CIQ Technology WebServices Specification FinancialData Version 3.0

Date Created: 7.31.2007
Last Updated: 11.2.2010

Business Owner:	Capital IQ
Technology Owner:	William Murphy (CIQ),
Business Analysts	Shawn West
Service Version	3.0
Support	e: webservicessupport@capitaliq.com

Scenario

Capital IQ (CIQ) currently provides company information and financial data though various feeds via FTP. A number of clients are starting to utilize Web Services as a way to exchange data as an alternative to data feeds. Web Services allows our clients to access this content in a more dynamic and compact form so they can integrate it to their portals or similar internal data management systems.

The use case is the client wants current and or historical data to store on their site for internal development. A separate specification will be developed for an even narrower subset of data, based on key section of the CapitalIQ Web Site

This document represents functionality for **Company Financials**, and will focus on the following data, data sets and template types.

1. Financials: Full financials providing a complete set of data for a given company. This includes all of the Income Statement, Balance Sheet, Cash Flow and Supplemental.

2. Current and Historical Periods

3. Across all of the Capital IQ template types

    a. Industrial
    b. Bank
    c. Insurance
    d. Utility
    e. REIT
    f. Financial Services
    g. Capital Markets

4. Reference Data: Any and all reference data can be augmented by Capital IQ; clients should access this data to have the most up-to-date information available for their systems.

The **Extensible Markup Language** (**XML**) is a general-purpose markup language. Its primary purpose is to facilitate the sharing of data across different information systems, particularly via the Internet.

Application Framework

The primary technology for this solution is XML Web Services (SOAP). Capital IQ hosts an API that responds to XML requests according to this API, and returns XML structured data in response. These XML requests are encrypted via the standard HTTPS protocol.

A secondary technology for this solution is the integration of CIQ DataFeeds on client database tier. This allows for reduced network traffic for common items that change infrequently.

Capital IQ hosts this data on Windows-based servers, powered by Microsoft SQL Server in an active-passive failover cluster configuration. Data is stored in multiple fully redundant EMC Storage Area Networks (SANs). The servers that run the platform are hosted at Quality Technology Services with a disaster recovery site at XO. At all levels, these environments are redundant, fault tolerant, and backed up to industry standards.

Web Services Description Language (WSDL) documents describe the detailed Services & Ports (Function Calls) available in this specification. See http://www.w3.org/TR/wsdl for more on WSDL.

Please note that all Web Service and WSDL URLs in this document are subject to change based on changing infrastructure requirements. CIQ will provide sufficient advanced notice to the client before changing any URL, hostname, IP address, etc. It is recommended that these URLs be configurable (via config files, etc.) on the client application so that changes can be handled with minimal user downtime. CIQ monitors activity on Production systems and may shut down improper-use processes or user accounts as required to preserve overall system health.

All Web Services requests and responses in this solution are encoded in the UTF-8 character set (http://en.wikipedia.org/wiki/UTF-8). Some string data in this solution is expected to only contain Windows-1252 characters (http://en.wikipedia.org/wiki/Windows-1252); these are labeled with "(W1252)" in this document. Other string data in this solution allows full UTF-8 characters; these are labeled with "(UTF-8)" in this document. Email addresses (labeled "(email)" in this document) and website URLs (labeled "(URL)" in this document) have more limited valid character sets. See http://en.wikipedia.org/wiki/Email_address and http://en.wikipedia.org/wiki/URL for more information.

All the web services have a WSDL definition that external developers will code against and pull in data that is served from the same Capital IQ data repository as our web platform. For a full menu of our Web Services and implementation documentation, please contact your account manager.

Web Service Versioning

Versioning Web Services: Over time, Capital IQ may need to extend the tags or datasets supported by our Web services. As a results we have created a URL based versioning solution provides a scalable framework for the future. Versioning provides a way for to accommodate these enhancements in a graceful manner.

Recommendation: Capital IQ recommends that all users upgrade to version 1.0 if they are using legacy services, to conform to the new URL formats.

How versioning works: Please note in the example below *<ServiceName.asmx>* is replaced with the name of the service and is used for illustration purposes only.

1. Web Service changes are captured as a new version of the file in a new directory.

   a. **Version 1** - https://api.capitaliq.com/ciqdotnet/api/1.0/*<ServiceName.asmx>* - Represents the first release of the service
   b. **Version 2** - https://api.capitaliq.com/ciqdotnet/api/2.0/*<ServiceName.asmx>* - Represents the second release and breaking change or significant enhancement.
   c. Clients have the ability to transition to the new version of the service or stay on the original version until they can transition older code.

2. Latest version of the Service will be located at the following URL. https://api.capitaliq.com/ciqdotnet/api/current/*<ServiceName.asmx>*. Using the example in section i above https://api.capitaliq.com/ciqdotnet/api/2.0/*<ServiceName.asmx>* would be in its own directory and referenced in the current directory.

Service Changes

Financial Data

Service	Version	Comments
URL	Current	https://api.capitaliq.com/ciqdotnet/api/Current/FinancialData.asmx?WSDL
URL	3.0	https://api.capitaliq.com CIQDotNet/api/3.0/FinancialData.asmx?WSDL
URL	2.0	https://api.capitaliq.com CIQDotNet/api/2.0/FinancialData.asmx?WSDL
URL	1.0	https://api.capitaliq.com/CIQDotNet/api/1.0/FinancialData.asmx?WSDL
URL	Legacy	https://api.capitaliq.com/CIQDotNet/Financials/FinancialData.asmx?WSDL

Release	Version	Comments
11/2009	3.0	New Input/Output item: financialDataSetId. This provides client application with the vendor providing the financial datapoint value.
11/2008	2.0	Fixed <anyType> in XML output replaced with <ArrayOfCompanyFinancials> in XML output
9/2008	1.0	Updated to conform to Capital IQ new versioning criteria

Financial Pro

Financial Summary

This function allows the client application to retrieve fundamental financial information based on a select set of data points that exist on the Capital IQ web platform. This data is reported by public companies on regulatory agency filings, as well as in press releases.

The client will receive a complete list of all Capital IQ financial data items for a list of company ID's. The client will use the GetFinancialDataItems function to access any template specific data items.

Functions

Web Service Data Ports (Functions)

```
CompanyFinancials() GetFinancials(
       Integer companyIdList(), Integer dataItemList(), Integer periodTypeIDList(),
       Integer restatementTypeID, DateTime startPeriodDate, DateTime endPeriodDate,
       Integer currencyID, Integer currencyConversionMethod, Integer financialDataSetId();)
```

Comments:

This function returns Standardized financial data for the given company or companies, limited by other criteria such as data item, period type, and relative Period. Data can be returned in the reported currency or converted to a supported currency. Only data points with non-NULL data are returned by this function. "Empty" periods and/or DataItems are not returned.

To prevent performance problems, the following formula is used to calculate the maximum amount of data that can be returned by a single call: **Formula**: (# of Companies) x (# of DataItems) x (# of Periods)
If the result of this formula is greater than 10,000, then a "too much data" exception will be returned.

Client Note: Download restrictions subject to change, to account for server and client performance considerations.

Exceptions:

1. Exception thrown if the request cannot be authenticated via a session cookie.
2. Exception thrown if any parameter is out of range.
3. Exception if the account is not configured on the CIQ with the proper web service authorization.
4. Exception if the password is incorrect.
5. Exception if the web service call more than 80 TTM (Trailing Twelve Months)
6. Exception if the web service call more than 80 YTD (Year to Date)
7. Exception if more than 2500 data points are sent in a request.
    a. **Data Points** = DataItemID + Year + Company + Period
       Example = Revenue + 1999 + MSFT + Annual

Reference Function: Get Financial Data Items for Specific Templates

```
FinancialDataItem() GetFinTemplateDataItems(
       Integer templateTypeID(),Integer financialDataSetId());)
```

Comments:

This function GetFinTemplateDataItems returns an array of FinancialDataItem that contain dataitemID's and Descriptions for each template type. This allows the client application to identify all data points associated with a specific industry templates delivered through the GetFinancials function. Input Types below (continues on page 6).

ID	Name
1	Standard
2	Banks
3	Insurance

Template Input Types (continued).

4	Utility
5	REIT
7	Financial Services
8	Capital Markets
10	FFIEC

Input Parameters:

1. An Array of Integer **companyIdList()** – Each item of the array represents a single companyID corresponding to a company that has financial data. At least one valid companyId is required. **Input** [Required], [Multiple].

2. An Array of Integer **dataItemList()** – Unique identifier for the different pieces of data. This identifies the type of the value (Revenue, Expense, etc.) Limits the data points that are returned to only the DataItemIDs in the array. For a complete lists of dataItemID's and there names see getRefdata function to accessing the full list dataItemID's. At least one valid dataItemID is required. **Input** [Required], [Multiple].

ID	**Name**
42	Extraordinary Item & Accounting Change
43	Extraordinary Item & Accounting Change
44	Diluted EPS - Extraordinary Items & Accounting Change

Developer Note: Refer to Reference Data ID 36.

3. An Array of Integer **periodTypeIDList()** – Each item of the array represents a single periodicity of the financial data to be returned. Users can have multiple periods returned in a request. **Input** [Required], [Multiple] possible values:

ID	**Name**
1	Annual
2	Quarterly
3	YTD
4	LTM (Trailing 12 Months)
7	Calendar Year
10	Semi-Annual

Developer Note: Refer to Reference Data ID 18.
Client Note: Calendar Year (ID7) and Semi-Annual (ID10) will be supported in a future release.

4. Integer **restatementTypeID** – This controls what subsequent Capital IQ filing data should be returned. **Input** [Required] [Single], possible values:

ID	**Name**
0	All Filings
1	Latest Filings
2	Original Filings

Developer Note: Refer to Reference Data ID 37.

5. DateTime **startPeriodDate** – The oldest period for which financial data should be returned. **Input** [Required], [Single].

6. DateTime **endPeriodDate** – The period for which financial data should be returned that is farthest in the future. **Input** [Required], [Single].

7. `Integer` **`currencyID`** – The currency in which to display the financial data. Monetary data will be converted to this currency, if collected in a different currency. **Default**: [0] Reported Currency, **Input** [Single], [Optional] possible values:

ID	Name
0	Reported Currency
55	British Pound
27	Canadian Dollar
50	European Union Euro
64	Hong Kong Dollar
79	Japanese Yen
160	US Dollar

Developer Note: Refer to Reference Data ID 26.

8. `Integer` **`currencyConversionMethod`** – If the currency is not the reported currency, this parameter controls how the data should be currency converted. **Default**: [0] Historical , [Single], [Optional] possible values:

ID	Name
0	Historical
1	Today’s Spot Rate

Developer Note: Refer to Reference Data ID 38.

9. `Integer` **`financialDataSetID`** – Represents the data vendor that provides the financial data item value. If the client application sends option “0” (Default) as input data from all supported vendors listed below will be returned as output. **Client Note**: It is possible to have multiple data item vendors for a specific data item. **Default**: [None],[Single], [Required] possible values:

ID	Name
0	Default
1	CIQ Standardized
2	D&B
6	FFIEC
11	Compustat

DevNote: Refer to Reference Data ID 127.

Returns XML Output

1. `An array of CompanyFinancials`, a container element with one unique `CompanyID`. Each unique `CompanyFinancials tag contains` one or more `FinancialInstanceInfo` tags, and each `FinancialInstanceInfo` tag contains a list of `FinancialDataPoint elements` capturing all related dataitems per `FinancialInstanceInfo` instance.

GetFinancials

CompanyFinancials (multiple)
Elements:

a. **CompanyId** - CIQ Company ID of the company to which the financial data pertains.
b. **FinanciaInstanceInfoList** - List of Financial instances

FinancialInstanceInfo (multiple)
Elements:

i. `Integer` **`FinancialInstanceId`** – This identifies a set of financial data.
ii. `Integer` **`PeriodTypeId`** – See Parameters for explanation.
iii. `Integer` **`CalendarYear`** – Calendar year the financial data is for.
iv. `Integer` **`CalendarQuarter`** – Calendar quarter (or semi-annual period).
v. `Integer` **`InstanceTypeId`** – Possible values

ID	Name
1	Press Release
2	Original
3	Restated

vi. `Integer` **`CurrencyConversionMethod`** – Relates to the conversion Rate
vii. `Integer` **`CurrencyId`** – Relates to Currency reference data.
- See **Reference Data ID** 26 for a full list of currencies.

viii. `Integer` **`ReportedCurrencyId`** – Relates to Currency in which the data is actually reported
ix. `Integer` **`FiscalYear`** – Fiscal year the estimate is for.
x. `Integer` **`FiscalQuarter`** – Fiscal quarter (or semi-annual period).
xi. `DateTime` **`PEODate`** – The last calendar date of the period that the set of financial data is for.
xii. `DateTime` **`FilingDate`** – The date when this set of financials were filed with a regulatory agency or reported via a press release.
xiii. `Integer` **`financialDataSetID`** – DataID for the vendor that provides the financial data.

Client Note PEODate & FilingDate: The DateTime value for each returns date only. Time values will be incorporated in a future release (TBD). The current time value will be 00:00:00.

xiv. DataItemList ()– List of FinancialDatapoint tags

DataItemList
Elements:

c. **FinancialDataPoint** (optional, multiple)
Elements:

i. `String` **`DataItemValue`** – The financial value (W1252)
ii. `Integer` **`ScaleId`** – Possible values

ID	Name
1	Actual
2	Thousands
3	Millions
4	Billions

DevNote: Refer to Reference Data ID 37

iii. `Integer` **`UnitTypeId`** – Possible values

ID	Name
1	Currency
2	Ratio
3	Percentage
4	Date
5	Text
6	Enumeration
7	Boolean
8	Other

DevNote: Refer to Reference Data ID 40

iv. `Integer` **`DataItemId`** – Relates to DataItem reference data. See **Reference Data ID** 36 for a full list.for a full list.
v. `Boolean` **`AuditableFlag`** –Flag indicating if the data item is auditable or not. **True** data item is auditable **False** it is not.
vi. `Boolean` **`Subtotal`** – if true dataitemID represents a subtotal.
  - **DevNote**: Used with Financial Summary Web service only

vii. `Integer` **`Order`** – sequence to present a series of dataitemsID.
  - **DevNote**:: Used with Financial Summary Web service only

viii. `Integer` **`financialDataSetID`** – DataID for the vendor that provides the financial data.

Exceptions:

1. Exception thrown if the request cannot be authenticated via a session cookie.
   a. User can request financial data for list of companies, period types and data items
2. Exception thrown if any parameter is out of range.
   a. Input Validations – at least 1 company id, 1 period type, 1 data item id must be passed
   b. Period type id has to be either Annual, Quarter, YTD or LTM
   c. Restatement type must be either 0 (all filings), 1 (original filings) or 2 (latest)
3. Exception if the web service call more than 80 TTM (Trailing Twelve Months)
   a. Exception if the web service call more than 80 YTD (Year to Date)
4. Exception if more than 2500 data points are sent in a request.
   a. **Data Points** = DataItemID + Year + Company + Period
   Example = Revenue + 1999 + MSFT + Annual

Web Services Description Language (WSDL)

Notwithstanding anything to the contrary in this Agreement, Capital IQ reserves the right to change, expand or modify Web Services Definitions and corresponding Web Services Description Language files (WSDL) at any time. Any such modifications will be done in accordance with industry standards that support backwards compatibility with previous WSDL files. If possible, Clients will be notified in advance of any modifications.

Client Note: Login using the supplied UserName and Password provided by Client Support or your Client Development representative.

Appendices

1. **Windows-1252 A character encoding of the Latin alphabet**, used by default in the legacy components of Microsoft Windows in English and some other Western languages. The encoding is a superset of ISO 8859-1, but differs from the IANA's ISO-8859-1 by using displayable characters rather than control characters in the 0x80 to 0x9F range. It is known to Windows by the code page number 1252, and by the IANA-approved name "windows-1252". This code page also contains all the printable characters that are in ISO 8859-15 (though some are mapped to different code points).

2. **Extensible Markup Language (XML)** is a general-purpose markup language. Its primary purpose is to facilitate the sharing of data across different information systems, particularly via the Internet.

3. **dateTime [Definition:]** values may be viewed as objects with integer-valued year, month, day, hour and minute properties, a decimal-valued second property, and a boolean timezoned property. Each such object also has one decimal-valued method or computed property, timeOnTimeline, whose value is always a decimal number; the values are dimensioned in seconds, the integer 0 is 0001-01-01T00:00:00 and the value of timeOnTimeline for other dateTime values is computed using the Gregorian algorithm as modified for leap-seconds. The timeOnTimeline values form two related "timelines", one for timezoned values and one for non-timezoned values. Each timeline is a copy of the ·value space· of decimal, with integers given units of seconds.

   The ·value space· of dateTime is closely related to the dates and times described in ISO 8601. For clarity, the text above specifies a particular origin point for the timeline. It should be noted, however, that schema processors need not expose the timeOnTimeline value to schema users, and there is no requirement that a timeline-based implementation use the particular origin described here in its internal representation. Other interpretations of the ·value space· which lead to the same results (i.e., are isomorphic) are of course acceptable.

   All timezoned times are Coordinated Universal Time (UTC, sometimes called "Greenwich Mean Time"). Other timezones indicated in lexical representations are converted to UTC during conversion of literals to values. "Local" or untimezoned times are presumed to be the time in the timezone of some unspecified locality as prescribed by the appropriate legal authority; currently there are no legally prescribed timezones which are durations whose magnitude is greater than 14 hours. The value of each numeric-valued property (other than timeOnTimeline) is limited to the maximum value within the interval determined by the next-higher property. For example, the day value can never be 32, and cannot even be 29 for month 02 and year 2002 (February 2002). For more details http://www.w3.org/TR/xmlschema-2/#dateTime