|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | |  |  |
| Standard Business Reporting  Australian Taxation Office –  Family Trust Election, Revocation or Variation (fter.0001)  Message Implementation Guide  Program name: Standard Business Reporting  Date: 21 June 2012  Production Release – 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](mailto:SBRServiceDesk@sbr.gov.au) or call 1300 488 231. International callers may use +61-2-6216 5577 | | |

VERSION CONTROL

|  |  |  |
| --- | --- | --- |
| Version | Release date | Description of changes |
| 1.0 | 21/06/2012 | Initial release  Update the following validation rules. VR.ATO.FTER.408087 VR.ATO.FTER.408113 VR.ATO.FTER.408115 VR.ATO.FTER.408161  Also see ATO Family Trust Election, revocation or Variation 2012 Release Notes for more details on updates to the fter.0001 service |

|  |  |  |
| --- | --- | --- |
| ENDORSEMENT  APPROVAL |  |  |
|  | Chief Solutions Architect  Standard Business Reporting | |
| Michael Ferris | Project Manager  Strategic Web Services  Australian Taxation Office | |

Copyright

© Commonwealth of Australia 2012 (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/). 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 6](#_Toc327363642)

[1.1 Purpose 6](#_Toc327363643)

[1.2 Audience and Scope 6](#_Toc327363644)

[1.3 References 6](#_Toc327363645)

[1.4 Change Management 6](#_Toc327363646)

[2 General Instructions 7](#_Toc327363647)

[2.1 Authorisation of Intermediaries 7](#_Toc327363648)

[2.1.1 Authorisation of Intermediaries for FTER Stand-alone Request Messages 7](#_Toc327363649)

[2.1.2 Authorisation of Intermediaries where FTER is a Schedule 7](#_Toc327363650)

[2.2 Monetary Amount 7](#_Toc327363651)

[2.3 Declarations 7](#_Toc327363652)

[2.3.1 Sender Declaration for Schedule 7](#_Toc327363653)

[2.3.2 Sender Declaration for Stand-alone 8](#_Toc327363654)

[2.3.3 Trustee declarations 10](#_Toc327363655)

[2.3.3.1 Trustee declarations for an FTER schedule 10](#_Toc327363656)

[2.3.3.2 Trustee declarations for an FTER stand-alone submitted by a self lodger 10](#_Toc327363657)

[2.3.3.3 Trustee declarations for an FTER stand-alone submitted by an intermediary 10](#_Toc327363658)

[2.4 SBDH Variations 10](#_Toc327363659)

[2.4.1 Business Documents 11](#_Toc327363660)

[2.4.2 Attachments 11](#_Toc327363661)

[2.4.3 Document Identifiers 11](#_Toc327363662)

[2.5 Response Messages 12](#_Toc327363663)

[2.5.1 Messages Described in the MIG 12](#_Toc327363664)

[2.6 Validation Phasing 12](#_Toc327363665)

[2.7 Rule expression 12](#_Toc327363666)

[2.7.1 Form Prefix Labels 12](#_Toc327363667)

[2.7.2 Context MIG Labels 12](#_Toc327363668)

[2.7.3 No Form or Context Prefix 13](#_Toc327363669)

[2.7.4 Use of xx.xx in Fact Names 13](#_Toc327363670)

[2.7.5 Use of Aliases 13](#_Toc327363671)

[2.7.6 Interpretation of NULL in Calculations and Comparisons 13](#_Toc327363672)

[2.7.7 Use of Domain in rules 13](#_Toc327363673)

[2.7.8 Case Sensitivity 13](#_Toc327363674)

[2.7.9 Tuples and Context 13](#_Toc327363675)

[2.7.10 Format Rules 14](#_Toc327363676)

[3 Business Overview 15](#_Toc327363677)

[3.1 Income tax suite overview 15](#_Toc327363678)

[3.1.1 Message Structure 15](#_Toc327363679)

[3.1.2 Taxonomy and MIG Structure 15](#_Toc327363680)

[3.1.3 Schema Use 15](#_Toc327363681)

[3.2 Business Context Model 15](#_Toc327363682)

[3.3 Financial Year and Substituted Accounting Periods 16](#_Toc327363683)

[3.4 Business Applicability Period 16](#_Toc327363684)

[3.5 Report Version 17](#_Toc327363685)

[4 XBRL Context Specifications 18](#_Toc327363686)

[4.1 Context Specification Dimension 1: ReportPartyType, Period: Duration 18](#_Toc327363687)

[4.1.1 Context Instances 20](#_Toc327363688)

[5 research & development tax incentive schedule Interaction Model 21](#_Toc327363689)

[5.1 Prerequisites 21](#_Toc327363690)

[5.2 Service Summary 21](#_Toc327363691)

[5.3 FTER.PRELODGE Specifications 21](#_Toc327363692)

[5.3.1 FTER.PRELODGE Request - Message 22](#_Toc327363693)

[5.3.1.1 Discoverable Taxonomy Set References 22](#_Toc327363694)

[5.3.1.2 Standard Business Document Header Content 22](#_Toc327363695)

[5.3.1.3 Standard Business Document Body Content 22](#_Toc327363696)

[5.3.2 FTER.PRELODGE Response - Message 22](#_Toc327363697)

[5.3.2.1 Discoverable Taxonomy Set References 22](#_Toc327363698)

[5.3.2.2 Standard Business Document Header Content 22](#_Toc327363699)

[5.3.2.3 Standard Business Document Body Content 22](#_Toc327363700)

[5.4 FTER.LODGE Specifications 23](#_Toc327363701)

[5.4.1 FTER.LODGE Request - Message 23](#_Toc327363702)

[5.4.1.1 Discoverable Taxonomy Set References 23](#_Toc327363703)

[5.4.1.2 Standard Business Document Header Content 23](#_Toc327363704)

[5.4.1.3 Standard Business Document Body Content 24](#_Toc327363705)

[5.4.1.4 FTER.LODGE Request Message Content Table 25](#_Toc327363706)

[5.4.2 FTER Lodge Response - Message 53](#_Toc327363707)

[5.4.2.1 Discoverable Taxonomy Set References 53](#_Toc327363708)

[5.4.2.2 Standard Business Document Header Context 53](#_Toc327363709)

[5.4.2.3 Standard Business Document Body Context 53](#_Toc327363710)

[Appendix A – The Message Content Table Explained 54](#_Toc327363711)

[Appendix B – Tax Office Structured English 57](#_Toc327363712)

[Appendix C – Validation rules alias definitions 69](#_Toc327363713)

[Appendix D – Common module validation rules 71](#_Toc327363714)

[Appendix E – Domain Definitions 80](#_Toc327363715)

Terminology

For definition of the terminology and acronyms used within this document please refer to the glossary on the SBR website – Click here <http://www.sbr.gov.au/software-developers/developer-tools/glossary>to go to the glossary.

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 **http://www.ietf.org/rfc/rfc2119.txt**. The use of the word “Mandatory” is to be read as “MUST”.

1. Introduction
   1. Purpose

The purpose of this document is to support software developers in the implementation of the SBR reporting service for the Australian Taxation Office (ATO) Family trust Election, Revocation or Variation (FTER).

The FTER reporting service can be submitted alone or it can accompany one of the following Income Tax Return forms:

* Self-Managed Superannuation Fund Annual Return (SMSFAR), or
* Trust Return (TRT).
  1. Audience and Scope

This document contains the necessary information required to support ATO Family Trust Election, Revocation or Variation (fter.0001) implementation. This document replaces the ATO FTER Message Implementation Guide (MIG) published on 23 August 2011.

* 1. References

|  |  |  |
| --- | --- | --- |
| **Ref** | **Document Link** | **Document description** |
|  | The SBR Web Service Implementation Guide document can be downloaded at  <http://www.sbr.gov.au/software-developers/developer-tools/web-services> | Technical interface data that is common to all business processes and messages that use the SBR channel:   * Web service protocol specifications * Standard message header structure * Standard error codes * Authentication protocol and trust broker |
|  | The SBR Taxonomy Architecture document can be downloaded at  <http://www.sbr.gov.au/software-developers/developer-tools/re-usable-components> | Reference document that describes the structure of the SBR taxonomy, its naming conventions, release management and change control, and how each business interaction fits within the architecture. |
|  | The Software Developer Kit documentation can be accessed at  <http://www.sbr.gov.au/software-developers/enabling-sbr-in-my-application/productivity-tools> | Reference information for software developers using the SBR software developer kit |
|  | The Message Implementation Guides for TRT and SMSFAR can be downloaded at  <http://www.sbr.gov.au/software-developers/developer-tools/ato> | Information to support the implementation of parent returns for FTER when used as a schedule. |

* 1. Change Management

If a material change is required to the ATO FTER Message Implementation Guide (MIG) the document will be re-released. The Taxonomy Approval Committee must approve any change.

1. General Instructions

This section provides instructions that are relevant across all collaborations and messages specified within this MIG.

* 1. Authorisation of Intermediaries
     1. Authorisation of Intermediaries for FTER Stand-alone Request Messages

For all FTER stand-alone request messages, the Tax Office will check against its records that the sender is authorised to perform the requested action for the reporting party (taxpayer).

The checking will compare the identity and permissions associated with the AUSkey against the identity provided in the business document for the reporting party. If the sender is acting in their role as an agent, then the Tax Agent Number must also be provided otherwise authorisation will fail.

If authorisation fails, then a response message communicating this in the Standard Business Document Header (SBDH) Message Event section will be returned with SBR Error Code: CMN.ATO.GEN.200000.

* + 1. Authorisation of Intermediaries where FTER is a Schedule

When FTER is a used as a schedule to a SMSFAR or TRT return request message, authorisation is managed on the parent form message. Refer to the MIG for the parent form (SMSFAR or TRT) for information on authorisation.

* 1. Monetary Amount

All data elements of type xbrli:MonetaryItemType supplied within messages associated with this report are required to be in Australian dollars. In adherence with the XBRL standard this is achieved by using the following unit declaration:

<xbrli:unit id="u1">

    <xbrli:measure>iso4217:AUD</xbrli:measure>

</xbrli:unit>

XBRL processors will validate that the measure adheres to the ISO standard but the agency will ensure that this is set to Australian Dollars.

* 1. Declarations

Unlike most other forms and schedules, the FTER may require more than one declaration to be submitted, that is, a sender declaration and one or more trustee declarations.

* + 1. Sender Declaration for Schedule

If the FTER is lodged as a schedule, the sender (intermediary - such as a tax agent, or the taxpayer) will make the declaration that is required by the main parent form. The requirements for the sender declaration are described in the MIGs for TRT or SMSFAR messages. No separate sender declaration is required. However Trustee declarations will be required – see Section 2.3.3 below.

* + 1. Sender Declaration for Stand-alone

If the FTER message is lodged as a stand-alone form then a declaration is required from the reporting party (taxpayer) indicating that the information contained is true and correct. The stand-alone form may also be lodged by an agent making a different declaration.

To make a declaration, the sender needs to be cognisant of two things:

* The statement they are making, and
* That it becomes a declaration by them ‘signing’ it.

As a result, in every case that a declaration is required to accompany a transaction, the sender must have displayed to them:

- Specific statement(s) describing what they are about to declare, and

- An acknowledgement that the declaration is made by signing the statement(s) in a particular way.

The sender signs by actively confirming what constitutes their ‘signature’ using a tick-box, submit button, or similar mechanism. Their signature must be some information sent with the transaction that enables the sender to be uniquely identified within the business.

How this declaration is made and handled via SBR varies depending on who sends the form and what type of AUSkey they use to do so. The instructions for each case are described in the tables below.

|  |  |
| --- | --- |
| Case 1: A **reporting party** (i.e. taxpayer), or an intermediary who is not an agent, is lodging via SBR using an AUSkey assigned to an **individual**. | |
| Declaration statement | The statement that the sender is declaring shall be:  “*I declare that the information transmitted in this Family Trust Election, Revocation or Variation Form is true and correct and that I am authorised to make this declaration”.* |
| Signing statement | The text describing the way that they are ‘making’ the declaration by ‘signing’ it in a particular way shall include reference to signing with the AUSkey.  For example: “*Tick this box to sign this declaration with the AUSkey you used to log in*.”  A statement “Tick this box to sign this declaration” would not be acceptable as it does not state the identity the sender is using to make the declaration. |

|  |  |
| --- | --- |
| Case 2: A **reporting party** (i.e. taxpayer), or an intermediary who is not an agent, is lodging via SBR using an AUSkey assigned to a **device**. | |
| Declaration statement | The statement that the sender is declaring shall be:  “*I declare that the information transmitted in this Family Trust Election, Revocation or Variation Form is true and correct and that I am authorised to make this declaration.*” |
| Signing statement | The text describing the way that they are ‘making’ the declaration by ‘signing’ it in a particular way shall include reference to signing with the AUSkey for the device *and* the field giving a unique user identifier.  For example: “*Tick this box to sign this declaration with the AUSkey used by this software and your full name inserted above*.”  A statement “Tick this box to sign this declaration” would not be acceptable as it does not state the identity the sender is using to make the declaration.  The user identifier must allow the AUSkey owner or an external auditor to uniquely identify the individual who made the declaration.  The identifier used can be specified by the AUSkey owner providing it allows identification as mentioned above. Examples of suitable identifiers include a user login, a full name, or an email address. |

|  |  |
| --- | --- |
| Case 3: An **intermediary** who is an agent (i.e. Tax agent) is lodging via SBR using an AUSkey assigned to an **individual**. | |
| Declaration statement | The statement that the sender is declaring shall be:  “*I declare that:*   * *I have prepared this Family Trust Election, Revocation or Variation Form in accordance with the information supplied by the entity;* * *I have received a declaration made by the entity that the information provided to me for the preparation of this return is true and correct; and* * *I am authorised by the entity to give information in this return to the Commissioner.”* |
| Signing statement | The text describing the way that they are ‘making’ the declaration by ‘signing’ it in a particular way shall include reference to signing with the AUSkey.  For example: “*Tick this box to sign this declaration with the AUSkey you used to log in.”*  A statement “Tick this box to sign this declaration” would not be acceptable as it does not state the identity the sender is using to make the declaration. |

|  |  |
| --- | --- |
| Case 4: An **intermediary** who is an agent (that is, a Tax agent) is lodging via SBR using an AUSkey assigned to a **device**. | |
| Declaration statement | The statement that the sender is declaring shall be:  “*I declare that:*   * *I have prepared this Family Trust Election, Revocation or Variation Form in accordance with the information supplied by the entity;* * *I have received a declaration made by the entity that the information provided to me for the preparation of this return is true and correct; and* * *I am authorised by the entity to give information in this return to the Commissioner.”* |
| Signing statement | The text describing the way that they are ‘making’ the declaration by ‘signing’ it in a particular way shall include reference to signing with the AUSkey for the device *and* the field giving a unique user identifier.  For example: “*Tick this box to sign this declaration with the AUSkey used by this software and your full name inserted above.”*  A statement “Tick this box to sign this declaration” would not be acceptable as it does not state the identity the sender is using to make the declaration.  The user identifier must allow the AUSkey owner or an external auditor to uniquely identify the individual who made the declaration.  The identifier used can be specified by the AUSkey owner providing it allows identification as mentioned above. Examples of suitable identifiers include a user login, a full name, or an email address. |

* + 1. Trustee declarations

Whether sent as a stand-alone form or as a schedule, a separate declaration must be made on behalf of the trustees to declare that each trustee has verified the details of the form.

* + - 1. Trustee declarations for an FTER schedule

If an FTER schedule is submitted with a Trust Return (TRT) or Self Managed Superannuation Fund Annual Return (SMSFAR) then the declaration is made by each of the trustee. The “Declaration of Trustee” tuple (includes person name details, organisation name details and declaration) must be completed.

Please note that there can be up to 10 Trustees (Trustee tuple) and the FTER document must have same number of trustee declarations. One trustee declaration must be completed as “TrueAndCorrect”.

If the parent return (TRT or SMSFAR) is submitted by an intermediary, then an intermediary declaration is not required for any attached FTER schedule. However trustee declarations are still required for each FTER schedule.

* + - 1. Trustee declarations for an FTER stand-alone submitted by a self lodger

If the FTER form is submitted by a self lodger (only Reporting Party context) then the Declaration of Trustee tuple (includes person name details, organisation name details and declaration) must be completed. There must be the number of trustee declarations must be equal to the number of trustee tuples. One declaration must be completed as “TrueAndCorrect”.

* + - 1. Trustee declarations for an FTER stand-alone submitted by an intermediary

If the FTER form is submitted by an Intermediary (Reporting Party and Intermediary context) then only Intermediary (sender) declaration must be completed.

* 1. SBDH Variations

The Web Services Implementation Guide (WIG) describes the Standard Business Document Header (SBDH) content in detail. Described in this section are only variations from what is defined in the WIG.

* + 1. Business Documents

When submitted as a stand-alone form, only one FTER Business Document (XBRL instance) will be accepted per message.

When submitted as a schedule to the TRT or SMSFAR, a limit of 50 FTER Business Documents (XBRL instances) may be included within the same message as the TRT or SMSFAR business document.

* + 1. Attachments

No attachments will be accepted or provided for FTER interactions.

* + 1. Document Identifiers

The sbdm:BusinessDocument.GovernmentGeneratedIdentifier.Text field will not be used for FTER interactions.

* 1. Response Messages
     1. Messages Described in the MIG

Where business rules associated with data elements could be reasonably implemented by a Software Developer they have been described in the business content tables below along with an associated response message code. A description of response message codes can be found in Appendix A of this document.

If there is mention of a label or a section in the error message, it is referring to the label or section as displayed on the paper form (PDF available at [www.ato.gov.au](http://www.ato.gov.au/)).

* 1. Validation Phasing

The validation rules described in this document will be applied to the Business Document in phases. Validation will not progress to the next phase until the current phase is completely passed. This is implemented so as to avoid multiple unnecessary messages being returned. As an example, if a FTER business document was not provided this would be an error. If a phased validation approach were not used, then potentially an error for each of the missing mandatory fields could be returned as well.

The phases implemented in the Income Tax suite will be as follows:

1. SBDH checks
2. XBRL contexts, Formats, Data types, lengths and enumerations
3. presence of mandatory fields
4. cross field rules, calculations, comparisons
5. cross form (cross Business Document) rules
   1. Rule expression

Many of the rules in the tables below are written in Tax Office Structured English. This is a type of pseudo code and has been used to ensure clarity in rule expression. For explanations of terms used in Tax Office Structured English see Appendix B.

* + 1. Form Prefix Labels

Since an FTER document may one of multiple business documents existing in one parent return message, cross-form rules may be applied. To ensure it is clear which business document an XBRL fact is from, a form prefix is included in any fact description. In particular, the expression ‘PARENT RETURN’ is used to reference facts in parent return for which RSPT is a listed schedule. For example:   
 PARENT RETURN:RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier   
means this field is in the parent return message, whether that is a SMSFAR or TRT message.

* + 1. Context MIG Labels

Context Instance labels will be given to each possible instance of an XBRL context and will be used within the MIG to describe the context and link a fact to a context.

For example, a fact may appear in a business rule with a prefix TRT:RP.TOFA:bafpr1.xx.xx:Income.FinancialArrangementsUnrealisedGains.Amount

This indicates this fact is being reported in the context where the dimension ReportPartyType is set to “ReportingParty” and the dimension FinancialArrangementType is set to “TOFA”.

* + 1. No Form or Context Prefix

Where no form or context prefix (as described above) applied to a fact within a rule, it is because the rule is reused on multiple forms or it applies regardless of context within the form.

* + 1. Use of xx.xx in Fact Names

In the actual Business Document, an XBRL fact will have a namespace prefix including the version of the element, for example:

bafpr1.02.02:Income.FinancialArrangementsUnrealisedGains.Amount

For the purposes of the MIG, the version contained within the prefix has been replaced with xx.xx. The correct version can be derived from the Discoverable Taxonomy Set.

* + 1. Use of Aliases

In order to make the validation rules more readable aliases have been used in some rules for both fields and formulas. Field alias is enclosed in square brackets e.g. [FTER23] and Formula aliases are described as functions with no parameters e.g. SumCYCL(). A full list of aliases used in this MIG and their definitions are enclosed in Appendix C.

* + 1. Interpretation of NULL in Calculations and Comparisons

Where a rule involves a calculation or a comparison with a number, we will consider NULL (xsi:nil=true) or absent XBRL facts as zero for the purposes of the calculation or comparison.

* + 1. Use of Domain in rules

Where a rule compares an XBRL fact to a range of values, this range may be defined as a Domain. In this case the values within the domain will be specified in Appendix E.

* + 1. Case Sensitivity

Many rules contain a comparison with a string value enclosed in inverted commas e.g. “Australia”. The case used in these comparisons reflects the most common usage, however any comparisons are case insensitive. That means that the test IF <a> = “Australia” would be true if <a> were “australia”, “AUSTRALIA”, “Australia” or “AuStRaLiA”.

* + 1. Tuples and Context

In most SBR ATO interactions, all facts reported within a tuple instance (including nested tuples) use the same context. In the fter.0001 interaction there are some facts within the same tuple that use different contexts.

* + 1. Format Rules

There are format rules that will be applied to facts provided in request documents. Given that they are heavily reused and very simple, they have been included here rather than provided in the message tables below for each fact.

| XBRL data type | Rules | Rule Imp. | SBR Msg Code |
| --- | --- | --- | --- |
| Any | Data supplied does not match the xsd:pattern value | Schematron ID = VR.ATO.GEN.002xxx | CMN.ATO.GEN.001017 |
| xbrli:dateItemType | Date invalid or format incorrect | Schematron ID = VR.ATO.GEN.001008 | CMN.ATO.GEN.001008 |
| xbrli:monetaryItemType | Monetary format incorrect | Schematron ID = VR.ATO.GEN.001009 | CMN.ATO.GEN.001009 |
| xbrli:booleanItemType | Boolean format incorrect | Schematron ID = VR.ATO.GEN.001010 | CMN.ATO.GEN.001010 |
| ID | ID format incorrect | Schematron ID = VR.ATO.GEN.001012 | CMN.ATO.GEN.001012 |
| xbrli:positiveIntegerItemType | Positive integer format incorrect | Schematron ID = VR.ATO.GEN.001013 | CMN.ATO.GEN.001013 |
| xbrli:nonNegativeIntegerItemType | Non negative integer format incorrect | Schematron ID = VR.ATO.GEN.001014 | CMN.ATO.GEN.001014 |
| xbrli:pureItemType | Pure format incorrect | Schematron ID = VR.ATO.GEN.001015 | CMN.ATO.GEN.001015 |
| xbrli:decimalItemType | Decimal format incorrect | Schematron ID = VR.ATO.GEN.001018 | CMN.ATO.GEN.001018 |
| xbrli:fractionItemType | numerator must be a decimal and the denominator must be a non-zero decimal (xbrli:nonZeroDecimal) | Schematron ID = VR.ATO.GEN.001031 | CMN.ATO.GEN.001031 |
| xbrli:floatItemType | Float invalid or format incorrect | Schematron ID = VR.ATO.GEN.001032 | CMN.ATO.GEN.001032 |

1. Business Overview

This section will describe the transmission of the FTER – both as a form and a schedule – in scope for SBR.

* 1. Income tax suite overview

The FTER may be lodged through SBR as a stand-alone form, or as a schedule accompanying a Trust tax return (TRT) or a Self-managed superannuation fund annual return (SMSFAR).

When lodged as a schedule, up to 50 separate FTER may be lodged with the one return form. The schedule may be used for an election, a revocation or a variation.

When lodged as a stand-alone form, the form may be used for an election or a variation (but not a revocation). No schedules may be submitted with the FTER stand-alone form.

* + 1. Message Structure

The message design of the Tax Office Income Tax suite specifies that the return and associated schedules (as required), are all included in one SBR message. The return and each schedule will be separate Business Documents (XBRL instances) within the Standard Business Document Body structure as defined in the web services implementation guide (WIG).

Where the return is lodged with one or more schedules, the return must always be the first business document. Therefore, the BusinessDocumentSequence.Number for the return must be 1 (ie. BusinessDocument.Sequence.Number = 1).

* + 1. Taxonomy and MIG Structure

Within the Income Tax suite each return and schedule has its own reporting taxonomy and MIG. To enable lodgment of an Income Tax Return a software developer must consider these taxonomies and MIGs together.

This MIG covers information relating to the schema for the FTER – both the stand-alone form and schedule versions.

* + 1. Schema Use

Please note that the lodge schema for this report will be used for both the pre-lodge and lodge interactions.

* 1. Business Context Model

This section provides a high level context for all ITR (including FTER) interactions between a business, or an intermediary, and the Tax Office for the purpose of meeting an income tax obligation.

The interactions are broadly divided into five phases:

*Register and maintain* – Obtaining an AUSkey from the Australian Business Register (ABR) to be used by the Tax Office to authenticate interactions. Intermediaries must also register with the Tax Office to allow the Tax Office to create an intermediary relationship with any taxpayers they will lodge for via SBR. These interactions are out of scope for SBR web services implementation at this time. This document does not describe these interactions any further, but includes them here to provide context.

*Notify obligations –* The Tax Office will notify a business of any impending obligations. This interaction is external to the SBR lodgement process and therefore is out of scope for SBR web services implementation and will not be described further in this document; however it will be included for context.

*Prepare and lodge interactions –* The Tax Office provides, via SBR, services to validate the ITR before lodging the form (prelodge), and the lodgement of ITR. These interactions are within the scope for SBR implementation and will be described further in this document.

*Post Lodgement phase* – The Tax office will notify the business/intermediary of the outcome of the trust tax assessment. The interactions within this phase may include enquiries made by the business or intermediary regarding the progress of a trust tax assessment. These interactions are outside of the SBR lodgement process and therefore are out of scope for SBR web services implementation and will not be described further in this document; however it will be included for context.

*Payment interactions* – Monetary transactions between the business and the Tax Office, in the form of payment or refunds, are determined by their trust tax assessment. Payments are made through various mechanisms and are outside of the scope for SBR implementation and will not be described further in this document; however has been included for context.

Each of the phases described above could be performed by a number of different channels such as paper, Business portal, Tax agent portal, ELS and Web services. In the context of income tax interactions the Tax Office will enable the third phase (Prepare and Lodge) via the SBR channel. The remaining phases are out of scope of this document since they are performed by other existing Tax Office channels or the ABR.

Details of legislative requirements and other background on all ITR are available at [www.ato.gov.au](http://www.ato.gov.au/).

* 1. Financial Year and Substituted Accounting Periods

Most reporting parties will report Income Tax over the standard Australian financial year (1st July to 30th June). Some Reporting Parties will have a specific arrangement with the Tax Office to report Income Tax over a different financial year period called a Substituted Accounting Period (SAP). If the Reporting Party operates on a SAP, wherever duration is specified for the report (for example in the RP context instance), instead of the financial dates, the **SAP** start and end dates must be supplied for the Period Start and Period End dates.

If lodged as a schedule, the Period Start and End dates must be the same as on the parent return.

When lodged as a stand-alone form, if the income year for the FTER does not end on 30 June, the SAP start and end dates must be supplied for the Period Start and Period End dates, instead of the financial year dates. These SAP dates are as would be supplied in response to Question 6 on the paper FTER form.

* 1. Business Applicability Period

In SBR, the FTER may not necessarily have a new reporting taxonomy released each year. A new reporting taxonomy will be published where updates reflect legislative changes required for that particular year. For example, in 2011 the FTER.0001 reporting taxonomy will be published on 23 August 2011, and in 2012 the reporting taxonomy has changes. The FTER.0001 reporting taxonomy will be valid for both 2011 and 2012. Where a new reporting taxonomy is published, an expiry date for the previous reporting taxonomy will not be specified as it will continue to be valid for the relevant income years into the future.

Within an income year’s reporting taxonomy, if a schema is versioned (a new one is released in production) the previous schema will be supported for an appropriate transition period

* 1. Report Version

The SBR report version for FTER 2012 is **fter.0001.02.00**

1. XBRL Context Specifications

The following sections define the context specifications that will be used within this MIG. The context instances are allocated to the individual data elements within the message specifications below.

* 1. Context Specification Dimension 1: ReportPartyType, Period: Duration

| **XBRL Instance Context Data Concept** | **Requirement** | **Instructions/Rules** | **Rule Imp** | **SBR Msg code** | |
| --- | --- | --- | --- | --- | --- |
| Context Identifier | Mandatory | This is a unique identifier used to link the data element to a defined XBRL context. SBR is recommending a four character id starting with ‘C’ and a three digit sequential number for each context e.g. C001.  1. IF context id = NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.000241 | | 1. CMN.ATO.GEN. 430298 |
| Entity Identifier | Mandatory | An identifier must be supplied.  1. IF entity.identifier.TFN = NULLORBLANK WHERE CONTEXT <> "INT"  RETURN VALIDATION MESSAGE ENDIF  2. IF entity.identifier.ABN = NULLORBLANK WHERE CONTEXT = "INT"  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.430317  2. Schematron ID = VR.ATO.GEN.430316 | | 1. CMN.ATO.GEN.430317  2. CMN.ATO.GEN.430316 |
| Entity Identifier Scheme | Mandatory | This field must be set to **http://www.ato.gov.au/abn** or **http://www.ato.gov.au/tfn**. TFN must be used for the reporting party and ABN must be used for the Intermediary. | See below | | See below |
| Entity Segment | Mandatory | Explicit member dimension ReportPartyType set to “ReportingParty” or “Intermediary”  1. IF (RprtPyType.xx.xx:ReportPartyTypeDimension <> "RprtPyType.02.03:Intermediary") AND (RprtPyType.xx.xx:ReportPartyTypeDimension <> "RprtPyType.02.03:ReportingParty")   RETURN VALIDATION MESSAGE ENDIF  2. IF (RprtPyType.xx.xx:ReportingPartyTypeDimension = “RprtPyType.02.03:ReportingParty”) AND (entity.identifier.scheme <> http://www.ato.gov.au/tfn)  RETURN VALIDATION MESSAGE ENDIF  3. IF (RprtPyType.xx.xx:ReportingPartyTypeDimension = “RprtPyType.02.03:Intermediary”) AND (entity.identifier.scheme <> http://www.ato.gov.au/abn)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.410197  2. Schematron ID = VR.ATO.GEN. 432265  3. Schematron ID = VR.ATO.GEN. 432266 | | 1. CMN.ATO.GEN.430299  2. CMN.ATO.GEN.001024  3. CMN.ATO.GEN.001025 |
| Period Date - Start Date | Mandatory | The start date should describe the start of the period that the return applies to. If the company operates on a standard financial year, this date would be set to 1/7/10 for a 2011 tax return. If the company operates on a Substitute Accounting Period (SAP), this date should be set to the start of the SAP. (The dates supplied here are used to record the SAP dates referred to in Question 6 of the paper form).  1. IF period.startDate = NULL WHERE NOT CONTEXT(ALL)  RETURN VALIDATION MESSAGE ENDIF  2. IF period.startDate >= period.endDate WHERE NOT CONTEXT(ALL)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.428252  2. Schematron ID = VR.ATO.GEN.000201 | | 1. CMN.ATO.GEN.001001  2. CMN.ATO.GEN.200009 |
| Period Date - End Date | Mandatory | The end date should describe the end of the period that the return applies to. If the company operates on a standard financial year, this date would be set to 30/06/11 for a 2011 tax return. If the company operates on a Substitute Accounting Period (SAP), this date should be set to the end of the SAP. . (The dates supplied here are used to record the SAP dates referred to in Question 6 of the paper form).  1. IF period.endDate = NULLORBLANK WHERE NOT CONTEXT(ALL)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.000237 | | 1.CMN.ATO.GEN.001001 |

* + 1. Context Instances

| **Context Instance MIG Label** | **Dimensions with Constrained Values** | **Instructions/Rules** | **Rule Imp** | **SBR Msg code** |
| --- | --- | --- | --- | --- |
| **ReportPartyTypeDimension** |
| RP | RprtPyType.xx.xx:ReportingParty | Every FTER document must have an ‘RP’ context that is identified by tax file number  of the reporting party. When used as a schedule, the FTER document must apply to the same reporting party as that on the parent form.  1. IF COUNT(RP) <> 1  RETURN VALIDATION MESSAGE ENDIF  2. IF (RP:entity.identifier.TFN <> NULLORBLANK) AND (TFNALGORITHM (RP:entity.identifier.TFN) = FALSE)  RETURN VALIDATION MESSAGE ENDIF  3. WHERE PARENT RETURN EXISTS  IF (FTER:RP:entity.identifier.TFN <> NULLORBLANK) AND (FTER:RP:entity.identifier.TFN <> PARENT RETURN:RP:entity.identifier.TFN)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.000209  2. Schematron ID = VR.ATO.GEN.428016  3. Schematron ID = VR.ATO.FTER.408006 | 1. CMN.ATO.GEN.430296  2. CMN.ATO.GEN.428016  3. CMN.ATO.FTER.408006 |
| INT | RprtPyType .xx.xx:Intermediary | An FTER document can be submitted by not more than one intermediary who is indentified by their Australian Business Number. A schedule must not have its own intermediary as the intermediary on the main form will submit and declare any schedules attached to the main form.  1. IF COUNT(INT) > 1  RETURN VALIDATION MESSAGE ENDIF  2. IF (INT:entity.identifier.ABN <> NULLORBLANK) AND (ABNALGORITHM(INT:entity.identifier.ABN) = FALSE)  RETURN VALIDATION MESSAGE ENDIF  3. WHERE PARENT RETURN EXISTS IF COUNT (INT) > 0  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.000227  2. Schematron ID = VR.ATO.GEN.428253  3. Schematron ID = VR.ATO.FTER.408001 | 1. CMN.ATO.GEN.436280  2. CMN.ATO.GEN.200011  3. CMN.ATO.GEN.410115 |

1. Interaction Model
   1. Prerequisites

Prior to performing any of the SBR FTER interactions described in this document, the sender must have:

Registered as an Australian business and obtained an ABN;

Registered for business roles (for eg: ITR (Income Tax Return) role) with the ABR or ATO;

Obtained an AUSkey from the ABR; and

Registered people with ‘User’ level SBR credentials to perform FTER tasks with the Tax Office or ABR. ‘Admin’ level SBR credentials will be authorised for any task for that ABN.

If an intermediary is lodging on behalf of a business, then the intermediary must have:

Registered as an Australian business and obtained an ABN;

Obtained an AUSkey from the ABR; and

Registered an ‘intermediary-taxpayer’ relationship with the Tax Office for FTER lodgements, or had the taxpayer business do so.

On submitting a request via SBR, the Tax Office will verify that the sender is authorised to perform that task for the business that is reporting the FTER.

If lodging via an intermediary, the business (taxpayer) may not need an ABN unless their activities (eg: GST) require them to have one. Under this scenario, the business’ TFN will be used to identify the reporting party within the XBRL instance.

* 1. Service Summary

The workflow defined above represents the abstract process model for the FTER. The following defines the sequence, optionality and repeatability of the service execution. Where a service is identified as 'Mandatory' then the Software Developer has to implement this service as the service needs to be executed by the business in order to complete the transaction. Repeatability indicates if the service can be executed more than once – 'Yes' meaning that a service can be executed multiple times within the sequence.

|  | Service Name | Mandatory | Repeatability |
| --- | --- | --- | --- |
| 1 | **prelodge**  The purpose of this transaction is to request pre-submission validation of a FTER, as a stand-alone form or with accompanying a parent form, without processing the return. | No | Yes |
| 2 | **lodge**  The purpose of this transaction is to allow a client to lodge a FTER for processing, either as a stand-alone form or with accompanying a parent form. | Yes | No |

* 1. FTER.PRELODGE Specifications

|  |  |
| --- | --- |
| Interaction Name | fter.prelodge |
| Description | The purpose of this transaction is to request pre-submission validation of a FTER and any accompanying schedule(s) without processing the return. This transaction is optional. |
| Stakeholders | Reporting party, Intermediary, Tax Office |
| Pre-conditions | See Pre-conditions for fter.lodge |
| Post-conditions | For a successful pre-lodgement validation the Tax Office will return:   * A message event item informing successful validation.   For an unsuccessful pre-lodgement validation the Tax Office will return:   * One or more message event item(s) containing a list of errors.   The Reporting party/Intermediary is then able to correct their Return and/or schedule(s) for re-submission. |
| Initiating party | Reporting party, Intermediary |
| Channel | SBR |
| Core Service Map | Prelodge |

* + 1. FTER.PRELODGE Request - Message
       1. Discoverable Taxonomy Set References

Refer to FTER.LODGE Request – Message (Section 5.4.1)

* + - 1. Standard Business Document Header Content

The WIG provides the specification of the SBDH. The following table specifies the message specific data element values or any variations to the WIG.

| Attribute Name | Instructions / Rules | Rule Imp | SBR Msg code |
| --- | --- | --- | --- |
| sbdm:Message.Type.Text | Mandatory - Value must be “fter.0001.prelodge.request” | MIG | CMN.ATO.GEN.100002 |

* + - 1. Standard Business Document Body Content

##### FTER.PRELODGE Request Message Content Table

Refer to FTER.LODGE Request – Message (Section 5.4.1)

* + 1. FTER.PRELODGE Response - Message
       1. Discoverable Taxonomy Set References

Refer to FTER.LODGE Response – Message

* + - 1. Standard Business Document Header Content

The WIG provides the specification of the SBDH. The following table specifies the message specific data element values or any variations to the WIG.

| Attribute Name | Instructions / Rules | Rule Imp | SBR Msg code |
| --- | --- | --- | --- |
| sbdm:Message.Type.Text | Mandatory - Value must be “fter.0001.prelodge.response” | MIG | CMN.ATO.GEN.100002 |

* + - 1. Standard Business Document Body Content

The following describes the facts and context required to be supplied within the XBRL instance document populated into the SBDB element BusinessDocument.instance.text

##### FTER.PRELODGE Response XBRL Context

Refer to FTER.LODGE Response – Message (Section 5.4.2)

* 1. FTER.LODGE Specifications

|  |  |
| --- | --- |
| Interaction Name | fter.lodge |
| Description | The purpose of this transaction is to allow a client to lodge a FTER and its accompanying schedules(s) for processing. |
| Stakeholders | Reporting party, Intermediary, Tax Office |
| Pre-conditions | See Pre-requisites for fter.prelodge (Section 5.1) |
| Post-conditions | For a successful lodgement the Tax Office will return:   * A message event item with receipt details informing successful lodgement.   For an unsuccessful lodgement validation the Tax Office will return:   * One or more message event item(s) containing a list of errors.   The Reporting party/Intermediary is then able to correct their Return and/or schedule(s) for re-submission or to confirm their lodgement has been accepted for processing. |
| Initiating party | Reporting party, Intermediary |
| Channel | SBR |
| Core Service Map | Lodge |

* + 1. FTER.LODGE Request - Message
       1. Discoverable Taxonomy Set References

|  |  |
| --- | --- |
| **Schema** | fter.0001.lodge.request.02.00.report.xsd  fter.0001.private.02.00.module.xsd |
| **Linkbases** | fter.0001.lodge.request.02.00.defLink.xml  fter.0001.private.02.00.defLink.xml |
| fter.0001.lodge.request.02.00.labLinkInfoCls.xml  fter.0001.private.02.00.labLinkInfoCls.xml |
| fter.0001.lodge.request.02.00.presLink.xml |
| fter.0001.lodge.request.02.00.refLink.xml |
| **Example Instance** | To be advised |
| **Schematron** | To be advised |

* + - 1. Standard Business Document Header Content

##### The WIG provides the specification of the SBDH. The following table specifies the message specific data element values or any variations to the WIG.

| Attribute Name | Instructions/Rules | SBR Msg Code |
| --- | --- | --- |
| sbdm:Message.Type.Text | Mandatory - value must be “fter.0001.lodge.request” | CMN.ATO.GEN.100002 |
| Sbdm:BusinessDocument.BusinessGenerated Identifier.Text | Mandatory - should be set to a unique identifier for the document being lodged. | CMN.ATO.GEN.100001 |

* + - 1. Standard Business Document Body Content

The following describes the facts and context required to be supplied within the XBRL instance document populated into the SBDB element BusinessDocument.instance.text

##### FTER.LODGE Response XBRL Context

Refer to Section 4 (XBRL Context Specifications)

* + - 1. FTER.LODGE Request Message Content Table

The following table contains the facts – tuples and elements – required in the instance document, together with their context and applicable validation rules. To improve readability, aliases are used in the rules to define most elements. These appear as a letter-number combination enclosed between square brackets, such as ‘[FTER]’. Validation rule aliases are fully defined in Appendix C.

Rules that apply to all instances of some common modules, such as Address, Organisation Name and Person Name tuples, are defined separately as ‘Common module validation rules’ in Appendix D. A reference to the ‘common ruleset’ appears in the Instructions / Rules column against the first element within each applicable tuple.

| **Context - INT** | | | | |
| --- | --- | --- | --- | --- |
| **Seq. No** | **XBRL Fact** | **Instructions / Rules** | **Rule Imp** | **SBR Msg Code** |
| 1 | pyid.xx.xx:Identifiers.TaxAgentNumber.Identifier | 1. IF ((pyid.xx.xx:Identifiers.TaxAgentNumber.Identifier) <> NULLORBLANK) AND (TANALGORITHM(pyid.xx.xx:Identifiers.TaxAgentNumber.Identifier) = FALSE) RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.428247 | 1. CMN.ATO.GEN.410009 |
| 2 | declaration2.xx.xx:Declaration (Tuple 0..1) | N/A | N/A | N/A |
| 2.1 | pyin.xx.xx:Declaration.StatementType.Code | Common rule set declaration2.xx.xx:Declaration does not apply to FTER  1. WHERE IN TUPLE(Declaration2.xx.xx:Declaration) IF pyin.xx.xx:Declaration.StatementAccepted.Indicator = FALSE OR Declaration.SignatoryIdentifier.Text = NULLORBLANK OR Declaration.Signature.Date = NULL   RETURN VALIDATION MESSAGE ENDIF  2. IF ([FTER159] <> NULLORBLANK) AND ([FTER159] <> "TrueAndCorrect" )  RETURN VALIDATION MESSAGE ENDIF  3. IF INT <> NULLORBLANK AND INT:pyin.xx.xx:Declaration.StatementAccepted.Indicator = NULL  RETURN VALIDATION MESSAGE ENDIF  4. IF pyde.xx.xx:Declaration.Signature.Date <> NULL AND pyde.xx.xx:Declaration.Signature.Date > DATE(TODAY)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408166  2. Schematron ID = VR.ATO.FTER.408167  3. Schematron ID = VR.ATO.FTER.408168  4. Schematron ID = VR.ATO.GEN.430255 | 1. CMN.ATO.FTER.408166  2. CMN.ATO.GEN.430250  3. CMN.ATO.GEN.430249  4. CMN.ATO.GEN.430255 |
| 2.2 | pyin.xx.xx:Declaration.StatementAccepted.Indicator | N/A | N/A | N/A |
| 2.3 | pyin.xx.xx:Declaration.Statement.Text | N/A | N/A | N/A |
| 2.4 | pyin.xx.xx:Declaration.Signature.Date | N/A | N/A | N/A |
| 2.5 | pyin.xx.xx:Declaration.SignatoryIdentifier.Text | N/A | N/A | N/A |
| 2.6 | personunstructuredname1.xx.xx:PersonUnstructuredName (Tuple 0..1) | N/A | N/A | N/A |
| 2.6.1 | pyde.xx.xx:PersonUnstructuredName.Usage.Code | N/A | N/A | N/A |
| 2.6.2 | pyde.xx.xx:PersonUnstructuredName.FullName.Text | 1. IF LENGTH(pyde.xx.xx:PersonUnstructuredName.FullName.Text) > 200  RETURN VALIDATION MESSAGE ENDIF  2. IF pyde.xx.xx:PersonUnstructuredName.FullName.Text <> NULLORBLANK AND pyde.xx.xx:PersonUnstructuredName.FullName.Text <> SET("a-z", "A-Z", "-", " ")  RETURN VALIDATION MESSAGE ENDIF  3. WHERE IN TUPLE (personunstructuredname1.xx.xx:PersonUnstructuredName) IN TUPLE(declaration2.xx.xx:Declaration) IF pyde.xx.xx:PersonUnstructuredName.Usage.Code <> "DeclarationSignatory"  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.430252  2. Schematron ID = VR.ATO.GEN.430253  3. Schematron ID = VR.ATO.GEN.436279 | 1. CMN.ATO.GEN.430252  2. CMN.ATO.GEN.430253  3. CMN.ATO.GEN.436279 |
| 3 | personunstructuredname1.xx.xx:PersonUnstructuredName (Tuple 0..1) | N/A | N/A | N/A |
| 3.1 | pyde.xx.xx:PersonUnstructuredName.Usage.Code | N/A | N/A | N/A |
| 3.2 | pyde.xx.xx:PersonUnstructuredName.FullName.Text | 1. WHERE IN TUPLE (xbrli\personunstructuredname1.xx.xx:PersonUnstructuredName) IF INT:pyde.xx.xx:PersonUnstructuredName.Usage.Code <> "Contact"  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.432391 | 1. CMN.ATO.GEN.432391 |
| 4 | electroniccontacttelephone1.xx.xx:ElectronicContactTelephone (Tuple 0..1) | N/A | N/A | N/A |
| 4.1 | pyde.xx.xx:ElectronicContact.Telephone.Usage.Code | N/A | N/A | N/A |
| 4.2 | pyde.xx.xx:ElectronicContact.Telephone.ServiceLine.Code | N/A | N/A | N/A |
| 4.3 | pyde.xx.xx:ElectronicContact.Telephone.Area.Code | N/A | N/A | N/A |
| 4.4 | pyde.xx.xx:ElectronicContact.Telephone.Minimal.Number | Common rule set electroniccontacttelephone1.xx.xx:ElectronicContactTelephone applies to this tuple | Ruleset electroniccontacttelephone1 | N/A |

| **Context - RP** | | | | |
| --- | --- | --- | --- | --- |
| **Seq. No** | **XBRL Fact** | **Instructions / Rules** | **Rule Imp** | **SBR Msg Code** |
| 1 | pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier | 1. WHERE PARENT RETURN DOES NOT EXISTS IF COUNT(INT) = 0 AND RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier = NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF  2. IF (RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier <> NULLORBLANK) AND (PARENT RETURN:RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier <> NULLORBLANK) AND (RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier <> PARENT RETURN:RP:pyid.xx.xx:Identifiers.AustralianBusinessNumber.Identifier) RETURN VALIDATION MESSAGE ENDIF  3. IF (pyid.xx.xx:IdentifiersAustralianBusinessNumberIdentifer <> NULLORBLANK) and (ABNALGORITHM(pyid.xx.xx:IdentifiersAustralianBusinessNumber.Identifer) = FALSE) RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408165  2. Schematron ID = VR.ATO.GEN.402010  3. Schematron ID = VR.ATO.GEN.428040 | 1. CMN.ATO.FTER.408165  2. CMN.ATO.GEN.402010  3. CMN.ATO.GEN.000477 |
| 2 | pyin.xx.xx:Lodgment.PeriodYear.Year | 1. WHERE PARENT RETURN DOES NOT EXIST IF (RP:pyin.xx.xx.Lodgment.PeriodYear.Year = NULL)  RETURN VALIDATION MESSAGE ENDIF  2. IF (RP:pyin.xx.xx.Lodgment.PeriodYear.Year <> NULL) AND (RP:pyin.xx.xx:Lodgment.PeriodYear.Year < 2011)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408023  2. Schematron ID = VR.ATO.FTER.408161 | 1. CMN.ATO.FTER.408023  2. CMN.ATO.FTER.408161 |
| 3 | rvctc3.xx.xx:Elections.FamilyTrustElectionRevocation.Code | 1. IF [FTER95] = NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF  2. WHERE PARENT RETURN EXISTS IF [FTER95] <> NULLORBLANK AND [FTER95] <> SET("E", "R", "V")  RETURN VALIDATION MESSAGE ENDIF  3. WHERE PARENT RETURN DOES NOT EXIST IF [FTER95] <> "E"  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408022  2. Schematron ID = VR.ATO.FTER.408024  3. Schematron ID = VR.ATO.FTER.408025 | 1. CMN.ATO.FTER.408022  2. CMN.ATO.FTER.408024  3. CMN.ATO.FTER.408025 |
| 4 | organisationname2.xx.xx:OrganisationNameDetails (Tuple 0..2) | N/A | N/A | N/A |
| 4.1 | pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code | Common rule set organisationname2.xx.xx:OrganisationNameDetails applies to this tuple | Ruleset:organisationname2 | N/A |
| 4.2 | pyde.xx.xx:OrganisationNameDetails.Currency.Code | N/A | N/A | N/A |
| 4.3 | pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text | 1. WHERE IN TUPLE(xbrli\organisationname2.xx.xx:OrganisationNameDetails) IF pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code <> "MN"   RETURN VALIDATION MESSAGE ENDIF  2. WHERE PARENT RETURN EXISTS IF [FTER4] <> NULLORBLANK AND [FTER4] <> (PARENT RETURN:xbrli\organisationname2.xx.xx.OrganisationName.Text WHERE TUPLE ELEMENT EXPLICIT OrganisationNameDetails.Currency.Code = "C" AND TUPLE ELEMENT EXPLICIT OrganisationNameDetails.OrganisationalNameType.Code = "MN")  RETURN VALIDATION MESSAGE ENDIF  3. WHERE PARENT RETURN DOES NOT EXIST IF COUNT (TUPLE(xbrli\organisationname2.xx.xx:OrganisationNameDetails) WHERE TUPLE ELEMENT EXPLICIT OrganisationNameDetails.Currency.Code = "C" AND TUPLE ELEMENT EXPLICIT OrganisationNameDetails.OrganisationalNameType.Code = "MN" AND TUPLE ELEMENT EXPLICIT OrganisationNameDetails.OrganisationalName.Text <> NULLORBLANK) <> 1  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408009  2. Schematron ID = VR.ATO.FTER.408144  3. Schematron ID = VR.ATO.FTER.408145 | 1. CMN.ATO.FTER.408009  2. CMN.ATO.FTER.408144  3. CMN.ATO.FTER.408145 |
| 5 | address2.xx.xx:AddressDetails (Tuple 1..2) | N/A | N/A | N/A |
| 5.1 | pyde.xx.xx:AddressDetails.OverseasAddress.Indicator | Common ruleset address2.xx.xx:AddressDetails applies to this tuple | Ruleset:address2 | N/A |
| 5.2 | pyde.xx.xx:AddressDetails.Usage.Code | N/A | N/A | N/A |
| 5.3 | pyde.xx.xx:AddressDetails.Currency.Code | N/A | N/A | N/A |
| 5.4 | pyde.xx.xx:AddressDetails.Line1.Text | 1. WHERE IN TUPLE (xbrli\address2.xx.xx.AddressDetails) IF (AddressDetails.Usage.Code <> "POS")  RETURN VALIDATION MESSAGE ENDIF  2. WHERE PARENT RETURN DOES NOT EXIST IF COUNT (TUPLE(xbrli\address2.xx.xx.AddressDetails) WHERE TUPLE ELEMENT EXPLICIT pyde.xx.xx:AddressDetails.Currency.Code = "C" AND TUPLE ELEMENT EXPLICIT pyde.xx.xx:AddressDetails.Usage.Code = "POS" AND TUPLE ELEMENT EXPLICIT pyde.xx.xx:AddressDetails.Line1.Text <> NULLORBLANK) <> 1   RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408008  2. Schematron ID = VR.ATO.FTER.408146 | 1. CMN.ATO.FTER.408008  2. CMN.ATO.FTER.408146 |
| 5.5 | pyde.xx.xx:AddressDetails.Line2.Text | N/A | N/A | N/A |
| 5.6 | pyde.xx.xx:AddressDetails.Line3.Text | N/A | N/A | N/A |
| 5.7 | pyde.xx.xx:AddressDetails.Line4.Text | N/A | N/A | N/A |
| 5.8 | pyde.xx.xx:AddressDetails.LocalityName.Text | N/A | N/A | N/A |
| 5.9 | pyde.xx.xx:AddressDetails.Postcode.Text | N/A | N/A | N/A |
| 5.10 | pyde.xx.xx:AddressDetails.StateOrTerritory.Code | N/A | N/A | N/A |
| 5.11 | pyde.xx.xx:AddressDetails.CountryName.Text | N/A | N/A | N/A |
| 5.12 | pyde.xx.xx:AddressDetails.Country.Code | N/A | N/A | N/A |
| 6 | pyde.xx.xx:Residency.CentralManagementAndControlOutsideAustralia.Indicator | 1. IF [FTER95] = "V" AND ( [FTER16] <> NULL OR [FTER17] <> NULL OR COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod")) > 0 OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:Trustee)) > 0 )  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER16] = TRUE AND [FTER17] <> TRUE AND COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod")) = 0  RETURN VALIDATION MESSAGE ENDIF  3. IF [FTER16] = FALSE AND ([FTER17] <> NULL OR COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod"))) > 0  RETURN VALIDATION MESSAGE ENDIF  4. IF [FTER95] = "E" AND ([FTER16] = NULL OR [FTER40] = NULL OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual)) <> 1 OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:Trustee)) = 0)  RETURN VALIDATION MESSAGE ENDIF  5. IF ([FTER16] <> NULL OR [FTER17] <> NULL OR COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails) WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod") > 0 OR COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails) WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "SubstitutedAccountingPeriod") > 0 OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:Trustee)) > 0 OR [FTER43] <> NULL OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual)) > 0) AND ([FTER57] <> NULL OR [FTER96] <> NULL OR [FTER58] <> NULL OR [FTER60] <> NULLORBLANK OR [FTER61] <> NULL OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity)) > 0)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408031  2. Schematron ID = VR.ATO.FTER.408051  3. Schematron ID = VR.ATO.FTER.408052  4. Schematron ID = VR.ATO.FTER.408053  5. Schematron ID = VR.ATO.FTER.408054 | 1. CMN.ATO.FTER.408031  2. CMN.ATO.FTER.408051  3. CMN.ATO.FTER.408052  4. CMN.ATO.FTER.408053  5. CMN.ATO.FTER.408054 |
| 7 | pyde.xx.xx:Residency.CentralManagementAndControlOutsideAustraliaFullPeriod.Indicator | 1. IF [FTER17] = TRUE AND COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod")) > 0  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408055 | 1. CMN.ATO.FTER.408055 |
| 8 | perioddetails1.xx.xx:PeriodDetails (Tuple 0..13) | N/A | N/A | N/A |
| 8.1 | pyin.xx.xx:Period.Type.Code | Common rule set perioddetails1.xx.xx.PeriodDetails applies to this tuple  1. WHERE IN TUPLE (xbrli\perioddetails1.xx.xx.PeriodDetails) IF pyin.xx.xx:Period.Type.Code <> NULLORBLANK AND pyin.xx.xx:Period.Type.Code <> SET("ControlOutsideAustraliaPartPeriod", "SubstitutedAccountingPeriod")  RETURN VALIDATION MESSAGE ENDIF | Ruleset:perioddetails1  1. Schematron ID = VR.ATO.GEN.408058 | 1. CMN.ATO.GEN.408058 |
| 8.2 | pyin.xx.xx:Period.Start.Date | 1. IF COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod")) > 12  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER18] <> NULL AND [FTER41] <> NULL AND (ANY OCCURRENCE OF [FTER18]) < [FTER41]  RETURN VALIDATION MESSAGE ENDIF  3. IF [FTER18] <> NULL AND [FTER43] <> NULL AND (ANY OCCURRENCE OF [FTER18]) < [FTER43]  RETURN VALIDATION MESSAGE ENDIF  4. IF [FTER95] = SET("E", "V") AND [FTER40] <> NULL AND [FTER41] = NULL AND [FTER18] <> NULL AND (ANY OCCURRENCE OF [FTER18] < (([FTER40] - 1)&"-07-01"))  RETURN VALIDATION MESSAGE ENDIF  5. IF COUNT (TUPLE(xbrli\perioddetails1.xx.xx.PeriodDetails WHERE TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "SubstitutedAccountingPeriod")) > 1  RETURN VALIDATION MESSAGE ENDIF  6. IF [FTER43] <> NULL AND [FTER41] <> NULL AND [FTER43] < [FTER41]  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408056  2. Schematron ID = VR.ATO.FTER.408061  3. Schematron ID = VR.ATO.FTER.408062  4. Schematron ID = VR.ATO.FTER.408063  5. Schematron ID = VR.ATO.FTER.408091  6. Schematron ID = VR.ATO.FTER.408096 | 1. CMN.ATO.FTER.408056  2. CMN.ATO.FTER.408061  3. CMN.ATO.FTER.408062  4. CMN.ATO.FTER.408063  5. CMN.ATO.FTER.408091  6. CMN.ATO.FTER.408096 |
| 8.3 | pyin.xx.xx:Period.Type.Code | Common rule set perioddetails1.xx.xx.PeriodDetails applies to this tuple  1. WHERE IN TUPLE (xbrli\perioddetails1.xx.xx.PeriodDetails) IF pyin.xx.xx:Period.Type.Code <> NULLORBLANK AND pyin.xx.xx:Period.Type.Code <> SET("ControlOutsideAustraliaPartPeriod", "SubstitutedAccountingPeriod")  RETURN VALIDATION MESSAGE ENDIF | Ruleset:perioddetails1  1. Schematron ID = VR.ATO.GEN.408058 | 1. CMN.ATO.GEN.408058 |
| 9 | fter.0001.lodge.req.xx.xx:Trustee (Tuple 0..10) | N/A | N/A | N/A |
| 9.1 | pyid.xx.xx:Identifiers.TaxFileNumber.Identifier | 1. IF COUNT (TUPLE(fter.0001.lodge.req.xx.xx:Trustee)) > 10  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) IF ([FTER21] = NULL OR [FTER21] = FALSE) AND [FTER20] = NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF  3. IF (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier <> NULLORBLANK) AND (TFNALGORITHM (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier) = FALSE)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408064  2. Schematron ID = VR.ATO.FTER.408066  3. Schematron ID = VR.ATO.GEN.410031 | 1. CMN.ATO.FTER.408064  2. CMN.ATO.FTER.408066  3. CMN.ATO.GEN.410031 |
| 9.2 | pyid.xx.xx:Identifiers.TaxFileNumberInexistent.Indicator | 1. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) IF [FTER21] = TRUE AND [FTER20] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408065 | 1. CMN.ATO.FTER.408065 |
| 9.3 | personstructuredname3.xx.xx:PersonNameDetails (Tuple 0..1) | N/A | N/A | N/A |
| 9.3.1 | pyde.xx.xx:PersonNameDetails.PersonNameType.Code | Common rule set personstructuredname3.xx.xx:PersonNameDetails applies to this tuple | Ruleset:personstructuredname3 | N/A |
| 9.3.2 | pyde.xx.xx:PersonNameDetails.Usage.Code | N/A | N/A | N/A |
| 9.3.3 | pyde.xx.xx:PersonNameDetails.Currency.Code | N/A | N/A | N/A |
| 9.3.4 | pyde.xx.xx:PersonNameDetails.Title.Text | N/A | N/A | N/A |
| 9.3.5 | pyde.xx.xx:PersonNameDetails.NameSuffix.Text | N/A | N/A | N/A |
| 9.3.6 | pyde.xx.xx:PersonNameDetails.FamilyName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IF (COUNT (TUPLE(personstructuredname3.xx.xx:PersonNameDetails)) = 0 AND COUNT (TUPLE(organisationname2.xx.xx:OrganisationNameDetails)) = 0) OR COUNT (TUPLE(address2.xx.xx:AddressDetails)) = 0  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IF COUNT (TUPLE(personstructuredname3.xx.xx:PersonNameDetails)) > 0 AND COUNT (TUPLE(organisationname2.xx.xx:OrganisationNameDetails)) > 0  RETURN VALIDATION MESSAGE ENDIF  3. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee/personstructuredname3.xx.xx:PersonNameDetails) IF (PersonNameDetails.PersonNameType.Code <> "LGL") OR (OrganisationNameDetails.Usage.Code <> "Contact") OR (OrganisationNameDetails.Currency.Code <> "C")   RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408003  2. Schematron ID = VR.ATO.FTER.408071  3. Schematron ID = VR.ATO.FTER.408157 | 1. CMN.ATO.FTER.408003  2. CMN.ATO.FTER.408071  3. CMN.ATO.GEN.432435 |
| 9.3.7 | pyde.xx.xx:PersonNameDetails.GivenName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IN TUPLE(prsnstrcnm3.xx.xx:PersonNameDetails)  IF pyde.xx.xx:PersonNameDetails.FamilyName.Text <> NULLORBLANK AND pyde.xx.xx:PersonNameDetails.GivenName.Text = NULLORBLANK  RETURN VALIDATION MESSAGE   ENDIF | 1. Schematron ID = VR.ATO.FTER.408072 | 1. CMN.ATO.FTER.408072 |
| 9.3.8 | pyde.xx.xx:PersonNameDetails.OtherGivenName.Text | N/A | N/A | N/A |
| 9.4 | organisationname2.xx.xx:OrganisationNameDetails (Tuple 0..1) | N/A | N/A | N/A |
| 9.4.1 | pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code | Common rule set organisationname2.xx.xx:OrganisationNameDetails applies to this tuple | Ruleset:organisationname2 | N/A |
| 9.4.2 | pyde.xx.xx:OrganisationNameDetails.Currency.Code | N/A | N/A | N/A |
| 9.4.3 | pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee\organisationname2.xx.xx:OrganisationNameDetails)  IF (OrganisationNameDetails.OrganisationalNameType.Code <> "MN") OR (OrganisationNameDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408007 | 1. CMN.ATO.FTER.408007 |
| 9.5 | pyid.xx.xx:Identifiers.AustralianCompanyNumber.Identifier | 1. IF [FTER86] <> NULLORBLANK AND [FTER29] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF  2. IF (pyid.xx.xx:Identifiers.AustralianCompanyNumber.Identifier <> NULLORBLANK) AND (ACNALGORITHM(pyid.xx.xx:Identifiers.AustralianCompanyNumber.Identifier) = FALSE)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408121  2. Schematron ID = VR.ATO.GEN.428021 | 1. CMN.ATO.FTER.408122  2. CMN.ATO.GEN.428021 |
| 9.6 | pyid.xx.xx:Identifiers.AustralianRegisteredBodyNumber.Identifier | 1. IF (pyid.xx.xx:Identifiers.AustralianRegisteredBodyNumber.Identifier <> NULLORBLANK) AND (ARBNALGORITHM (pyid.xx.xx:Identifiers.AustralianRegisteredBodyNumber.Identifier) = FALSE)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.410105 | 1. CMN.ATO.GEN.410105 |
| 9.7 | address2.xx.xx:AddressDetails (Tuple 1..1) | N/A | N/A | N/A |
| 9.7.1 | pyde.xx.xx:AddressDetails.OverseasAddress.Indicator | Common ruleset address2.xx.xx:AddressDetails applies to this tuple | Ruleset:address2 | N/A |
| 9.7.2 | pyde.xx.xx:AddressDetails.Usage.Code | N/A | N/A | N/A |
| 9.7.3 | pyde.xx.xx:AddressDetails.Currency.Code | N/A | N/A | N/A |
| 9.7.4 | pyde.xx.xx:AddressDetails.Line1.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IF COUNT (TUPLE(address2.xx.xx.AddressDetails)) <> 1  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee\address2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.Usage.Code <> "POS") OR (pyde.xx.xx:AddressDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408004  2. Schematron ID = VR.ATO.FTER.408075 | 1. CMN.ATO.FTER.408004  2. CMN.ATO.FTER.408075 |
| 9.7.5 | pyde.xx.xx:AddressDetails.Line2.Text | N/A | N/A | N/A |
| 9.7.6 | pyde.xx.xx:AddressDetails.Line3.Text | N/A | N/A | N/A |
| 9.7.7 | pyde.xx.xx:AddressDetails.Line4.Text | N/A | N/A | N/A |
| 9.7.8 | pyde.xx.xx:AddressDetails.LocalityName.Text | N/A | N/A | N/A |
| 9.7.9 | pyde.xx.xx:AddressDetails.Postcode.Text | N/A | N/A | N/A |
| 9.7.10 | pyde.xx.xx:AddressDetails.StateOrTerritory.Code | N/A | N/A | N/A |
| 9.7.11 | pyde.xx.xx:AddressDetails.CountryName.Text | N/A | N/A | N/A |
| 9.7.12 | pyde.xx.xx:AddressDetails.Country.Code | N/A | N/A | N/A |
| 9.8 | pyde.xx.xx:Residency.TaxPurposesPersonStatus.Indicator | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IF [FTER36] = NULL  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) IF [FTER36] = FALSE AND ([FTER37] <> NULL OR COUNT (TUPLE(perioddetails1.xx.xx:PeriodDetails)) > 0)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408005  2. Schematron ID = VR.ATO.FTER.408077 | 1. CMN.ATO.FTER.408005  2. CMN.ATO.FTER.408077 |
| 9.9 | pyde.xx.xx:Residency.NonResidentFullPeriod.Indicator | 1. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) IF [FTER36] = TRUE AND ( ([FTER37] = NULL OR [FTER37] = FALSE) AND COUNT (TUPLE(perioddetails1.xx.xx:PeriodDetails)) = 0)  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) IF [FTER37] = TRUE AND COUNT (TUPLE(perioddetails1.xx.xx:PeriodDetails)) > 0  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408078  2. Schematron ID = VR.ATO.FTER.408079 | 1. CMN.ATO.FTER.408078  2. CMN.ATO.FTER.408079 |
| 9.10 | perioddetails1.xx.xx:PeriodDetails (Tuple 0..4) | N/A | N/A | N/A |
| 9.10.1 | pyin.xx.xx:Period.Type.Code | Common rule set perioddetails1.xx.xx.PeriodDetails applies to this tuple  1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee\perioddetails1.xx.xx.PeriodDetails) IF pyid.xx.xx:Period.Type.Code <> "NonResidentPartPeriod"  RETURN VALIDATION MESSAGE ENDIF | Ruleset:perioddetails1  1. Schematron ID = VR.ATO.FTER.408081 | 1. CMN.ATO.FTER.408081 |
| 9.10.2 | pyin.xx.xx:Period.Start.Date | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:Trustee) IF COUNT (TUPLE(perioddetails1.xx.xx.PeriodDetails)) > 4  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER41] <> NULL AND (ANY OCCURRENCE OF [FTER38] <> NULL AND [FTER38] < [FTER41])  RETURN VALIDATION MESSAGE ENDIF  3. IF [FTER43] <> NULL AND (ANY OCCURRENCE OF [FTER38] <> NULL AND [FTER38] < [FTER43])  RETURN VALIDATION MESSAGE ENDIF  4. IF [FTER40] <> NULL AND [FTER41] = NULL AND [FTER43] = NULL AND (ANY OCCURRENCE OF [FTER38] <> NULL) AND ([FTER38] < ([FTER40] - 1)&"-07-01")  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408080  2. Schematron ID = VR.ATO.FTER.408084  3. Schematron ID = VR.ATO.FTER.408085  4. Schematron ID = VR.ATO.FTER.408086 | 1. CMN.ATO.FTER.408080  2. CMN.ATO.FTER.408084  3. CMN.ATO.FTER.408085  4. CMN.ATO.FTER.408086 |
| 9.10.3 | pyin.xx.xx:Period.End.Date | N/A | N/A | N/A |
| 10 | rvctc3.xx.xx:Elections.FamilyTrustElectionStatus.Year | 1. WHERE PARENT RETURN EXISTS IF [FTER95] = "V" AND [FTER40] <> NULL AND [FTER40] <> PARENT RETURN:RP:pyin.xx.xx:Report.TargetFinancial.Year  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER95] = "V" AND (([FTER40] = NULL AND [FTER43] = NULL) OR COUNT (TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual)) = 0)  RETURN VALIDATION MESSAGE ENDIF  3. WHERE PARENT RETURN DOES NOT EXIST IF [FTER95] = "E" AND [FTER40] <> NULL AND [FTER40] < 2005  RETURN VALIDATION MESSAGE ENDIF  4. IF [FTER40] <> NULL AND [FTER43] <> NULL AND [FTER41] = NULL AND [FTER43] < (([FTER40] - 1)&"-07-01")  RETURN VALIDATION MESSAGE ENDIF  5. WHERE PARENT RETURN EXISTS IF [FTER95] = "E" AND [FTER40] <> PARENT RETURN:RP:Elections.FamilyTrustElectionStatus.Year  RETURN VALIDATION MESSAGE ENDIF  6. IF [FTER95] = "R" AND [FTER40] <> NULL AND [FTER40] < 1995  RETURN VALIDATION MESSAGE ENDIF  7. IF [FTER95] = "R" AND [FTER58] <> NULL AND [FTER40] > [FTER58]  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408029  2. Schematron ID = VR.ATO.FTER.408030  3. Schematron ID = VR.ATO.FTER.408087  4. Schematron ID = VR.ATO.FTER.408088  5. Schematron ID = VR.ATO.FTER.408112  6. Schematron ID = VR.ATO.FTER.408115  7. Schematron ID = VR.ATO.FTER.408116 | 1. CMN.ATO.FTER.408029  2. CMN.ATO.FTER.408030  3. CMN.ATO.FTER.408087  4. CMN.ATO.FTER.408088  5. CMN.ATO.FTER.408112  6. CMN.ATO.FTER.408115  7. CMN.ATO.FTER.408116 |
| 11 | rvctc3.xx.xx:Elections.Commencement.Date | N/A | N/A | N/A |
| 12 | fter.0001.lodge.req.xx.xx:SpecifiedIndividual (Tuple 0..1) | N/A | N/A | N/A |
| 12.1 | pyid.xx.xx:Identifiers.TaxFileNumber.Identifier | 1. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual) IF [FTER44] = NULLORBLANK AND [FTER45] <> TRUE  RETURN VALIDATION MESSAGE ENDIF  2. IF (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier <> NULLORBLANK) AND (TFNALGORITHM (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier) = FALSE)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408107  2. Schematron ID = VR.ATO.GEN.410031 | 1. CMN.ATO.FTER.408107  2. CMN.ATO.GEN.410031 |
| 12.2 | pyid.xx.xx:Identifiers.TaxFileNumberInexistent.Indicator | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:SpecifiedIndividual) IF [FTER45] = TRUE AND [FTER44] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408100 | 1. CMN.ATO.FTER.408100 |
| 12.3 | pyde.xx.xx:PersonDemographicDetails.Birth.Date | 1. IF [FTER40] <> NULL AND [FTER46] <> NULL AND [FTER41] = NULL AND [FTER43] = NULL AND [FTER46] > (([FTER40] - 1)&"-07-01")  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER46] <> NULL AND [FTER41] <> NULL AND [FTER46] > [FTER41]  RETURN VALIDATION MESSAGE ENDIF  3. IF [FTER46] <> NULL AND [FTER43] <> NULL AND [FTER46] > [FTER43]  RETURN VALIDATION MESSAGE ENDIF  4. ([FTER43] <> NULL AND [FTER46] > [FTER43]) OR ([FTER43] = NULL AND [FTER41] <> NULL AND [FTER46] > [FTER41])  RETURN VALIDATION MESSAGE ENDIF  5. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:SpecifiedIndividual) IF PersonDemographicDetails.Birth.Date = NULL  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408090  2. Schematron ID = VR.ATO.FTER.408095  3. Schematron ID = VR.ATO.FTER.408097  4. Schematron ID = VR.ATO.FTER.408098  5. Schematron ID = VR.ATO.FTER.408099 | 1. CMN.ATO.FTER.408090  2. CMN.ATO.FTER.408095  3. CMN.ATO.FTER.408097  4. CMN.ATO.FTER.408098  5. CMN.ATO.FTER.408099 |
| 12.4 | personstructuredname3.xx.xx:PersonNameDetails (Tuple 1..1) | N/A | N/A | N/A |
| 12.4.1 | pyde.xx.xx:PersonNameDetails.PersonNameType.Code | Common rule set personstructuredname3.xx.xx:PersonNameDetails applies to this tuple | Ruleset:personstructuredname3 | N/A |
| 12.4.2 | pyde.xx.xx:PersonNameDetails.Usage.Code | N/A | N/A | N/A |
| 12.4.3 | pyde.xx.xx:PersonNameDetails.Currency.Code | N/A | N/A | N/A |
| 12.4.4 | pyde.xx.xx:PersonNameDetails.Title.Text | N/A | N/A | N/A |
| 12.4.5 | pyde.xx.xx:PersonNameDetails.NameSuffix.Text | N/A | N/A | N/A |
| 12.4.6 | pyde.xx.xx:PersonNameDetails.FamilyName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:SpecifiedIndividual) IF COUNT (TUPLE personstructuredname3.xx.xx.PersonNameDetails WHERE TUPLE ELEMENT EXPLICIT PersonNameDetails.PersonNameType.Code = "LGL" AND WHERE TUPLE ELEMENT EXPLICIT PersonNameDetails.Usage.Code = "Contact" AND WHERE TUPLE ELEMENT EXPLICIT PersonNameDetails.Currency.Code = "C") <> 1  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408089 | 1. CMN.ATO.FTER.408089 |
| 12.4.7 | pyde.xx.xx:PersonNameDetails.GivenName.Text | N/A | N/A | N/A |
| 12.4.8 | pyde.xx.xx:PersonNameDetails.OtherGivenName.Text | N/A | N/A | N/A |
| 12.5 | address2.xx.xx:AddressDetails (Tuple 1..1) | N/A | N/A | N/A |
| 12.5.1 | pyde.xx.xx:AddressDetails.OverseasAddress.Indicator | Common ruleset address2.xx.xx:AddressDetails applies to this tuple | Ruleset:address2 | N/A |
| 12.5.2 | pyde.xx.xx:AddressDetails.Usage.Code | N/A | N/A | N/A |
| 12.5.3 | pyde.xx.xx:AddressDetails.Currency.Code | N/A | N/A | N/A |
| 12.5.4 | pyde.xx.xx:AddressDetails.Line1.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:SpecifiedIndividual\address2.xx.xx.AddressDetails) IF (AddressDetails.Usage.Code <> "RES") OR (AddressDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:SpecifiedIndividual) IF COUNT (TUPLE(address2.xx.xx.AddressDetails)) <> 1  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408108  2. Schematron ID = VR.ATO.FTER.408150 | 1. CMN.ATO.FTER.408108  2. CMN.ATO.FTER.408150 |
| 12.5.5 | pyde.xx.xx:AddressDetails.Line2.Text | N/A | N/A | N/A |
| 12.5.6 | pyde.xx.xx:AddressDetails.Line3.Text | N/A | N/A | N/A |
| 12.5.7 | pyde.xx.xx:AddressDetails.Line4.Text | N/A | N/A | N/A |
| 12.5.8 | pyde.xx.xx:AddressDetails.LocalityName.Text | N/A | N/A | N/A |
| 12.5.9 | pyde.xx.xx:AddressDetails.Postcode.Text | N/A | N/A | N/A |
| 12.5.10 | pyde.xx.xx:AddressDetails.StateOrTerritory.Code | N/A | N/A | N/A |
| 12.5.11 | pyde.xx.xx:AddressDetails.CountryName.Text | N/A | N/A | N/A |
| 12.5.12 | pyde.xx.xx:AddressDetails.Country.Code | N/A | N/A | N/A |
| 13 | rvctc3.xx.xx:Elections.Revocation.Date | 1. IF [FTER95] = "R" AND [FTER57] = NULL AND [FTER96] = NULL  RETURN VALIDATION MESSAGE ENDIF  2. WHERE PARENT RETURN EXISTS IF [FTER57] > ((PARENT RETURN:RP:pyin.xx.xx:Report.TargetFinancial.Year)&"-06-30")  RETURN VALIDATION MESSAGE  ENDIF | 1. Schematron ID = VR.ATO.FTER.408032  2. Schematron ID = VR.ATO.FTER.408162 | 1. CMN.ATO.FTER.408032  2. CMN.ATO.FTER.408110 |
| 14 | rvctc3.xx.xx:Elections.FamilyTrustRevocationStatusEffective.Year | 1. IF [FTER95] = "R" AND [FTER96] <> NULL AND [FTER96] <> PARENT RETURN:RP:pyin.xx.xx:Report.TargetFinancial.Year  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408033 | 1. CMN.ATO.FTER.408033 |
| 15 | rvctc3.xx.xx:Elections.NotificationOfElectionBeingRevoked.Year | 1. IF [FTER95] = SET("E", "V") AND ([FTER57] <> NULL OR [FTER96] <> NULL OR [FTER58] <> NULL OR [FTER60] <> NULLORBLANK OR [FTER61] <> NULL OR (COUNT TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) > 0))  RETURN VALIDATION MESSAGE ENDIF  2. IF [FTER95] = "R" AND [FTER58] = NULL AND [FTER40] = NULL  RETURN VALIDATION MESSAGE ENDIF  3. IF [FTER58] <> NULL AND [FTER58] < 1998  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408027  2. Schematron ID = VR.ATO.FTER.408102  3. Schematron ID = VR.ATO.FTER.408113 | 1. CMN.ATO.FTER.408027  2. CMN.ATO.FTER.408102  3. CMN.ATO.FTER.408113 |
| 16 | pyid.xx.xx:Identifiers.TaxFileNumber.Identifier | 1. IF [FTER95] = "R" AND [FTER60] = NULLORBLANK AND [FTER61] <> TRUE  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408114 | 1. CMN.ATO.FTER.408114 |
| 17 | pyid.xx.xx:Identifiers.TaxFileNumberInexistent.Indicator | 1. IF [FTER61] = TRUE AND [FTER60] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408101 | 1. CMN.ATO.FTER.408101 |
| 18 | fter.0001.lodge.req.xx.xx:InterposedEntity (Tuple 0..120) | N/A | N/A | N/A |
| 18.1 | pyde.xx.xx:Party.Type.Code | 1. IF [FTER62] <> NULLORBLANK AND [FTER62] <> SET("051","161","212","217")  RETURN VALIDATION MESSAGE ENDIF  2. IF COUNT (TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity)) > 120  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408117  2. Schematron ID = VR.ATO.FTER.408151 | 1. CMN.ATO.FTER.408117  2. CMN.ATO.FTER.408151 |
| 18.2 | pyid.xx.xx:Identifiers.TaxFileNumber.Identifier | 1. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) IF [FTER64] = NULLORBLANK AND [FTER63] <> TRUE  RETURN VALIDATION MESSAGE ENDIF  2. IF (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier <> NULLORBLANK) AND (TFNALGORITHM (pyid.xx.xx:Identifiers.TaxFileNumber.Identifier) = FALSE)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408119  2. Schematron ID = VR.ATO.GEN.410031 | 1. CMN.ATO.FTER.408119  2. CMN.ATO.GEN.410031 |
| 18.3 | pyid.xx.xx:Identifiers.TaxFileNumberInexistent.Indicator | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:InterposedEntity) IF [FTER63] = TRUE AND [FTER64] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408118 | 1. CMN.ATO.FTER.408118 |
| 18.4 | organisationname2.xx.xx:OrganisationNameDetails (Tuple 0..1) | N/A | N/A | N/A |
| 18.4.1 | pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code | Common rule set organisationname2.xx.xx:OrganisationNameDetails applies to this tuple | Ruleset:organisationname2 | N/A |
| 18.4.2 | pyde.xx.xx:OrganisationNameDetails.Currency.Code | N/A | N/A | N/A |
| 18.4.3 | pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:InterposedEntity\organisationname2.xx.xx:OrganisationNameDetails)  IF (OrganisationNameDetails.OrganisationalNameType.Code <> "MN") OR (OrganisationNameDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408120 | 1. CMN.ATO.FTER.408120 |
| 18.5 | pyid.xx.xx:Identifiers.AustralianCompanyNumber.Identifier | 1. IF [FTER87] <> NULLORBLANK AND [FTER67] <> NULLORBLANK  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408122 | 1. CMN.ATO.FTER.408122 |
| 18.6 | pyid.xx.xx:Identifiers.AustralianRegisteredBodyNumber.Identifier | N/A | N/A | N/A |
| 18.7 | address2.xx.xx:AddressDetails (Tuple 0..1) | N/A | N/A | N/A |
| 18.7.1 | pyde.xx.xx:AddressDetails.OverseasAddress.Indicator | Common ruleset address2.xx.xx:AddressDetails applies to this tuple | Ruleset:address2 | N/A |
| 18.7.2 | pyde.xx.xx:AddressDetails.Usage.Code | N/A | N/A | N/A |
| 18.7.3 | pyde.xx.xx:AddressDetails.Currency.Code | N/A | N/A | N/A |
| 18.7.4 | pyde.xx.xx:AddressDetails.Line1.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:InterposedEntity\address2.xx.xx.AddressDetails) IF (AddressDetails.Usage.Code <> "POS") OR (AddressDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408123 | 1. CMN.ATO.FTER.408123 |
| 18.7.5 | pyde.xx.xx:AddressDetails.Line2.Text | N/A | N/A | N/A |
| 18.7.6 | pyde.xx.xx:AddressDetails.Line3.Text | N/A | N/A | N/A |
| 18.7.7 | pyde.xx.xx:AddressDetails.Line4.Text | N/A | N/A | N/A |
| 18.7.8 | pyde.xx.xx:AddressDetails.LocalityName.Text | N/A | N/A | N/A |
| 18.7.9 | pyde.xx.xx:AddressDetails.Postcode.Text | N/A | N/A | N/A |
| 18.7.10 | pyde.xx.xx:AddressDetails.StateOrTerritory.Code | N/A | N/A | N/A |
| 18.7.11 | pyde.xx.xx:AddressDetails.CountryName.Text | N/A | N/A | N/A |
| 18.7.12 | pyde.xx.xx:AddressDetails.Country.Code | N/A | N/A | N/A |
| 19 | fter.0001.lodge.req.xx.xx:DeclarationOfTrustee (Tuple 0..10) | N/A | N/A | N/A |
| 19.1 | personstructuredname3.xx.xx:PersonNameDetails (Tuple 0..1) | N/A | N/A | N/A |
| 19.1.1 | pyde.xx.xx:PersonNameDetails.PersonNameType.Code | Common rule set personstructuredname3.xx.xx:PersonNameDetails applies to this tuple | Ruleset:personstructuredname3 | N/A |
| 19.1.2 | pyde.xx.xx:PersonNameDetails.Usage.Code | N/A | N/A | N/A |
| 19.1.3 | pyde.xx.xx:PersonNameDetails.Currency.Code | N/A | N/A | N/A |
| 19.1.4 | pyde.xx.xx:PersonNameDetails.Title.Text | N/A | N/A | N/A |
| 19.1.5 | pyde.xx.xx:PersonNameDetails.NameSuffix.Text | N/A | N/A | N/A |
| 19.1.6 | pyde.xx.xx:PersonNameDetails.FamilyName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:DeclarationOfTrustee)  IF [FTER75] = NULLORBLANK AND [FTER79] = NULLORBLANK   RETURN VALIDATION MESSAGE ENDIF  2. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:DeclarationOfTrustee\personstructuredname3.xx.xx:PersonNameDetails)  IF (PersonNameDetails.PersonNameType.Code <> "LGL") OR (PersonNameDetails.Usage.Code <> "DeclarationSignatory") OR (PersonNameDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE  ENDIF  3. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) IF ([FTER79] <> NULL) AND ([FTER75] <> NULL)  RETURN VALIDATION MESSAGE  ENDIF  4. WHERE IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) IF [FTER75] <> NULL AND [FTER77] = NULL  RETURN VALIDATION MESSAGE  ENDIF | 1. Schematron ID = VR.ATO.FTER.408130  2. Schematron ID = VR.ATO.FTER.408131  3. Schematron ID = VR.ATO.FTER.408135  4. Schematron ID = VR.ATO.FTER.408136 | 1. CMN.ATO.FTER.408130  2. CMN.ATO.FTER.408131  3. CMN.ATO.FTER.408135  4. CMN.ATO.FTER.408158 |
| 19.1.7 | pyde.xx.xx:PersonNameDetails.GivenName.Text | N/A | N/A | N/A |
| 19.1.8 | pyde.xx.xx:PersonNameDetails.OtherGivenName.Text | N/A | N/A | N/A |
| 19.2 | organisationname2.xx.xx:OrganisationNameDetails (Tuple 0..1) | N/A | N/A | N/A |
| 19.2.1 | pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code | Common rule set organisationname2.xx.xx:OrganisationNameDetails applies to this tuple | Ruleset:organisationname2 | N/A |
| 19.2.2 | pyde.xx.xx:OrganisationNameDetails.Currency.Code | N/A | N/A | N/A |
| 19.2.3 | pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text | 1. WHERE IN TUPLE (fter.0001.lodge.req.xx.xx:DeclarationOfTrustee\organisationname2.xx.xx:OrganisationNameDetails)  IF (OrganisationNameDetails.OrganisationalNameType.Code <> "MN") OR (OrganisationNameDetails.Currency.Code <> "C")  RETURN VALIDATION MESSAGE  ENDIF | 1. Schematron ID = VR.ATO.FTER.408137 | 1. CMN.ATO.FTER.408137 |
| 19.3 | declaration2.xx.xx:Declaration (Tuple 1..1) | N/A | N/A | N/A |
| 19.3.1 | pyin.xx.xx:Declaration.StatementType.Code | **Common rule set declaration2.xx.xx:Declaration does not apply to FTER**  1.. WHERE IN TUPLE(fter.000.lodge.req.xx.xx:DeclarationOfTrustee\declaration2.xx.xx:Declaration) IF (pyin.xx.xx:Declaration.StatementType.Code <> NULLORBLANK) AND (pyin.xx.xx:Declaration.StatementType.Code <> SET ("HardCopy", "TrueAndCorrect" ))  RETURN VALIDATION MESSAGE ENDIF  2. IF COUNT(INT) = 0 AND   COUNT (pyin.xx.xx:Declaration.StatementType.Code ="TrueAndCorrect" IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee\declaration2.xx.xx:Declaration)) <> 1  RETURN VALIDATION MESSAGE ENDIF  3. WHERE IN TUPLE(Declaration2.xx.xx:Declaration) IF pyin.xx.xx:Declaration.StatementAccepted.Indicator = FALSE OR Declaration.SignatoryIdentifier.Text = NULLORBLANK OR Declaration.Signature.Date = NULL   RETURN VALIDATION MESSAGE ENDIF  4. IF pyde.xx.xx:Declaration.Signature.Date <> NULL AND pyde.xx.xx:Declaration.Signature.Date > DATE(TODAY)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408163  2. Schematron ID = VR.ATO.FTER.408164  3. Schematron ID = VR.ATO.FTER.408166  4. Schematron ID = VR.ATO.GEN.430255 | 1. CMN.ATO.FTER.408163  2. CMN.ATO.FTER.408164  3. CMN.ATO.FTER.408166  4. CMN.ATO.GEN.430255 |
| 19.3.2 | pyin.xx.xx:Declaration.StatementAccepted.Indicator | N/A | N/A | N/A |
| 19.3.3 | pyin.xx.xx:Declaration.Statement.Text | N/A | N/A | N/A |
| 19.3.4 | pyin.xx.xx:Declaration.Signature.Date | 1. IF COUNT(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) > 10  RETURN VALIDATION MESSAGE ENDIF  2. IF COUNT(INT) = 0 AND [FTER95] = "E" AND COUNT(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) <> COUNT(fter.0001.lodge.req.xx.xx:Trustee)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.FTER.408154  2. Schematron ID = VR.ATO.FTER.408155 | 1. CMN.ATO.FTER.408154  2. CMN.ATO.FTER.408155 |
| 19.3.5 | pyin.xx.xx:Declaration.SignatoryIdentifier.Text | N/A | N/A | N/A |
| 19.3.6 | personunstructuredname1.xx.xx:PersonUnstructuredName (Tuple 0..1) | N/A | N/A | N/A |
| 19.3.6.1 | pyde.xx.xx:PersonUnstructuredName.Usage.Code | N/A | N/A | N/A |
| 19.3.6.2 | pyde.xx.xx:PersonUnstructuredName.FullName.Text | 1. IF LENGTH(pyde.xx.xx:PersonUnstructuredName.FullName.Text) > 200  RETURN VALIDATION MESSAGE ENDIF  2. IF pyde.xx.xx:PersonUnstructuredName.FullName.Text <> NULLORBLANK AND pyde.xx.xx:PersonUnstructuredName.FullName.Text <> SET("a-z", "A-Z", "-", " ")  RETURN VALIDATION MESSAGE ENDIF  3. WHERE IN TUPLE (personunstructuredname1.xx.xx:PersonUnstructuredName) IN TUPLE(declaration2.xx.xx:Declaration) IF pyde.xx.xx:PersonUnstructuredName.Usage.Code <> "DeclarationSignatory"  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.430252  2. Schematron ID = VR.ATO.GEN.430253  3. Schematron ID = VR.ATO.GEN.436279 | 1. CMN.ATO.GEN.430252  2. CMN.ATO.GEN.430253  3. CMN.ATO.GEN.436279 |
| 20 | pyin.xx.xx:Report.CompletionHours.Number | 1. IF (pyin.xx.xx:Report.CompletionHours.Number) <> SET(1-999)  RETURN VALIDATION MESSAGE ENDIF | 1. Schematron ID = VR.ATO.GEN.410082 | 1. CMN.ATO.GEN.410082 |

* + 1. FTER Lodge Response - Message
       1. Discoverable Taxonomy Set References

No XBRL instance will be returned.

* + - 1. Standard Business Document Header Context

##### The WIG provides the specification of the SBDH. The following table specifies the message specific data element values or any variations to the WIG.

| **Attribute Name** | **Instructions/Rules** |
| --- | --- |
| sbdm:Message.Type.Text | 1. Mandatory – Value must be “fter.0001.lodge.response” |
| sbdm:Lodgement.Receipt.Identifier | 1. N/A – Will not be provided |
| sbdm:Lodgement.Receipt.Datetime | 1. Optional – Will be provided for successful lodgements |

* + - 1. Standard Business Document Body Context

No XBRL instance will be returned for this schedule FTER

Appendix A – The Message Content Table Explained

This section defines the table structure that must be used to define the context, structure, and rules of the data elements contained within the XBRL instance document – referred to as the message content table.

There will be a message content table for each context within the message. The grouping of the data elements in accordance to the context aligns to how the data elements are built into the XBRL taxonomy and this consistent presentation will assist Software Developers.

The message content table uses the following rows and columns:

**Context Type (row at top of table)**

This is the name of the XBRL Context Specification or Context Instance which has been defined early in the MIG document.

**Sequence Number**

This is a sequential number used to indicate the expected order of the data elements within the instance document and to describe the structure of tuples. Data elements within a tuple are allocated a multilevel number to indicate the expected structure and order of the data elements contained within the tuple. If there are nested tuples then the sequence number goes to next numbering level. The following is an extract from a message content table which illustrates the sequence numbering and multi-levelling required to cater for tuples and nested tuples.

In the following example, the ‘smsfar.0001.lodge.req.xx.xx:TrusteeOrDirector’ fact is a tuple that includes a ‘declaration2.xx.xx:Declaration’ tuple. This declaration2.xx.xx:Declaration contains all the elements listed below it and the personunstructuredname1.xx.xx:PersonUnstructuredName’ tuple. The fact at Seq No 133 is not within a tuple, as indicated by the numbering level.

| Context - RP | | | | |
| --- | --- | --- | --- | --- |
| Seq No. | XBRL Fact | Instructions / Rules | Rule Imp | SBR Msg Code |
| 132 | smsfar.0001.lodge.req.xx.xx:TrusteeOrDirector (Tuple 1..1) |  |  |  |
| 132.1 | declaration2.xx.xx:Declaration (Tuple 1..1) |  |  |  |
| 132.1.1 | pyin.xx.xx:Declaration.StatementType.Code |  |  |  |
| 132.1.2 | pyin.xx.xx:Declaration.StatementAccepted.Indicator |  |  |  |
| 132.1.3 | pyin.xx.xx:Declaration.Statement.Text |  |  |  |
| 132.1.4 | pyin.xx.xx:Declaration.Signature.Date |  |  |  |
| 132.1.5 | pyin.xx.xx:Declaration.SignatoryIdentifier.Text |  |  |  |
| 132.1.6 | personunstructuredname1.xx.xx:PersonUnstructuredName (Tuple 0..1) |  |  |  |
| 132.1.6.1 | pyde.xx.xx:PersonUnstructuredName.Usage.Code |  |  |  |
| 132.1.6.2 | pyde.xx.xx:PersonUnstructuredName.FullName.Text |  |  |  |
| 133 | pyin.xx.xx:Report.CompletionHours.Number |  |  |  |

**XBRL Fact**: This is the name of the data element to be reported.

For example: Identifiers.AustralianBusinessNumber.Identifier

**Instructions / Rules**: This column describes all the instructions / rules applicable to the data element. Each rule needs to be given a sequential number which links the rule to its implementation and message code. Rules would include information such as optionality, presentation criteria and the use of XML attributes such as IsVisible.

**Rule Implementation**: This column informs Software Developers how the rules specified in the Rules column will be provided.

NOTE: This column is only applicable for request messages and the column will not be present in the table for Response Messages.

There can only be the following options:

* XBRL – validation provided via the XBRL schemas and linkbases. Typically rules implemented via XBRL do not need to be specified within the MIG. Only those rules that are considered to provide necessary information to software developers should be defined within the MIG. An example of this is the rules associated with the domain values of a dimension within a context specification.
* Schematron ID – for rules that cannot be implemented using XBRL some agencies will provide a Schematron implementation of the rule. When Schematron is provided then the unique ID used to identify the rule within the Schematron file must be provided within the MIG. The following is an example of how this should appear in the column :.Schematron ID = VICMIG001
* MIG - There will be situations where rules will not be provided to Software Developers in a machine readable format and the description of the rule in the MIG is all that will be provided. In this situation the Software Developer has the choice of either implementing the rule as specified within the MIG or to rely on the agency to validate the data element (the expectation is that the Agency will always test for this rule)
* Agency – This rule cannot be implemented by the Software Developer and can only be executed by the agency.

**SBR Message Code**: All messages returned via the SBR channel will contain a code to uniquely identify the condition that has occurred.

NOTE: This column is only applicable for request messages and the column will not be present in the table for Response Messages.

In order to allow codes to be managed in a distributed fashion, codes will take the following format:

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

represented by the regular expression

**([A-Z0-9])+.([A-Z0-9])+.([A-Z0-9])+.([A-Z0-9])+**

Initially

**Jurisdiction** = SBR | CMN | QLD | NSW | ACT | VIC | SA | WA | NT | TAS

**Agency** = Jurisdiction specific agency code

For CMN (Commonwealth), = ATO, ASIC, APRA, ABS

For SBR = GEN (i.e. SBR wide codes)

For States = OSR of Offices of State Revenue

**Function** = Agency specific functional area or GEN for agency wide codes

For SBR = GEN or FAULT

**Id** = function specific identifier (format may vary across agencies).

Examples are shown below;

SBR.GEN.FAULT.TOOMANYINSTANCES

CMN.ATO.TFN.OK

QLD.OSR.PRL.000001

The above structure recognises and caters for the current situation where agency errors are unharmonised, and will need to be passed through to client software.

The expectation is that for each rule identified within the message content table will have a corresponding message code however depending on the rule implementation a message code may not be relevant in which case Not Applicable (N/A) should be inserted into the rule’s corresponding message code to make this clear to Software Developers. The follow table summaries what must be provided in the message code column in relation to the rules implementation choice.

|  |  |
| --- | --- |
| **Rule Implementation** | **Message Code** |
| Schematron | Message Code needs to be provided against corresponding rule. |
| XBRL | Message Code not relevant – place N/A against corresponding rule. |
| MIG | Message Code needs to be provided against corresponding rule. The only exception is if the rule is associated with a rendering instruction to the software developer. |
| Agency | Message Code needs to be provided against corresponding rule. |

The expectation is that each agency will populate a message repository with all error, warning and information messages that could be returned via the SBR channel. These messages will be allocated an SBR message code using the above mentioned code format. The Software Developer would then use the SBR message code provided via the MIG and the message repository to obtain the full details associated with the message.

Appendix B – Tax Office Structured English

The validation rules are expressed in structured English. The following table defines terms and characters used throughout these rules.

| Structured english term | Intended interpretation | Examples |
| --- | --- | --- |
| -  (as in <a> - <b>) | Minus. | <a> - <b> Means value of ‘a’ minus the value of ‘b’ |
| -  (as in SET(0-9) | Specifies a range of numeric or alphabetic values within a ‘SET’ construct. | = SET(0-3) Means is equal to 0, 1, 2 or 3  = SET(a-z) Means is equal to a, b, c, d …(etc)… or z  DOES NOT CONTAIN SET(a-z) Means the field does not include any incidence of a lower case alphabetic character between a and z |
| <> SET ("0","1","2",",") | Set of acceptable characters |  |
| & | Concatenate. Joins the value of a field to a literal or other field | [CGTS1]&"-06-30" Where [CGTS1] is 2010, means 2010-06-30 |
| +/- | In numerical calculations, allows for an allowable deviation | <a> <> <b> +/- 1 Means (<a> > <b> + 1) OR (<a> < <b> - 1)  <a> = <b> +/- 1 Means (<a> = <b>) OR (<a> = <b> + 1) OR (<a> = <b> - 1) |
| <> | Not equal to | IF <a> <> <b> Means if the value of ‘a’ is not equal to the value of ‘b’ |
| ABSVALUE | Absolute value. Ignore the signage of a numeric value. Both negative and positive values are included as positive values | IF <a> > ABSVALUE(<b>) Where the value of <b> is 25, means IF <a> > 25. Where the value of <b> is -20, means IF <a> > 20 |
| ALGORITHM (with <IDtype> prefix) | Defines a test against a standard algorithm – as indicated by the <IDtype> prefix. <IDtype> can be ABN, TFN, TAN, ARBN or ACN | IF TFNALGORITHM(<a>) = FALSE Means if the TFN field <a> fails the TFN algorithm.  IF ABNALGORITHM(<a>) = FALSE Means the ABN field <a> fails the ABN algorithm. |
| ALL OCCURENCES OF | All instances of a given field, where the field is from a repeatable tuple. (For testing values in repeating tuples.) | IF SUM(ALL OCCURRENCES OF(<a> - <b>)) <> <c> Means if the sum of every instance of <a>, minus the sum of every instance of <b> is not equal to the value of <c> (where <a> and <b> are from a repeatable tuple). |
| AND | Logical AND |  |
| ANY CHARACTER OF | Any character within a field |  |
| ANY OCCURRENCE OF | Any instances of a given field, where the field is from repeatable tuple. (For testing values in repeating tuples.) | IF ANY OCCURRENCE OF(<a>) > 10 Means if any instance of <a> is greater than 10 (where <a> is from a repeatable tuple). |
| ANY OTHER OCCURRENCE OF | Used to check if any given value for a particular element is repeated in the same element, where the element is part of a repeatable tuple. . (For testing a value in one occurrence against other occurrences) | IF <a> = ANY OTHER OCCURRENCE OF <a> Means if the value of <a> from one instance of the tuple, is equal to the value of another instance of the tuple. |
| CONTAINS | A string search for text within a field  Usage: <a> CONTAINS <B> | IF <a> CONTAINS "UNKNOWN" Means if <a> contains or equals the word ‘unknown’. |
| CONTAINS MORE THAN ONE | A text field contains more than one incidence of a given string with the field | IF pyde.xx.xx:ElectronicContact.ElectronicMail.Address.Text CONTAINS MORE THAN ONE "@"  Means if there is more than one ‘@’ symbol within the email address. |
| CONTEXT | Used to refer to a context instance  Usage: CONTEXT(<A>)  where <A> is a context instance abbreviation eg RP.GST.CC | WHERE CONTEXT = “RPI.Closing” Means in instances where the context instance is ‘RPI.Closing’. |
| COUNT | A count of the number of occurences of a field or context | IF COUNT(RPI) > 1 Means if the number of occurrences of the RPI context is more than 1.  IF COUNT(SCHEDULE = "IEE") > 50 Means if the number of occurrence of an IEE schedule in the business document is greater than 50. |
| COUNT(SCHEDULE) =1 | to describe if a schedule is attached to a parent return.  Usage: COUNT(SCHEDULE = <A>) = 1  Where <A> is a schedule abbreviation eg DIS, IEE |  |
| COUNT(SCHEDULE) = 0 | to describe if a schedule is not attached to a parent return.  Usage: COUNT(SCHEDULE = <A>) = 0  Where <A> is a schedule abbreviation eg DIS, IEE |  |
| DATE(TODAY) | Compares a date against the current date | IF <a> > DATE(TODAY) Means if <a> is a date in the future. |
| DECODE\_TO\_FDFGROUP | Translate the value in the Source Field to the corresponding Group Code decode value using the translation table in the specified worksheet in the Request Spreadsheet.  Where no corresponding value is found, no return value is to be returned.  Usage:  DECODE\_TO\_FDFGROUP(Source field, <Group Code>, <Worksheet Name in Request Spreadsheet>) |  |
| Dimension | Test against a specific set of metadata for a particular context | IF (RprtPyType.xx.xx:ReportingPartyTypeDimension = “RprtPyType.xx.xx:Intermediary”) Means if the Reporting Party Type context is ‘Intermediary’. |
| DOES NOT CONTAIN | An element has no instance of the stated value or set of values | DOES NOT CONTAIN SET("a-z", "A-Z", "0-9") Means that the field has no instance of an alphabetical character (excepting special characters), nor a numeric character. |
| DOMAIN | A globally defined set of values  EXAMPLE USAGE  <a> = SET(DOMAIN(<B>))  <a> is one of the values defined in <B> | EXAMPLE:  SET (DOMAIN(COUNTRY CODES) Means the complete set of country codes. Each set of domain values is defined in the Standard enumerations section within this document. |
| ENCODE\_TO\_FDFGROUP | Translate the value in the Source Field to the corresponding Numeric Group Code value using the translation table in the specified worksheet in the Request Spreadsheet.  Where no corresponding value is found, no return value is to be returned.  Usage:  ENCODE\_TO\_FDFGROUP(Source field, <Group Code>, <Worksheet Name in Request Spreadsheet>) |  |
| ENDSWITH | A string searches for text at the end of a field  Usage: <a> ENDSWITH <B> | IF <a> ENDSWITH " T/A" Means the condition is true if field <a> contains a value that ends with the text string ‘. T/A’. |
| FOUND | A string search for text within a field by performing the set, contains, startswith and endswith functions:  USAGE: <A> = FOUND(<B>,<C>)  The following functions is case insensitive is performed:  <a> = SET("<B>","<C>") (exact match)  <a> CONTAINS SET(" <B> "," <C> ") (a space on each side of the variable)  <a> STARTSWITH SET("<B> ","<C> ") (a space after the variable)  <a> ENDSWITH SET(" <B>"," <C>") (a space before the variable)  Where multiple elements have been provided, each element will need to be checked using the above functions.. | IF <a> = FOUND("The trustee","The Exec") Means if <a>:   * equals ‘The trustee’ or ‘The Exec’ (exact match), or * contains ‘ The trustee ’ or ‘ The Exec ’ (a space on each side of the variable), or * starts with ‘The trustee ’ or ‘The Exec ’ (with a space after), or * ends with ‘ The trustee’ or ‘ The Exec’ (with a space before). |
| IN TUPLE | Restricts a test to the value of a field within a particular tuple. (Where the field may exist in more than one tuple). | IF <a> IN TUPLE(<b>) Means if the value of <a> within the tuple <b>. (Where <a> may also exist outside tuple <b>).  (See also: ‘WHERE’) |
| LENGTH | Used to define the contraints on the length of a field.  (See also TEXT). | IF LENGTH(<a>) < 6 Means if the value of <a> does not contain at least 6 characters. |
| MONETARY() | Defines a monetary field pattern where a true response is given when a value passes all conditions.  As in: MONETARY(<a>,<b>,<c>) Where:   * <a> = S or U to indicate if field can be signed or not * <b> = Maximum number of digits (including decimal places) * <c> = Maximum number of decimal places   Notes: For <a> an S indicates a field can be prefixed with a sign, but may be omitted.  Where <a> is a U, the field must not be prefixed with a sign.  The value of <b> does not include a decimal point or sign in the total character limit. | <a> <> MONETARY(U,11,0) Field <a> is not equal to a number in the range of 0 to 99999999999, or includes a character other than 0 to 9.  <a> <> MONETARY(S,11,0) Field <a> is not equal to a number in the range -99999999999 to 99999999999, or includes a character other than 0 to 9, or ‘+’ or ‘–‘ as the first (left-most) character.  <a> <> MONETARY(U,13,2) Field <a> is not equal to a number in the range 0.00 and 99999999999.99, or includes a character other than 0 to 9 or a decimal point. (Decimal point may be absent). |
| NOT | Reverses the value of a boolean. i.e turns TRUE to FALSE and vice versa. |  |
| NULL | Fact is not there, or has been specified to be null with xsi:nil indicator or is a null non-textual value. | IF <a> = NULL Means if a (non-textual) value for <a> is blank or if <a> does not exist. |
| NULLORBLANK | Fact is not there, is null with xsI:nil = true or is a null string. (Applied to Text, Code, Description and Identifier facts). | IF <a> = NULLORBLANK Means if a (textual) value for <a> is blank or if <a> does not exist. |
| NUMBER() | Definition of a valid numeric field pattern where a true response is given when a value passes all conditions.  Usage: NUMBER(<a>,<b>,<c>)  Where <a> = S or U to indicate if field can be signed or not  <b> = Maximum number of digits including decimal places  <c> = Maximum number of decimal places | Examples: NUMBER(S,13,2)  NUMBER(U,13,2)  NUMBER(S,11,0)  Note: for <a> an S indicates a field can be prefixed with a sign, but it does not need to include one. However where <a> is a U the field cannot be prefixed with a sign.  Maximum amount of significant digits (i.e. non-decimal) is determined by the maximum number of digits minus the maximum number of decimal places (<b>-<c>) |
| NUMERIC | Contains only digits between 0..9 |  |
| OR | Logical OR |  |
| PARENT RETURN | Some schedule rules depend on the return it is attached to.  Usage: IF PARENT RETURN = <A>  <A> could be CTR or PTR or other return | IF <a> <> PARENT RETURN:<a> Means if the value of <a> on the schedule is not equal to the value of <a> on the main form.  WHERE PARENT RETURN EXISTS Means apply the test if this is a business document containing a schedule as a part of a main form. (Applies only to IEE and FTER whch may be submitted either as a form on its own or as a schedule as part of a form). |
| SCHEDULE | To describe a schedule that could be attached to a parent return.  Usage: SCHEDULE = <A>  Where <A> is a schedule abbreviation eg DIS, IEE  One of a series of ATO forms used to provide additional information to that contained in a main tax form.  In terms of SBR validation rules, refers to a business document containing a tax schedule submitted within the same standard business document body structure as a business document for a main tax return form. | IF COUNT(SCHEDULE = "S25A") = 0 Means if there is no instance of a Schedule 25A included in the business document body.  IF COUNT(SCHEDULE = "RSPT") > 50 Means if the number of occurrence of a Rental schedule in the business document body is greater than 50. |
| SET | Definition of an explicit set of values where if one value meets the criteria for comparation, a true response is given. | IF <a> <> SET("a","b","c") Means if <a> does not equal a or b or c.  IF <a> = SET(“a”,”b”,”c”) Means if <a> is equal to a or b or c.  Note: No Spaces exist between the SET and values are in brackets and are comma separated  IF <a> = SET(0-3) Means if <a> is equal to 0 or 1 or 2 or 3 |
| STARTSWITH | A string searches for text at the start of a field  Usage: <a> STARTSWITH <B> | IF <a> STARTSWITH "T/A" Means the condition is true if field <a> contains a value that starts with the text string ‘T/A’ |
| SUM | The sum of all instances of an element.  Usage: SUM(<A>)  where <A> is an element that appears in a repeating tuple or is a repeating element. | SUM(<a>) The total value of all instances of <a>, when each <a> is added up. (Where <a> is an element that is part of a repeating tuple or is a repeating element). |
| TEXT() | Used to define the maximum length of a textual field.  Definition of a valid text field pattern where a true response is given when a value passes all conditions.  Usage: TEXT(<a>)  Where <a> = Maximum number of characters  TRUE if the tested field is less than or equal to length <a>  (See also LENGTH) | <a> <> TEXT(150) Means the maximum number of characters allowable within field <a> is 150. |
| TUPLE | Concepts that contain a group of two or more fields. Generally, although not always, these concepts are a set of two or more fields that may be repeated, as a group, within a single business document. | TUPLE(addressdetails2.xx.xx:AddressDetails) Means the fields that have been defined as belonging to the ‘addressdetails2.xx.xx:AddressDetails’ module. |
| TUPLE (ELEMENT) EXPLICIT | Tuple element explicits are used to define a particular contextualisation of a tuple.  This data element (or elements) within a tuple are be used to specify the circumstance in which the tuple is interpreted, they contribute to the meaning of the tuple instance. | Example: orgname2  <xsd:element name="OrganisationNameDetails" substitutionGroup="xbrli:tuple" id="RT665" nillable="true">   <xsd:complexType>   <xsd:sequence>   <xsd:element ref="pyde.02.00:OrganisationNameDetails.OrganisationalNameType.Code"/>   <xsd:element ref="pyde.02.00:OrganisationNameDetails.Currency.Code"/>   <xsd:element ref="pyde.02.00:OrganisationNameDetails.OrganisationalName.Text"/>   </xsd:sequence>   <xsd:attribute name="id" type="xsd:ID" use="optional"/>   </xsd:complexType>  </xsd:element>  In this example, the tuple element explicits are the Currency Code and the Organisation Name Type Code as they provide meaning (or context) to the tuple as a whole. For example if the currency code value was "P" and the Organisation Name Type Code was "MTR", then we will then understand that the tuple represents a Previous Main Trading name. |
| (WHERE) IN TUPLE (element definition) | The element including the tuple definition is to be considered as a whole for the purposes of rule execution  This means that if the tuple definition can not be met, the element is considered NULL. | **EXAMPLE 1:**  where <B> is a fact in tuple <A>  IF (<B> IN TUPLE(<A>)) = NULLORBLANK  RETURN VALIDATION MESSAGE  END IF  This example will trigger if tuple <A> does not exist or if <B> in <A> is null or blank.  **EXAMPLE 2:**  where <B> is a fact in tuple <A>  IF COUNT (<B> IN TUPLE(<A>)) > 1  RETURN VALIDATION MESSAGE  END IF  This example will trigger when the occurrence of <B> in <A> is more than one. If tuple <A> does not exist this rule will not trigger, as “<B> in <A>” does not exist and therefore the count equals 0.  **EXAMPLE 3:**  The WHERE keyword is used when tuple element explicits are required:  e.g. ((RP:pyde.xx.xx:AddressDetails.Line1.Text WHERE(TUPLE ELEMENT Address Usage = "BUS") IN TUPLE(address2)) = NULLORBLANK)  Rule will trigger if RP is not present  Rule will trigger if address2 tuple is not present  Rule will trigger if address2 tuple with Address Usage = BUS is not present (i.e. a business address is not present)  Rule will trigger if Line1.Text in the address2 tuple with Address Usage = BUS is not present |
| (WHERE) IN TUPLE (rule prefix) | Rule is to be executed within the "context" of a defined tuple  This indicates that the rule execution is dependent on the tuple existence.  USAGE  IN TUPLE(<A>)  IF <B>…. | **EXAMPLE:**  where <B> is a fact in tuple <A>  IN TUPLE(<A>)  IF <B> = NULLORBLANK  RETURN VALIDATION MESSAGE  END IF  In this example the rule will only trigger if <A> exists and if <B> (in <A>) is null or blank, as the rule is conditional on the existance of tuple <A>  WHERE keyword is optional |
| xbrli (element definition) | xbrli is used to denote the reporting taxonomy root.(Indicates the tuple is not within another tuple)  Due to the ability of facts to be repeated at different levels of the reporting taxonomy (e.g. embedded in tuples)  xbrli keyword has been used to describe specific facts in relation to their location  Used where a particular tuple appears more than once within a form or schedule. | **EXAMPLE 1:**  Example from the TFND reporting taxonomy:  <A> = TUPLE(xbrli\declaration1.xx.xx:Declaration)  <B> = TUPLE(tfnd.0001.xx.xx:Payee\declaration1.xx.xx:Declaration)  As the declaration tuple is used twice, the above definitions can be used to refer to specific tuples  **EXAMPLE 2:**  IN TUPLE (xbrli\organisationname2.xx.xx:OrganisationNameDetails) Means in the tuple ‘organisationname2.xx.xx:OrganisationNameDetails’ that is not within another tuple.  In this example, the implication is that the ‘organisationname2.xx.xx:OrganisationNameDetails’ also exists under another tuple within the same form or schedule. |

Appendix C – Validation rules alias definitions

Field aliases are short identifiers for reporting taxonomy elements in ATO SBR messages, for example ‘FTER18’.

Field aliases are used instead of the full XBRL element in validation rules and other documentation to improve readability. Formula aliases are short identifiers for repeated calculations used in validation rules.

Field aliases and corresponding full XBRL element expansions for all elements used in fter.0001 reporting taxonomy are listed below.

| **Alias** | **Definition** |
| --- | --- |
| FTER95 | FTER:RP:Elections.FamilyTrustElectionRevocation.Code |
| FTER4 | FTER:RP:OrganisationNameDetails.OrganisationalName.Text WHERE ((TUPLE ELEMENT EXPLICIT pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code = "MN") AND (TUPLE ELEMENT EXPLICIT pyde.xx.xx:OrganisationNameDetails.Currency.Code = "C")) IN TUPLE(organisationname2.xx.xx:OrganisationNameDetails) |
| FTER16 | FTER:RP:Residency.CentralManagementAndControlOutsideAustralia.Indicator |
| FTER17 | FTER:RP:Residency.CentralManagementAndControlOutsideAustraliaFullPeriod.Indicator |
| FTER40 | FTER:RP:Elections.FamilyTrustElectionStatus.Year |
| FTER43 | FTER:RP:Elections.Commencement.Date |
| FTER57 | FTER:RP:Elections.Revocation.Date |
| FTER96 | FTER:RP:Elections.FamilyTrustRevocationStatusEffective.Year |
| FTER58 | FTER:RP:Elections.NotificationOfElectionBeingRevoked.Year |
| FTER60 | FTER:RP:Identifiers.TaxFileNumber.Identifier |
| FTER61 | FTER:RP:Identifiers.TaxFileNumberInexistent.Indicator |
| FTER18 | FTER:RP:Period.Start.Date WHERE (TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "ControlOutsideAustraliaPartPeriod") IN TUPLE(perioddetails1.xx.xx:PeriodDetails) |
| FTER41 | FTER:RP:Period.Start.Date WHERE (TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "SubstitutedAccountingPeriod") IN TUPLE(perioddetails1.xx.xx:PeriodDetails) |
| FTER21 | FTER:RP:Identifiers.TaxFileNumberInexistent.Indicator IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER20 | FTER:RP:Identifiers.TaxFileNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER86 | FTER:RP:Identifiers.AustralianCompanyNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER29 | FTER:RP:Identifiers.AustralianRegisteredBodyNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER36 | FTER:RP:Residency.TaxPurposesPersonStatus.Indicator IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER37 | FTER:RP:Residency.NonResidentFullPeriod.Indicator IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER38 | FTER:RP:Period.Start.Date WHERE (TUPLE ELEMENT EXPLICIT pyin.xx.xx:Period.Type.Code = "NonResidentPartPeriod") IN TUPLE(perioddetails1.xx.xx:PeriodDetails) IN TUPLE(fter.0001.lodge.req.xx.xx:Trustee) |
| FTER44 | FTER:RP:Identifiers.TaxFileNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual) |
| FTER45 | FTER:RP:Identifiers.TaxFileNumberInexistent.Indicator IN TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual) |
| FTER46 | FTER:RP:PersonDemographicDetails.Birth.Date IN TUPLE(fter.0001.lodge.req.xx.xx:SpecifiedIndividual) |
| FTER62 | FTER:RP:Party.Type.Code IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) |
| FTER64 | FTER:RP:Identifiers.TaxFileNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) |
| FTER63 | FTER:RP:Identifiers.TaxFileNumberInexistent.Indicator IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) |
| FTER87 | FTER:RP:Identifiers.AustralianCompanyNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) |
| FTER67 | FTER:RP:Identifiers.AustralianRegisteredBodyNumber.Identifier IN TUPLE(fter.0001.lodge.req.xx.xx:InterposedEntity) |
| FTER75 | FTER:RP:PersonNameDetails.FamilyName.Text WHERE ((TUPLE ELEMENT EXPLICIT pyde.xx.xx:PersonNameDetails.PersonNameType.Code = "LGL") AND (TUPLE ELEMENT EXPLICIT pyde.xx.xx:PersonNameDetails.Usage.Code = "DeclarationSignatory")) IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails) IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) |
| FTER79 | FTER:RP:OrganisationNameDetails.OrganisationalName.Text WHERE ((TUPLE ELEMENT EXPLICIT pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code = "MN") AND (TUPLE ELEMENT EXPLICIT pyde.xx.xx:OrganisationNameDetails.Currency.Code = "C")) IN TUPLE(organisationname2.xx.xx:OrganisationNameDetails) IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) |
| FTER77 | FTER:RP:PersonNameDetails.GivenName.Text WHERE ((TUPLE ELEMENT EXPLICIT pyde.xx.xx:PersonNameDetails.PersonNameType.Code = "LGL") AND (TUPLE ELEMENT EXPLICIT pyde.xx.xx:PersonNameDetails.Usage.Code = "DeclarationSignatory")) IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails) IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) |
| FTER168 | FTER:RP:Declaration.Signature.Date IN TUPLE(declaration2.xx.xx:Declaration) IN TUPLE(fter.0001.lodge.req.xx.xx:DeclarationOfTrustee) |

Appendix D – Common module validation rules

The common module validation rules within this appendix apply to each occurrence of specific common module tuple. For example, for each incidence of an ‘addressdetails2.xx.xx:addressdetails ‘ tuple within a message, the ‘address2’ set of common module rules will apply.

These are referred to within Section 5.4.1.3.2: FTER.LODGE Request Message Content Table against the first element within the respective tuple.

The common modules used in fter.0001 service are:

* addressdetails2.xx.xx:AddressDetails
* electroniccontacttelephone1.xx.xx:ElectronicContactTelephone
* organisationname2.xx.xx:OrganisationNameDetails
* perioddetails1.xx.xx.PeriodDetails
* personstructuredname3

The following rules apply to each addressdetails2.xx.xx:AddressDetailstuple.

| **Rule Implementation Schematron ID** | **Rule** | **Associated Response Message Code** |
| --- | --- | --- |
| VR.ATO.GEN.300003 | IF (pyde.xx.xx:AddressDetails.StateOrTerritory.Code <> NULLORBLANK) AND (pyde.xx.xx:AddressDetails.StateOrTerritory.Code <> SET ("ACT","NSW","NT","QLD","SA","VIC","WA","TAS","AAT"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.300003 |
| VR.ATO.GEN.410002 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF ((pyde.xx.xx:AddressDetails.Line1.Text CONTAINS "C/-") AND (pyde.xx.xx:AddressDetails.Line2.Text = NULLORBLANK))   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410002 |
| VR.ATO.GEN.410008 | IF (pyde.xx.xx:AddressDetails.Line3.Text <> NULLORBLANK)   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410008 |
| VR.ATO.GEN.410013 | IF (pyde.xx.xx:AddressDetails.Line4.Text <> NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410013 |
| VR.ATO.GEN.434147 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF pyde.xx.xx:AddressDetails.OverseasAddressIndicator = FALSE AND (pyde.xx.xx:AddressDetails.StateOrTerritory.Code = NULLORBLANK OR pyde.xx.xx:AddressDetails.Postcode.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.434147 |
| VR.ATO.GEN.410148 | IN TUPLE(addressdetails2.xx.xx:AddressDetails)IF pyde.xx.xx:AddressDetails.Currency.Code = NULLORBLANK  RETURN VALIDATION MESSAGE END IF | CMN.ATO.GEN.410148 |
| VR.ATO.GEN.410167 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.OverseasAddress.Indicator = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410167 |
| VR.ATO.GEN.410191 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.OverseasAddress.Indicator = TRUE) AND (pyde.xx.xx:AddressDetails.Country.Code = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410191 |
| VR.ATO.GEN.410192 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.Country.Code <> NULLORBLANK) AND (pyde.xx.xx:AddressDetails.Country.Code <> SET(DOMAIN(COUNTRY CODES)))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410192 |
| VR.ATO.GEN.410194 | IF LENGTH(pyde.xx.xx:AddressDetails.Line1.Text) > 38   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410194 |
| VR.ATO.GEN.410195 | IF LENGTH(pyde.xx.xx:AddressDetails.Line2.Text) > 38  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410195 |
| VR.ATO.GEN.410205 | IF (pyde.xx.xx:AddressDetails.Line1.Text = FOUND("AS ABOVE"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410001 |
| VR.ATO.GEN.410211 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.Country.Code = "au") AND (pyde.xx.xx:AddressDetails.OverseasAddress.Indicator = TRUE)   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410211 |
| VR.ATO.GEN.410212 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails)IF (pyde.xx.xx:AddressDetails.Country.Code <> NULLORBLANK) AND (pyde.xx.xx:AddressDetails.Country.Code <> "au") AND (pyde.xx.xx:AddressDetails.OverseasAddress.Indicator = FALSE)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410212 |
| VR.ATO.GEN.410213 | IF (pyde.xx.xx:AddressDetails.CountryName.Text <> NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410213 |
| VR.ATO.GEN.410214 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.Line1.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410214 |
| VR.ATO.GEN.428202 | IF (pyde.xx.xx:AddressDetails.Currency.Code <> NULLORBLANK)AND (pyde.xx.xx:AddressDetails.Currency.Code <> SET("C","P"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.428202 |
| VR.ATO.GEN.428230 | IF (pyde.xx.xx:AddressDetails.LocalityName.Text = FOUND("QLD","NSW","VIC","SA","WA","NT ","ACT","TAS"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000411 |
| VR.ATO.GEN.428240 | IF (pyde.xx.xx:AddressDetails.Line1.Text = FOUND("C/O","C/","Care Of","CO"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000406 |
| VR.ATO.GEN.428241 | IF (pyde.xx.xx:AddressDetails.Line2.Text = FOUND("C/-","C/O","C/","Care Of","CO"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000408 |
| VR.ATO.GEN.428254 | IF (pyde.xx.xx:AddressDetails.Line1.Text CONTAINS "UNKNOWN")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000405 |
| VR.ATO.GEN.428255 | IF (pyde.xx.xx:AddressDetails.Line2.Text CONTAINS "UNKNOWN")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000407 |
| VR.ATO.GEN.428256 | IF (pyde.xx.xx:AddressDetails.LocalityName.Text CONTAINS "UNKNOWN")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000409 |
| VR.ATO.GEN.430245 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.LocalityName.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.430245 |
| VR.ATO.GEN.430246 | WHERE IN TUPLE(addressdetails2.xx.xx:AddressDetails) IF (pyde.xx.xx:AddressDetails.OverseasAddress.Indicator = FALSE) AND (pyde.xx.xx:AddressDetails.LocalityName.Text CONTAINS NUMERICS SET(0-9))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000410 |

The following rules apply to each **electroniccontacttelephone1.xx.xx:ElectronicContactTelephone** tuple.

| **Rule Implementation Schematron ID** | **Rule** | **Associated Response Message Code** |
| --- | --- | --- |
| VR.ATO.GEN.432016 | WHERE IN TUPLE(electroniccontacttelephone1.xx.xx:ElectronicContactTelephone)  IF pyde.xx.xx:ElectronicContact.Telephone.Minimal.Number = NULL OR pyde.xx.xx:ElectronicContact.Telephone.Usage.Code <> "03" OR pyde.xx.xx:Electronic Contact.Telephone.ServiceLine.Code <> SET ("01", "02")  RETURN VALIDATION MESSAGE  ENDIF | CMN.ATO.GEN.432016 |

The following rules apply to each **organisationname2.xx.xx:OrganisationNameDetails** tuple.

| **Rule Implementation Schematron ID** | **Rule** | **Associated Response Message Code** |
| --- | --- | --- |
| VR.ATO.GEN.410039 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text <> NULLORBLANK) AND (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text DOES NOT CONTAIN SET("a-z", "A-Z", "0-9"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410039 |
| VR.ATO.GEN.428258 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text CONTAINS " - " )  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000413 |
| VR.ATO.GEN.428259 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text CONTAINS "P/L")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000414 |
| VR.ATO.GEN.428260 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text CONTAINS SET("--","'’"," ") )  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000416 |
| VR.ATO.GEN.410038 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text STARTSWITH "T/A ") AND (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text ENDSWITH SET(" Pship"," P'ship"," P/ship"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410038 |
| VR.ATO.GEN.410140 | IN TUPLE(organisationname2.xx.xx:OrganisationNameDetails) IF (pyde.xx.xx:OrganisationNameDetails.Currency.Code = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410140 |
| VR.ATO.GEN.410147 | WHERE IN TUPLE(organisationname2.xx.xx:OrganisationNameDetails) IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalNameType.Code = NULLORBLANK)   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410147 |
| VR.ATO.GEN.410206 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text = FOUND("The trustee","The Exec","exec","The TTE"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.428042 |
| VR.ATO.GEN.410215 | WHERE IN TUPLE(organisationname2.xx.xx:OrganisationNameDetails) IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410215 |
| VR.ATO.GEN.430850 | IF pyde.xx.xx:OrganisationNameDetails.Currency.Code <> NULLORBLANK AND pyde.xx.xx:OrganisationNameDetails.Currency.Code <> SET ("C", "P")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.430850 |
| VR.ATO.GEN.428261 | IF (pyde.xx.xx:OrganisationNameDetails.OrganisationalName.Text ENDSWITH SET(" T/A"," T/A P'ship"," T/A Pship"," T/A P/Ship"," T/A Partnership"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000417 |

The following rules apply to each **perioddetails1.xx.xx.PeriodDetails** tuple.

| **Rule Implementation Schematron ID** | **Rule** | **Associated Response Message Code** |
| --- | --- | --- |
| VR.ATO.GEN.408057 | WHERE IN TUPLE (perioddetails1.xx.xx.PeriodDetails) IF (pyin.xx.xx:Period.Type.Code = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.408057 |
| VR.ATO.GEN.408059 | WHERE IN TUPLE(perioddetails1.xx.xx.PeriodDetails) IF pyin.xx.xx:Period.Start.Date = NULL OR pyin.xx.xx:Period.End.Date = NULL  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.408059 |
| VR.ATO.GEN.408060 | WHERE IN TUPLE (perioddetails1.xx.xx.PeriodDetails) IF pyin.xx.xx:Period.End.Date <= pyin.xx.xx:Period.Start.Date  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.408060 |

The following rules apply to each **personstructuredname3.xx.xx:PersonNameDetails** tuple.

| **Rule Implementation Schematron ID** | **Rule** | **Associated Response Message Code** |
| --- | --- | --- |
| VR.ATO.GEN.000458 | IF (pyde.xx.xx:PersonNameDetails.NameSuffix.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.NameSuffix.Text <> SET(DOMAIN(SUFFIX CODES)))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000458 |
| VR.ATO.GEN.000459 | IF (pyde.xx.xx:PersonNameDetails.Title.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.Title.Text <> SET(DOMAIN(TITLE CODES)))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000459 |
| VR.ATO.GEN.410040 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.FamilyName.Text DOES NOT CONTAIN SET("A-Z","a-z"))  RETURN VALIDATION MESSSAGE ENDIF | CMN.ATO.GEN.410040 |
| VR.ATO.GEN.410063 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.GivenName.Text DOES NOT CONTAIN SET("A-Z","a-z"))  RETURN VALIDATION MESSSAGE ENDIF | CMN.ATO.GEN.410063 |
| VR.ATO.GEN.410131 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text DOES NOT CONTAIN SET("A-Z","a-z")) RETURN VALIDATION MESSAGEENDIF | CMN.ATO.GEN.410131 |
| VR.ATO.GEN.410183 | IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails)IF (pyde.xx.xx:PersonNameDetails.Currency.Code = NULLORBLANK)  RETURN VALIDATION MESSAGE END IF | CMN.ATO.GEN.410183 |
| VR.ATO.GEN.410200 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text = FOUND("Exec for","Rep for","Trustee for"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000424 |
| VR.ATO.GEN.410201 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text = FOUND("MR","MRS","MISS","MS"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000426 |
| VR.ATO.GEN.410202 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text = FOUND("ESQ","II","III","IV","JNR","JP","MHA","MHR","MLA","MLC","MP","QC","SNR"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000436 |
| VR.ATO.GEN.410203 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text = FOUND("Exec for","Rep for","Trustee for"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000437 |
| VR.ATO.GEN.410204 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text = FOUND("MR","MRS","MISS","MS")) RETURN VALIDATION MESSAGEENDIF | CMN.ATO.GEN.000438 |
| VR.ATO.GEN.410207 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text = FOUND("ESQ","II","III","IV","JNR","JP","MHA","MHR","MLA","MLC","MP","QC","SNR"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000448 |
| VR.ATO.GEN.410208 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text = FOUND("Exec for","Rep for","Trustee for"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000449 |
| VR.ATO.GEN.410209 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text = FOUND("MR","MRS","MISS","MS"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000450 |
| VR.ATO.GEN.410216 | WHERE IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails) IF (pyde.xx.xx:PersonNameDetails.PersonNameType.Code = NULLORBLANK)   RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410216 |
| VR.ATO.GEN.410217 | WHERE IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails) IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.410217 |
| VR.ATO.GEN.410218 | IN TUPLE(personstructuredname3.xx.xx:PersonNameDetails) IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text <> NULLORBLANK) AND (pyde.xx.xx:PersonNameDetails.GivenName.Text = NULLORBLANK)  RETURN VALIDATION MESSAGE   ENDIF | CMN.ATO.GEN.410218 |
| VR.ATO.GEN.428231 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text = FOUND("ESQ","II","III","IV","JNR","JP","MHA","MHR","MLA","MLC","MP","QC","SNR"))  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000422 |
| VR.ATO.GEN.428262 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text CONTAINS " - ")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000423 |
| VR.ATO.GEN.428263 | IF (pyde.xx.xx:PersonNameDetails.FamilyName.Text CONTAINS SET("--","'’"," ") )  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000427 |
| VR.ATO.GEN.428264 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text CONTAINS " - ")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000434 |
| VR.ATO.GEN.428265 | IF (pyde.xx.xx:PersonNameDetails.GivenName.Text CONTAINS SET("--","'’"," ") )  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000439 |
| VR.ATO.GEN.428266 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text CONTAINS " - ")  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000446 |
| VR.ATO.GEN.428267 | IF (pyde.xx.xx:PersonNameDetails.OtherGivenName.Text CONTAINS SET("--","'’"," ") )  RETURN VALIDATION MESSAGE ENDIF | CMN.ATO.GEN.000451 |

Appendix E – Domain Definitions

**DOMAIN(COUNTRY Codes)**

The domain of valid Country codes is defined in the SBR Definitional Taxonomy for the element pyde.xx.xx:AddressDetails.Country.Code

**DOMAIN(TITLE CODES)**

| **Code** | **Title** | **Code** | **Title** |
| --- | --- | --- | --- |
| 2LT | Second Lieutenant | LBDR | Lance Bombardier |
| AB | Able Seaman | LCPL | Lance Corporal |
| ABBOT | Abbot | LORD | Lord |
| AC | Airman/Aircraftman | LS | Leading Seaman |
| ACM | Air Chief Marshal | LT | Lieutenant |
| ADM | Admiral | LT CMDR | Lieutenant Commander |
| AIR CDRE | Air Commodore | LT COL | Lieutenant Colonel |
| ALDERMAN | Alderman | LT GEN | Lieutenant General |
| AM | Air Marshal | MADAM | Madam |
| ARCHBISHOP | Archbishop | MAJ | Major |
| ARCHDEACON | Archdeacon | MAJ GEN | Major General |
| ASSOC PROF | Associate Professor | MASTER | Master |
| AVM | Air Vice Marshal | MATRON | Matron |
| BARON | Baron | MAYOR | Mayor |
| BARONESS | Baroness | MAYORESS | Mayoress |
| BISHOP | Bishop | MIDN | Midshipman |
| BR | Brother | MISS | Miss |
| BRIG | Brigadier | MON | Monsignor |
| CANON | Canon | MOST REV | Most Reverend |
| CAPT | Captain | MR | Mr |
| CARDINAL | Cardinal | MRS | Mrs |
| CDRE | Commodore | MS | Ms |
| CDT | Cadet | PASTOR | Pastor |
| CHAP | Chaplain | PATRIARCH | Patriarch |
| CMDR | Commander | PLT OFF | Pilot Officer |
| CMM | Commissioner | PO | Petty Officer |
| COL | Colonel | PRIOR | Prior |
| CONST | Constable | PROF | Professor |
| COUNT | Count | PTE | Private |
| COUNTESS | Countess | RABBI | Rabbi |
| CPL | Corporal | RADM | Rear Admiral |
| CPO | Chief Petty Officer | RECTOR | Rector |
| DAME | Dame | REV | Reverend |
| DEACON | Deacon | RF | Representative for |
| DEACONESS | Deaconess | RT HON | Right Honourable |
| DEAN | Dean | RT REV | Right Reverend |
| DEPUTY SUPT | Deputy Superintendent | RT REV BISHOP | Right Reverend Bishop |
| DR | Doctor | RT REV MON | Right Reverend Monsignor |
| DUCHESS | Duchess | SBLT | Sub Lieutenant |
| DUKE | Duke | SEN | Senator |
| EARL | Earl | SGT | Sergeant |
| EF | Executor for | SIR | Sir |
| FLGOFF | Flying Officer | SMN | Seaman |
| FLT LT | Flight Lieutenant | SNR CONST | Senior Constable |
| FR | Father | SQN LDR | Squadron Leader |
| FSGT | Flight Sergeant | SR | Sister |
| GEN | General | SSGT | Staff Sergeant |
| GNR | Gunner | SUPR | Superintendent |
| GP CAPT | Group Captain | SWAMI | Swami |
| HON | Honourable | TF | Trustee for |
| HON JUDGE | Honourable Judge | VADM | Vice Admiral |
| HON JUST | Honourable Justice | VERY REV | Very Reverend |
| HRH | His/Her Royal Highness | VICAR | Vicar |
| INSP | Inspector | VISCOUNT | Viscount |
| JUDGE | Judge | WG CDR | Wing Commander |
| JUST | Justice | WO | Warrant Officer |
| LAC | Leading Aircraftman | WO1 | Warrant Officer Class 1 |
| LACW | Leading Aircraftwoman | WO2 | Warrant Officer Class 2 |
| LADY | Lady |  |  |

**DOMAIN(SUFFIX CODES)**

| **Code** | **Suffix** | **Code** | **Suffix** |
| --- | --- | --- | --- |
| ESQ | Esquire | MHR | Member House of Representatives |
| II | The Second | MLA | Member Legislative Assembly |
| III | The Third | MLC | Member Legislative Council |
| IV | The Fourth | MP | Member of Parliament |
| JNR | Junior | QC | Queens Council |
| JP | Justice of the Peace | SNR | Senior |
| MHA | Member House of Assembly |  |  |