

**CIQ Technology WebServices
Specification: ProfessionalsSummary
Version 2.0**

Date Created: 10.31.2007

Last Updated: 11.20.2008

Business Owner:
Technology Owner:
Business Analysts
Version

Jay Zachter, Michael Yusko
William Murphy (CIQ),
Shawn West
2.0

CIQ Technology Services Specification: GetProfessionalsSummary

Scenario	3
Application Framework	4
Web Service Versioning	5
Service Changes	5
Professionals GetProfessionalsSummary	6
Get ProfessionalsSummary	6
Professionals Ports (Functions):	6
Appendices	10

CIQ Technology Services Specification: GetProfessionalsSummary

Scenario

Capital IQ (CIQ) currently provides company information and financial data through 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.

This specification covers how to use the GetProfessional Web Service to retrieve basic professional information; this includes name, address, boards and education.

For detailed professional information CapitalIQ provides GetProfessionalDetail for a more granular representation based on PersonId or ProfessionalId. The PersonId and Professional are required input for GetProfessionalDetail. The client application can retrieve the PersonID via our datafeed or by running the GetProfessional function.

CIQ Technology Services Specification: GetProfessionalsSummary

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.

CIQ Technology Services Specification: GetProfessionalsSummary

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 result 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

Professional Summary

Service	Version	Comments
URL	Current	https://api.capitaliq.com/ciqdotnet/api/Current/Professionals.asmx?WSDL
URL	2.0	https://api.capitaliq.com/ciqdotnet/api/2.0/Professionals.asmx?WSDL
URL	1.0	https://api.capitaliq.com/ciqdotnet/api/1.0/Professionals.asmx?WSDL
URL	Legacy	https://api.capitaliq.com/ciqdotnet/Company/ProfessionalsDetail.asmx?WSDL

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

CIQ Technology Services Specification: GetProfessionalsSummary

Professionals GetProfessionalsSummary

Get ProfessionalsSummary

This function allows the client application to retrieve information about professionals within a company or group of companies. The function will return company name, and detailed information about professionals within the organization. This includes their board memberships, affiliations, year born, title and background. The data in this section is collected through 3rd parties and CIQ. Therefore a 3rd party licenses may be required to access this data set.

Professionals Ports (Functions):

```
Professional() GetProfessionalsSummary(
    Integer CompanyId(), Integer ProfessionalListFilter(),
    Integer ProfessionalStatusFilter, Integer ProfessionalFunctionId());
```

Comments:

This function returns an Array of professionals within a company and includes their professional type, status and function. Non-NULL data is not returned by this function

Client Note: To prevent performance problems the function will throw an exception if there are more than 1000 companyID entries per Web Service request

Input Parameters:

1. An Array of Integer **CompanyId()** – The companyID represents for the main locations for a company the Professional is associated with. At least one valid companyID is required. **Input** [Required], [Multiple]
2. Integer **ProfessionalListFilter** – Controls the selection of professional, board type and director type. Possible values: **Input** [Optional], [Single] **Default** [0] All

ID	Name
0	All
1	Professionals
2	Board Members
3	Key Professionals Only
4	Alumni

3. Integer **ProfessionalStatusFilter** – Controls the occurrence of the professional. Possible values: **Input** [Optional], [Single] **Default** [0] All

ID	Name
0	All
1	Current
2	Prior

CIQ Technology Services Specification: GetProfessionalsSummary

4. Integer **ProfessionalFunctionId()** – Each item represents a professional job function or role. Possible values:
Input [Optional], [Multiple].

Client Note: To receive all Professional Functions only a single entry of “0 – All Professional Functions” or no entry will provide the client application with all values. **it is not necessary to enter additional values even though the input allows for [Multiple]**. If the client application includes values in addition to “0” in the input for ProfessionalId all other values will be ignored.

ID	Name
0	All Professional Functions
1	Chief Executive Officer
2	Co-Chief Executive Officer
3	President
4	Co-President
5	Top Key Executive
6	Chief Financial Officer
7	Chief Operating Officer
8	Unit CEO
9	Unit President
10	Senior Key Executive
11	Chief Investment Officer
12	Chief Accounting Officer
13	Head of Investment Banking
14	Head of Corporate Finance
15	Chief Technology Officer
16	Chief Information Officer
17	Treasurer
18	Secretary
19	Assistant Secretary
20	Assistant Treasurer
21	Chief Administrative Officer
22	Head of Investor Relations
23	Chief Legal Officer
24	Head of Corporate Communications
25	Head of Sales and Marketing
26	Head of Human Resources
27	Head of Corporate Development
28	Controller
29	Other Key Executive
261	Member of Asset-Liability Committee
262	Member of Pension/Benefit Fund Investments Committee
263	Member of Budget/Planning Committee
264	Member of Real Estate/Facilities/Building Committee

CIQ Technology Services Specification: GetProfessionalsSummary

Return Output

```
Professional();
```

Comments: This function return will return an array of ProfessionalData container Objects with details about the Professionals within an array of Companies.

Professional (multiple) Elements:

- a. Integer **ProfessionalId** – Unique identifier for a "professional". A professional is the instance of a person being associated to a company (or educational institution, etc.)
- b. Integer **CompanyId** – The unique identifier of the Primary CIQ Company that the Professional relates to.
- c. String **CompanyName** – (0-200) a name that identifies the primary Company. (W1252)
- d. Integer **BoardStatusId** – Status in the Board

ID	Name
0	Not A Board Member
1	Current
2	Prior

- e. Integer **ProfessionalStatusId** – Status in the organization

ID	Name
0	Not A Professional
1	Current
2	Prior

- f. Boolean **AlumniFlag** – True if the professional is and alumni, False if the professional is not.
- g. Integer **PersonId** – This is a unique identifier for every person in the Capital IQ data set.
- h. String **Prefix** – (1-50) such as "Mr.", "Ms.", etc. (W1252)
- i. String **Salutation** – For some individuals, the name or nickname they prefer to be called. (W1252)
- j. String **FirstName** – (1-50) First Name. (W1252)
- k. String **MiddleName** – (0-50) Middle name (W1252)
- l. String **LastName** – (1-50) Last name (W1252)
- m. String **Suffix** – (0-50) Tag, such as "Jr.", "III", "Ph.D.", etc. (W1252)
- n. Date **YearBorn** – Year the professional was born
- o. String **PersonBiography** – (0-8000) A paragraph describing the experience of the person profiled (W1252)
- p. String **EmailAddress** – (6-100) email address for this professional (W1252)
- q. Boolean **DeceasedFlag** – is the professional deceased True.
- r. String **Title** – (0-500) Professional title of the Person. Example: "Senior Director".
- s. Integer **ProRank** – Returns an integer that provides a sort order for the professional. This can effectively order the company by where the people stack up in the organization.
- t. Integer **BoardRank** – The "rank" of this individual on the board. the type of board member is ranked (chairman = 1) and then ordered.

CIQ Technology Services Specification: GetProfessionalsSummary

U. **ProfessionalFunction()** (optional, multiple) - attributes:

- i. Integer **ProFunctionId** - This is the unique identifier of a given proFunction. A proFunction is a standardized way to describe titles since titles can be different across firms.
- ii. String **ProFunctionName** - (0-100) Name that identifies profunctionID (W1252)
- iii. DateTime **ProfunctionStartDateUTC** - Date professional started

V. **otheraffiliations()**(optional, multiple)- attributes:

- i. Integer **ProfessionalId** - Unique identifier for a "professional". A professional is the instance of a person being associated to a company (or educational institution, etc.)
- ii. Integer **CompanyId** - This is the unique company identifier for where the professional has affiliation
- iii. String **CompanyName** - (0-100) Name that represents the company where the professional has affiliations. (W1252)
- iv. Boolean **AlumniFlag** - True if the professional is and alumni, False if the professional is not.
- v. Integer **BoardStatusId** - Status in the Board

ID	Name
0	Not A Board Member
1	Current
2	Prior

vi. Integer **ProfessionalStatusId** - Status in the organization

ID	Name
0	Not A Professional
1	Current
2	Prior

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 if the total of companies exceed the Application setting
 - a. Base setting will be 1000 CompanyId.

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.

Login: Using the supplied Username and Password this is 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>