

Request for Proposal

XBRL US Consistency Checks

September 12, 2008

Prepared by Campbell Pryde

Chief Standards Officer

Table of Contents

Proposal Purpose	5
Business Background	5
Business Objectives.....	9
Business Overview	9
Market Participants	9
Filers of Instance Documents.....	9
Regulators	9
Auditors.....	9
End Users	10
Software Vendors	10
Risks Associated with the Implementation.....	11
Inaccurate Consistency Checks	11
Rollout of Consistency Checks	11
Type of Validations	12
Sign Checking	12
Consistency of Content across Disclosures.....	12
Mutual Exclusivity	12
Consistency of Content	12
Extension Consistency.....	12
Dimensional Checking.....	12
Dimensional Addition.....	13
Industrial Applicability	13
SEC Validation Criteria	13
System Components	14
Formula Editor	14
Formula Processing Engine & API	14
Formula Distribution System and Infrastructure	14
Business Content Components	15
Formula Ruleset	15

Validation Taxonomy	15
XBRL Instance and Associated DTS	15
Output of the API	15
System Requirements	17
Formula Creation Environment (Formula Editor)	17
User Friendly Grammar Format	18
Supports Cross and inter Dimensional Calculations	18
Supports Cross Table (Hypercube) Calculations	19
Recognize Attributes in the DTS and Instance	19
Changes to Hypercubes do not Cause Formula to Fail	20
Formulas are Easily Updated for Changes in Namespaces	20
Supports User Defined Functions	20
Allows Linkbase Navigation to Return Required Sets	20
Supports Auto Completion	21
Supports a Mechanism to Control and Manage the Order of Formula Execution	21
Supports Operators to Restrict Formula Execution	21
Validation of Formula Syntax	21
Formula Processing Engine & API	21
Incorporation of Functions in the Processing Model	23
Rule Distribution System and Infrastructure	23
Authentication	24
Scenario Management and Testing Infrastructure	25
Proposal Administration	27
Proposal Due Date	27
RFP Distribution	27
Proposal Eligibility	27
Proposal Acceptance	27
Presentations or Demonstrations	27
Pre-Proposal Conference	27
Project Management & Reporting	27
Deliverables/Measurable Standards Schedule	28
Training	28

Documentation	28
Support and Maintenance	28
Method of Payment & Pricing Options.....	29
Component Pricing Table.....	29
Phased Delivery.....	29
Architecture Options.....	30
Commonly Asked Questions	31
Appendix A - SEC Validation Criteria.....	33
Appendix B - Phased System Requirements	37

Proposal Purpose

The ultimate goal of this product is to provide a common core to enable the editing, distribution and processing of validation criteria or consistency checks initially for use by US filers, but also longer term for other regulatory and reporting entities using taxonomies outside the scope of US GAAP. This Request for Proposal defines the requirements for the creation and implementation of the following:

- A rules creation environment
- A rules distribution system and infrastructure
- A rules processing engine and API
- A rules management and testing infrastructure

The system will be used to evaluate a set of consistency checks against XBRL data submitted to the SEC, by an SEC registrant, in XBRL format (including instance documents and XBRL extension taxonomies). For the purposes of this proposal, the terms *rules* and *formulas* can be used interchangeably. Formulas also include specific functions that are written to navigate logical XBRL structures.

Business Background

XBRL US has identified a gap in the XBRL document validation stack at a level between purely technical (XML specification, XBRL taxonomy and instance) criteria provided by software products, and business rules and criteria provided by public accountants and auditors, and intermediary companies such as financial printers.

This RFP requests bids to build an interoperable component that will serve as a common core to enable the editing, distribution and processing of validation criteria or consistency checks.

These consistency checks will be maintained along with the taxonomy and offered on a subscription basis by XBRL US using a web services delivery model (See Fig 4).

This RFP describes the mechanism for developing and deploying a “common core” of checks to achieve consistency of instance documents and XBRL extensions, regardless of the company originating the document, the software products used, or the professional service providers enlisted.

The instance document consistency problem exists because issuers may create an XBRL instance that not only passes traditional XBRL validation tests, but is acceptable in an accounting and regulatory context as well - and yet still not be consistent from company to company.

Therefore we will develop a set of XBRL US consistency checks that can be deployed to identify data inconsistencies in the interactive disclosure initiative. . Such a mechanism would check the values in the instant against a set of criteria defined in a validation criteria-set managed centrally by XBRL US. In Fig 1 this layer is called “XBRL US Consistency Checks”. This layer ensures that filers have prepared their instance documents consistently within and across industries. The following table distinguishes each layer in the stack.

Fig 1: The XBRL Validation Stack

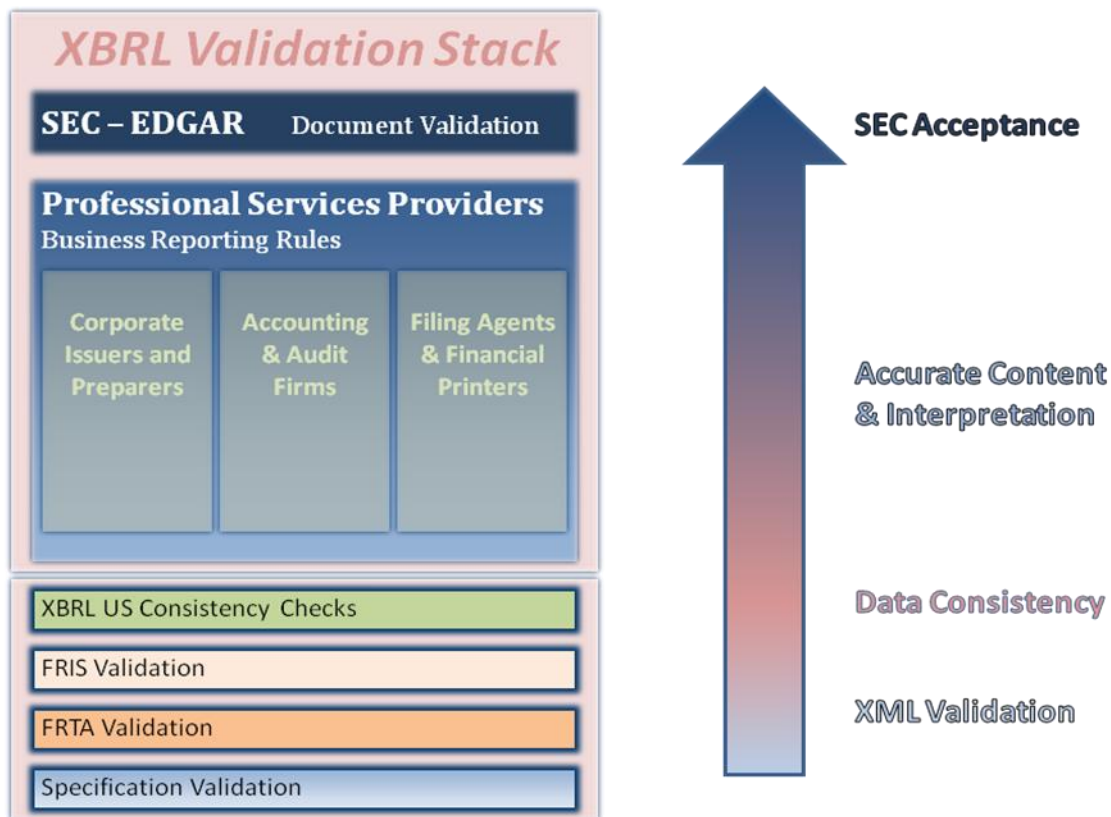


Table 1: Layers of the Validation Stack

Diagram Description	Business Description	Explanation
SEC Acceptance	Regulator Validation	Regulatory rules to validate for processing and analytical purposes. Similar to the FDIC processes, these may be proprietary and/or public domain in nature. The regulator will probably deploy other rules in the stack.
Accurate Content and Interpretation	Content Validation	Rules to validate compliance with specific content requirements and/or analytical rules associated with preparer, consumer or third party requirements. These rules are largely proprietary in nature and can include rules for content disclosure requirements and content analysis.
Data Consistency	XBRL US Consistency Checks	Criteria to validate compliance with the US GAAP Taxonomies. These criteria are public domain in nature and target compliance with the XBRL US GAAP Taxonomies to ensure

		consistency in the application of the taxonomy to facilitate comparability assessments. These criteria could be considered an automated extension of the Preparers Guide materials.
XML/XBRL Validation	XBRL Specification Validation FRTA Validation FRIS Validation Dimension Validation	Criteria to validate compliance with XBRL International Specifications and Standards. These criteria are public domain and target XML compliance. Similar to other Internet language consistency checks and generic to all taxonomies.

The current US GAAP taxonomy allows users to represent the same data in multiple ways in an instance document. The consistency checks indicate to the filer how a value should be tagged and how it should not. These criteria ensure that the instance is built in a manner that allows downstream analytics and analysis to be performed without coding for every variation in reporting the same facts. It helps filers from misinterpreting the taxonomy and allows feedback in real time eliminating the need to fix errors after consulting written guidance. The ability to provide consistency checking also means that XBRL US can provide real time guidance to filers as common inconsistencies in tagging arise.

For example, a taxonomy does not indicate if values should be tagged as negative or positive values. In filings to date treasury stock has been tagged both as a negative value and a positive value in different filings. However, they are identical. The way the taxonomy is designed, a negative value for treasury stock would mean that a company has issued more stock out of treasury than it actually has. An impossible situation as the company would exceed the outstanding shares or possibly authorized shares. Secondly, the use of dimensions in the taxonomy allows the tagging of every intersection in a table. In most cases only a proportion of these intersection points are valid. Those invalid points are not indicated in the taxonomy. In Fig 2 below, the valid intersection points for the statement of shareholders equity are shown in white, the invalid intersection points are black, and the yellow intersection points are dependent on other criteria. (Please note that this table is used to illustrate the valid intersection points by color distribution and is not intended to be read)

Fig 2: Tagging Statement of Shareholders Equity

For example, in the statement of shareholders equity, the majority of intersection points should not be tagged. There are over 150 such tables in the US GAAP Taxonomies v1.0 which define intersection points that currently do not have specific guidance added in the definition of an element’s label. This mechanism is required to communicate invalid tagging options to the user.

Thirdly, there are a number of cases where items are mutually exclusive; if you tag one item you would not tag another. For example, if the filer tags APIC in the statement of shareholders equity they would not also tag the “Common Stock and APIC” group.

XBRL US would create and maintain a set of consistency checks that would be released with taxonomy versions and updated on a continuous basis as consistency problems in instances and company extensions are identified and made available to all market participants.

Business Objectives

Business Overview

The proposed system will support the collection of a yearly license fees from various market participants and simultaneously create feasible revenue models or cost savings for XBRL US members.

Filers will pay for direct access to consistency checks that can be processed locally on a continuous basis in an automated way. For the same reason that a filer will use a validation tool to check that a filing complies with the XBRL specification, we also want to enable the filer to be able to check the consistency of their instance with the taxonomy in house before they submit their filing to a regulator.

Market Participants

The system must be flexible enough to be deployed by the following users:

- Registrants filing instance documents (Primary)
- Regulators such as the SEC (Secondary)
- Auditors (Secondary), and
- Users of the financial information (Secondary).

Filers of Instance Documents

The filers of XBRL instance documents include individual companies and financial printers such as Bowne, Merrill Corp and RR Donnelly. The consistency checks will be made available to all of these participants in an electronic format in exchange for a yearly license fee that is due for any legal entity running the consistency checks during the course of a year. The consistency checks will be available on a central server controlled by XBRL US that is updated on a continuing basis.

The proposed system will check that the latest set of checks are available and download the latest set of checks from XBRL US if necessary. The consumer of the consistency checks will pay a fee based on the checks they run on their own behalf or on behalf of a client. The intermediary is free to charge a higher amount to their client. Whatever the member filing agent charges its customers, XBRL US will always receive a fixed amount for every filer that uses the checks to validate their instance. Downstream users offering services to customers should be able to use the local version of the system to expand the checks to differentiate themselves in the market place.

Regulators

The system must be designed to enable regulators to use the system for validation of incoming filings and as a test service so companies can test their filings prior to their final submission.

Auditors

The system must be designed to enable Auditors performing an audit of an instance to run consistency checks of the instance. In addition they must be able to add their own more extensive business and validation checks.

End Users

The system must be designed to enable investors and consumers of XBRL information to run consistency checks. They should be able check an instance using the standard checks or add their own checks and validation criteria.

Software Vendors

XBRL US wants to enable an open market that allows software vendors to incorporate the standard consistency checking software into their products but at the same time restrict the consistency checks from being separated from the processing engine. The proposed system must be capable of being integrated into tools offered by various software vendors irrespective of the operating system used. Software Vendors should be able to integrate the proposed software into their tools to allow updates of the consistency checks on a continuous basis. The checks published by XBRL US must be distributed in an encrypted format so that the checks cannot be freely distributed.

The proposed system must have the necessary infrastructure to support the distribution of the consistency checks to the software that processes these checks.

Risks Associated with the Implementation

Inaccurate Consistency Checks

One of the risks associated with consistency checks are false positives creating extra work for users interpreting the results of the checks. To eliminate the occurrence of this the proposed system must include:

- A robust testing methodology that tests the checks against quantifiable data sets before implementation.
- A formula language that has sufficient expressiveness to allow for the removal of false positives.
- The ability to build consistency checks integrated into the taxonomy so that the impact of taxonomy changes on consistency checks can be adequately assessed.
- A mechanism to detect duplicate checks that create numerous alerts for the same error.
- A mechanism to categorize errors by type.

Rollout of Consistency Checks

It is anticipated that checks will be released on a continuous basis. As a rule is released, it will be automatically available to all parties running the consistency checks. The system must therefore allow the update and release of consistency checks on a continuous basis.

It is imagined that new consistency checks will be added as problems are identified in instances. Once the rule is implemented, registrants can then follow the instructions to correct any exceptions if they are deemed relevant by the company.

Type of Validations

The validation rules have a large utility to registrants and auditors, and allow for automated checking that would otherwise be time consuming and tedious if performed manually based on information buried in a paper document such as a preparers guide.

Proposed Validations would include the following types:

Sign Checking

These screens will test that items that should be positive are always positive and items that are negative are always negative.

Consistency of Content across Disclosures

This would include screens to test that the balance sheet balances, the cash flow statement reconciles from period to period, the statement of shareholders equity aligns with the income statement, balance sheet and cash flows, and the notes align with the line items on the face of the financial statements.

Mutual Exclusivity

These screens test that mutually exclusive items are not both reported in the same instance. A company cannot have a pension benefit and a pension expense in the same period if they have only one pension plan.

Consistency of Content

This would include screens that test that mutually dependent disclosures appear together, and if a company has disclosed service costs it will test that other elements relevant to defined benefit plans have also been disclosed. It will also test consistency between items such as debt and interest expense and commonly reported disclosures that are required within an industry. E.g., if a company has recorded discontinued operations, the tests will ensure that the appropriate splits are recorded in the income statement cash flow and disclosures, etc.

Extension Consistency

These check the taxonomy extensions to ensure that new elements and links are appropriately defined to confer meaning to systems processing the information. For example, these rules could check that extended elements of monetary types that appear in the financial statements are included in the calculation linkbase.

Dimensional Checking

This would include screens to test the consistency of dimensional members. For example, it will check that dimension members are not shared across conceptually different axes (i.e., entities and products on the same axis), that members are consistent on an axis, and that inappropriate members are not created. It will also check that restatements, and scenarios are modeled consistently and correctly across different parts of the instance document. The system must allow the checking of invalid dimensional intersections.

Dimensional Addition

These check that dimensional addition has been calculated correctly across members.

Industrial Applicability

These checks test that items are not tagged for an industry where they are not applicable, such as oil reserves for an insurance company.

SEC Validation Criteria

Refer to Appendix A for a listing of these criteria.

System Components

Formula Editor

The validation rules would be created in a Formula Editor. The primary requirement of the Formula Editor is that it is user friendly and does not require any knowledge of XPath or the XBRL formula specification. The editor would allow users to create formulas as easily as they can create formulas in Excel (formulas may be more complicated than Excel as rule creators need to define dimensional qualifiers). The Formula Editor will need to support dimensional formula within an individual hypercube and the entire instance. The formula implementation will also need to allow formula calculation across hypercubes (unless the entire instance is thought of as a massive hypercube). The Formula Editor will need to support a syntax that allows the definition of arc navigation and resolution. For example, the formula language should be able to return groups of dimensionally defined data that can be executed upon as a group. The Formula Editor will NOT be required to output the formula in the XBRL formula language initially. The formula can be defined in a proprietary syntax or contain proprietary extensions. The authoring and management of validation formula will require a user friendly interface that allows the construction of formulas with no knowledge of XML, XPath or the formula specification.

Formula Processing Engine & API

In addition to a Formula Editor, software needs to be either purchased or built to process the defined formula against an instance and company extension. The software must be able to read the submitted instance and company extension and run it against the appropriate formula set. The processor will create various output files as an exception report that details the formula exception, the reason it is an exception, and a suggested resolution to the problem. The exception report will be available as an XBRL instance so that it can be run into other XBRL applications for processing.

Each formula will have a unique identifier and definition associated with it where this information is captured. We will create Boolean elements associated with each rule in a data validations extension taxonomy that will allow for the capturing of additional attributes such as error criticality, workflow handling and multiple labels for capturing resolution information, error condition information, and other information that may be required.

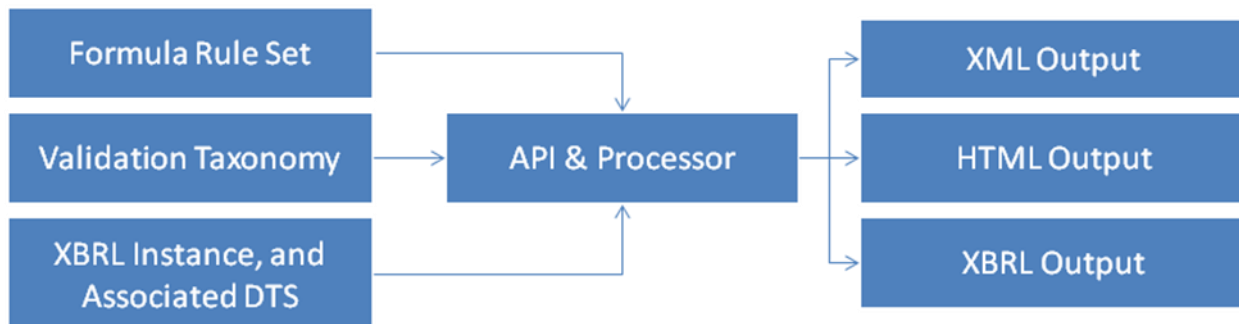
Formula Distribution System and Infrastructure

The Formula Processor must be accessible to software from numerous vendors in a standard way. This is essential so that vendors who want to build the validation rules into their tool can download the rules and have the processing engine process them. The processing engine must be standardized so that various software products do not produce different results. Since we cannot distribute the rules in a standard format the rules must be encrypted. The option proposed in this RFP is that a XBRL US member downloads software and installs it locally. They then feed the instance documents into the software and it produces exception reports based on the rules downloaded to the software.

Business Content Components

The business content components define the inputs and outputs to the Processor through the API.

Fig 3: API Input and Output



Formula Ruleset

The Formula Ruleset are the formulas defined in the Formula Editor.

Validation Taxonomy

The Validation Taxonomy defines the following components:

- The elements or concepts used to store the value generated from the execution of a rule. (These will usually be a Boolean item type.)
- Resources associated with each of the rules. This includes labels and references such as a standard name for the rule, a definition of the rule, and an explanation of how to resolve the issue identified and links to associated documentation if applicable.
- Attributes of the elements that store the results of executing the rule. This includes the elements type such as Boolean, if it is a period or duration, or other attributes that may be added such as severity, its type (Black, White, Grey) and work flow options.

XBRL Instance and Associated DTS

This component represents the input provided from the user. The API should be able to allow the transfer of physical files, a URI or an in memory data model to the processor.

Output of the API

The API will produce output in XML, HTML or XBRL, depending on the choice of the user. The HTML output should be generated based on the output defined in the XML file. In the XML file the following fields will be provided:

- Element
- Element Attributes
 - Period Type
 - Severity (1-5)
 - Status (Active, Inactive)
- Element labels

- Standard Label
 - Error Description
 - Action to Correct
- Element References
 - URI to explanation
- Data Values of the element for all periods, entities and other dimensional qualifiers.

System Requirements

Formula Creation Environment (Formula Editor)

All XBRL validation rules will be created and maintained in an integrated Formula Editor. The editor must meet the following requirements:

1. User friendly grammar format
2. Supports cross and inter-dimensional calculations
3. Supports cross table (hypercube) calculations
4. Changes to the definition of hypercubes do not cause the formula to fail
5. Support for tuples is not required
6. Formulas are easily updated for changes in namespaces
7. The formula expression interface supports functions defined by the user (a number of functions will need to be created by a software vendor)
8. The formula expression language should be able to support importing off the shelf function libraries.
9. The formula must be able to navigate linkbases and return required sets
10. The formula builder must have an interface that supports defining different calculations based on either/or:
 - the value of *any* element in an instance
 - the results of any other calculation
 - the context of an element in an instance
 - the dimensions of an element in an instance, and
 - the location of an element in a taxonomy network
11. The formula expression language must support implicit matching
12. A single formula can calculate multiple values for different contextual and dimensional qualifiers
13. The formula language cannot be ambiguous
14. Supports auto-completion of variables
15. The formula language should give notification of circular references

16. The formula must be able to recognize attributes contained in the DTS and the instance document
17. Standard date functions need to be available for the manipulation of the Period context
18. Standard mathematical and string functions need to be available for the manipulation of values contained in the instance
19. Formula must be individually identifiable and referenceable as resources
20. Formula must be able to navigate label links to determine if labels have been overridden or prohibited.
21. Formula must be able to be expressed as XBRL formula at some point in the future
22. All formula must be able to have a label to provide a description of their purpose
23. The Formula Editor must support a mechanism to allow the user to control and manage the order of formula execution
24. The formula language must contain operators that allow the exclusion of a subset of a calculation for a given Cartesian product to avoid the occurrence of multiple results for one intersection in a dimension. A NOT operator must allow the definition of elements:
 - Not in a defined network
 - Not in a given namespace
 - Not having a given attribute
25. The Formula Editor must validate that the formula syntax is correct.

User Friendly Grammar Format

The Formula Editor interface must allow efficient creation and maintenance of validation criteria. This means the user should not have to concern themselves with binding variables, tree navigation, XPath nomenclature and the like. Preferably the interface will be graphical in nature (i.e. GUI not command line) and will help the user to build taxonomies quickly and easily. In addition the interface will need to show dependencies on other Formulas both up and down a calculation hierarchy. The editor should allow the user to switch between seeing the components of a formula as elements or labels. The editor should use expressions that people are familiar with such as +, -, *, and /. In addition, the formula should look like a formula; i.e., “Enterprise Value = Number of Shares Outstanding * Share Price + Cash – Short term debt – debt”.

Supports Cross and inter Dimensional Calculations

The Formula Editor must allow the expression of formula between dimension members. For example, a user should be able to define “Foreign Operations” (member1) + “Domestic Operations” (member 2) =

“All Operations” (member 3). This formula would apply to all primary items and other dimensions defined in a specific hypercube in a specific extended link. If the user wanted this formula to apply to all hypercubes and all extended links the user needs a mechanism to define this, without linking the formula to every hypercube and extended link manually.

The ability to qualify formula components to valid dimensional Cartesian products is also required. I.e.,

```
Hypercube1:Extendedlink1 →
  Dimension1.DimensionMember3:Dimension2.DimensionMember1:PrimaryItem1
  =
  Dimension1.DimensionMember1:Dimension2.DimensionMember1:PrimaryItem1
  +
  Dimension1.DimensionMember2:Dimension2.DimensionMember1:PrimaryItem1.
```

In the hypercube PPE which has dimensions of PPE Type and Legal Entity we may want to express that the member “Land & Land Improvements” is equal to the members “Land” + “Land Improvements” for the legal entity “ABC”. For the entity Forest Corp. the value of “Land & Land Improvements” is equal to the members “Land” + “Land Improvements” + “Timberlands”.

Hypercube & Extended Link

PPE. Standard→

Formula 1

```
PPE Type.'Land & Land Improvements':LegalEntity.'ABC'
=
PPE Type.'Land':LegalEntity.'ABC'
+
PPE Type.' Land Improvements':LegalEntity.'ABC'
```

Formula 2

```
PPE Type.'Land & Land Improvements':LegalEntity.'Forest Corp.'
=
PPE Type.'Land':LegalEntity.'Forest Corp.'
+
PPE Type.' Land Improvements':LegalEntity.'Forest Corp.'
+
PPE Type.'TimberLands':LegalEntity.'Forest Corp.'
```

Supports Cross Table (Hypercube) Calculations

The formula expression language must allow the definition of formula that can take data from one hypercube and calculate a value in another hypercube. The editor must support the following combinations:

```
Primary item Value = Value in hypercube1 + Value in Hypercube 2
Primary item Value = Value in hypercube1 + Value in Hypercube 1
Value in hypercube1 = Value in hypercube1 + Value in Hypercube 2
Value in hypercube1 = Value in hypercube2 + Value in Hypercube 3
```

This requirement only applies where there is a single value in the Cartesian product and would not be valid for a Cartesian product that returns more than a single value. This expression of course would apply to all contexts that are not dimensions. A value would be generated for Primary items in all periods for example.

Recognize Attributes in the DTS and Instance

The formula language should allow for filtering of a formula set based on any attribute contained with the DTS or Instance. For example, if we want to indicate that a formula is only applicable for items of a duration type this should be expressible in the formula language.

```
PPE.Standard →
  Attribute(PeriodType=Duration):Period:FullYear
  =
  Period:Q1
  +
  Period:Q2
  +
  Period:Q3
  +
  Period:Q4
```

This formula will create a new context that represents the sum of four quarters only for those elements that are of type duration in the PPE hypercube within the standard extended link. This formula would only calculate for the primary items disposal, acquisition and adjustments if these were duration concepts linked to the PPE hypercube.

In an instance we may only want to perform a calculation on values which are denominated in USD. This is an attribute in the instance and should be able to be qualified in the formula.

Changes to Hypercubes do not Cause Formula to Fail

Changes to hypercube definitions occur when a company extends a hypercube by adding domain members or adding new axis. The formula expression interface should use implicit matching so that all dimensions in a hypercube do not have to be defined in the formula. If dimensions are added to a taxonomy or a taxonomy extension then the formulas will execute on the extended dimensional axis.

Formulas are Easily Updated for Changes in Namespaces

It is anticipated that the US GAAP Taxonomies will be released once a year with a new name space. For this reason the ability to change the namespace should be centrally configurable so that every formula does not have to be changed when a new taxonomy is released.

Supports User Defined Functions

The user should be able to define their own functions using standard functions available in the formula expression interface. For example, a standard function may be ReturnElementsInNetwork(). The parameters in the function may be Network type, Extended Link, Element, Descendents or Parents, or levels of tree. For example, ReturnElementsInNetwork(DefinitionLinkbase, ConceptExtendedLink, Assets, Descendents, all) will return all children of assets defined in the definition linkbase. The user could then take this expression and create their own function called Assets() which references this defined function. This allows the construction of higher level functions by users and allows a user to change one abstracted function rather than making changes multiple times across formula.

Allows Linkbase Navigation to Return Required Sets

The user should be able to define formulas that allow the navigation of standard and generic linkbases to return a set of values. For example, the user should be able to select all children of a concept with an Industry Group arc such as Industries and use this result set to define that the formula will operate only for a given set of industries rather than having to define a separate formula for every industry.

Supports Auto Completion

The user should be able to type the first few letters of an element and have it auto-completed by the Formula Editor. This would include taxonomy components (element names, group names and arc names) and language components (Function names and operators)

Supports a Mechanism to Control and Manage the Order of Formula Execution

The results derived for formula will differ depending on the order of execution if the Cartesian product results in formula overlap. The editor must be able to define which Cartesian product overlaps take precedent over others. For example, the results of one formula may create 20 values, and another formula creates 5. However, two of these results exist in the same dimensional intersect of a given hypercube. The results from the formula with the highest priority will be placed in the dimensional intersect. However, we would encourage users not to create intersecting formula results. (See next requirement.)

Supports Operators to Restrict Formula Execution

The formula language needs operators to restrict formula execution such as “Union”, “Intersection” and “Not”. For example, if we had a formula for the banking industry but did not want the formula to run for all other industries then the “Not” operator could be used. For example, interest expense is part of expenses for a nonbank and offset against Interest Income for a Bank. To calculate profit before expenses we would have two formulas.

1. “SelectBanks()”. Profit before Expenses = Income – Interest Expense
2. Not “SelectBanks()”. Profit before Expenses = Income – COGS

The first formula applies to only banks and the second selects all industries except Banks.

Validation of Formula Syntax

Specifically it should check at least the following:

- Operators are used in the correct location
- Element names actually exist in the taxonomy
- The syntax of the formula is correct
- Functions are defined correctly and have appropriate variables
- User defined parameters are correct and have appropriate variables.

Formula Processing Engine & API

This component processes a defined formula against an instance and company extension. The requirements for this component are as follows:

1. There must be an API that allows access to the processing engine.
2. The processing engine must be able to read a submitted instance and company extension and process it against the latest formula set from XBRL US. The engine should be able to allow the transfer of physical files, a URI or an in memory data model to the processor.

3. The engine must be able to create an XBRL instance output file as an exception report that details the formula exception, the reason it is an exception and a suggested resolution to the problem.
4. The exception report must be available as an XBRL instance so that it can be run into other XBRL applications for processing.
5. The engine and API must be operating system independent.
6. The formula processing engine must be optimized to only run the formulas that are relevant for the instance and extension submitted.
7. The engine must have logic to handle special values (Nil) and duplicate elements in an instance.
8. The engine software must check every time it is run if a later version of the software is available. If a later version is available it must initiate a process to download the latest version.
9. The engine software must be able to communicate across corporate firewalls.
10. The engine must support an option to check for rules updates.
11. Each distributed copy of the engine must have a unique identifier that stays the same if it is copied and cannot be changed. This identifier must be available upon request.
12. Every downloaded version of the engine must have a unique identifier that it uses when communicating with the XBRL US server.
13. The engine will not run the validation checks unless a CIK number is provided. The system must notify the user to provide the CIK number in the instance.
14. The engine must validate that the CIK number matches with the unique identifier assigned to that company.
15. The engine must ensure that the validation checks are not made available to software interfacing through the API.
16. The engine should store rules provided from the central server in either an encrypted or encoded format that cannot be easily accessed by external software.
17. The engine must allow third party software or a user to manually **submit** an instance and/or an extension for validation.
18. The engine should include a simple GUI to allow users to run the consistency checks without third party software.
19. The API must allow third party software or a user to manually **request** that the engine does not check for new rules when a validation check is requested to be run.
20. The API must allow third party software or a user to manually **request** that the engine does not check for updated software to be downloaded from the XBRL US server.
21. The API must allow third party software or a user to manually **add** rules to the engine and store them. (These do not have to be encrypted.)
22. The API must allow third party software or a user to manually **request** that only XBRL US rules are run, all rules are run or only custom rules are run.
23. The API should allow a user to switch off pre-defined rules or groups of rules that are not applicable and add an explanation of why the rule was switched off.
24. The engine must be able to download newly defined functions loaded to the central server and incorporate that function into its processing model. (See explanation below.)

25. The engine should support the base set of consistency checks provided by XBRL US, but should also have a mechanism to permit the user to find and select rules from a list of other vendors and download those rules from either XBRL US servers or third party servers.
26. The engine should support transparency into how a calculation was derived. The transparency should allow drilldown through formula used to the facts that derived the result. This should be provided in a format that can be processed by other software.

Incorporation of Functions in the Processing Model

From time to time new functions will need to be added to enable the identification of problems in extensions and instances. For example, We may decide to check that only certain members from the US GAAP Taxonomy existed on an axis, but don't want to report new members that a filer had added as part of their extension process. To do this we would create a function which defines the dimension, the namespace and the allowable elements.

OnlyElementsInGivenNamespaceThatCanBeOnDimension(*Dimension, Namespace, Element1, Element2, ...*) These functions would be defined in the Formula Editor, but the code that makes them work would be written separately by a qualified developer. The code would be published and could be used in rules. When the engine reads new rules it will also look to new functions and load and incorporate these.

Rule Distribution System and Infrastructure

All XBRL validation Checks (formula and functions) will be stored and distributed from a central repository. The distribution system must meet the following requirements:

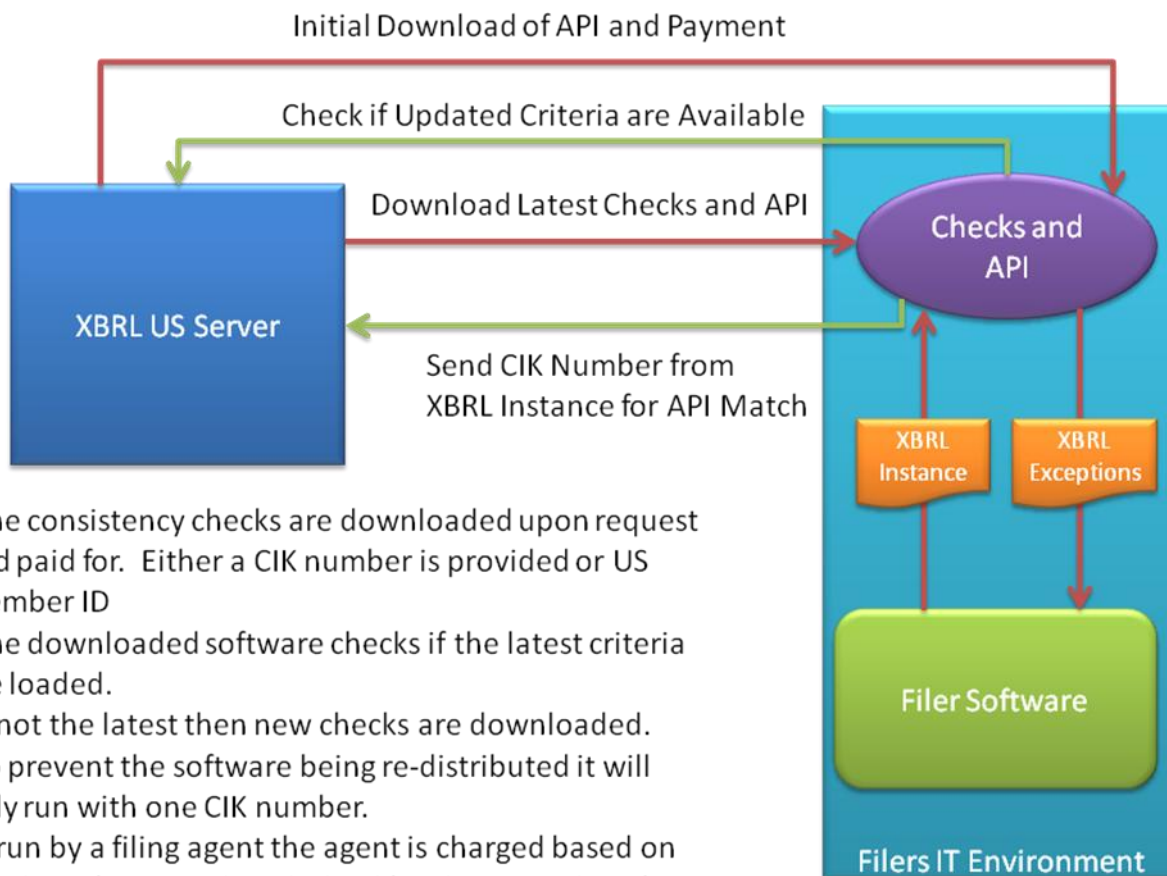
1. All checks must be encrypted at all times.
2. The server environment must support the distribution of checks as well as the software to process the checks.
3. The system must track which API software has been distributed and the CIK (or other identifier) for which that software can be used.
4. The system must also support the distribution of the Validation taxonomy.
5. The complete Validation Taxonomy must remain encrypted however components of it are used to report error messages from the API. I.e., Labels and explanations.
6. The system must have a mechanism to monitor the performance of the remote API for trouble shooting purposes.
7. The system must segregate checks that have been uploaded but not yet published. This mechanism allows the users to load checks into a staging area before publication. Once approved the check can be published. These will usually be performed in a batch.
8. The system must allow the distribution of multiple sets of rules that could be provided from various vendors.
9. The system must support payment collection upon request for an account.
10. The system must provide a portal that allows the downloading of the software, registration and payment.
11. The system must support the collection of the following information:
 - a. Contact Name
 - b. Contact organization

- c. CIK Number
- d. Payment Information
- e. Primary contact email address.

Authentication

The system requires integration with an external authentication provider, which will be provided by XBRL US and is not in the scope of this proposal.

Fig 4: Distribution Infrastructure



- The consistency checks are downloaded upon request and paid for. Either a CIK number is provided or US member ID
- The downloaded software checks if the latest criteria are loaded.
- If not the latest then new checks are downloaded.
- To prevent the software being re-distributed it will only run with one CIK number.
- If run by a filing agent the agent is charged based on number of CIK numbers lodged for their API identifier.

Scenario Management and Testing Infrastructure

In order to ensure that all consistency checks are working correctly a scenario management and testing infrastructure is required to ensure that the rules are functioning as intended. The requirements for these subsystems are as follows:

1. The user must be able to test rules against a test dataset.
2. The user must be able to debug the rule and view the results of the rule as it runs through the dataset.
3. The user must be able to configure the testing system to be able to test an individual rule, all rules or a subset of rules at one time.
4. The user must be able to restrict the testing dataset that will be tested, to one filing, a group of filings or all filings.
5. The user must be able to pass user defined element values through the debugging interface into an instance in the test dataset.

6. The end user must be able to refresh the dataset from latest filings and select specific filings to include in the dataset.
7. The user must be able to generate test instances for a new taxonomy based on the previous release of the taxonomy and the instances filed under it.
8. The environment must use the latest API at its core so that testing is always performed on the latest API that has been deployed. The infrastructure should also allow old APIs to be loaded so that testing can be done on previous versions of the API if necessary.

The scenario management and testing infrastructure is platform independent, however it must use the same processing engine that is deployed or will be deployed. This infrastructure will be used to deploy all new releases of the API.

Proposal Administration

Proposal Due Date

A technical and price proposal for this RFP must be received on or before October 17, 2008, 2:00 PM, Eastern Time and shall be emailed to Campbell Pryde at Campbell.Pryde@XBRL.US.

RFP Distribution

The RFP will be distributed from the XBRL web site and will be made available to all members of the Public. The proposal can be freely distributed and is not subject to any restrictions on distribution.

Proposal Eligibility

Any organization or individual is free to submit a proposal. Preference will be given to proposals received from members of XBRL US.

Proposal Acceptance

The proposal will be accepted one week from the conclusion of presentations. Vendors will be advised on October 31, 2008 as to the status of their proposal.

Presentations or Demonstrations

After submission of a proposal, XBRL US may ask selected vendors to present a presentation or demonstration of their proposal to XBRL US. The presentations will be conducted one week after the submission of the proposal.

Pre-Proposal Conference

XBRL US will conduct a pre-proposal conference call on September 26, 2008 at Midday Eastern Time to answer questions that prospective bidders have about the RFP. Statements made at the conference are informational only. Changes resulting from the conference are official only if issued through an amendment to the RFP. Questions for the conference must be submitted in advance. The cutoff date for questions is September 24th 2008. All questions are considered to be anonymous.

The call in details of the conference is as follows:

Dial in US: +1 (877) 339-0018

Dial in Int: +1 (205) 263-0801

Participant Code: *8493053*

Project Management & Reporting

The Contractor shall prepare bi-weekly status reports that describe the work performed during the period in question and the progress of the work against the Project Plan and Schedule. Each status report shall include any significant technical or substantive issues that reasonably might bear on the timely, successful completion of any deliverable or work required. Each status report shall include a

statement of the total costs incurred through the end of the period covered by the report, together with a statement of the prior estimated costs through that period and an explanation for any variance from the estimate. In addition the reports should discuss progress against all the deliverables listed herein.

In addition to the written status reports, the contractor shall participate in weekly status meetings with XBRL US to discuss project status and significant issues.

Deliverables/Measurable Standards Schedule

The contractor will aim to provide the following deliverables on the following dates:

Project Plan and Schedule	November 7, 2008
Formula and Function Editor	February 28, 2009
Formula Processor	February 28, 2009
API software	March 31, 2009
Distribution Software	March 31, 2009
Rules Management and Testing Infrastructure	April 30, 2009

If the contractor believes these dates cannot be met they should indicate what dates can be met in their proposal for each deliverable.

XBRL US will provide the following deliverables:

- Consistency checks using the Formula described in the proposal
- Functions defined based on base functions provided in the Formula Editor
- Validation taxonomy

Training

This proposal does not anticipate any training services provided by the vendor other than demos of the system.

Documentation

The vendor will provide user documentation for the following:

- API documentation (technical aimed at developers)
- User document (For users who download the software)
- Operation documentation (Run book for XBRL US service)

Support and Maintenance

The company developing the software will be responsible for maintenance of the software and should anticipate being available within an hour, 5 days a week for 16 hours a day. The proposal should include

the hourly rate for maintenance costs and the proposed response times for maintenance. Separate rates should be provided for cost of maintenance and the cost of further development activities over time such as writing new functions. Either a fixed price or hourly maintenance and support contract will be considered viable alternatives for this RFP.

Method of Payment & Pricing Options

We will consider various financing options proposed by the vendor for the completion of the project. Bidders are encouraged to propose cost and revenue sharing options. For example:

- The cost of the project will be financed by the software vendor and payment will be made upon receipt of revenue from filers using the system. 100 percent of all revenue received from filers will be paid to the vendor until such time that the costs of development, overhead, profit, compensation for business risk and financing costs are met. These costs must be included in the proposal.
- The cost of the project will be financed by the software vendor and XBRL US will pay on an on-going royalty to the software vendor based on licenses provided to filers which will reduce over time in alignment with depreciation of the software code. The royalty over the period should match the cost of development plus profit percentage plus financing risk and financing costs. These costs must be included in the proposal.
- The cost of the project will be financed by the software vendor and XBRL US in a shared arrangement with XBRL US granted a perpetual license to the software code.

Contractors should also include various cost scenarios of developing the software if financed in whole or in part by XBRL US.

If the bidder believes there are other arrangements that would help to subsidize the costs these should be included in the proposal.

Component Pricing Table

Bidders must bid on the entire proposal but must list the price of doing each component separately and in aggregate.¹ A table of various pricing options for each component and financing option should be provided.

Phased Delivery

The scope of this project is expansive; however the need to get validation rules in place is urgent. We will prioritize those proposals that support a cut down version of consistency checks that can be deployed quickly. It is NOT anticipated that these checks meet all the requirements, but will allow the validation of simple checks in a very short timeframe. These simple checks would include the following:

- Sign Checking

¹ The testing and scenario management may be done as a follow-up stage as this can be performed temporarily using the initial three infrastructure components.

- Consistency of Content across Disclosures
- Mutual Exclusivity

The deployment of these checks should however be consistent with the proposal. This early delivery only need support a rules processing engine. Please list out the requirements that would be delivered in a phased approach using the schedule attached in Appendix B.

Architecture Options

This RFP should be considered a starting point for a qualified bidder to use in formulating with XBRL US a systems design. The bidder should assume they will be working in collaboration with, and under the oversight of XBRL US members and XBRL US management.

Commonly Asked Questions

Do preparers upload their draft company report to a server for validation?

It is not proposed that filers submit their filings to a server. The checks will be distributed by XBRL US and the checks will be processed locally. No filings will travel outside of the organization's secure infrastructure. XBRL US will only distribute the rules and the proposed software to process the rules. The code must be a standard API that allows software vendors to submit a company's instance to the API and the API will return an XBRL exception instance in an XBRL and HTML format. Before the rules are transmitted the user will have to be authenticated. (See Fig 4)

Why is XBRL US developing software?

XBRL US does not intend to develop software. XBRL US will develop the checks. The API to process the rules, the editor to define the rules, and the distribution system will be developed by a third party software vendor.

Will other systems be able to process the consistency checks?

The proposed system will be the only system that will actually process the rules. The code must be provided as an API. Only one software vendor will provide the API. The code and API will be owned by XBRL US. Software vendors who provide instance creation tools must be able to submit an instance to this API to get a current listing of exceptions. Over the longer term we aim to make this code open source.

Does XBRL US need to own the proposed system?

XBRL US needs to own the proposed system for the following reasons:

1. We need to ensure consistency of rule processing.
2. Our revenue model is based on charging for the checks. If we only distribute the checks any processor will be able to process them without the need to pay for their use. We could encrypt the checks and require the purchase of a key to use them. However, this will require significant coordination with software vendors to make the system work in a short timeframe.

What are the data quality risks known to exist?

We have created instances based on the US GAAP Taxonomies and found that different taggers have tagged comparable information in different ways. The same items were tagged as negative values by some taggers and positive values by others. Specifically gain (loss) items, treasury stock, cash flow items. Dimension intersection points were tagged inconsistently. Specifically the statement of shareholders equity had numerous inconsistencies. An example is conversion of preferred stock to common. We had three different ways that this element was tagged. All these inconsistencies could be easily addressed with consistency checks.

How will users be able to use the common infrastructure for building their own rules?

The API will be able to process checks defined by XBRL US, but will also be able to process checks, rules and formula defined by end users. The checks defined by XBRL US will be locked so they cannot be easily shared.

Will users be able to turn rules off?

The API will have functionality that allows users to view the description of the rules and turn off rules that they deem to be inappropriate.

Appendix A - SEC Validation Criteria

1. All XBRL documents shall be XBRL 2.1 compliant.
2. Facts in instance documents shall be checked for consistency against the calculation link bases to detect inconsistencies between calculated totals and reported totals.
3. Filers shall use tags from the Document Entity Information (DEI) taxonomy to tag required company information as described below:
 - a. Filers shall use the DEI *EntityRegistrantName* element to enter the company's name.
 - b. Filers shall use the DEI *EntityCentralIndexKey* element to identify the company's CIK (central index key) code.
 - c. Filers shall use the DEI *DocumentType* element to identify the document (form) type that is represented by this document.
 - d. Filers shall use the DEI *DocumentReportType* element to identify whether the document is an annual report or a transition report.
 - e. Filers shall use the DEI *DocumentPeriodEndDate* element to tag the document's end date of report or period of filing.
 - f. Filers shall use the DEI *AmendmentFlag* element to indicate the whether the document is an amendment.
 - g. Filers shall use the DEI *AmendmentDescription* element to give a reason for the amendment.
 - h. Filers shall use the DEI *CurrentFiscalYearEndDate* element to identify the company's fiscal year end date.
 - i. Filers shall use the DEI *EntityWellKnownSeasonedIssuer* element to indicate whether the company meets the well-known seasoned issuer criteria noted in Rule 405 of Regulation C under the Securities Act of 1933 [17 CFR 230.405].
 - j. Filers shall use the DEI *EntityVoluntaryFilers* element to indicate whether the company is not required to file reports pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934.
 - k. Filers shall use the DEI *EntityCurrentReportingStatus* element to indicate whether the company has filed all reports required to be filed by Section 13 or 15(d) during the preceding 12 months and has been subject to such filing requirements for the past 90 days.
 - l. Filers shall use the DEI *EntityFilerCategory* element to identify whether the filer is a large accelerated filer, accelerated filer, non-accelerated filer, or a smaller reporting company.

- m. Filers shall use the DEI *EntityPublicFloat* element to disclose the company's public float amount.
 - n. Filers shall use the DEI *EntityCommonStockSharesOutstanding* element to disclose the number of shares outstanding for each of the registrant's classes of common stock.
4. XBRL data submitted to EDGAR will have document type and file extension as follows:

Filing Requirement	XBRL Document Type	EDGAR Document Type	File Extension
XBRL-Related Documents (as proposed in releases 33-8924 and 33-8929)	Instance Document	EX-100.INS	.xml
	Schema	EX-100.SCH	.xsd
	Presentation Linkbase	EX-100.PRE	.xml
	Calculation Linkbase	EX-100.CAL	.xml
	Label Linkbase	EX-100.LAB	.xml
	Definition Linkbase	EX-100.DEF	.xml
	Reference Linkbase	EX-100.REF	.xml
Filing Requirement	XBRL Document Type	EDGAR Document Type	File Extension
Interactive Data Files (as proposed in releases 33-8924 and 33-8929)	Instance Document	EX-101.INS	.xml
	Schema	EX-101.SCH	.xsd
	Presentation Linkbase	EX-101.PRE	.xml

	Calculation Linkbase	EX-101.CAL	.xml
	Label Linkbase	EX-101.LAB	.xml
	Definition Linkbase	EX-101.DEF	.xml
	Reference Linkbase	EX-101.REF	.xml

5. XBRL documents shall not contain any invalid or extended ASCII characters (e.g. foreign language characters), inconsistent schema, inconsistent reference link base, inconsistent label link base, inconsistent presentation link base, inconsistent definition link base, or inconsistent calculation link base.
6. EDGAR Type 1 Modules (partial documents) are not allowed in XBRL format. Only EDGAR Type 2 Modules (complete documents) can be submitted in XBRL format.
7. The number of modules per CIK shall not exceed ten (10).
8. Extended-ASCII characters or SGML tags shall not be used within XBRL document text.
9. The entire 10-digit Central Index Key (CIK) code assigned by the SEC as the unique company identifier shall be used as the company identifier in the following way:
`<identifier scheme="http://www.sec.gov/CIK">xxxxxxxxxx</identifier>`.
10. Instance document contexts shall use the *decimal* attribute instead of the *precision* attribute.
11. Ensure there are no documentation labels (definitions) in the extension taxonomy which were added to existing elements in the standard taxonomy.
12. Ensure there are no references in the extension taxonomy which were added to existing elements in the standard taxonomy.
13. Ensure that every new extension element should contain at a minimum a presentation link and a standard label.
14. Ensure there are no elements with the *Tuple* type in instance documents or in extension taxonomies.
15. Companies shall provide absolute paths for references to base taxonomies and relative paths for extension taxonomies.

16. Taxonomy extensions shall not reference the *statement-dis-all* entry point in the base taxonomies.
17. Ensure there are no duplicate elements (same value) in the instance document (same taxonomy concept, same context, same units, same value).
18. Ensure there are no duplicate elements (different values) in the instance document (same taxonomy concept, same context, same units, different value).
19. An element in an extension taxonomy shall not have the same name as an element in the base taxonomy.
20. Extensions created in the instance document shall contain, at a minimum, the following in the name space: name space:/{company name}/{version date}.
21. Element declarations for concepts shall contain an "id" attribute whose value begins with the recommended namespace prefix of the taxonomy, followed by an underscore, followed by the element name.
22. Taxonomy schemas shall not contain embedded link bases.
23. Dimensions shall only be used within segment and not scenario of contexts.
24. Typed members shall not be used in extension taxonomies.
25. Active Content (Actions, embedded JavaScript, etc.) in XBRL documents shall not be accepted.
26. External References (Destinations, Hyperlinks, etc.) shall not be included in instance documents or linkbases except for references to approved XBRL taxonomies (that can be found by going to the SEC's Information for EDGAR Filers webpage — <http://www.sec.gov/info/edgar.shtml>) and core XBRL files (that are listed in Volume II of the EDGAR filing manual - <http://www.sec.gov/info/edgar/edmanuals.htm>).
27. All extended-type links in a single link base shall have the same namespace and local name.

Appendix B - Phased System Requirements

#	Formula Creation Environment	Rollout			
		Current	Phase 1	Phase 2	Phase 3
1	The formula expression language supports a user friendly grammar format.				
2	The formula expression language supports cross and inter-dimensional calculations.				
3	The formula expression language supports cross table (hypercube) calculations.				
4	Changes to the definition of hypercubes do not cause the formula to fail.				
5	Support for tuples is not required.				
6	Formulas are easily updated for changes in namespaces.				
7	The formula expression interface supports functions defined by the user (a number of functions will need to be created by a software vendor).				
8	The formula expression language should be able to support importing off the shelf function libraries.				
9	The formula must be able to navigate linkbases and return required sets.				
10	The formula builder must have an interface that supports defining different calculations based on either/or: <ul style="list-style-type: none"> • the value of <i>any</i> element in an instance • the results of any other calculation • the context of an element in an instance • the dimensions of an element in an instance, and • the location of an element in a taxonomy network. 				
11	The formula expression language must support implicit matching.				
12	A single formula can calculate multiple values for different contextual and dimensional qualifiers.				
13	The formula language cannot be ambiguous.				
14	The formula expression language supports auto-completion of variables.				
15	The formula language should give notification of circular references.				
16	The formula must be able to recognize attributes contained in the DTS and the instance document.				
17	Standard date functions need to be available for the manipulation of the Period context.				

- 18 Standard mathematical and string functions need to be available for the manipulation of values contained in the instance.
- 19 Formula must be individually identifiable and referenceable as resources.
- 20 Formula must be able to navigate label links to determine if labels have been overridden or prohibited.
- 21 Formula must be able to be expressed as XBRL formula at some point in the future.
- 22 All formula must be able to have a label to provide a description of their purpose.
- 23 The Formula Editor must support a mechanism to allow the user to control and manage the order of formula execution.
- 24 The formula language must contain operators that allow the exclusion of a subset of a calculation for a given Cartesian product to avoid the occurrence of multiple results for one intersection in a dimension. A NOT operator must allow the definition of elements:
 - Not in a defined network
 - Not in a given namespace
 - Not having a given attribute
- 25 The Formula Editor must validate that the formula syntax is correct.

Formula Processing Engine & API

- 26 There must be an API that allows access to the processing engine.
- 27 The processing engine must be able to read a submitted instance and company extension and process it against the latest formula set from XBRL US. The engine should be able to allow the transfer of physical files, a URI or an in memory data model to the processor.
- 28 The engine must be able to create an XBRL instance output file as an exception report that details the formula exception, the reason it is an exception and a suggested resolution to the problem.
- 29 The exception report must be available as an XBRL instance so that it can be run into other XBRL applications for processing.
- 30 The engine and API must be operating system independent.

- 31 The formula processing engine must be optimized to only run the formulas that are relevant for the instance and extension submitted.
- 32 The engine must have logic to handle special values (Nil) and duplicate elements in an instance.
- 33 The engine software must check every time it is run if a later version of the software is available. If a later version is available it must initiate a process to download the latest version.
- 34 The engine software must be able to communicate across corporate firewalls.
- 35 The engine must support an option to check for rules updates.
- 36 Each distributed copy of the engine must have a unique identifier that stays the same if it is copied and cannot be changed. This identifier must be available upon request.
- 37 Every downloaded version of the engine must have a unique identifier that it uses when communicating with the XBRL US server.
- 38 The engine will not run the validation checks unless a CIK number is provided. The system must notify the user to provide the CIK number in the instance.
- 39 The engine must validate that the CIK number matches with the unique identifier assigned to that company.
- 40 The engine must ensure that the validation checks are not made available to software interfacing through the API.
- 41 The engine should store rules provided from the central server in either an encrypted or encoded format that cannot be easily accessed by external software.
- 42 The engine must allow third party software or a user to manually **submit** an instance and/or an extension for validation.
- 43 The engine should include a simple GUI to allow users to run the consistency checks without third party software.
- 44 The API must allow third party software or a user to manually **request** that the engine does not check for new rules when a validation check is requested to be run.
- 45 The API must allow third party software or a user to manually **request** that the engine does not check for updated software to be downloaded from the XBRL US server.

- 46 The API must allow third party software or a user to manually **add** rules to the engine and store them. (These do not have to be encrypted.)
- 47 The API must allow third party software or a user to manually **request** that only XBRL US rules are run, all rules are run or only custom rules are run.
- 48 The API should allow a user to switch off pre-defined rules or groups of rules that are not applicable and add an explanation of why the rule was switched off.
- 49 The engine must be able to download newly defined functions loaded to the central server and incorporate that function into its processing model. (See explanation below.)
- 50 The engine should support the base set of consistency checks provided by XBRL US, but should also have a mechanism to permit the user to find and select rules from a list of other vendors and download those rules from either XBRL US servers or third party servers.
- 51 The engine should support transparency into how a calculation was derived. The transparency should allow drilldown through formula used to the facts that derived the result. This should be provided in a format that can be processed by other software.

Rule Distribution System and Infrastructure

- 52 All checks must be encrypted at all times.
- 53 The server environment must support the distribution of checks as well as the software to process the checks.
- 54 The system must track which API software has been distributed and the CIK (or other identifier) for which that software can be used.
- 55 The system must also support the distribution of the Validation taxonomy.
- 56 The complete Validation Taxonomy must remain encrypted however components of it are used to report error messages from the API. I.e., Labels and explanations.
- 57 The system must have a mechanism to monitor the performance of the remote API for trouble shooting purposes.

- 58 The system must segregate checks that have been uploaded but not yet published. This mechanism allows the users to load checks into a staging area before publication. Once approved the check can be published. These will usually be performed in a batch.
- 59 The system must allow the distribution of multiple sets of rules that could be provided from various vendors.
- 60 The system must support payment collection upon request for an account.
- 61 The system must provide a portal that allows the downloading of the software, registration and payment.
- 62 The system must support the collection of the following information:
 - Contact Name
 - Contact organization
 - CIK Number
 - Payment Information
 - Primary contact email address.

Scenario Management and Testing Infrastructure

- 63 The user must be able to test rules against a test dataset.
- 64 The user must be able to debug the rule and view the results of the rule as it runs through the dataset.
- 65 The user must be able to configure the testing system to be able to test an individual rule, all rules or a subset of rules at one time.
- 66 The user must be able to restrict the testing dataset that will be tested, to one filing, a group of filings or all filings.
- 67 The user must be able to pass user defined element values through the debugging interface into an instance in the test dataset.
- 68 The end user must be able to refresh the dataset from latest filings and select specific filings to include in the dataset.
- 69 The user must be able to generate test instances for a new taxonomy based on the previous release of the taxonomy and the instances filed under it.

70 The environment must use the latest API at its core so that testing is always performed on the latest API that has been deployed. The infrastructure should also allow old APIs to be loaded so that testing can be done on previous versions of the API if necessary.

Type of Validations

- 71 The system must support Sign Checking
- 72 The system must be able to check consistency of Content across Disclosures
- 73 The system must support checking Mutual Exclusivity
- 74 The system must support checking Consistency of Content
- 75 The system must support checking Extension Consistency

- 76 The system must support Dimensional Checking
- 77 The system must support checking Dimensional Addition
- 78 The system must support checking Industrial Applicability

- 79 The system must support running the SEC Validation Criteria

Inaccurate Consistency Checks

- 80 The system must support robust testing methodology that tests the checks against quantifiable data sets before implementation.
- 81 The system must support a formula language that has sufficient expressiveness to allow for the removal of false positives.
- 82 The system must support the ability to build consistency checks integrated into the taxonomy so that the impact of taxonomy changes on consistency checks can be adequately assessed.
- 83 The system must support a mechanism to detect duplicate checks that create numerous alerts for the same error.
- 84 The system must support a mechanism to categorize errors by type.

Rollout of Consistency Checks

- 85 The system must allow the update and release of consistency checks on a continuous basis

Appendix C – Q & A on the Request for Proposal

1. Is it your intention that all requirements must be fulfilled by a single vendor or can the RFP be split between “best of breed” providers?
 - Yes the proposal can be split between vendors.
2. The stated timeline suggests that the Formula and Function Editor is the first software deliverable. Given the statement on page 29 about the urgency of getting validation rules in place, will XBRL US consider proposals that instead deliver a baseline set of fully-tested consistency checks that enable the service to go live and deliver the formal editing tool later in the schedule?
 - Yes the order or timeline is loose guidance that should be considered in the proposal but is not an absolute date that should be met. Proposals that can deliver the critical requirements closer to these dates will get priority. However if the system can be made to function earlier and the proposal details non critical components that can be delivered later that would be preferable. The criticality of the component of course will depend in large part on the vendors proposed solution and approach.
3. The RFP section entitled “Commonly Asked Questions” mentions the possibility of making the consistency checks software available as open source. Is it correct that this only applies to the “API Software” referred to in the RFP and not to the Formula and Function Editor, Formula Processor, Distribution Software, Rules Management and Testing Infrastructure, or any other software except the “API Software”?
 - The API software is intended to be freely available and freely licensed to allow all software vendors to use it to make calls to it. The formula editor and formula processor should be available for companies to write and process their own rules but will not be open source. The other components (Distribution Software, Rules Management and Testing Infrastructure) will not be made freely available or be distributed with an open source license.
4. According to the RFP, the “API Software” leverages a “Formula Processor” as part of the distributable software. Is it your expectation that the “Formula Processor” is a proprietary component that is licensed by XBRL US or Market Participants and not subject to the open source license that may apply to the API Software that uses the component?
 - The proposal must include in the price the cost to XBRL to license the formula processor in perpetuity to redistribute for use in processing rules published by XBRL US. The processor is not required to be open source and the proposal could include the option that companies writing their own rules or others publishing rules have to pay a separate license fee to process these rules. However the cost to the preparer or other users of the processor for the next 5 years should be included in the proposal.
5. If XBRL US proceeds with open source for the software, under which licensing model will this occur?
 - The software is not planned to be open source, however the API will be freely licensable.
6. The RFP section entitled “Commonly Asked Questions” mentions the possibility of making the consistency checks software available as open source. The plan to encrypt the validation rules seems at odds with the goal of providing the software as open source. It would seem that in providing access to the source code that programmers could code around the encryption. Can you please comment?
 - The processor code will not be open source. The rules should not be able to be directly accessed through the API.
7. With regard to the software delivery model, will XBRL US consider a server-based and/or thin-client application, provided that filing data interacts with the server in a secure manner?
 - Yes, so long as the following conditions are met:
 - a) The filing is secure through-out the submission process
 - b) The results of the rules are secure

- c) The cost of maintenance is minimal
 - d) Minimal IT infrastructure is required to support it.
 - e) It supports all the requirements defined
8. The RFP mentions four types of market participants. Can you please identify the relative priority for each and help us understand the timeline for addressing their needs?
 - The main priority is the filer. This is the only participant the proposal should focus on. The other participants were included in the proposal so that any proposed designs considers their longer term needs that at this point are undefined.
 9. Given the complex nature of XBRL and the US GAAP, IFRS, and ICI taxonomies, is there a set of technical qualification criteria that a vendor must meet in order to be awarded the contract? For example, should prospective vendors be able to demonstrate the ability to execute consistency checks like the SEC Public Validation Criteria?
 - References from previous work performed should be provided. However given the limited history with the standard it is expected that this will be generally limited.
 10. On page 22 in the section entitled, "Formula Processing Engine & API" the RFP states, "The engine and API must be operating system independent". The assumption made in the RFP seems to be that the API runs on the client machines of the Market Participant. It is not explicitly stated but it thus seems like "operating system independent" is referring to client operating systems. To best understand the targeted operating systems, can you please list each and their relative priority?
 - The assumption is correct; it refers to the client's operating system. The principle is that the software should be able to run in the client's environment. The priority if there was one would match the general distribution of operating systems used by filers.
 11. Is it realistic to expect that the software will be able to ensure that consistency checks are kept encrypted? Surely users need to know what the checks are so they can fix them. Why would someone use checks they cannot see or understand?
 - The point of encrypting the consistency checks is to discourage people copying them and freely distributing them. It is not expected that the solution will ensure the secrecy of the rules. In fact the explanation of all the rules and how to fix issues identified by the checks will be freely available to anyone. The syntax that defines the rules will not. Obviously someone would be able to hack the software and get the syntax of the current set of rules. The rules will be licensed by those using them with a yearly fee per company.
 12. There is some assumption—or maybe desire—that the vendor might assume some risk (reduced cost) for delayed gratification (royalties). However, there is absolutely no indication of the business model in terms of projections. That is, there is no way to evaluate the cost-benefit of taking any risk. It would be appreciated if you could comment on this.
 - Concentrate your proposal on providing the best platform for delivery of the service. Based on the merits of each proposal submitted the business model will be discussed with those vendors who have submitted an acceptable proposal. In your proposal you should define your own assumptions as to what you believe the market share and cost and risk sharing could be and the methodology you would use to price the proposal based on the given variables of risk, market cost and yearly license fee.
 13. *Whatever the member filing agent charges its customers, XBRL US will always receive a fixed amount for every filer that uses the checks to validate their instance.* Will the filing agents be required to pay a separate fee for every client of theirs for whom the consistency checks are utilized? The picture on page 25 seems to make the point that multiple CIK numbers will be associated with the filing agents identifier which seems to suggest that the filing agent's payments will be based on the number of clients the agent is running the checks for.
 - Yes that is correct, a license would be paid for each individual client of a filing agent.
 14. *Proposed Validations would include the following types:* While tests like Sign checking are easily understood by a non-accounting person (programmer), there are numerous test described in the RFP that are deeply accounts specific. Will XBRL-US be providing guidance and assistance to the software vendors in defining the specifics of such tests or will the software vendor be required to do this on their own?

- XBRL-US will provide guidance on this. XBRL US will write the formula's in a manner that complies with the syntax provided from the vendor. However we will work with the vendor to ensure that the syntax meets the requirements of the consistency checks that are required.
15. *The exception report will be available as an XBRL instance so that it can be run into other XBRL applications for processing. Who will be creating, publishing and owning the XBRL taxonomy for use in creating these exception report XBRL instances? Is this taxonomy the "Validation Taxonomy" described on page 15?*
- This will be provided by XBRL US. XBRL US will provide all business information related to the construction of the checks. It is anticipated that this will include the following:
 - i. Definition of the checks in a formulaic expression
 - ii. Creation of the documentation surrounding the check, i.e. explanation of the error, attributes associated with the error and an explanation of the resolution of the error.
 - iii. Creation of the Validation taxonomy.
 - Yes it is the same.
16. *The API will produce output in XML, HTML or XBRL, depending on the choice of the user. This is in contradiction to the previous requirement of output available in the form of XBRL instance. It would seem that the user choice option supersedes the XBRL only option - would you concur?*
- Yes the user choice supersedes the XBRL option only.
17. *The formula expression language should be able to support importing off the shelf function libraries. Which off the shelf function libraries are being referred here? Can you please provide an example?*
- This is at the discretion of the software vendor, different vendors may have different preferences, and for example the vendor may be easily able to incorporate statistical functions from an open source library such as R.
18. *Standard date functions need to be available for the manipulation of the Period context. Please provide example(s) of what is meant by "standard date functions".*
- This is the ability to manipulate dates in the formula, Examples would include the following:
 - i. Days difference between 2 dates.
 - ii. Return the Month of a date
 - iii. Compare two dates for equivalence within a given tolerance
 - iv. Date addition and subtraction
 - v. Determine if a data is a quarter, half year or an annual year
 - vi. Etc.

This is an example of functions, but the functions available should allow dates defined in the instance and as a domain member to be manipulated by a function. This includes dates defined in a typed dimension.
19. *Formula must be able to be expressed as XBRL formula at some point in the future. There is a potential risk here in the fact that XBRL formula may not support all the aspects of the formula defined as part of the implementation. Is that ok?*
- Yes.
20. *All formula must be able to have a label to provide a description of their purpose. This seems to reflect the need for the formula to be defined using XBRL constructs, is that correct?*
- Yes. A formula should be comprised of XBRL constructs. XBRL items should be able to be incorporated into the formula syntax. In addition the result of the formula would be available in an XBLR concept that can have labels attached to it.
21. *All information is included in submitted instance? ex) P.21 sample 1 and 2 need type of company (bank or non-bank). Will such information be provided from the Instance? Or the formula engine must get the company type information from outside of the instance such as the table which maps CIK code to company type code.*

- Yes. All information should be obtained from within the instance. In a number of cases the information will not be available and the creator of the rules will have to incorporate that into the execution of the rules. For example if the filer has not indicate the industry a check will need to be included to indicate that the industry has not been defined. It is NOT anticipated that the formula engine will have to go and look data up.
22. *The processing engine must be able to read a submitted instance and company extension and process it against the latest formula set from XBRL US. The engine should be able to allow the transfer of physical files, a URI or an in memory data model to the processor. Can you clarify the reference to "processing engine" and "processor"? An example or scenario description would be most helpful.*
- The processing engine and processor refer to the same component.
23. *The engine and API must be operating system independent. So, if the engine runs within a Java VM and the API is in Java, is that acceptable?*
- Yes
24. *The formula processing engine must be optimized to only run the formulas that are relevant for the instance and extension submitted. Will XBRL-US be providing guidance on which formulas are relevant for which case?*
- The point of this requirement becomes relevant if the processor takes large amounts of time to process a filing. For example if a check checks if Treasury stock is less than zero and the instance does not contain treasury stock then the check should not be run. Note that the vendor will not have to write the rules. Once the syntax is agreed XBRL US will provide rules in the syntax for testing purposes. XBRL US will also provide the rules in an excel like format to help the vendor to build or modify their syntax if required. In addition we can add attribute to the formula to indicate processing options, such as do not run this check if a certain element does not appear in the instance.
25. *The engine will not run the validation checks unless a CIK number is provided. The system must notify the user to provide the CIK number in the instance. How does this work in the context of filing agents? The instance document will contain the CIK number of the client, but the unique identifier will be assigned to the filing agent - so both of these will not match. The picture on page 25 seems to make the point that multiple CIK numbers will be associated with the filing agent's identifier and that is how this would be resolved, correct?*
- Yes that is correct the CIK number should be unique for each instance. Of course the filing agent could change the CIK to be the same for all filings so there may be some additional checks to ensure the CIK matches the name of the entity. It is not anticipated that the vendor should spend significant time on this as we do not expect filing agents to game the system.
26. *The engine should support the base set of consistency checks provided by XBRL US, but should also have a mechanism to permit the user to find and select rules from a list of other vendors and download those rules from either XBRL US servers or third party servers. This makes the assumption that*
- a. The other vendors have developed the rules using the same engine
 - b. There is a pre-agreed protocol already established and known to all vendors for making their rules available to others (with or without a fee).
- Can you please elaborate on this?
- A. Yes the rules will have to be in the same format.
 - B. Yes, XBRL US will publish that protocol and make it available to market participants who want to make their own rules available to the market.
27. *The engine should support transparency into how a calculation was derived. The transparency should allow drilldown through formula used to the facts that derived the result. This should be provided in a format that can be processed by other software. Can you please provide an example of what is expected in terms of processing by other software?*
- An example is expected to be an XML format showing the hierarchy of how the calculation was derived. The format should allow different software sufficient information to provide a graphical drill down into how the result of the calculation was derived.

28. *The company developing the software will be responsible for maintenance of the software and should anticipate being available within an hour, 5 days a week for 16 hours a day. Start time for 16 hours a day is fixed by XBRL US? Will it be based on EST time zone?*

- Yes

29. Maintenance cost will be pay by hourly only? Annual maintenance fee is acceptable?

- Either is acceptable

30. *Standard mathematical and string functions need to be available for the manipulation of values contained in the instance. While mathematical functions would be add, subtract, multiple and divide; which string manipulation functions are expected here?*

- At a minimum string matching should be available so that enumerated values can be checked. For example the DEI taxonomy specifies different form types, and these are expressed as strings. Checks will be created that will apply for 10-K versus a 10-Q. The formula language may state where <form type> ="10-K "then execute the following check. In addition additional string functions like left(), Right() etc should be considered.

If you need clarification on any of the above questions please email Campbell.Pryde@xbri.us.