

**CIQ Technology Services
Specification: GetCompetitors
Version 2.0**

**Date Created: 1.22.2008
Last Updated: 11.20.2008**

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

CIQ Technology Services Specification: Get Competitors

Scenario	3
Application Framework	4
Web Service Versioning	5
Service Changes	5
Competitors	6
Competitors Summary	6
Competitors Ports (Functions):	6
Appendices	8

CIQ Technology Services Specification: Get Competitors

Scenario

This document represents the specifications for functionality, known as **GetCompetitors** the service will provide the client with a list of competitors for a specific company based on CompanyID and or subsidiaries. The source can be from the Company, Competitor or Third Party. These relationships are collected from sources such as SEC filings, press releases, and direct company contact.

CIQ Technology Services Specification: Get Competitors

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: Get Competitors

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 that 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

Get Competitors

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

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

CIQ Technology Services Specification: Get Competitors

Competitors

Competitors Summary

This section of the specification returns information about competitive relationships between Companies. These relationships are collected from sources such as SEC filings, press releases, and direct company contact. 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 the client to purchase a separate license.

Competitors Ports (Functions):

```
CompetitorsInfo () GetCompetitors(
    Integer companyID, Boolean IncludeSubsidiariesFlag,
    Integer BusinessRelationshipSourceTypeId);
```

Based on a CompanyID and a Boolean SubsidiariesFlag this function returns information about Companies that compete with the given CompanyID. The data returned is sorted based on `competitorCompanyName`.

Parameters

1. Integer **CompanyID** – The CIQ Company ID of the Company for which competitors should be returned. **Input** [Required], [Single]
2. Boolean **IncludeSubsidiariesFlag** **Input** [Optional], [Single], **Default:** [False].
 - a. If True, Competitors of current subsidiaries of the given CompanyID will be returned.
 - b. If False, only Competitors of the given CompanyID will be returned..
3. Integer **BusinessRelationshipSourceTypeId** – Describes which Company reported the competitive relationship. **Input** [Optional], [Single], **Default:** [0] All. Possible values:

ID	Name
0	All
4	Named By Third Party
5	Named By Company
6	Named By Competitor

Client Note: When calling the Data ReferenceService for **BusinessRelationshipSourceTypeId** please note that only Id's 4, 5, 6 apply to Competitors as listed above **BusinessRelationshipSourceTypeId's** 1, 2, and 3 apply to datasets out side the scope of this Web Service..

Exceptions:

1. An exception will be thrown if the request cannot be authenticated via a session cookie.
2. An exception will be thrown if the given `companyID` is not provided.

CIQ Technology Services Specification: Get Competitors

Returns:

```
CompetitorInfo();
```

An Array of CompetitorInfo Objects, each contains information about a single competitive relationship.

CompetitorInfo

Attributes: (optional, multiple)

- Integer **CompanyId** – If includeSubsidiariesFlag = False, this is always equivalent to the companyId parameter. If includeSubsidiariesFlag = True, this may be a current subsidiary of the given companyId.
- String **CompanyName** – (1-100 characters) The name of a specific Company (W1252).

Client Note Logic: If includeSubsidiariesFlag = False. This is always equivalent to the name of the Company supplied in the companyId parameter. If includeSubsidiariesFlag = True, this will represent the name of a current subsidiary supplied in the companyId parameter.

- Integer **CompetitorCompanyId** – The CIQ Company ID of the Company that competes with the Company provided in the companyId parameter.
- String **CompetitorCompanyName** – (1-100 characters) The name of a specific Company (W1252). The name of the Company that competes with the Company provided in the companyId parameter.
- Integer **BusinessRelationshipSourceTypeID** – Describes which Company reported the competitive relationship. Possible values:

ID	Name
4	Named By Third Party
5	Named By Company
6	Named By Competitor

Client Note: BusinessRelationshipSourceTypeID's 1, 2, and 3 apply to datasets outside the scope of this Web Service. Only Id's 4, 5, 6 apply to Competitors as listed above.

Exceptions:

- An exception will be thrown if the request cannot be authenticated via a session cookie.

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>