



**Suruhanjaya Sekuriti**  
**Securities Commission**  
**Malaysia**

Suruhanjaya Sekuriti  
Securities Commission Malaysia  
**SC XBRL Taxonomy Framework Architecture**

---

Document version: 1.0  
Document date: 2017-03-31

## Index

1	Introduction.....	3
2	Scope of the framework.....	3
3	Data model.....	3
4	Relation to XBRL specifications.....	4
5	Publication and distribution.....	4
6	Location and modularisation in folder and files.....	5
6.1	Location.....	5
6.2	Physical modularisation of the taxonomy.....	5
6.2.1	Definition layer.....	6
6.2.2	Reports layer.....	6
7	Definitions of concepts, dimensional constructs and other.....	6
7.1	Namespaces.....	7
7.2	Concepts and constructs.....	7
7.2.1	Reportable concepts (metrics).....	7
7.2.2	Abstract constructs.....	8
7.2.3	Dimensional constructs.....	8
7.3	Roles used on extended links.....	9
8	Sets of relations.....	10
9	Labels.....	11
10	Application of dimensions.....	12
10.1	Typed dimensions.....	12
11	Business rules.....	12
12	Relation to other taxonomies.....	13
13	Extensibility.....	13
14	Versions.....	13
15	Instance documents.....	13
15.1	Schema reference.....	13
15.2	Multilingual values.....	13
15.3	Precision of values.....	13
16	Enquiries.....	14
17	Appendix.....	15

## 1 Introduction

The purpose of this document is to present and explain the architecture of the framework of XBRL taxonomy created by the Securities Commission Malaysia (SC). In particular, it explains the scope (coverage of information requirements), modularisation in files, manner of defining concepts and relations, and other important design aspects.

This document is aimed at users of the SC XBRL Taxonomy, in particular, business users working with the taxonomy in order to produce instance documents (by applying mappings to internal systems or assigning XBRL tags with values in any other manner) as well as developers of IT solutions facilitating reporting in the XBRL format or analysis of XBRL data.

## 2 Scope of the framework

The SC XBRL Taxonomy framework reflects information requirements for filings to the SC. Under the current version, the scope of the framework is limited to the following fund management industry submission reports:

- FM01.M.UTF – UTF Statistical Returns
- FM01.M.WF – WF Statistical Returns
- FM02.M.UTF – UTF Investment Returns
- FM02.M.WF – WF Investment Returns
- FM03.M.PRS – PRS Statistical and Investment Returns
- FM04.M.UTF – UTF Statistical Returns - Foreign
- FM04.M.WF – WF Statistical Returns - Foreign
- FM05.Q.ETF – ETF Statistical and Investment Returns
- FM05.Q.REIT – REIT Statistical and Investment Returns
- FM06.Q.ETFa – ETF Statistical Returns - Foreign ACMF
- FM06.Q.ETFb – ETF Statistical Returns - Foreign Non-ACMF
- FM06.Q.REIT – REIT Statistical Returns - Foreign
- FM08.M.FMC – Monthly FMC Report
- FM09.H.FMC – Half-yearly FMC Report
- FM10.M.FMC – Monthly Complaints Disclosure
- FM11.H.FMC – Compliance Report
- FM12.A.FMC – Company Profile
- FM14.O.ALL – Audited Financial Statements
- FM15.H.ALL – Employee Report
- FM16.M.UTF – Compliance Review Report – UTF
- FM16.M.PRS – Compliance Review Report – PRS
- FM16.Q.ETF – Compliance Review Report – ETF
- FM16.Q.REIT – Compliance Review Report - REIT

## 3 Data model

Prior to the development of taxonomy, information requirements need to be analysed in order to identify reportable concepts and relations between them. This is done in the form of data model.

The data model for the SC XBRL Taxonomy framework has been created as a result of analysis of the existing submission reports used for fund management industry. For modelling of data (in terms of

methodology and format), SC followed the best practice approaches of the Eurofiling project<sup>1</sup>. In particular, the SC applied the Data Point Modelling (DPM) methodology in defining the data model for the description of the exchanged data.

#### 4 Relation to XBRL specifications

Taxonomy developed by the SC is compliant with the XBRL 2.1 specification<sup>2</sup> as of December 31, 2003 with errata corrections to February 20, 2013, and the Dimensions 1.0 specification<sup>3</sup> as of September 18, 2006 with errata corrections to January 25, 2012.

The business rules layer in the form of linkbase files is defined according to the XBRL Formula 1.0 specification<sup>4</sup> (2009 – 2013) and supporting specifications (Generic Links – June 22, 2009). Assertion tests may also use XPath/XQuery and XBRL Functions.

For enumerated metric's dropdowns, the taxonomy utilises the Extensible Enumerations 1.0 specification<sup>5</sup> from October 30, 2014.

#### 5 Publication and distribution

The SC XBRL Taxonomy is distributed as a package (called as "SC XBRL Taxo Pack") according to the Taxonomy Packages 1.0 specification<sup>6</sup> as of April 19, 2016. This allows users to quickly identify relevant entry points and enables software to automatically configure the necessary remapping.

SC XBRL Taxo Pack contents are defined under the *META-INF* container within the package. The *META-INF* includes the following configuration files:

- *catalog.xml* that provides URL remapping to the taxonomy folder stored locally;
- *taxonomyPackage.xml* that provides general information about the taxonomy identification, versioning, publisher and available entry points.

---

<sup>1</sup> Eurofiling is an open joint initiative in collaboration with the EBA, EIOPA and XBRL Europe, as well as stakeholders like central and commercial banks, supervisors over banking systems, data exchange solutions providers and others. Deliverables of the Eurofiling project can be found on <http://www.eurofiling.info/>.

<sup>2</sup> <http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html>

<sup>3</sup> <https://www.xbrl.org/specification/dimensions/rec-2012-01-25/dimensions-rec-2006-09-18+corrected-errata-2012-01-25-clean.html>

<sup>4</sup> <https://specifications.xbrl.org/work-product-index-formula-formula-1.0.html>

<sup>5</sup> <https://specifications.xbrl.org/work-product-index-extensible-enumerations-extensible-enumerations-1.0.html>

<sup>6</sup> <https://www.xbrl.org/Specification/taxonomy-package/REC-2016-04-19/taxonomy-package-REC-2016-04-19.html>

## 6 Location and modularisation in folder and files

### 6.1 Location

The official location of all components in the framework is <https://www.sc.com.my/xbrl/taxonomy/> (base URL).

The components of the framework are kept in the files placed in folders under the base URL according to the pattern: *{base URL}/{taxonomy layer}/{abbreviated name of a component of the framework}/{file name}*.

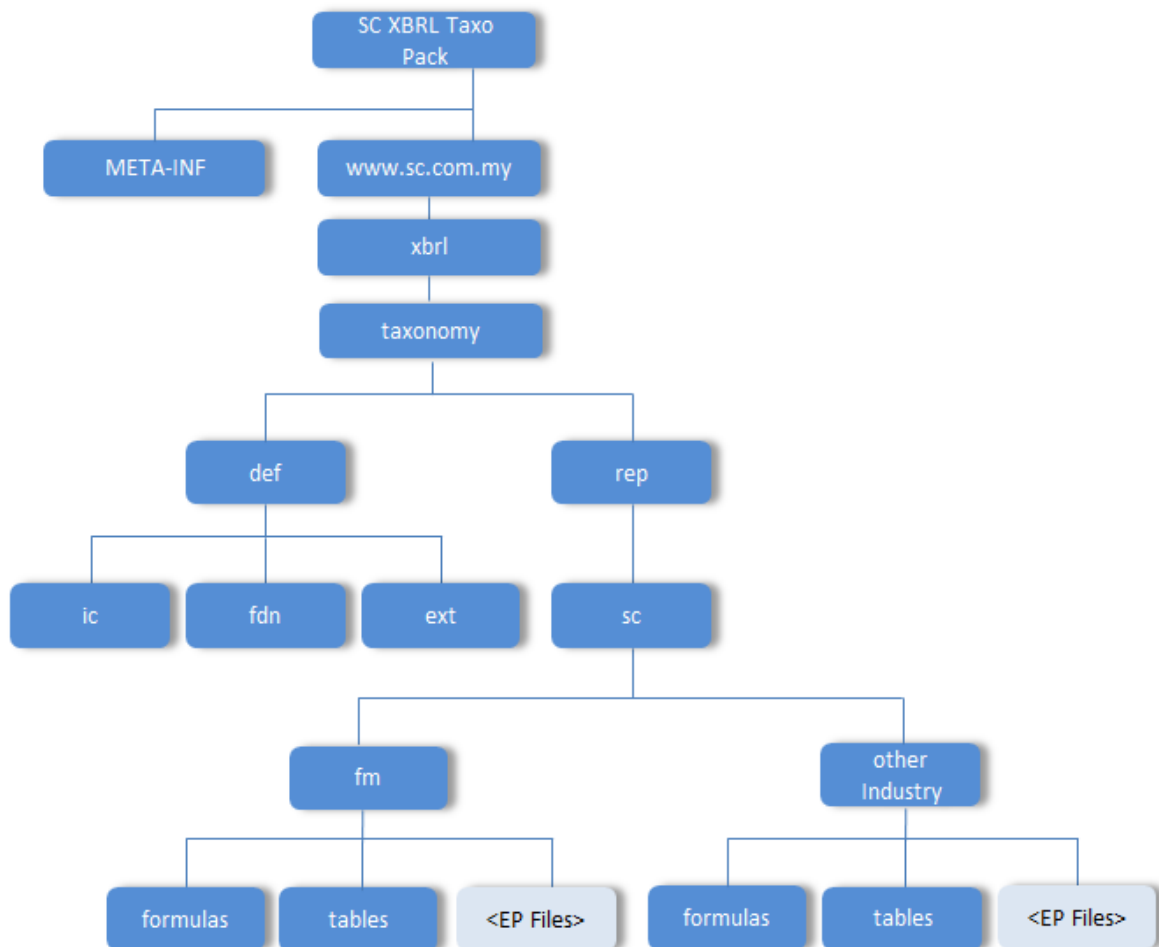
### 6.2 Physical modularisation of the SC XBRL Taxo Pack

The SC XBRL Taxonomy consists of sets of taxonomy files divided into:

- Definition layer (*def*) which stores common dictionary, reusable by different report types along with locally stored external schemas that are used across the taxonomy;
- Reports layer (*rep*) reflecting representations of submission reports containing all necessary relations between elements defined in the common dictionary

Figure 1 presents high-level physical modularisation of the SC XBRL Taxo Pack.

Figure 1: High-level physical modularisation of the SC XBRL Taxo Pack



### 6.2.1 Definition layer

All parts of the SC XBRL Taxonomy are defined within *def* folder as common dictionaries, reusable by different submission reports. Definition layer consists of three subfolders:

- *ext* folder which is a dedicated placeholder for any external schema files to be stored locally for mapping purposes. Currently, no external schemas are stored within the SC taxonomy package as all of them are imported directly from their official locations (absolute references). In the future versions, *ext* folder may contain external taxonomies that can be reused and/or extended by SC XBRL Taxonomy;
- *fdn* folder which is a dedicated placeholder for any custom schemas to be defined by the SC in case of standard specifications not providing the necessary constructs/mechanisms to meet the SC-specific requirements. In the current version of the taxonomy, no custom attributes, data types or other constructs were defined;
- *ic* folder that stores the core schema (*scm\_2017-03-31.xsd*) file with the definition of business concepts and their XBRL properties along with Label Linkbase file (*lab\_scm-en\_2017-03-31.xml*) that provides human readable etiquettes for all elements defined in the dictionary and Definition Linkbase file (*def\_scm-en\_2017-03-31.xml*) that defines hierarchical structures for the dimensional elements described in the taxonomy.

### 6.2.2 Reports layer

The reports layer contains all the information requirements, including the visualisation of the reporting templates. The reporting templates are constructed in the form of linkbase files that provide the relationships between concepts defined in the core schema. Each industry will have its own report layer (e.g. fund management industry report layer will be specified under */rep/sc/fm/* folder). From the modularisation perspective, the SC XBRL Taxonomy defines three main components within the reports layer:

- *formula* folder that contains Formula Linkbase file with the definition of business rules and validation checks using the Formula 1.0 Specification;
- *tables* folder that contains definition of relationships between the elements from the core schema in order to reflect the view of original submission templates.
- *entry point files* which is a filing report that contains a set of reporting templates (as defined in *tables* folder) and reporting scenario (e.g. frequency, entity type, etc.) of a particular filing. Please refer to [Appendix](#) for the full list of entry point files.

## 7 Definitions of concepts, dimensional constructs and other

Concepts, dimensional constructs and other artefacts are defined in XBRL schema files.

In particular, XBRL schema files contain definitions of:

- reportable concepts (items);
- non-reportable concepts (abstracts) used to support browsing of the taxonomy tree structures (relations);

- dimensional constructs (hypercube items, dimension items and domain members) that reflect breakdowns or special characteristics applicable for reportable concepts;
- enumeration drop down list elements;
- `roleTypes` used on definition extended links that combine relationships of a specific nature or application (e.g. *General Information and Declaration*).

## 7.1 Namespaces

Namespaces are used to differentiate between concepts defined by different regulations or applicable for different purposes. They are used in order to avoid name clashes and indicate the origin of each defined concept or construct.

Namespaces in this framework are constructed using the base URL of `https://www.sc.com.my/xbml/taxonomy/` followed by a particular component of a *rep* layer framework. To facilitate better versioning and maintenance of the submission templates, a date component {YYYY-MM-DD} was introduced to the namespace of the *rep* layer (not used in *def*). The list of the currently used namespaces and prefixes (or their patterns) applied consistently across the framework is defined in Table 1.

Table 1: Namespaces and prefixes currently used in SC XBRL taxonomy

Recommended prefix	Namespace
scm	https://www.sc.com.my/xbml/taxonomy/scm
fm{report_code}	https://www.sc.com.my/xbml/taxonomy/{YYYY-MM-DD}/fm{report_code}
t{table_code}	https://www.sc.com.my/xbml/taxonomy/{YYYY-MM-DD}/fm/t{table_code}

## 7.2 Concepts and constructs

As described above, the SC XBRL Taxonomy contains definitions of reportable concepts (items), non-reportable concepts (abstracts) and dimensional constructs (hypercubes, dimensions and domain members).

All concepts are defined in `item` substitution group or derived from it (`hypercubeItem` for hypercubes and `dimensionItem` for dimensions). All concepts are nillable (`@nillable="true"`) hence, they can be reported as nil (`@xsi:nil="true"`). Although semantically unimportant, values of `@id` attribute (used for the purpose of creating links in XLink) are constructed basing on the pattern: `{recommended prefix}_{element name}`<sup>7</sup>.

### 7.2.1 Reportable concepts (metrics)

Definition of a reportable (non-abstract, i.e. `@abstract="false"`) concept must at minimum consist of:

- unique local (within a namespace) name;
- indication of a period type;
- identification of a data type.

<sup>7</sup> As defined in <http://www.xbrl.org/technical/guidance/FRTA-RECOMMENDATION-2005-04-25+corrected-errata-2006-03-20.htm> (FRTA) rule 2.1.5. This pattern applies to items, abstracts, dimensional constructs, `roleTypes`, data types and enumerations for custom data types.

Value of @periodType attribute is either `instant` for these concepts that are reported at a point of time (as of specified date) or `duration` for concepts representing flows and changes (between specified dates or infinite). In cases where the period type is not obvious, the period type attribute is set to `duration`<sup>8</sup>. This information must be taken into consideration in instance document when constructing contexts for facts based on reportable concepts. For textual concepts (i.e. concepts in which data type is `stringItemType`), start and end dates should reflect the boundary dates for which a report is created. For example, if the statement covers the third quarter of 2010, then the dates are 2010-07-01 and 2010-09-30 respectively. Similar reasoning applies for concepts that represent dates. For instance, in an annual report commencing June 2009, the context start and end dates for date concepts such as “Reporting period - From”, “Reporting period - To” would be 2009-07-01 and 2010-06-30 respectively.

The code ({name}) for each metric is composed of three components:

- a letter that represents the data type in lowercase (for available options, see below table);
- a letter that represents the period type characteristics (*i* for instant and *d* for duration);
- a number that corresponds to the numeric code in the model (no zero padding or predetermined length).

**Table 2: Model and XBRL data type for metrics and local name codification letter**

Model data type	XBRL data type	Local name codification letter
Monetary (currency)	xbri:monetaryItemType	m
Percent or Ratio	num:percentItemType	p
Decimal	xbri:decimalItemType	r
Integer	xbri:integerItemType	i
Text	xbri:stringItemType	s
Date	xbri:dateItemType	d
Boolean	xbri:booleanItemType	b
Enumeration	enum:enumerationItemType	e

### 7.2.2 Abstract constructs

All non-reportable concepts have @abstract="true".

Names of abstract constructs correspond to the following pattern: `Abstract{subsequent number}`

Although it is semantically unimportant, all abstracts are @periodType="duration" and @dataType="stringItemType".

### 7.2.3 Dimensional constructs

Definition of dimensional constructs follows the rules of the XBRL Dimensions 1.0 specification and the recommendations and deliverables of the XBRL International Working Group on Interoperable Taxonomy Architecture.

<sup>8</sup> This is for the reasons that it is always possible to indicate a moment in time using two identical dates (more precisely data and time) while it is not possible to describe a period of time using just a single date. The same approach was taken by the IFRS taxonomy.



According to the above, all dimensional constructs are abstracts (`@abstract="true"`). Moreover, hypercubes are defined in `hypercubeItem` and dimensions in `dimensionItem` substitutions groups<sup>9</sup>, data type of domain members is `domainItemType`, and, although semantically unimportant, all dimensional constructs are `@periodType="duration"` and hypercubes and dimensions are `@dataType="stringItemType"`.

The code (`{name}`) of each domain member corresponds to its code assigned in the data model which is in general starts with two letter domain code followed by an underscore and lowercase letter *x* (due to XML naming restrictions disallowing digit as the starting character) followed by a sequential number (with *x0* reserved for the default member). However, if the concept represented already has a widely accepted standard codification like ISO codes, the name will match the existing codification, for example:

- ISO 4217: standard currency codes composed of three alphabetical characters;

SC XBRL Taxonomy defines a number of typed dimensions and typed domains (see Section 10.1 for details). Definition of a typed domain must consist of a `@name`, `@dataType` and `@id`. Names of typed domains reflect the camel-case description of their content followed by the component `identifier`. `@dataType` used for all typed domains is set to `string` (XML Schema data type).

### 7.3 Roles used on extended links

XBRL schema files contain declarations of `roleTypes` used on extended links. They distinguish and name the sets of relations defined in linkbase files.

Declaration of a `roleType` consists of a `@roleURI` attribute (which is a unique identification of a role in a form a Unified Resource Identifier<sup>10</sup> that allows creation of networks of relations spread across many files), an `@id` attribute (used for linking purposes with no semantic meaning) and subelements: `definition` (containing human readable description in a single language), and `usedOn` specifying the kinds of extended links (or resources) where the declared `roleType` can be applied.

In general, `roleTypes` are created due to business reasons - in order to support browsing of taxonomy by separating information sets (concepts and relations between them) that represent certain parts of reporting requirements (e.g. all concepts from *General Information and Declaration* and relations between them).

In terms of structuring of relationships in the definition linkbase, the SC XBRL Taxonomy distinguishes between the following two types of `roleTypes`, which are used to define:

- a) **reporting templates** containing items linked to dimensional information defined in the data model;
- b) **hierarchies** containing relationships of domain members defined in the data model.

Description of a type of information set (regarding which group it applies to) is expressed by the value of `@roleURI` attribute and `definition` element.

<sup>9</sup> As defined in <http://www.xbrl.org/2005/xbrldt-2005.xsd>.

<sup>10</sup> [http://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://en.wikipedia.org/wiki/Uniform_Resource_Identifier)

For all `roleTypes` from group a), `definition` element contains a number component that consists of a three-digit identifier followed by a dot, followed by two-digit identifier, all within square brackets for a given information set (e.g. `[010.01]`). First three digits identify a particular reporting template and the remaining two digits provide information about the variant of the template that is required for a reporting scenario.

Construction of `@roleURIS` for reporting templates begins with the following base URI: `https://www.sc.com.my/xbrl/taxonomy/{date of publication in format YYYY-MM-DD}/{module}/` followed by the template code (e.g. `https://www.sc.com.my/xbrl/taxonomy/2017-03-31/fm/t110_06`).

Relationships of domain members (hierarchies from group b) defined in the data model are represented using XBRL extended link roles in which role type URI is built according to the following pattern: `https://www.sc.com.my/xbrl/taxonomy/dom/{DC}/{relationship code}`, where `{DC}` represents the code of the domain in uppercase and `{relationship code}` is the numeric code of the hierarchy. The value of the `definition` element contains the component `{DC_relationship code}` followed by the name of the hierarchy as described in the data model (e.g. `CU_1: Currency`).

`roleTypes` defined in the SC XBRL Taxonomy can be used on `presentationLink` and `definitionLink`. However, in some cases they can be used on generic link for formula linkbase and `labelLink` for table-specific labels (described later in this document).

Example of `roleType` declaration defined in the SC XBRL Taxonomy is presented on Code example 1 and 3.

#### Code example 1: Custom roleType declaration for reporting template

```
<link:roleType roleURI="https://www.sc.com.my/xbrl/taxonomy/2017-03-31/fm/role/t110_06" id="t110_06">
  <link:definition>[110.06] Financial - Others</link:definition>
  <link:usedOn>link:presentationLink</link:usedOn>
  <link:usedOn>link:definitionLink</link:usedOn>
  <link:usedOn>link:labelLink</link:usedOn>
  <link:usedOn>gen:link</link:usedOn>
</link:roleType>
```

#### Code example 2: Custom roleType declaration for hierarchy

```
<link:roleType roleURI="https://www.sc.com.my/xbrl/taxonomy/2017-03-31/dom/CU/1" id="CU_1">
  <link:definition>CU_1: Currency</link:definition>
  <link:usedOn>link:definitionLink</link:usedOn>
</link:roleType>
```

## 8 Sets of relations

The SC XBRL Taxonomy contains relations in the presentation, definition linkbases and generic linkbases. Relations in the presentation linkbase are constructed as defined in the XBRL 2.1 Specification. Definition linkbase relations are constructed using arcs with `arcroles` defined in XBRL

Dimensions 1.0<sup>11</sup>. Generic linkbase relations should be constructed using arcs with arcroles defined in the Generic Labels 1.0. and Formula 1.0. with the supporting specifications.

Relations provide semantic information on:

- placement of concepts (items with or without dimensional information) in particular information sets (such as *Fund Overview*, *Fund Contribution through EPF*, etc.);
- application of dimensional information for items;
- hierarchical structures between items and members of dimensions;
- compliance checks between items and members of dimensions.

In order to facilitate browsing and discovery of the content of the SC XBRL Taxonomy, relations have been split in sets according to certain rationale.

In case of presentation and definition linkbases that are storing structures for the templates, the reportable information can be found in extended links of which the `definition` element contains the template code following pattern of [XXX.XX]. With regards to the hierarchies of domain members that are used, for example, as enumeration lists, their definition linkbase structures can be found in extended links of which the `definition` element contains {DC\_relationship code}.

For better alliance with different software solutions regarding generation of Excel forms and visualisation of the templates, presentation and definition linkbases are sharing the same hierarchies (presentation for non-dimensional structures and definition for dimensional structures) with exception to hierarchies that are stored only in the definition linkbase (not meant for reporting).

Each type of linkbase is stored in a separate file following the below naming convention:

- **pre\_scm\_{table code}\_{date stamp in format YYYY-MM-DD}.xml** for presentationLink type of the extended link, storing complete hierarchies reflecting a particular reporting template;
- **def\_scm\_{table code}\_{date stamp in format YYYY-MM-DD}.xml** for definitionLink type of the extended link, storing complete hierarchies reflecting particular template;
- **scm-{entry point name}-formula.xml** for formula linkbase set of relationships, storing business rules and compliance checks described later in this document.

## 9 Labels

Currently, the SC XBRL Taxonomy contains English labels for each concept for easier browsing of the taxonomy contents.

Each reporting template stores a separate list of labels for the concepts defined in the taxonomy that are specific to the particular form and defined under the extended link role with the presentation and definition hierarchy of the table; naming convention for these files is as follows:

- **lab-{language code}\_scm\_{table code}\_{date stamp in format YYYY-MM-DD}.xml**; these linkbase files are referenced from the schema file.

An example of these form-specific labels is presented on Code example 3.

---

<sup>11</sup> [http://www.xbrl.org/Specification/XDT-REC-2006-09-18+Corrected-Errata-2009-09-07.htm#\\_Toc243301749](http://www.xbrl.org/Specification/XDT-REC-2006-09-18+Corrected-Errata-2009-09-07.htm#_Toc243301749)

### Code example 3: Example of form-specific label

```
<link:labelLink xlink:type="extended" xlink:role="https://www.sc.com.my/xbrl/taxonomy/2017-03-31/Em/role/t010_01">  
<link:loc xlink:type="locator" xlink:href="../../../def/ic/scm_2017-03-31.xsd#scm_si23" xlink:label="si23" xlink:title="si23"/>  
<link:label xlink:type="resource" xlink:label="label_si23" xlink:role="http://www.xbrl.org/2003/role/label" xlink:title="label_si23"  
xml:lang="en" id="label_si23">Contact Number</link:label>  
<link:labelArc xlink:type="arc" xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label" xlink:from="si23" xlink:to="label_si23"  
xlink:title="label: si23 to label_si23"/>
```

## 10 Application of dimensions

The SC XBRL Taxonomy makes an extensive use of XBRL dimensions<sup>12</sup> of both typed and explicit type.

The designated container for dimensional information in instance documents is `scenario` element as indicated on definition arcs with `http://xbrl.org/int/dim/arcrole/all` arcrole. The taxonomy does not contain any arc with `http://xbrl.org/int/dim/arcrole/notAll`. Every hypercube is closed (`@closed="true"`).

### 10.1 Typed dimensions

Typed dimensions are used for allowing disclosing certain information of a predefined type that needs to be further characterized in order to identify its exact meaning.

In particular, they are used to identify line numbers for particular open tables.

As described in section 7.2.3, typed domains of all typed dimension are always simple constructs which data type is `integer`. In an instance document, typed domain must be instantiated as a unique “key” value linking facts that have something in common.

For example, reporting entities are required to disclose information about their fund contribution through Employee Provident Fund under *[070.01] Fund Contribution through EPF*. Therefore, the taxonomy defines six primary items: *Fund Code*, *Sales – Value (RM)*, *Sales – Volume (unit)*, *Repurchases – Value (RM)*, *Repurchases – Volume (unit)* and *Net Asset Value (RM)* that are linked to a hypercube containing *Line number* typed dimension. In instance document *Line number* should contain a unique typed domain value for a given fund. This is due to the fact that the number of funds cannot be predicted, therefore the table should not be closed.

## 11 Business rules

The SC XBRL Taxonomy defines business rules with use of Formula Linkbase 1.0 and supportive specifications<sup>13</sup>. Currently, all validation checks that are provided in the SC XBRL Taxonomy are value assertions. These are, in most cases, cross-contextual multidimensional checks that could not be covered with the use of standard XBRL 2.1 Calculation Linkbase.

Each business rule provides a human readable error message in English, informing the reporting entities about any inconsistencies that might occur in the instance document.

<sup>12</sup> <http://www.xbrl.org/Specification/XDT-REC-2006-09-18+Corrected-Errata-2009-09-07.htm>

<sup>13</sup> <http://specifications.xbrl.org/work-product-index-formula-formula-1.0.html>

Use of Formula Linkbase is to assure the validity and completeness of reported information submitted to SC by the reporting entities.

## 12 Relation to other taxonomies

Currently, the SC XBRL Taxonomy framework is independent from any external taxonomy.

## 13 Extensibility

The SC XBRL Taxonomy with the current coverage of the information requirements is not extensible. This means that instance documents submitted must not contain any extension concepts or structures.

## 14 Versions

Apart from the versioning information that is provided as part of the use of Taxonomy Packages 1.0 specification (as described in the section 5 of this document), the SC XBRL Taxonomy also provides a date stamp on particular entry point and reporting template. For those components, the publication date is used on file names, namespaces and `roleURIS`.

With each update of any entry point or particular reporting template, the current and the previous versions of the files will be maintained for historical purposes.

## 15 Instance documents

The content of instance documents is determined by the taxonomy that is referenced. There is however a set of other characteristics defined in instance documents that the taxonomy doesn't control<sup>14</sup>. The set of rules and hints below shall help reporting entities to comply with the reporting requirements of the SC XBRL Taxonomy.

### 15.1 Schema reference

Instance documents must reference (using `link:schemaRef` element) one of the entry schema files as listed in the [Appendix](#).

### 15.2 Multilingual values

The SC XBRL Taxonomy at current provides all information in English, i.e. all concepts and roles used on extended links are supplied with English labels.

### 15.3 Precision of values

Precision of reported monetary figures (i.e. information if the amount is in ringgit or thousands/millions of ringgit or with sen decimal places) is set in instance document for each tag by a reporting entity. This is for the reason that a reporting entity is the one that knows how accurate are the numbers being reported. Information about this accuracy is reflected using `@decimals` or `@precision` attributes as described in the XBRL 2.1 Specification<sup>15</sup>. It is therefore possible that an entity reports in an instance document the same information twice: in thousands (with `@decimals="-3"`) and in ringgit (with `@decimals="0"`). This approach however is not recommended

---

<sup>14</sup> These characteristics can be controlled by a formula linkbase that supports the taxonomy.

<sup>15</sup> <http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm>

as it hinders comparability of data and may lead to inconsistencies (two or more different values representing a single fact). Therefore, each fact must be tagged only once with reference to the ISO 4217 currency code and appropriate accuracy. It is the role of a user interface or rendering application to display the numbers properly (i.e. in millions, thousands, etc.).

## **16 Enquiries**

Any enquiries regarding the taxonomy, its architecture or application should be submitted to the following e-mail: [scxbrl@seccom.com.my](mailto:scxbrl@seccom.com.my)

## Appendix

Table 3: List of entry point schemas

Submission report	Entry point schema
FM01.M.UTF	{root folder}\rep\sc\fm\scm-fm01mutf_2017-03-31_full_entry_point.xsd
FM01.M.WF	{root folder}\rep\sc\fm\scm-fm01mwf_2017-03-31_full_entry_point.xsd
FM02.M.UTF	{root folder}\rep\sc\fm\scm-fm02mutf_2017-03-31_full_entry_point.xsd
FM02.M.WF	{root folder}\rep\sc\fm\scm-fm02mwf_2017-03-31_full_entry_point.xsd
FM03.M.PRS	{root folder}\rep\sc\fm\scm-fm03mprs_2017-03-31_full_entry_point.xsd
FM04.M.UTF	{root folder}\rep\sc\fm\scm-fm04mutf_2017-03-31_full_entry_point.xsd
FM04.M.WF	{root folder}\rep\sc\fm\scm-fm04mwf_2017-03-31_full_entry_point.xsd
FM05.Q.ETF	{root folder}\rep\sc\fm\scm-fm05qetf_2017-03-31_full_entry_point.xsd
FM05.Q.REIT	{root folder}\rep\sc\fm\scm-fm05qreit_2017-03-31_full_entry_point.xsd
FM06.Q.ETFa	{root folder}\rep\sc\fm\scm-fm06qetfa_2017-03-31_full_entry_point.xsd
FM06.Q.ETFb	{root folder}\rep\sc\fm\scm-fm06qetfb_2017-03-31_full_entry_point.xsd
FM06.Q.REIT	{root folder}\rep\sc\fm\scm-fm06qreit_2017-03-31_full_entry_point.xsd
FM08.M.FMC	{root folder}\rep\sc\fm\scm-fm08mfmc_2017-03-31_full_entry_point.xsd
FM09.H.FMC	{root folder}\rep\sc\fm\scm-fm09hfmc_2017-03-31_full_entry_point.xsd
FM10.M.FMC	{root folder}\rep\sc\fm\scm-fm10mfmc_2017-03-31_full_entry_point.xsd
FM11.H.FMC	{root folder}\rep\sc\fm\scm-fm11hfmc_2017-03-31_full_entry_point.xsd
FM12.A.FMC	{root folder}\rep\sc\fm\scm-fm12afmc_2017-03-31_full_entry_point.xsd
FM14.O.ALL	{root folder}\rep\sc\fm\scm-fm14oall_2017-03-31_full_entry_point.xsd
FM15.H.ALL	{root folder}\rep\sc\fm\scm-fm15hall_2017-03-31_full_entry_point.xsd
FM16.M.UTF	{root folder}\rep\sc\fm\scm-fm16mutf_2017-03-31_full_entry_point.xsd
FM16.M.PRS	{root folder}\rep\sc\fm\scm-fm16mprs_2017-03-31_full_entry_point.xsd
FM16.Q.ETF	{root folder}\rep\sc\fm\scm-fm16qetf_2017-03-31_full_entry_point.xsd
FM16.Q.REIT	{root folder}\rep\sc\fm\scm-fm16qreit_2017-03-31_full_entry_point.xsd