

This is an overview of the essence of the method<sup>1</sup> we have created for constructing a high-quality XBRL taxonomy and related XBRL instance. The method would not necessarily be obvious; it took us about 10 years to figure it out. We are applying the method to the IPSAS reporting standards to create an XBRL taxonomy for that reporting scheme to further demonstrate and test this method<sup>2</sup>.

Keep in mind that the objective is not to meet poorly defined, substandard quality standards. The objective is to meet a correctly defined quality hurdle; one that you would want to have if you are using XBRL internally for automation of accounting, reporting, auditing, or analysis internally within an organization; and further to be able to prove the quality of the XBRL taxonomy and resulting XBRL instance. Dumbing down the quality standards masks the problem, it does not solve the problem. Our objective is to guarantee high-quality.

For the next level of detail, please see the document *Open Source Framework for Implementing XBRL-based Digital Financial Reporting*<sup>3</sup>. That document points to example XBRL linkbases and other resources that shows exactly how these details are implemented. Ultimately, we will provide a complete IPSAS XBRL Taxonomy implemented using this method and related XBRL instances.

## Overview of Steps

First, recognize that an XBRL taxonomy is not one big thing, it is a collection of smaller pieces. You work with ONE PIECE AT A TIME. I will call each piece a “fragment”. A fragment could be a Network, a Table/Hypercube, or a Block. To create or modify an XBRL taxonomy fragment, the following steps are undertaken:

1. Create (or modify) the fragment following the defined taxonomy **architecture and model structure rules**. Provide all **properties**. Provide **constraints** such as data type, period type, balance type for each report element. Provide any other necessary constraints. Provide documentation, labels, and references to authoritative documentation necessary for report element. Provide any other documentation and references.
2. Create (or modify) the **mathematical rules** to which that fragment relates.
3. Create (or modify) the **continuity cross check rules** to which relate to that fragment.
4. Create (or modify) the **impute rules** to which relate to that fragment.
5. Create (or modify) the **class/type rules** to which that fragment relates.
6. Create (or modify) the **disclosure mechanics** rules to which exist for that fragment.

---

<sup>1</sup> Charles Hoffman, CPA and Rene van Egmond, *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*,

<http://xbrl.azurewebsites.net/2019/Library/MethodForImplementingStandardFinancialReportUsingXBRL.pdf>

<sup>2</sup> International Public Sector Accounting Standards XBRL Taxonomy Prototype Project,

<http://xbrl.squarespace.com/journal/2019/1/16/international-public-sector-accounting-standards-xbrl-taxono.html>

<sup>3</sup> Charles Hoffman, CPA, et. al., *Open Source Framework for Implementing XBRL-based Digital Financial Reporting*, <http://xbrl.azurewebsites.net/2018/Library/OpenSourceFrameworkForImplementingXBRLBasedFinancialReporting-2018-12-05.pdf>

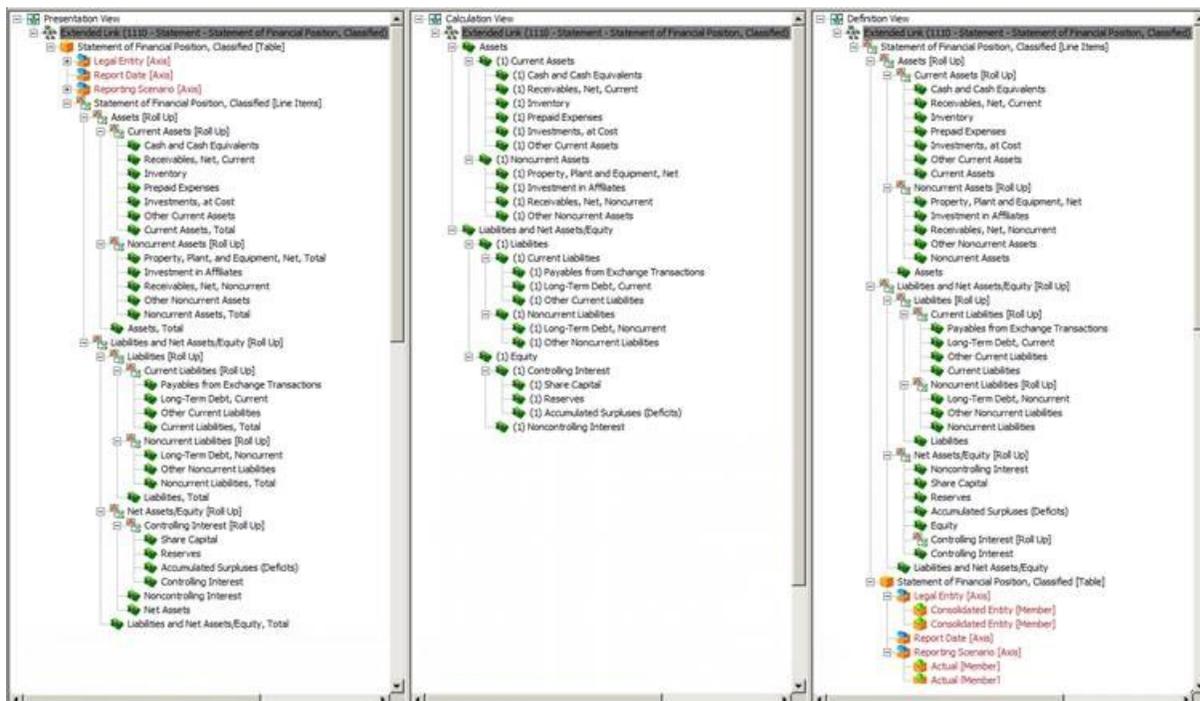
7. Create (or modify) the **reporting checklist rules** to which that fragment relates.
8. Create (or modify) the test (or real) **XBRL instance** to which that fragment relates.
9. **Test** (or re-test) the integrity of the individual fragment XBRL instance and XBRL taxonomy against all defined rules (structural, mathematical, continuity, class/type, mechanical, reporting).
10. **Test** (or re-test) the integrity of the entire report XBRL instance and XBRL taxonomy against all defined rules (structural, mathematical, continuity, class/type, mechanical, reporting).
11. **Test** (or re-test) the consistency of the entire report XBRL instance and XBRL taxonomy against all prior XBRL instances and related XBRL taxonomies.
12. **Test** (or re-test) the consistency of the entire report XBRL instance and XBRL taxonomy against peer or comparable reports XBRL instances and related XBRL taxonomies.

**Repeat** for each new fragment added.

Here is a more detailed walkthrough of the steps which are summarized above. Pretend that you have an empty XBRL taxonomy.

## Step 1: Create (or modify) the fragment following the defined taxonomy architecture and model structure rules.

In this walk through we will first create the fragment “Statement of Financial Position, Classified”. Notice that the XBRL presentation relations, XBRL calculation relations, and XBRL definition relations use the same Network URI to identify the fragment. Notice that an explicit [Table] (or hypercube) was used to define the fragment. How this is done I don’t want to get into, this is software implementation independent, but they do have to get this right.



Provide all required **properties** of a report element. Provide **constraints** such as data type, period type, balance type for each report element. Provide any other necessary constraints. Provide documentation, labels, and references to authoritative documentation necessary for report element. Provide any other necessary documentation and references.

Note that the most restrictive constraints possible should be provided. For example, rather than using a data type of “`xbri:monetaryItemType`”, it is better to use the data type “`dtr:nonNegativeMonetaryItemType`”<sup>4</sup> for concepts that would always be reported using a POSITIVE fact value.

Again, the details of exactly how to represent a fragment within an XBRL taxonomy is very important but is beyond the scope of this overview.

## **Step 2: Create (or modify) the mathematical rules to which that fragment relates.**

So, the majority of the mathematical rules were defined or auto-generated when the XBRL presentation and definition relations were defined in the form of XBRL calculation relations. But, there is ONE additional mathematical rules that needs to be defined:

**“Assets = Liabilities and Net Assets/Equity”**

## **Step 3: Create (or modify) the continuity cross check rules to which relate to that fragment.**

The following continuity cross check rules should be defined for the statement of financial position (balance sheet):

**“Assets = Liabilities and Net Assets/Equity”**

**“Assets = Current Assets + Noncurrent Assets”**

**“Liabilities = Current Liabilities + Noncurrent Liabilities”**

**“Net Assets = Controlling Interest + Noncontrolling Interest”**

**“Liabilities and Net Assets/Equity = Liabilities + Net Assets/Equity”**

---

<sup>4</sup> XBRL International Data Type Registry, <http://www.xbrl.org/dtr/type/numeric-2012-06-30.xsd>

## Step 4: Create (or modify) the impute rules to which relate to that fragment.

Impute rules are used to derive unreported facts. For example, if the line item “Noncurrent Assets” is not reported, the continuity cross check rule “**Assets = Current Assets + Noncurrent Assets**” can be leveraged to derive a value for the non-reported line item “Noncurrent Assets”. Impute rules only need to be provided to the extent that line items are not required to be reported. Or, saying this another way; if every high-level line item is reported than none need to be derived, therefore no impute rules need to be provided. Here are a handful example impute rules:

IF Noncurrent Assets is not reported and Assets was reported and Current Assets was reported  
THEN Noncurrent Assets = Assets - Current Assets.

IF Controlling Interest is not reported and Net Assets was reported and Noncontrolling Interest  
was reported THEN Controlling Interest = Net Assets - Noncontrolling Interest.

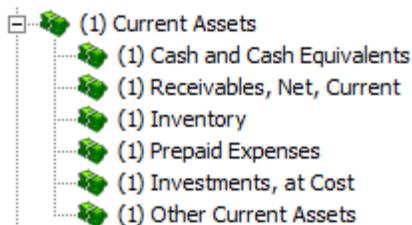
IF Noncurrent Liabilities is not reported and Liabilities was reported and Current Liabilities was  
reported THEN Noncurrent Liabilities = Liabilities - Current Liabilities.

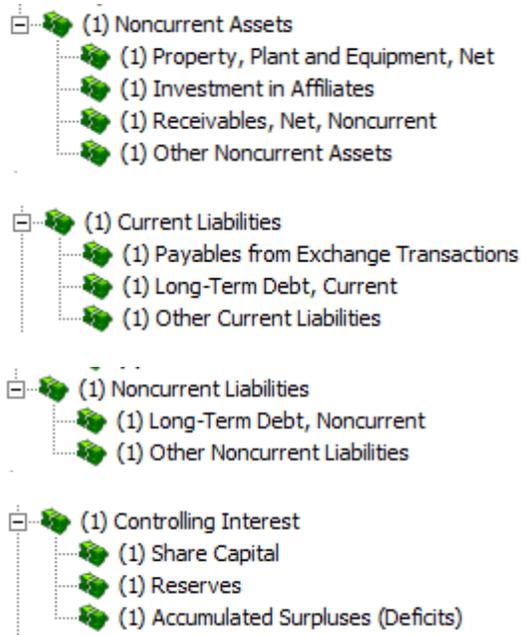
...and so forth.

## Step 5: Create (or modify) the class/type rules to which that fragment relates.

Class/type rules prevent the misuse of concepts in one role that were intended to be used for another role. For example, the concept “Receivables, Net, Noncurrent” is intended to be PART OF Noncurrent Assets and should NEVER be used to represent, say, a Current Asset. Another example of a class/type misuse is to use the concept “Reserves” as PART OF Noncurrent Liabilities. Class/type rules prevent these sorts of errors. This is critical when economic entities creating reports can modify the statement of financial position model.

For the balance sheet, the class/type rules are built into the XBRL calculation relations.





## Step 6: Create (or modify) the disclosure mechanics rules to which exist for that fragment.

Disclosure mechanics rules relate to the mechanical relationships between the report elements that make up a fragment. The following are examples of the disclosure mechanics rules that relate to the statement of financial position, classified (balance sheet).

1. A statement of financial position is made up of TWO disclosures: a roll up of Assets; a roll up of Liabilities and Equity.
2. A statement of financial position disclosure is REQUIRED to be represented using the TABLE “ipsas:StatementOfFinancialPotionClassifiedTable”.
3. Assets [Roll up] is defined to be a concept arrangement pattern of “[Roll Up]”.
4. Assets [Roll up] requires the concept “ipsas:Assets” as the TOTAL concept of the roll up.
5. Liabilities and Net Assets/Equity [Roll up] is defined to be a concept arrangement pattern of “[Roll Up]”.
6. Liabilities and Net Assets/Equity [Roll up] requires the concept “ipsas:NetAssetsEquity” as the TOTAL concept of the roll up.

Other additional rules could be provided such as any [Axis] that is REQUIRED to be reported, any line items that are required to be reported, and any other rules that make it possible to uniquely identify the statement of financial position within an XBRL-based financial report.

## **Step 7: Create (or modify) the reporting checklist rules to which that fragment relates.**

The reporting checklist rules are designed to indicate when a disclosure is required to be provided within a report. A disclosure could always be required such as the balance sheet, income statement, and cash flow statement; or the nature of economic entity, basis of reporting, or significant accounting policies. Or, a disclosure could be required when a specific line items is reported within a financial report. For example, if the “Inventories” line item is provided on a balance sheet then the “Inventories Subclassifications” disclosure is required to also be reported.

Currently, because we are only getting started creating taxonomy fragments, we only have one fragment. Also, the balance sheet is a required disclosure, so we would have the reporting checklist rule:

“A Statement of Financial Position is REQUIRED to exist in a financial report.”

## **Step 8: Create (or modify) the test (or real) XBRL instance to which that fragment relates.**

The next step is to enable the ability to TEST the XBRL taxonomy fragment which has been created as well as the RULES used to both describe how this fragment should be created within an extension XBRL taxonomy and its related XBRL instance and verify using automated machine-based processes that the XBRL instance and the XBRL taxonomy have, in fact, been created per these defined rules. And so, we create the XBRL instance. (Remember that all of the rules are represented in machine-readable form just like the XBRL instance and XBRL taxonomy. All rules are represented using the XBRL technical syntax.)

And so here is what the XBRL instance might look like within a software application:

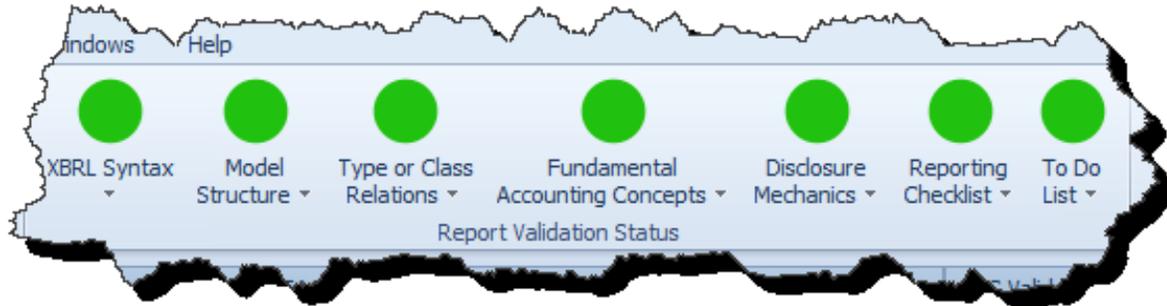
Component: (Network and Table)		
Network	1110 - Statement - Statement of Financial Position, Classified	
Table	Statement of Financial Position, Classified [Table]	
Reporting Entity [Axis]	1234567890 http://regulator.gov/id	
Reporting Scenario [Axis]	Actual [Member]	
Report Date [Axis]	Reported as of March 18, 2020 [Member]	
Legal Entity [Axis]	Consolidated Entity [Member]	
Unit [Axis]	USD	
	Period [Axis] ▼	
Statement of Financial Position, Classified [Line Items]	2010-12-31	2009-12-31
<b>Assets [Roll Up]</b>		
<b>Current Assets [Roll Up]</b>		
Cash and Cash Equivalents	1,000 <sup>1</sup>	1,000
Receivables, Net, Current	1,000 <sup>1</sup>	1,000
Inventory	1,000	1,000
Prepaid Expenses	500	500
Investments, at Cost	500	500
Other Current Assets	1,000	1,000
Current Assets, Total	5,000	5,000
<b>Noncurrent Assets [Roll Up]</b>		
Property, Plant, and Equipment, Net, Total	4,000	4,000
Investment in Affiliates	0	0
Receivables, Net, Noncurrent	0	0
Other Noncurrent Assets	3,000	1,000
Noncurrent Assets, Total	7,000	5,000
Assets, Total	12,000	10,000
<b>Liabilities and Net Assets/Equity [Roll Up]</b>		
<b>Liabilities [Roll Up]</b>		
<b>Current Liabilities [Roll Up]</b>		
Payables and Accruals	3,000	3,000
Long-Term Debt, Current	1,000	1,000
Other Current Liabilities	1,000	1,000
Current Liabilities, Total	5,000	5,000
<b>Noncurrent Liabilities [Roll Up]</b>		
Long-Term Debt, Noncurrent	500	500
Other Noncurrent Liabilities	500	500
Noncurrent Liabilities, Total	1,000	1,000
Liabilities, Total	6,000	6,000
<b>Net Assets/Equity [Roll Up]</b>		
<b>Controlling Interest [Roll Up]</b>		
Share Capital	1,000	1,000
Reserves	2,000	1,000
Accumulated Surpluses (Deficits)	1,000	1,000
Controlling Interest	4,000	3,000
Noncontrolling Interest	2,000	1,000
Net Assets	6,000	4,000
Liabilities and Net Assets/Equity, Total	12,000	10,000

**Step 9: Test (or re-test) the integrity of the individual fragment XBRL instance and XBRL taxonomy against all defined rules (structural, mathematical, continuity, class/type, mechanical, reporting).**

The next step is to run TEST of the fragment (XBRL instance and related XBRL taxonomy) and the RULES used to both describe how this fragment should be created and verify using automated machine-based processes that the XBRL instance and the XBRL taxonomy have, in fact, been created per these defined rules. Since the rules are (a) provided and (b) machine readable; then software can be used to verify the XBRL instance information against the specified rules. (This is as contrast to not having rules or not

having software which would mean that all of these details would need to be tested using manual human-based processes; you CANNOT simply ignore the details by not looking at them.)

There are the validation results provided by two different software applications for this fragment:



			List	List	List	List	List	List
All Components (Networks/Tables)	Status	Count of Relations	XBRL Technical Syntax Rules	Model Structure Rules	Business Rules <sup>(A)</sup>	Roll Up Rules <sup>(B)</sup>	Other Manual Review Tasks	Other Rules and Best Practice Tasks
Automated summary	Completed	108	OK	OK	OK	OK	Not Specified <sup>(C)</sup>	OK
Automated rules defined			205	7	24	6	0	0
Automated rules executed which PASSED			205	7	29	27	0	0
Automated rules executed which FAILED			0	0	0	0	0	0
Manual summary	Completed	108	OK	OK	OK	OK	Not Specified <sup>(C)</sup>	OK
Manual rules defined			0	0	9	6	0	31
Manual rules executed which PASSED			0	0	0	0	0	31
Manual rules executed which FAILED			0	0	0	0	0	0

Since there are no inconsistencies reported, we can rest assured that the report is consistent with all of the provided rules.

**Step 10: Test (or re-test) the integrity of the entire report XBRL instance and XBRL taxonomy against all defined rules (structural, mathematical, continuity, class/type, mechanical, reporting).**

Step 10 would be to put the fragment that we just created in with other fragments. While STEP 9 tests the integrity of the single fragment, STEP 10 tests the fragment relative to other existing fragments. Since we don't have any additional fragments yet, we don't need to perform this step because the results of STEP 9 and STEP 10 would be the same because there is only a single fragment at this point in time.

## Step 11: Test (or re-test) the consistency of the entire report XBRL instance and XBRL taxonomy against all prior XBRL instances and related XBRL taxonomies.

Step 11 we will skip for now because I am the only one creating reports at the moment. This step helps compare different interpretations of the XBRL taxonomy and the rules by comparing and contrasting reports created for multiple periods for the same economic entity.

## Step 12: Test (or re-test) the consistency of the entire report XBRL instance and XBRL taxonomy against peer or comparable reports XBRL instances and related XBRL taxonomies.

Step 12 we will skip for now because, similar to Step 11, I am the only one creating reports at the moment. This step helps compare different interpretations of the XBRL taxonomy and the rules by comparing and contrasting reports created for multiple economic entities for the same period.

## REPEAT! Repeat process for the next new fragment of for any modification of an existing fragment.

After all inconsistencies have been REMOVED by either (a) fixing the XBRL Taxonomy, (b) fixing the XBRL instance, or (c) fixing the rules; we move on to the next report fragment.

To follow this project, please see this blog post:

<http://xbrl.squarespace.com/journal/2019/1/16/international-public-sector-accounting-standards-xbrl-taxono.html>)

## VERIFY: Re-verify the result for the entire report for all automatable validation.

After everything is deemed correct, re-verify using automated processes to be sure that the report is consistent with all machine-readable rules. The following is a summary view of the verification:



**CCO 1.0 Universal (CCO 1.0)  
Public Domain Dedication**

CCO 1.0 Universal (CCO 1.0) Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0/>

The following are detailed validation results:

Overall process:

Line	Type	Message ID	Message	Location
0	Info	Report.Load	Loading was successful: no errors or warnings.	Filename: company-instance.xml
1	Info	Report.Validation.ModelStructure	Model Structure Validation: Validated successfully.	Filename: company-instance.xml
2	Info	Report.Validation.TypeOrClassRelations	TypeOrClassRelations: Validated successfully.	Filename: company-instance.xml
3	Info	Report.Validation.FundamentalAccountingConcepts	Fundamental Accounting Concepts Validation: Validated successfully.	Filename: company-instance.xml
4	Info	Report.Validation.DisclosureMechanics	Disclosure Semantic Validation: Validated successfully.	Filename: company-instance.xml
5	Info	Report.Validation.ReportingChecklist	Reporting Checklist Validation: Validated successfully.	Filename: company-instance.xml

Model structure detail:

Child	Parent						
	Network	Table	Axis	Member	LineItems	Abstract	Concept
[Network]	0	0	0	0	0	0	0
[Table]	14	0	0	0	0	0	0
[Axis]	0	43	0	0	0	0	0
[Member]	0	0	43	12	0	0	0
[LineItems]	0	14	0	0	0	0	0
[Abstract]	0	0	0	0	24	14	0
[Concept]	0	0	0	0	0	155	0

Type/class detail:

Line	Type	Message ID	Message	Location
2	Info	Report.Validation.TypeOrClassRelations	TypeOrClassRelations: Validated successfully.	Filename: company-instance.xml

Fundamental accounting concept relations detail:

Entity	Period	ID	Test	Result	Amount Of Inconsistency	Evaluation
1234567890	2020-FY	FAC_CONSISTENCY_1	fac:Equity = ( fac:EquityAttributableToParent + fac:EquityAttributableToNoncontrollingInterest )	True	0	fac:Equity[poses:Equity( 6,000 )] = ( fac:EquityAttributableToParent[poses:ControllingInterest( 4,000 )] + fac:EquityAttributableToNoncontrollingInterest[poses:NoncontrollingInterest( 2,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_16	fac:IncomeLossFromContinuingOperationsBeforeTax = ( fac:IncomeLossFromContinuingOperationsAfterTax + fac:IncomeTaxExpenseBenefit )	True	0	fac:IncomeLossFromContinuingOperationsBeforeTax[poses:SurplusDeficitFromContinuingOperations( 500 )] = ( fac:IncomeLossFromContinuingOperationsAfterTax[poses:SurplusDeficitFromContinuingOperations( 500 )] + fac:IncomeTaxExpenseBenefit[poses:ShareSurplusDeficitEquityMethodAssociatesJointVentures( 1,000 )] - fac:IncomeTaxExpenseBenefit[poses:ShareSurplusDeficitEquityMethodAssociatesJointVentures( 500 )] )
1234567890	2020-FY	FAC_CONSISTENCY_17	fac:NetIncomeLoss = ( fac:IncomeLossFromContinuingOperationsAfterTax + fac:IncomeLossFromDiscontinuedOperationsBeforeTax )	True	0	fac:NetIncomeLoss[poses:SurplusDeficit( 500 )] = ( fac:IncomeLossFromContinuingOperationsAfterTax[poses:SurplusDeficitFromContinuingOperations( 500 )] + fac:IncomeLossFromDiscontinuedOperationsBeforeTax[poses:GainLossFromDiscontinuedOperations( 0 )] )
1234567890	2020-FY	FAC_CONSISTENCY_18	fac:NetIncomeLoss = ( fac:NetIncomeLossAttributableToParent + fac:NetIncomeLossAttributableToNoncontrollingInterest )	True	0	fac:NetIncomeLoss[poses:SurplusDeficit( 500 )] = ( fac:NetIncomeLossAttributableToParent[poses:SurplusDeficitAttributableToOwnersControllingInterest( 400 )] + fac:NetIncomeLossAttributableToNoncontrollingInterest[poses:SurplusDeficitAttributableToNoncontrollingInterest( 100 )] )
1234567890	2020-FY	FAC_CONSISTENCY_25	fac:Assets = ( fac:CurrentAssets + fac:NoncurrentAssets )	True	0	fac:Assets[poses:Assets( 12,000 )] = fac:LiabilitiesAndEquity[poses:LiabilitiesAndAssetsEquity( 12,000 )]
1234567890	2020-FY	FAC_CONSISTENCY_28	fac:NetCashFlowFromOperatingActivities = ( fac:NetCashFlowFromInvestingActivities + fac:NetCashFlowFromFinancingActivities )	True	0	fac:NetCashFlowFromOperatingActivities[poses:NetCashFlow( 1,000 )] = ( fac:NetCashFlowFromInvestingActivities[poses:NetCashFlowFromUsedInInvestingActivities( 1,000 )] + fac:NetCashFlowFromFinancingActivities[poses:NetCashFlowFromUsedInFinancingActivities( 2,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_3	fac:Assets = ( fac:CurrentAssets + fac:NoncurrentAssets )	True	0	fac:Assets[poses:Assets( 12,000 )] = ( fac:CurrentAssets[poses:CurrentAssets( 5,000 )] + fac:NoncurrentAssets[poses:NoncurrentAssets( 7,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_38	fac:IncomeLossFromContinuingOperationsBeforeTax = ( fac:OperatingAndNonoperatingRevenues - fac:OperatingAndNonoperatingCostsAndExpenses )	True	0	fac:IncomeLossFromContinuingOperationsBeforeTax[poses:SurplusDeficitFromContinuingOperationsBeforeTax( 500 )] = ( fac:OperatingAndNonoperatingRevenues[poses:Revenue( 7,000 )] - fac:OperatingAndNonoperatingCostsAndExpenses[poses:Expenses( 6,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_4	fac:Liabilities = ( fac:CurrentLiabilities + fac:NoncurrentLiabilities )	True	0	fac:Liabilities[poses:Liabilities( 6,000 )] = ( fac:CurrentLiabilities[poses:CurrentLiabilities( 5,000 )] + fac:NoncurrentLiabilities[poses:NoncurrentLiabilities( 1,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_5	fac:LiabilitiesAndEquity = ( fac:Liabilities + fac:Equity )	True	0	fac:LiabilitiesAndEquity[poses:LiabilitiesAndAssetsEquity( 12,000 )] = ( fac:Liabilities[poses:Liabilities( 6,000 )] + fac:Equity[poses:Equity( 6,000 )] )
1234567890	2020-FY	FAC_CONSISTENCY_8	fac:NetCashFlowDiscontinued = ( fac:NetCashFlowFromOperatingActivitiesDiscontinued + fac:NetCashFlowFromInvestingActivitiesDiscontinued + fac:NetCashFlowFromFinancingActivitiesDiscontinued )	True	0	fac:NetCashFlowDiscontinued[ 0 ] = ( fac:NetCashFlowFromOperatingActivitiesDiscontinued[ 0 ] + fac:NetCashFlowFromInvestingActivitiesDiscontinued[ 0 ] + fac:NetCashFlowFromFinancingActivitiesDiscontinued[ 0 ] )

Disclosure mechanics detail:

#	Disclosure	Category	Level	Pattern	Disclosure Found	Disclosure Consist...	Applicable	Representation Concept [TEXT BLOO]	Representation Concept DETAIL
1	Assets Roll-Up	Unknown	Level4Detail	RollUp	True	N/A	False	NOT EXPECTED	Assets
2	Cash Flow Statement, Direct Method	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT EXPECTED	Net Cash Flow
3	Document Information	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT EXPECTED	Document Title
4	Entity Information	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT EXPECTED	Economic Entity Name
5	Liabilities and Net Assets/Equity Roll-Up	Unknown	Level4Detail	RollUp	True	N/A	False	NOT EXPECTED	Liabilities and Net Assets/Equity
6	Prior Period Adjustment	Unknown	Level4Detail	Adjustment	True	CONSISTENT	True	NOT EXPECTED	Accumulated Surplus (Deficit)
7	Property, Plant, and Equipment, Net Reconciliations	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT EXPECTED	Property, Plant and Equipment, Net
8	Property, Plant, and Equipment, Net Subclassifications	Unknown	Level4Detail	RollUp	False	N/A	False	NOT EXPECTED	NOT FOUND
9	Property, Plant, and Equipment, Net Subclassifications (Alternative)	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT EXPECTED	Property, Plant and Equipment, Net
10	Revenue Detail by Customer	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT EXPECTED	Revenue
11	Share Capital by Class	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT EXPECTED	Share Capital
12	Statement of Changes in Net Assets/Equity	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT EXPECTED	Equity
13	Statement of Financial Performance, by Function	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT EXPECTED	Surplus (Deficit)
14	Statement of Financial Performance, Comparison of Budget and Actual Amounts	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT EXPECTED	Surplus (Deficit) from Continuing Operations before Equity Method Associates and Joint Ventures
15	Statement of Financial Position, Classified	Statement	UNKNOWN	Component	True	CONSISTENT	True	NOT EXPECTED	Surplus (Deficit)
16	Surplus (Deficit) Breakdown	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT EXPECTED	Surplus (Deficit)

**CC0 1.0 Universal (CC0 1.0)  
Public Domain Dedication**

CC0 1.0 Universal (CC0 1.0) Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0/>

### Reporting checklist detail:

#	Disclosure	Checklist Category	Reason Disclosure Must Exist	Discovered	Expectation Met	Link to Disclosure Mechanics
0	Reporting Checklist					
1	Document Information	Required disclosure	Disclosure always required	True	CONSISTENT	Document Information
2	Entity Information	Required disclosure	Disclosure always required	True	CONSISTENT	Entity Information
3	Statement of Financial Position, Classified	Required disclosure	Disclosure always required, satisfied by Assets Roll Up and Liabilities and Net A...	True	CONSISTENT	Statement of Financial Position, Classified
4	Assets Roll Up	Part of disclosure	Satisfies Statement of Financial Position, Classified disclosure	True	N/A	Assets Roll Up
5	Liabilities and Net Assets/Equity Roll Up	Part of disclosure	Satisfies Statement of Financial Position, Classified disclosure	True	N/A	Liabilities and Net Assets/Equity Roll Up
6	Statement of Financial Performance, by Function	Required disclosure	Disclosure always required	True	CONSISTENT	Statement of Financial Performance, by Function
7	Cash Flow Statement, Direct Method	Required disclosure	Disclosure always required	True	CONSISTENT	Cash Flow Statement, Direct Method
8	Statement of Changes in Net Assets/Equity	Required disclosure	Disclosure always required	True	CONSISTENT	Statement of Changes in Net Assets/Equity
9	Surplus (Deficit) Breakdown	Line item exists, then disclosure requi...	Required because line item ppsas:SurplusDeficitAttributableToNoncontrollingInt...	True	CONSISTENT	Surplus (Deficit) Breakdown
10	Share Capital by Class	Line item exists, then disclosure requi...	Required because line item ppsas:ShareCapital was reported	True	CONSISTENT	Share Capital by Class
11	Property, Plant, and Equipment, Net Subclassifications	Line item exists, then disclosure requi...	Disclosure is required, satisfied by Property, Plant, and Equipment, Net Subclas...	False	N/A	Property, Plant, and Equipment, Net Subclassifications
12	Property, Plant, and Equipment, Net Subclassifications (Al...	Alternative representation	Alternative to required disclosure Property, Plant, and Equipment, Net Subclass...	True	CONSISTENT	Property, Plant, and Equipment, Net Subclassifications (Al...
13	Property, Plant, and Equipment, Net Reconciliations	Line item exists, then disclosure requi...	Required because line item ppsas:PropertyPlantAndEquipmentNet was reported	True	CONSISTENT	Property, Plant, and Equipment, Net Reconciliations
14	Prior Period Adjustment	Possible disclosure	Disclosure is present	True	CONSISTENT	Prior Period Adjustment
15	Revenue Detail by Customer	Possible disclosure	Disclosure is present	True	CONSISTENT	Revenue Detail by Customer
16	Statement of Financial Performance, Comparison of Budge...	Possible disclosure	Disclosure is present	True	CONSISTENT	Statement of Financial Performance, Comparison of Budge...

### Manual review (i.e. To Do list):

Message List				
L... ▲ ▼	Type	Message ID	Message	Location

[Edit Filter](#)