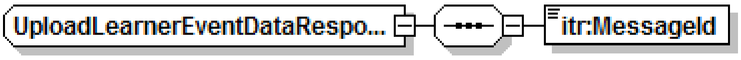
```
<?xml version=”1.0” encoding=”UTF-8”?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
     <urn:UploadLearnerEventDataRequest  xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
           < urn:MessageHeaders>
              < urn:EsaaUsername>[ESAA PROVIDED]</ urn:EsaaUsername>
              < urn:EsaaPassword>[ESAA PROVIDED]</ urn:EsaaPassword>
              < urn:ProviderNumber>123456</ urn:ProviderNumber>
              < urn:TMSUsername>[TMS User Name]</ urn:TMSUsername>
              < urn:MessageSentTime>2002-05-30T09:00:00</urn:MessageSentTime>
           </ urn:MessageHeaders>
           < urn:PerformanceDataMessage>
              < urn:SchemaVersion>1.0.0</SchemaVersion>
              < urn:TertiaryPerformanceData>
                 <!--
                   Learner Event Data Structures as described in the ITR Schema Definition Guide
                   -->
```

```
                </ urn:TertiaryPerformanceData>
            </urn:PerformanceDataMessage>
        </urn: UploadLearnerEventDataRequest >
    </soapenv:Body>
</soapenv:Envelope>
```

### 5.1.2 Response Message

The response message is used to acknowledge the data requested by your TMS system by providing the TMS system with a correlation ID, which is represented as a Global Unique Identifier (GUID). Successful submission will result in a response message with a correlation ID (GUID). TMS systems will have to re-send the message when no correlation ID was received or a SOAP exception was returned instead.

#### 5.1.2.1 XML Response Message Structure



| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageID | 1 | GUID Length:32 Characters | The messageID contains the correlation ID that uniquely identifies your event data request and should be stored with the corresponding record send. Any query for results requests to ITR will need this correlation ID.ITR will use this ID as the unique identifier within the ITR system and can always be used for any systematic or phone support queries.<br>The value will be a Global Unique Identifier (GUID) and will have the following regular expression format: [\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}<br>e.g.: **87762AA3-D38B-4F50-85A3-ECA8577553E3** |

#### 5.1.2.2 SOAP Response Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:UploadLearnerEventDataResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
            <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</ urn:MessageId>
        </urn:UploadLearnerEventDataResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

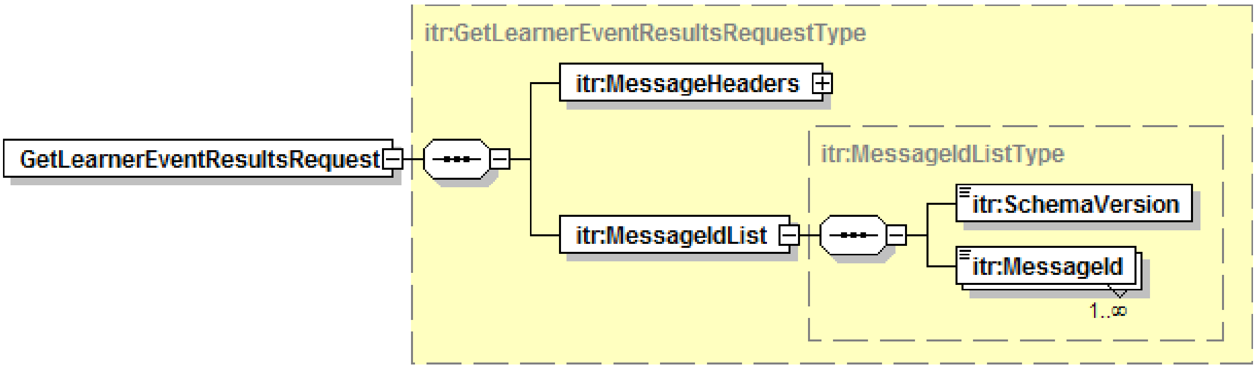
Any soap fault responses or responses not representing the structure above should be considered a failure. SOAP failures can be legitimate authentication or authorisation failures and should be logged at the TMS and resolved by providing the correct ESAA account information.

## 5.2 GetLearnerEventResults Operation

### 5.2.1 Request Message

This requesting message is used to get the processing results for previously submitted (see Section 5.1) learner event data. The message must provide the correlation ID (See MessageID in section 5.1.2.1)

#### 5.2.1.1 XML Request Message Structure



| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageHeaders | 1 | Complex | See Section 4.3 for a detailed description. |
| SchemaVersion | 1 | String | *Legacy field which is no longer used.* |
| MessageId | 1..n | GUID | Provide a list of correlation ID's you wish to request a result for. These ID's will have the same GUID format as described in Section 5.1.2.1. You must provide at least one correlation ID, but can provide as many as 10 ID's. |

#### 5.2.1.2 Soap Request Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. This example is requesting results for two learner event processing results.

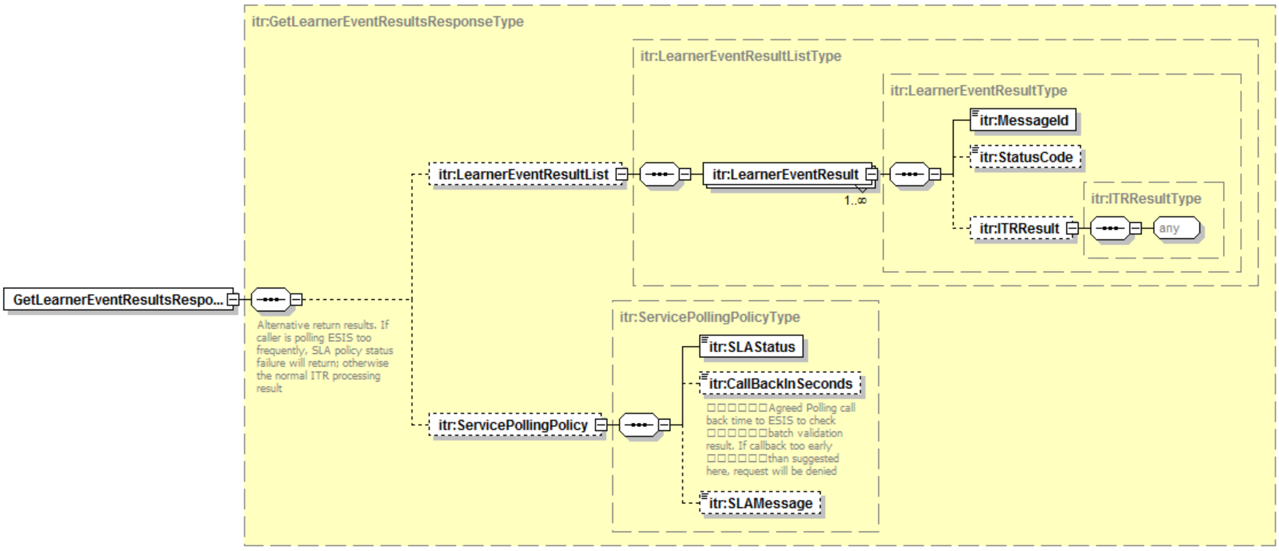
```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetLearnerEventResultsRequest  xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
        < urn:MessageHeaders>
          < urn:EsaaUsername>[ESAA PROVIDED]</ urn:EsaaUsername>
          < urn:EsaaPassword>[ESAA PROVIDED]</ urn:EsaaPassword>
          < urn:ProviderNumber>123456</ urn:ProviderNumber>
          < urn:TMSUsername>[TMS User Name]</ urn:TMSUsername>
          < urn:MessageSentTime>2002-05-30T09:00:00</ urn:MessageSentTime>
```

```
                </ urn:MessageHeaders>
                < urn:MessageIdList>
                  < urn:SchemaVersion>1.0.0</SchemaVersion>
                    < urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</ urn:MessageId>
                    < urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E4</ urn:MessageId>
                </ urn:MessageIdList>
        </urn: GetLearnerEventResultsRequest >
    </soapenv:Body>
</soapenv:Envelope>
```

## 5.2.2 Response Message

The response message describes the details information requested based on the correlation ID. The message is broken basically into two parts. Part one will provide you with a summary of the correlation ID and status code and part two provide a very detailed response message associated with the correlation ID if available at the time of requesting results.

### 5.2.2.1 XML Response Message Structure


itr:GetLearnerEventResultsResponseType
itr:LearnerEventResultListType
itr:LearnerEventResultType
itr:MessageId
itr:StatusCode
itr:ITRResultType
itr:LearnerEventResultList
itr:LearnerEventResult
1..∞
itr:ITRResult
any
GetLearnerEventResultsRespo...
Alternative return results. If caller is polling ESIS too frequently, SLA policy status failure will return; otherwise the normal ITR processing result
itr:ServicePollingPolicyType
itr:SLAStatus
itr:CallBackInSeconds
□□□□□□Agreed Polling call back time to ESIS to check □□□□□□batch validation result. If callback too early □□□□□□than suggested here, request will be denied
itr:ServicePollingPolicy
itr:SLAMessage

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| LearnerEventResultList | 0 | Complex | This element will contain a list of LearnerEventResult elements for each of the corresponding ID requested. (See Section 5.2). This element will not be part of the message if no results are available for the requested corresponding IDs. |
| LearnerEventResult | 1..n | Complex | This element contains the results applicable to the corresponding ID’s |
| MessageID | 1 | String | The correlation ID that the element information applies too. |
| StatusCode | 0 | Enum | Will be one of the following values:<br>**Received:** The tertiary event has been received and placed on a processing queue. This status means that MQS has not yet sent it to ITR or that ITR has not |

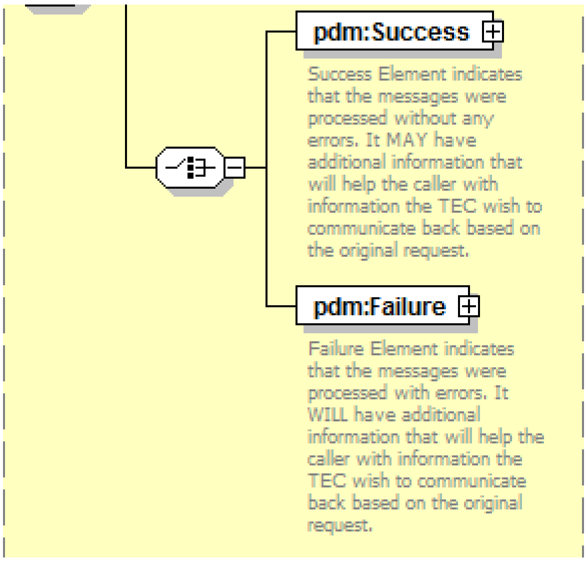
| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| | | | yet processed the request. TMS systems should not expect or process any structured data in the ITRResults element.<br>**Delivered**: The tertiary event has been received set to TEC ITR for processing. This status means that MQS has not yet been notified by ITR that processing is complete. TMS systems should not expect or process any structured data in the ITRResults element.<br>**Retrieved**: The tertiary event has been processed and the results are ready for retrieval. This status means that MQS successfully retrieved the results from ITR. See ITRResults element below.<br>**Message_Error**: an unrecoverable problem has occurred during the message delivery and/or processing process, TMS systems should try resubmitting the message.<br>**Unknown:** A corresponding tertiary event for the given correlation identifier could not be found. |
| ITRResults | 0 | Complex | This element describes the complete and detailed description of the processing results. See Section 5.3 for more information on the detailed results.<br>This element will not be available if no result data is available for processing. |
| ServicePollingPolicy | 0 | Complex | *Legacy field which is no longer used.* |
| SLAStatus | 1 | Enum | *Legacy field which is no longer used.* |
| CallBackInSeconds | 0 | Int | *Legacy field which is no longer used.* |
| SLAMessage | 0 | String | *Legacy field which is no longer used.* |

### 5.2.2.2 SOAP Response Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. The following message structures did omit some of the namespaces to make the response XML more readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetLearnerEventResultsResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
      <urn:LearnerEventResultList>
        <urn:LearnerEventResult>
          <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
          <urn:StatusCode>Processed</urn:StatusCode>
          <urn:ITRResult>
            <!-- See Section 5.3 for more information on the XML provided here -->
          </urn:ITRResult>
        </urn:LearnerEventResult>
      </urn:LearnerEventResultList>
      <urn:ServicePollingPolicy>
        <urn:SLAStatus>SLA_MEET</urn:SLAStatus>
        <urn:CallBackInSeconds>120</urn:CallBackInSeconds>
        <urn:SLAMessage>
             This message describes the details around the SLA
        </urn:SLAMessage>
      </urn:ServicePollingPolicy>
    </urn:GetLearnerEventResultsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

## 5.3 Detailed Result Message for ITRResult element


pdm:Success
Success Element indicates that the messages were processed without any errors. It MAY have additional information that will help the caller with information the TEC wish to communicate back based on the original request.
pdm:Failure
Failure Element indicates that the messages were processed with errors. It WILL have additional information that will help the caller with information the TEC wish to communicate back based on the original request.

| Element Name | | Cardinality | Data Type | Description |
|---|---|---|---|---|
| Success | **OR** | 0..1 | Complex | This element will be available if the message was processed successfully. Addition information may be provided. See Section 5.3.1.5 for detailed description on the information supplied. |
| Failure | | 0..1 | Complex | This element will be available if the message was processed with failures. Addition information will be provided to describe the failures. See Section 5.3.1.5 for detailed description on the information supplied. |

### 5.3.1.1 SOAP Response Message Structure – No Results

The following snippet just shows the soap parts of the message when no results are available from ITR. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. The following message structures did omit some of the namespaces to make the response XML more readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Header/>
 <soapenv:Body>
   <urn:GetLearnerEventResultsResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
     <urn:LearnerEventResultList>
       <urn:LearnerEventResult>
         <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
```

```
          <urn:StatusCode>Queued</urn:StatusCode>
        </urn:LearnerEventResult>
      </urn:LearnerEventResultList>
      <urn:ServicePollingPolicy>
        <urn:SLAStatus>SLA_MEET</urn:SLAStatus>
        <urn:CallBackInSeconds>120</urn:CallBackInSeconds>
        <urn:SLAMessage>
        This message describes the details around the SLA
        </urn:SLAMessage>
      </urn:ServicePollingPolicy>
    </urn:GetLearnerEventResultsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

#### 5.3.1.2 SOAP Response Message Structure – Success

The following snippet just shows the soap parts of the message when success results are available from ITR. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. The following message structures did omit some of the namespaces to make the response XML more readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetLearnerEventResultsResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
      <urn:LearnerEventResultList>
        <urn:LearnerEventResult>
          <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
          <urn:StatusCode>Retrieved</urn:StatusCode>
          <urn:ITRResult>
              <urn:Success/>
          </urn:ITRResult>
        </urn:LearnerEventResult>
      </urn:LearnerEventResultList>
      <urn:ServicePollingPolicy>
        <urn:SLAStatus>SLA_MEET</urn:SLAStatus>
        <urn:CallBackInSeconds>120</urn:CallBackInSeconds>
        <urn:SLAMessage>
              This message describes the details around the SLA
         </urn:SLAMessage>
      </urn:ServicePollingPolicy>
    </urn:GetLearnerEventResultsResponse>
  </soapenv:Body>

</soapenv:Envelope>
```

#### 5.3.1.3 SOAP Response Message Structure – Success with Additional data

The following snippet just shows the soap parts of the message when success results are available from ITR with additional information. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
```

```
    <urn:GetLearnerEventResultsResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
      <urn:LearnerEventResultList>
        <urn:LearnerEventResult>
          <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
          <urn:StatusCode>Retrieved</urn:StatusCode>
          <urn:ITRResult>
            <urn:Success>
              <PerformanceDataMessage>
                <ObjectCategoryDescription>General</ObjectCategoryDescription>
                <ObjectIdentifiers>
                  <ObjectIdentifier>
                    <Key>Programme Number</Key>
                    <Value>7024</Value>
                  </ObjectIdentifier>
                  <ObjectIdentifier>
                    <Key>Programme Version Number</Key>
                    <Value>1</Value>
                  </ObjectIdentifier>
                </ObjectIdentifiers>
                <TransactionResultCode>10000</TransactionResultCode>
                <TransactionResultDescription>Success</TransactionResultDescription>
                <OperationParameters/>
              </PerformanceDataMessage>
            </urn:Success>
          </urn:ITRResult>
        </urn:LearnerEventResult>
      </urn:LearnerEventResultList>
      <urn:ServicePollingPolicy>
        <urn:SLAStatus>SLA_MEET</urn:SLAStatus>
        <urn:CallBackInSeconds>120</urn:CallBackInSeconds>
        <urn:SLAMessage>This message describes the details around the SLA</urn:SLAMessage>
      </urn:ServicePollingPolicy>
    </urn:GetLearnerEventResultsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

### 5.3.1.4 SOAP Response Message Structure – Failure

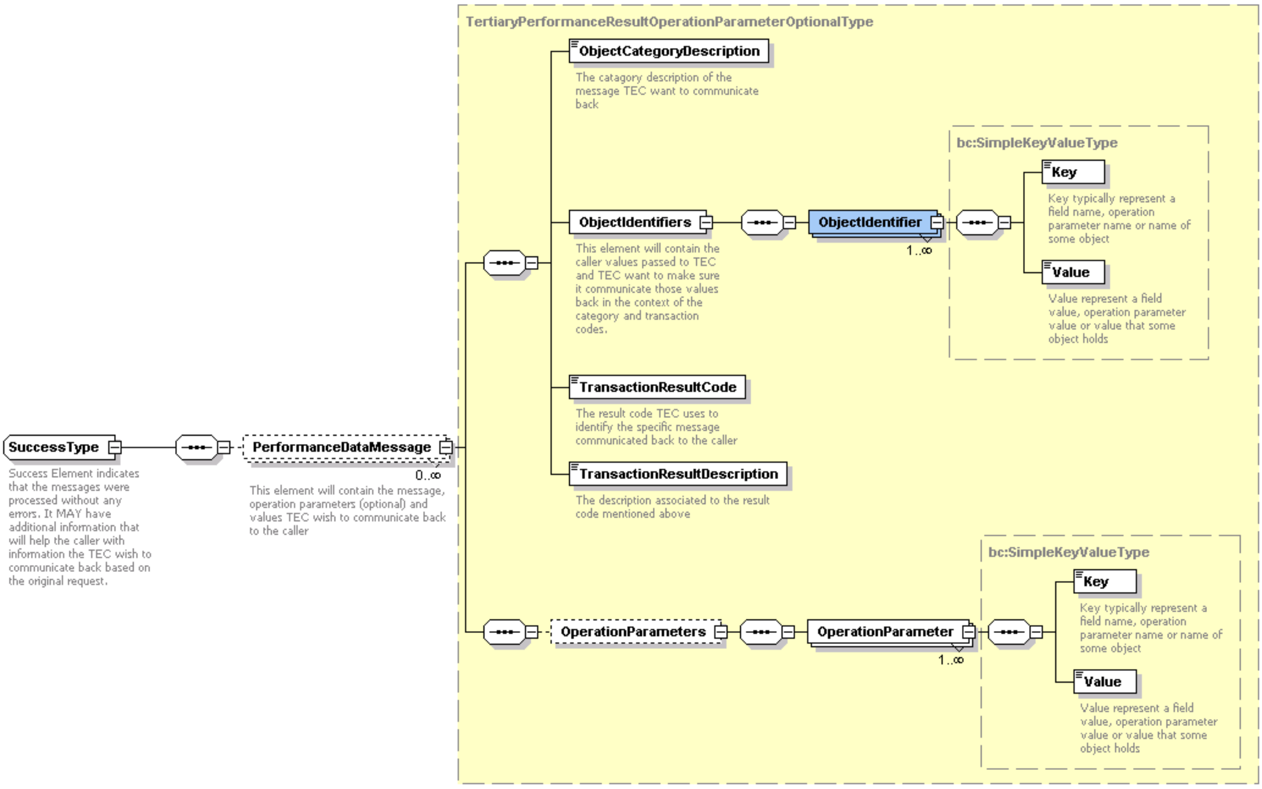
The following snippet just shows the soap parts of the message when failure results are available from ITR. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. The following message structures did omit some of the the namespaces to make the response XML more readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetLearnerEventResultsResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
      <urn:LearnerEventResultList>
        <urn:LearnerEventResult>
          <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
          <urn:StatusCode>Retrieved</urn:StatusCode>
          <urn:ITRResult>
            <urn:Failure>
              <urn:PerformanceDataMessage>
                <ObjectCategoryDescription>Programme</ObjectCategoryDescription>
                <ObjectIdentifiers>
                  <ObjectIdentifier>
                    <Key>Organisation Programme Identifier</Key>
                    <Value>NQ_ENR_100</Value>
                  </ObjectIdentifier>
                </ObjectIdentifiers>
                <TransactionResultCode>30010</TransactionResultCode>
              <TransactionResultDescription>Your organisation's programme identifier must be unique</TransactionResultDescription>
                <OperationParameters>
                  <OperationParameter>
                    <Key>Programme Number</Key>
                    <Value>5</Value>
                  </OperationParameter>
                </OperationParameters>
              </urn:PerformanceDataMessage>
              <urn:PerformanceDataMessage>
                <ObjectCategoryDescription>Programme</ObjectCategoryDescription>
                <ObjectIdentifiers>
                  <ObjectIdentifier>
                    <Key>Organisation Programme Identifier</Key>
                    <Value>NQ_ENR_100</Value>
                  </ObjectIdentifier>
                </ObjectIdentifiers>
                <TransactionResultCode>30020</TransactionResultCode>
              <TransactionResultDescription>Your organisation's programme name must be unique</TransactionResultDescription>
                <OperationParameters>
                  <OperationParameter>
                    <Key>Organisation Programme Name</Key>
                    <Value>NQ in Land Erosion Management - Arable</Value>
                  </OperationParameter>
                  <OperationParameter>
                    <Key>Programme Version Number</Key>
                    <Value>1</Value>
                  </OperationParameter>
                  <OperationParameter>
                    <Key>Programme Number</Key>
                    <Value>5</Value>
                  </OperationParameter>
                </OperationParameters>
              </urn:PerformanceDataMessage>
            </urn:Failure>
          </urn:ITRResult>
        </urn:LearnerEventResult>
      </urn:LearnerEventResultList>
      <urn:ServicePollingPolicy>
        <urn:SLAStatus>SLA_MEET</urn:SLAStatus>
        <urn:CallBackInSeconds>120</urn:CallBackInSeconds>
        <urn:SLAMessage>This message describes the details around the SLA  </urn:SLAMessage>
      </urn:ServicePollingPolicy>
    </urn:GetLearnerEventResultsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

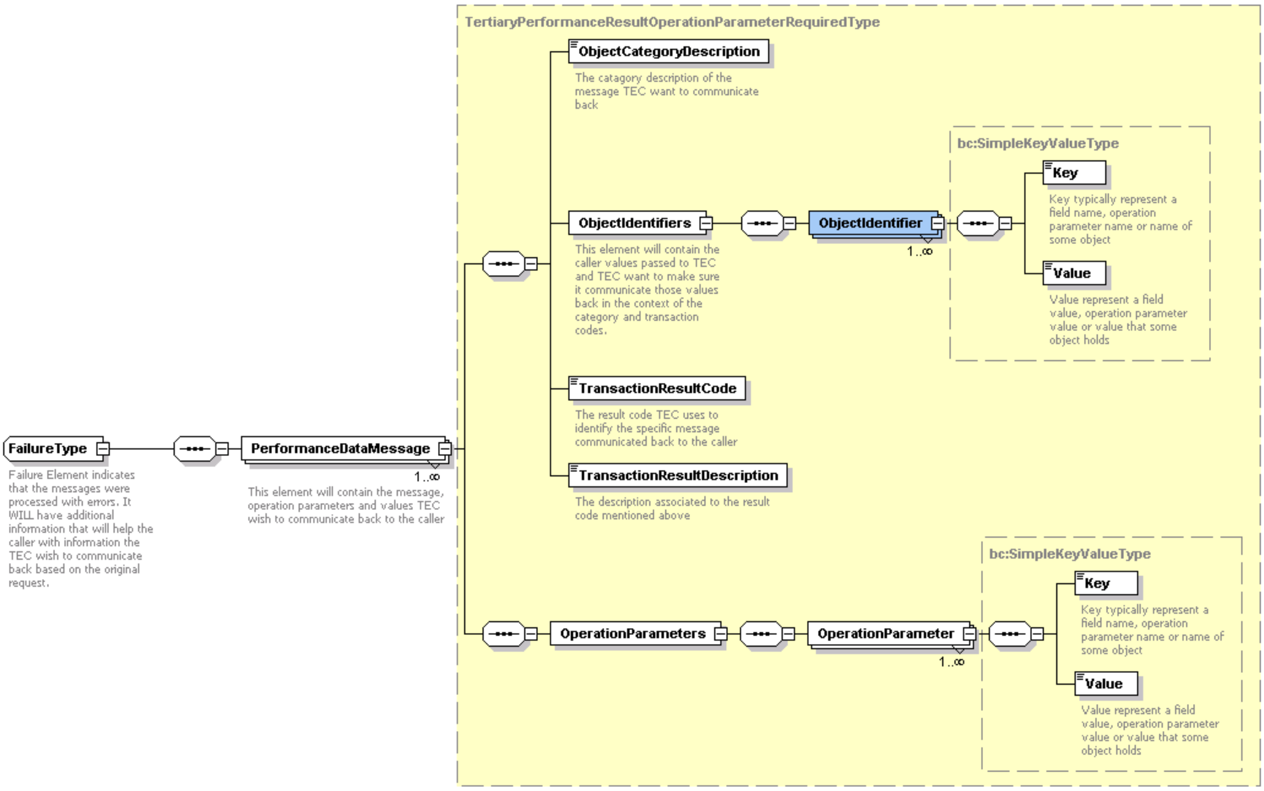
### 5.3.1.5 Result Error/ Failure Message Details

Learner event request results will use the following structures to describe detailed success or failure information that the ITR system will send back to the TMS. There are two similar types used by ITR ‘SuccessType’ and ‘FailureType’, both are defined in the ITR PerformanceDataCaptureMessages.xsd file. They differ only in that OperationParameters are optional on the SuccessType.

### 5.3.1.6 The ITR SuccessType structure


TertiaryPerformanceResultOperationParameterOptionalType
ObjectCategoryDescription
The catagory description of the message TEC want to communicate back
bc:SimpleKeyValueType
Key
Key typically represent a field name, operation parameter name or name of some object
ObjectIdentifiers
ObjectIdentifier
1..∞
This element will contain the caller values passed to TEC and TEC want to make sure it communicate those values back in the context of the category and transaction codes.
Value
Value represent a field value, operation parameter value or value that some object holds
TransactionResultCode
The result code TEC uses to identify the specific message communicated back to the caller
SuccessType
Success Element indicates that the messages were processed without any errors. It MAY have additional information that will help the caller with information the TEC wish to communicate back based on the original request.
PerformanceDataMessage
0..∞
This element will contain the message, operation parameters (optional) and values TEC wish to communicate back to the caller
TransactionResultDescription
The description associated to the result code mentioned above
bc:SimpleKeyValueType
Key
Key typically represent a field name, operation parameter name or name of some object
OperationParameters
OperationParameter
1..∞
Value
Value represent a field value, operation parameter value or value that some object holds

### 5.3.1.7 The ITR FailureType structure


TertiaryPerformanceResultOperationParameterRequiredType
ObjectCategoryDescription
The catagory description of the message TEC want to communicate back
bc:SimpleKeyValueType
Key
Key typically represent a field name, operation parameter name or name of some object
ObjectIdentifiers
ObjectIdentifier
1..∞
This element will contain the caller values passed to TEC and TEC want to make sure it communicate those values back in the context of the category and transaction codes.
Value
Value represent a field value, operation parameter value or value that some object holds
TransactionResultCode
The result code TEC uses to identify the specific message communicated back to the caller
FailureType
Failure Element indicates that the messages were processed with errors. It WILL have additional information that will help the caller with information the TEC wish to communicate back based on the original request.
PerformanceDataMessage
1..∞
This element will contain the message, operation parameters and values TEC wish to communicate back to the caller
TransactionResultDescription
The description associated to the result code mentioned above
bc:SimpleKeyValueType
Key
Key typically represent a field name, operation parameter name or name of some object
OperationParameters
OperationParameter
1..∞
Value
Value represent a field value, operation parameter value or value that some object holds

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| ObjectCategoryDescription | 1 | String | This description will describe the part of the message that ITR determined is |

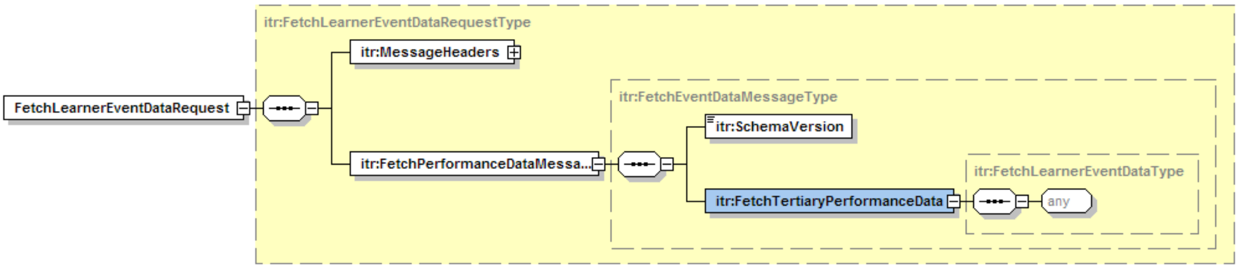
| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| | | | responsible for triggering the validation messages. |
| ObjectIdentifiers | 1 | Complex | This element contains a list of all the fields and values used and identified during validation. In most cases this will only contain a single key/value pair. An example will be the NSN number that is not valid. Please refer to the ITR Schema Definition document Validation Errors section for more information on the codes and descriptions that will be used. |
| ObjectIdentifier | 1 | String | This element is a basic key/value pair combination. The keys will be the name of the field, property or object while the value will contain the data for such keys. |
| TransactionResultCode | 1 | String | This will describe the code used to identify the transaction information. Error codes and messages are described in the ITR Schema Definition document. |
| TransactionResultDescription | 1 | String | This will be the plain English description used to describe the transaction result Code above. |
| OperationParameters | 0..1 for SuccessType<br>1 for FailureType | Complex | This element is a list of operational parameters which ITR used during the processing of the message and will help identify the data used that led up to the creation of this response.<br>Please refer to the ITR Schema Definition document Validation Errors section for more information on the codes and descriptions that will be used.<br>**Please Note:**<br>This element is optional when used as part of the success element and mandatory when used as part of a failure element. Please refer to section 5.3 for a description on the success and failure elements referenced here. |
| OperationParameter | 1 | Complex | This element is a basic key/value pair combination. The keys will be the name of the field, property or object while the value will contain the data for such keys. Value elements may not have a value in which case you will have an empty value element tag (e.g. <value/>), but you will always have a key element with data. |

## 5.4 FetchLearnerEventData Operation

### 5.4.1 Request Message

The requesting message is used to retrieve previously submitted learner event data from the ITR system using MQS. The fetch learner event data request will allow a single learner event data request to be made, for enrolment details, training agreement details etc. More information on the message data that can be used within this requesting message can be found in the ITR Schema Definition Guide.

### 5.4.1.1 XML Request Message Structure


itr:FetchLearnerEventDataRequestType
FetchLearnerEventDataRequest
itr:MessageHeaders
itr:FetchPerformanceDataMessa...
itr:FetchEventDataMessageType
itr:SchemaVersion
itr:FetchLearnerEventDataType
itr:FetchTertiaryPerformanceData
any

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageHeaders | 1 | Complex | See Section 4.3 for detailed description. |
| SchemaVersion | 1 | String | *Legacy field which is no longer used.* |
| FetchTertiaryPerformanceData | 1 | Complex | This element will contain the actual request for learner event data details XML as described in the ITR schema definition guide and should be validated against the PeformanceDataManagement.xsd schema definition before creating the FetchTertiaryPerformanceData element. |

### 5.4.1.2 Soap Request Message Structure

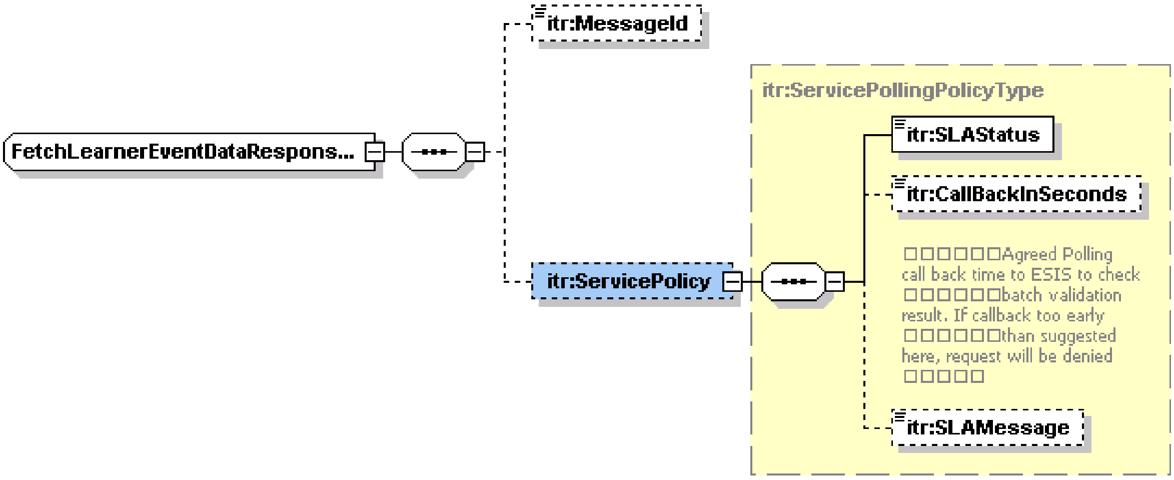
The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
     <urn:FetchLearnerEventDataRequest xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
          < urn:MessageHeaders>
             < urn:EsaaUsername>[ESAA PROVIDED]</ urn:EsaaUsername>
             < urn:EsaaPassword>[ESAA PROVIDED]</ urn:EsaaPassword>
             < urn:ProviderNumber>123456</ urn:ProviderNumber>
             < urn:TMSUsername>[TMS User Name]</ urn:TMSUsername>
             < urn:MessageSentTime>2002-05-30T09:00:00</urn:MessageSentTime>
          </ urn:MessageHeaders>
          < urn:FetchPerformanceDataMessage >
             < urn:SchemaVersion>1.0.0</SchemaVersion>
             < urn:FetchTertiaryPerformanceData >
                <!--
                  Fetch Learner Event Data Structures as described in the ITR Schema Definition Guide
                  -->
             </ urn:FetchTertiaryPerformanceData >
          </urn:FetchPerformanceDataMessage >
     </urn:FetchLearnerEventDataRequest >
  </soapenv:Body>
</soapenv:Envelope>
```

### 5.4.2 Response Message

The response message is used to acknowledge the data requested by your TMS system by providing the TMS system with a correlation ID, which is represented as a Global Unique Identifier (GUID). Successful submission will result in a response message with a correlation ID (GUID). TMS systems will have to re-send the message when no correlation ID was received or a SOAP exception was returned instead.

#### 5.4.2.1 XML Response Message Structure


itr:MessageId
FetchLearnerEventDataRespons...
itr:ServicePolicy
itr:ServicePollingPolicyType
itr:SLAStatus
itr:CallBackInSeconds
□□□□□□Agreed Polling call back time to ESIS to check □□□□□□batch validation result. If callback too early □□□□□□than suggested here, request will be denied □□□□□
itr:SLAMessage

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageId | 1 | GUID Length:32 Characters | The messageID contains the correlation ID that uniquely identifies your fetch event data request and should be stored with the corresponding record send. Any query for fetch learner event data results requests to ITR will need this correlation ID. ITR will use this ID as the unique identifier within the ITR system and can always be used for any systematic or phone support queries.<br>The value will be a Global Unique Identifier (GUID) and will have the following regular expression format: [\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}<br>e.g.: **87762AA3-D38B-4F50-85A3-ECA8577553E3** |
| ServicePolicy | 1 | Complex | *Legacy field which is no longer used.* |

#### 5.4.2.2 SOAP Response Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
     <urn:FetchLearnerEventDataResponse xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
         <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</ urn:MessageId>
     </urn:FetchLearnerEventDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

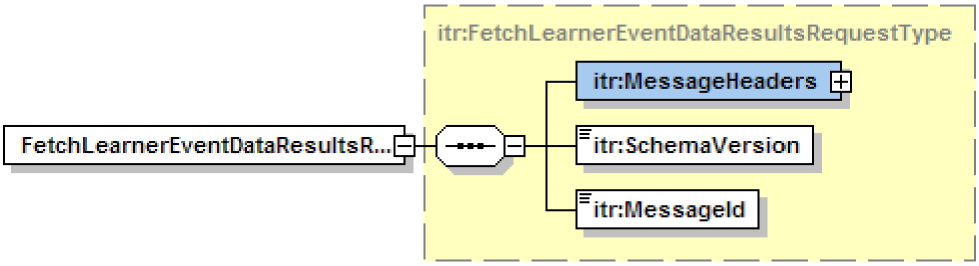
Any soap fault responses or responses not representing the structure above should be considered a failure. SOAP failures can be legitimate authentication or authorisation failures and should be logged at the TMS and resolved by providing the correct ESSA account information.

## 5.5 FetchLearnerEventDataResults Operation

### 5.5.1 Request Message

This requesting message is used to get the results for a previously submitted request (see Section 5.4) for learner event details. The message must provide the correlation ID (See MessageID in section 5.4.2.1)

#### 5.5.1.1 XML Request Message Structure


itr:FetchLearnerEventDataResultsRequestType
itr:MessageHeaders
FetchLearnerEventDataResultsR...
itr:SchemaVersion
itr:MessageId

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| MessageHeaders | 1 | Complex | See Section 4.3 for a detailed description. |
| SchemaVersion | 1 | String | *Legacy field which is no longer used.* |
| MessageId | 1 | GUID | The correlation ID you wish to request a result for. This ID has the same GUID format as described in Section 5.4.2.1. |

#### 5.5.1.2 Soap Request Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. This example is requesting results for two learner event processing results.

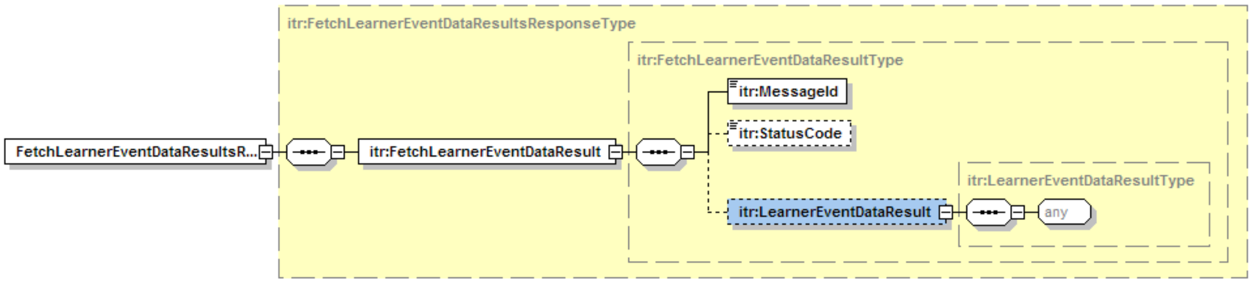
```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soapenv:Header/>
  <soapenv:Body>
     <urn:FetchLearnerEventDataResultsRequest xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
          <urn:MessageHeaders>
              <urn:EsaaUsername>[ESAA PROVIDED]</urn:EsaaUsername>
```

```
                <urn:EsaaPassword>[ESAA PROVIDED]</urn:EsaaPassword>
                <urn:ProviderNumber>123456</ urn:ProviderNumber>
                <urn:TMSUsername>[TMS User Name]</urn:TMSUsername>
                <urn:MessageSentTime>2002-05-30T09:00:00</urn:MessageSentTime>
            </urn:MessageHeaders>
            <urn:SchemaVersion>1.0.0</SchemaVersion>
            <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
        </urn:FetchLearnerEventDataResultsRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

### 5.5.2 Response Message

The response message describes the details information requested based on the correlation ID. The message is broken basically into two parts. Part one will provide you with a summary of the correlation ID and status code and part two provide a very detailed response message associated with the correlation ID if available at the time of requesting results.

#### 5.5.2.1 XML Response Message Structure


itr:FetchLearnerEventDataResultsResponseType
itr:FetchLearnerEventDataResultType
itr:MessageId
itr:StatusCode
FetchLearnerEventDataResultsR...
itr:FetchLearnerEventDataResult
itr:LearnerEventDataResultType
itr:LearnerEventDataResult
any

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
| LearnerEventDataResult | 0 | Complex | This element will contain the LearnerEventDataResult element for the corresponding ID requested. (See Section 5.4). This element will be empty, if no results are available for the requested corresponding ID. |
| MessageID | 1 | String | The correlation ID that the message information applies too. |
| StatusCode | 0 | Enum | Will be one of the following values:<br>**Received:** The fetch data request has been received and placed on a processing queue. This status means that MQS has not yet sent it to ITR or that ITR has not yet processed the request. TMS systems should not expect or process any structured data in the LearnerEventDataResult element.<br>**Delivered**: The fetch data request has been received set to TEC ITR for processing. This status means that MQS has not yet been notified by ITR that processing is complete. TMS systems should not expect or process any |

| Element Name | Cardinality | Data Type | Description |
|---|---|---|---|
|  |  |  | structured data in the LearnerEventDataResult element.<br>**Retrieved**: The fetch data request has been processed and the results are ready for retrieval. This status means that MQS successfully retrieved the results from ITR. See LearnerEventDataResult element below.<br>**Message_Error**: an unrecoverable problem has occurred during the message delivery and/or processing process, TMS systems should try resubmitting the message.<br>**Unknown:** A corresponding fetch data request for the given correlation identifier could not be found. |

### 5.5.2.2 SOAP Response Message Structure

The following snippet just shows the soap parts of the message. Most modern implementation technologies today will enable you to create proxy classes and associated soap messages using the supplied WSDL descriptions without having to create SOAP messages programmatically. The following message structures did omit some of the namespaces to make the response XML more readable.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
     <urn:FetchLearnerEventDataResultsResponse
xmlns:urn="urn:nzl:govt:educating:integration:tec:itr:1">
        <urn:FetchLearnerEventDataResult >
           <urn:MessageId>87762AA3-D38B-4F50-85A3-ECA8577553E3</urn:MessageId>
           <urn:StatusCode>Processed</urn:StatusCode>
           <urn:LearnerEventDataResult>
              <!-- See Section 6.2 of the ITR Schema Definition document for more information on the XML
provided here -->
           </urn:LearnerEventDataResult>
        </urn:FetchLearnerEventDataResult >
     </urn:FetchLearnerEventDataResultsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```