CIQ Technology WebServices
Specification: Key Developments
Version 2.0

Date Created: 7.31.2007
Last Updated: 11.20.2008

Business Owner: Jay Zachter, Michael Yusko
Technology Owner: William Murphy (CIQ),
Business Analysts Shawn West
Version 2.0

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

Key Developments

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

Release	*Version*	*Comments*
11/2008	2.0/Current	Fixed <anyType> in XML output replaced with ArrayOfKeyDevelopment Type in XML output
9/2008	1.0	Updated to conform to Capital IQ new versioning criteria

Key Developments

Key Developments Summary

This section of the specification returns information about significant changes, announcements, or news stories about or related to companies. Key Developments tend to represent large enough changes in the operation of a company that they are considered "market-moving" events.

The data in this section is collected and owned by CIQ. Therefore no 3rd party licenses are required. However, the data returned in this section would require a separate license.

Key Developments Ports (Functions):

```
KeyDevelopmentInfos() GetKeyDevelopments(
       Integer CompanyID(), DateTime StartDateUTC, DateTime EndDateUTC,
       Integer KeyDevelopmentEventTypeId(), Boolean IncludeSubsidiaryFlag,
       Boolean IncludeInvestmentsFlag);
```

Comments

This function returns a KeyDeveopmentInfos container object with headline, situation, keyDevelopmentDate, lastModifiedDate and an array of KeyDevelopmentCompanyEventTypeData.

Input Parameters

1. `Array of Integer` **`CompanyId()`**– Each item of the array represents a single companyID corresponding to a company that has a Key Development . At least one valid companyID is required. **Input** [Required], [Multiple].

2. `DateTimeUTC` **`StartDateUTC`** – The oldest date & time for which Key Developments data should be returned. **Default**: [Null] **Input** [Optional], [Single].

3. `DateTimeUTC` **`EndDateUTC`**– The most recent (or farthest in the future) date & time for which Key Developments data should be returned. **Default**: [Null] **Input** [Optional], [Single].

   Date Logic: To prevent performance degradation, the maximum amount of 1000 Key Developments will be returned by a single call. Based on this maximum return amount the **`StartDateUTC/EndDateUTC`** parameter will function as follows:

   i. **Null** - If the client application does not supply a value **`StartDateUTC`** or **`EndDateUTC`** the service will return only the first 1000 Key Developments from history. Results will start from the current date going back in history.

   ii. **Start Date** - If the application supplies only **`StartDateUTC`** the service will return 1000 results from **`StartDateUTC`** up to the current date.

   iii. **End Date** - If the application supplies only **`EndDateUTC`** the service will return 1000 results from **`EndDateUTC`** going back in history.

   iv. **Exception**: An exception will be thrown for greater than 1000 KeyDevelopments per service request.

   - **Start Date & End Date Exception** – If the application supplies a **`StartDateUTC`** **and** **`EndDateUTC`** the result exceeds 1000 key developments, only the 1000 results closest to the **`EndDateUTC`** will be returned.

4. An Array of Integer **KeyDevelopmentEventTypeId()** – Key Developments of the included Types will be returned. Possible values: **Input** [Optional], [Multiple].

   Client Note: If no input value is given, client application will receive all Key Developments.
   Client Note: Some Key Development type will require additional licensing agreements contact your client development representative for further information.
   Client Note: If a KeyDevelopmentEventTypeId is supplied by the client application and the content is not licenced, that content will not return as output.

ID	Name
0	All
1	Seeking to Sell/Divest
3	Seeking Acquisitions/Investments
5	Seeking Financing/Partners
7	Bankruptcy Related
11	Delayed SEC Filings
12	Delistings
16	Executive/Board Changes - Other
21	Discontinued Operations/Downsizings
22	Strategic Alliances
23	Client Announcements
24	Accounting Issues/SEC Inquiries
25	Lawsuits & Legal Issues
26	Corporate Guidance - Lowered
27	Corporate Guidance - Raised
28	Announcements of Operating Results
29	Corporate Guidance - New/Confirmed
31	Business Expansions
32	Business Reorganizations
36	Buybacks
41	Product-Related Announcements
42	Debt Financing Related
43	Restatements of Operating Results
44	Labor-related Announcements
45	Dividend Affirmations
46	Dividend Increases
47	Dividend Decreases
48	Earnings/Operating Results Calls
49	Guidance/Update Calls
50	Shareholder/Analyst Calls
51	Conference Presentation Calls
52	Special/M&A Calls
53	Stock Splits & Significant Stock Dividends
54	Stock Dividends (<5%)
55	Expected Earnings/Operating Results Release Date
56	Name Changes
57	Exchange Changes
58	Ticker Changes
59	Auditor Going Concern Doubts
60	Address Changes
61	Delayed Earnings Announcements
62	Annual General Meetings
63	Considering Multiple Strategic Alternatives
64	Ex-Div Date (Regular)

65	M&A Rumors and Discussions
68	Credit Rating - S&P - Upgrade
69	Credit Rating - S&P - Downgrade
70	Credit Rating - S&P - Not-Rated Action
71	Credit Rating - S&P - New Rating
72	Credit Rating - S&P - CreditWatch/Outlook Action
73	Impairments/Write Offs
74	Debt Defaults
75	Index Constituent Drops
76	Legal Structure Changes
77	Changes in Company Bylaws/Rules
78	Board Meetings
79	Fiscal Year End Changes
86	Follow-on Equity Offerings
87	Fixed Income Offerings
88	Derivative/Other Instrument Offerings
89	Bankruptcies Filed
90	Bankruptcies Concluded
91	Emerged from Bankruptcy
92	End of Lock-Up Period
93	Shelf Registration Filings
94	Special Dividend Announced
95	Index Constituent Adds
97	Special/Extraordinary Shareholders Meetings
99	Potential Privatization of Government Entities
100	Ex-Div Date (Special)
101	Executive Changes - CEO
102	Executive Changes - CFO

5. **IncludeSubsidaryFlag** Boolean – If True, Key Developments for current Subsidiaries of each CompanyID provided are returned. If False, only Key Developments for the given CompanyID returned. **Default**: [False] **Input** [Single].

6. **IncludeInvementsFlag** Boolean – If True, Key Developments for current Investments of each given CompanyID provided are returned. If False, only Key Developments for the given CompanyID returned. **Default**: [False] **Input** [Single].

Output Parameters

```
KeyDevelopmentInfo ( KeyEventTypeData());
```

Comments:

This function returns an array of KeyDevelopmentInfo objects, each of which contains an array of KeyDevelopmentEventTypeData objects.

KeyDevelopmentInfo (multiple)
Attributes:

1. String **Headline** – (1-500 characters) the summary text of the Key Development. (W1252)
2. String **Situation** – (0-5000 characters) the synopsis text of the Key Development. (W1252)
3. DateTimeUTC **KeyDevleopmentDateUTC –** Date the key development was created. (Define Best Date)
    a. Best Date Rules Example: Is the known as the most important date. If GE announced a split our most important date is not the announce date but the date on or after the stock traded after the split occurred or dividend was distributed.
4. DateTimeUTC **LastModifiedDateUTC –** Date the key development was modified.

5. **KeyDevCompanyEventTypeData () (not optional, could be multiple)** –

    a. Integer **CompanyId** – CIQ companyID the key development pertains to
    b. String **CompanyName** (1-100 characters) – Company name. (W1252)
    c. Integer **ParentCompanyId** – Parent Company parentID.
    d. String **ParentCompanyName** (1-100 characters) **–** Parent Company name. (W1252)
    e. String **CompanyRelTypeName** (1-100 characters) **–** the relationship between the parent (input company) and a related company (subsidiary or investment) to each other. It can also describe the relationship between a related company through a Key Development (not an input company) and its ultimate parent. (W1252)
    f. Integer **KeyDevelopmentEventTypeId –** Identifier for the type of key development
    g. Integer **KeyDevelopmentRoleTypeId – Indicates relationship between the CompanyId and KeyDevelopment.**

ID	Name
1	Target
2	Advisor
3	Buyer
4	Seller
5	Transaction
6	Transaction Consideration

Exceptions:

1. An exception will be thrown if the request cannot be authenticated via a session cookie.
2. An exception will be thrown if any parameter is out of range.
3. An exception will be thrown for more than 1000 key developments.

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