S&P Global Market Intelligence API Developer's Guide

Client Support

Email	APISupport@spglobal.com
Client Support Website	http://support.standardandpoors.com

February, 2017
V2.0.0.46

Contents

This page is intentionally left blank.

1 Introduction

The S&P Global Market Intelligence API offers fast, secure access to our extensive data sets on demand, and freedom from the overhead of processing bulk feeds or warehousing data on site.

Using the API, developers can build client library applications that use secure web services to dynamically access our data sets or bypass the client libraries and make requests using web service direct.

1.1 Data Sets

The S&P Market Intelligence API provides one of the industry's most comprehensive sets of global data— from a broad range of Fundamental, Industry Specific and Segment Data, Valuations & Pricing, Standard & Poor's Credit Ratings and Research to Reference and Terms & Conditions data. S&P Global Market Intelligence data supports a full spectrum of investor functions.

1.1.1 *S&P Capital IQ Premium Financials*

All instances of standardized financial data on an annual, quarterly, semi-annual, last twelve month and year-to-date basis are captured in S&P Capital IQ Premium Financials. This includes all representations of a financial period including press releases, original filings and all restatements. These files contain the seven industry templates of an S&P Capital IQ presentation (Industrial, Bank, Capital Markets, Financial Services, Insurance, REITs and Utilities) as well as over 5,000 financial, supplemental and industry-specific data items for active and inactive public companies.

1.1.2 *Compustat Financials*

Compustat Financials are comprised of standardized North American and global company fundamental and market data for active and inactive publicly-traded companies. North American Coverage includes annual fundamental data back to 1950, quarterly data and monthly market data back to 1962, and daily market data back to 1984. Global coverage includes annual and quarterly fundamental and daily market data back to 1986, with 20-year rolling monthly market data.

1.1.3 *S&P Capital IQ Industry-Specific Data*

Our industry-specific data delivers in-depth operational global data for companies in the following 16 industries: Airline, Bank, Hotels & Gaming, Healthcare Facilities, Homebuilding, Insurance, Internet Media, Managed Care (HMO), Mining, Oil & Gas, Pharmaceutical, Real Estate, Restaurant, Retail, Semiconductor Equipment and Telecom, Cable & Wireless..

1.1.4 *Private Company Financials*

S&P Market Intelligence Private Company Financials offers financial statement and ratio data on over 750,000 companies throughout Europe, North America and Korea. The data provided has history from 2004 forward and is consistent with S&P Capital IQ's high data quality standards.

1.1.5 *Debt and Equity Capital Structure Data*

We provide detailed information about the debt and equity aspects of a company's capital structure. Debt capital structure data covers over 51,000 public and private companies and equity capital structure data on over 81,000 public and select private companies worldwide.

1.1.6 *Bank Regulatory Data*

Our bank regulatory data provides bank regulatory and financial forms data filed with federal banking agencies, including the Federal Deposit Insurance Corporation (FDIC) as well as the Federal Financial Institutions Examination Council (FFIEC). FDIC regulatory filings cover over 20,000 banks and their subsidiaries, delivering information on over 900 unique data items with history available back to 1976. FFIEC regulatory filings include over 3,500 banks and their subsidiaries, delivering information on over 800 unique data items with history available back to 1986.

1.1.7 *Estimates Data*

S&P Capital IQ Estimates delivers standardized, global, real-time financial forecasting measures, such as upgrades / downgrades, target price revisions and market-moving news, as well as estimates based on projections, models, analysis and research from analysts, brokers and the companies themselves. History is available from 1996 for international companies and from 1999 for U.S. companies for annual, quarterly and semiannual time periods. Our new intra-day data gives you the most timely estimates and guidance, with updates and revisions available up to every five minutes.

1.1.8 *Pricing Data*

We provide independent evaluated prices daily on over 3 million global fixed income instruments, CDS price verification and OTC derivative valuations. Through our API service, you also gain access to over 1.8 million actual trade and indicative prices sourced from third-party providers on a daily basis. Gain insight into valuations based on trading data from over 80 exchanges around the world.

1.1.9 *Premium Company Data*

S&P Global Market Intelligence provides clients with access to one of the largest data sets of company information, including core and analytically-enhanced data for public and private companies globally. We combine proprietary research with select third-party content to provide profiles of public and private companies, investment firms, and professionals worldwide. Our Premium Company Data includes the following:

Dividends

The Dividends package provides all dividends paid in a security's history according to date and type.

General Company Details

Our General Company Details data provides profiles of public and private firms worldwide including company contact information, competitors, financial auditors and summary or detailed-level business descriptions.

Key Developments and Events

The Key Developments and Events data sets provide information on more than one million key developments from over 20,000 news sources including press releases, regulatory filings, company web sites, web mining and call transcripts.

Markit Short Interest

The Markit Short Interest data set includes stock loan availability and volume, short interest, and mnemonics reflecting the supply and demand in the securities lending market and the flow of funds driving market prices.

Multiples

S&P Global Market Intelligence Multiples package offers pre-calculated data points to accompany financial statements.

S&P Capital IQ Ownership

Our detailed equity ownership data delivers information for over 55,000 public and private companies worldwide consisting of institutional, mutual fund and insider/individual owners.

Private Equity Data

Our Private Equity data provides private equity firm profiles including investment professionals, limited partners, investment criteria, areas of interest and fund data.

Professionals

S&P Capital IQ Professionals data delivers profiles of professionals including biography, contact data, education, compensation, affiliations and corporate board memberships.

Transactions

Our Transactions data set provides profiles of M&A, private placement, venture capital, PIPE, spin-off, bankruptcies, share buyback programs and equity/debt public offerings. Each transaction has a synopsis, deal size, company participants, advisors coverage and associated filings. The global transaction database provides up to 10 years of history.

1.1.10 *Market, Commodity & Macro Data*

Our market data includes depositary receipts, equity market data, fixed income data, and foreign exchange rates.

Commodity data offers all the information you need on the commodities and futures markets at your fingertips, including details for over 300 securities with history as far back as 1970.

Our macroeconomic data includes coverage of historical and forecasted global economic indicators.

1.1.11 *Reference & Linking Data*

The Reference and Linking Master data set provides access to data at the security and entity level that help firms create customized data mapping systems, review and validate current identifiers and pinpoint the relationships that exist between standard and proprietary identifiers around the world. In addition, this data set provides a listing of data fields that are commonly used to describe a security, including (but not limited to) the security description, the issuer of the security, the currency it is quoted on, the country of issue, the symbols identifying the security and the exchange(s) that the security may be listed on, etc.

1.1.12 *Terms & Conditions Data*

Our Terms and Conditions data delivers near real-time and end of day updates for over 2 million global fixed income securities including corporates, governments, agencies, U.S. municipals and more than 1.4 million U.S. structured finance securities.

1.1.13 *S&P Global Credit Ratings & Research*

We offer credit ratings and related research on more than 9,000 global issuers including corporates, financial institutions and insurance companies; 600 sovereigns, and international and U.S. public finance entities; and 15,000 structured finance transactions.

1.1.14 *Credit Related Data*

Our Credit Indicators data helps you identify weakening credit and fortify your surveillance process for both rated and unrated entities with an array of quantitative tools for credit risk analysis. Credit Indicators data includes:

Probability of Default (PD)

Statistical measures that are calculated utilizing a proprietary model—PD Model Market Signals Corporates—that translates stock price volatility into credit risk indicators in order to provide daily estimates of the probability that an entity will not meet its contractual obligations. Data is available daily for all publicly traded companies.

Market Derived Signals (MDS)

Statistical measures that are calculated utilizing a proprietary model—Market Derived Signals Model— that evaluates CDS spreads in order to provide an early warning of potential credit changes and capture the market's daily view of a company's perceived risk. There are approximately 1,500 entities that have an MDS, a figure that fluctuates depending on the number of CDS that are in the marketplace. Data is available daily for scored entities.

Credit Default Swap (CDS)

CDS, although not credit risk indicators in themselves, are used for monitoring how the market views the credit risk across a wide range of companies and financial institutions and banks. The offering contains the 5-year CDS spread content.

Credit Stats data allows you to gain deeper insight into S&P Global rating methodologies with access to financial statement data, as adjusted by our credit analysts for evaluating North American and European corporate and utility issuers.

1.1.15 *Equity ETFs*

S&P Global Market Intelligence Equity ETF data includes a comprehensive system that classifies ETFs and ETNs with over 40 standardized descriptive characteristics. The data set also includes commentary and evaluations.

The S&P Capital IQ ETF Data package provides a comprehensive database of more than 2,900 publicly traded North American Exchange-Traded Products (ETP), including both Exchange-Traded Funds (ETF) and Exchange-Traded Notes (ETN).

1.1.16 *Global Funds*

Our Global Funds data includes a comprehensive classification system, commentary and evaluations.

1.1.17 *Industry, Sector & Security Groupings*

We provide extensive industry sector classifications linked at the company level. Industry classifications available in this service include Standard Industrial Classification (SIC) and the North American Industry Classification System (NAIC).

1.1.18 *Cross Reference Services*

The S&P Global Market Intelligence API now offers on-demand access to our robust cross reference capabilities. The Business Entity Cross Reference Service and the Global Instruments Cross Reference Service allow you to link multiple identifiers, identify relationships between instruments and issuers, and obtain a linear view of the corporate hierarchy.

For specific details on packages, functions and identifiers, please refer to the Release Notes on the API Support Site.

Business Entity Cross Reference Service

Our Business Entity Cross Reference Service provides immediate cross reference capabilities for 2 million public and private entities with standardized and proprietary identifiers, including the S&P Capital

IQ Company ID, S&P Capital IQ Ultimate Parent ID, Global Legal Entity Identifier (GLEI) codes, Rating Agency Identifiers, CUSIP Global Services Issuer Number, and more. With the Business Entity Cross Reference Service you can use S&P Capital IQ's data mapping to systematically update and maintain the multifaceted relationships between entities, companies and issuers.

Global Instruments Cross Reference Services

Our Global Instruments Cross Reference Service resolves the securities numbering problem in a global trading and processing environment with our comprehensive database of security identifiers, cross referencing over 3 million instruments from all over the globe. Now, both middle office and back office subscribers can streamline their operations and create customized data mapping systems for securities and trading items. With Global Instruments Cross Reference Service, you can keep up with the expanding markets and rapid growth of new securities by using this automated service to assist in linking securities to issuers.

For specific details on packages, functions and identifiers, please refer to the Release Notes on the API Support Site.

1.2 **Communication with the Client**

Communication with the API client is facilitated by web services, as shown in the following diagram:
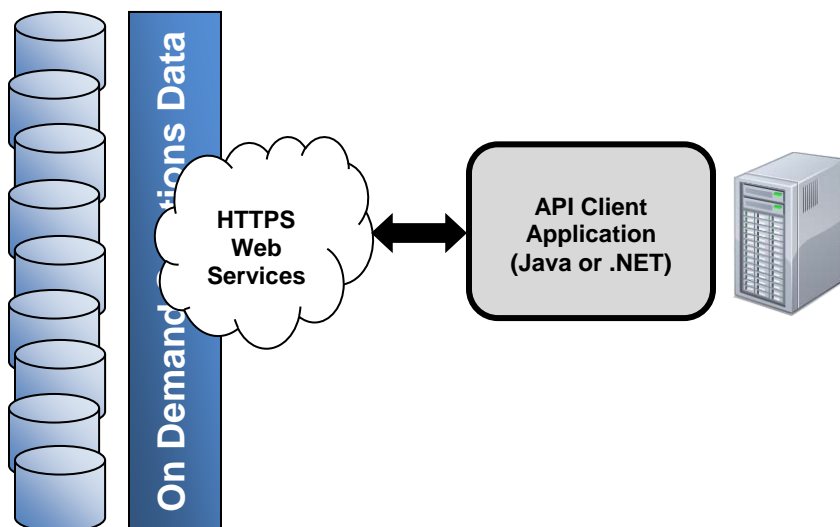
All communication between the API client and the platform is through HTTPS protocol using 256-bit encryption on port 443. The client is authenticated using encrypted HTTPS Basic Authentication, with the username and password credentials passed in as part of the header in the request.

API client libraries are provided for .**NET** and **Java**. In addition to using the API libraries, developers also have the option of communicating directly through the web services via REST/JSON (for more information on these web service standards, see Protocols, Standards and Message Formats).

1.3 **Current API Client Library Versions**

Library	Version	Download
Java	2.0.0.46	http://support.standardandpoors.com/gds/images/stories/Excel_Plug-in/SDK/SPCIQAPI_java_v2.0.0.46.zip
.NET	2.0.0.46	http://support.standardandpoors.com/gds/images/stories/Excel_Plug-in/SDK/SPCIQAPI_dotnet_v2.0.0.46.zip

1.4 **Minimum Supported Frameworks and Language Versions**

Library	Minimum Supported Version
Java	1.6
.NET	3.5
HTTPS	1.1

1.5 **Protocols, Standards and Message Formats**

Protocol	Website
JSON	http://www.json.org/
REST	http://en.wikipedia.org/wiki/Representational_State_Transfer
WSDL	http://www.w3.org/TR/wsdl
.NET	http://www.microsoft.com/net/
Java	http://www.java.com
TLS	http://en.wikipedia.org/wiki/Transport_Layer_Security
Compression	http://en.wikipedia.org/wiki/HTTP_compression

2 Downloading and Referencing the API Libraries

The next two sections outline how to download and reference the API Libraries for .NET and Java.

2.1 **API for .NET**

Tools You Will Need

Microsoft .NET Framework 3.5 or later

Download the API for .NET

1. Go to http://support.standardandpoors.com/gds/images/stories/Excel_Plug-in/SDK/SPCIQAPI_dotnet_v2.0.0.46.zip

2. Your browser will prompt you to save the zip file.

3. Once saved, unzip the API into a folder of your choice.

Files Included in the Download

Included in the zip download are three client DLLs – **SDKServices.dll, SDKWSDLServices.dll** and **SDKWSDLServices.XmlSerializers.dll –** which comprise the primary libraries, as well as several other DLLs that are optional.

SDKServices.dll, SDKWSDLServices.dll and SDKWSDLServices.XmlSerializers.dll include several classes that are not used in requesting data. Event Driven Alerts are available through a separate license; for more details, please request the Event Driven Alerts Developer's Guide.

Adding References to the .NET API Libraries

Within Visual Studio, references to the .NET API libraries can be added by right-clicking a given Project within Solution Explorer and adding references. The following example shows creating a new project and adding references to the appropriate API libraries.

1. Within Visual Studio menu, click Menu > New > Project.

2. Select the Console Application and press OK.

3. To add the API DLLs to the Project, Open Solution Explorer.

    a. Right-click the newly created Project (in this case **sdkDataSample**) and click **Add References**.

    b. Click the Browse tab. Navigate to the folder where the API libraries were unzipped. Select the three SDK DLLs and click OK.

    c. Right-click the Project and go to Properties. Make sure that the Target Framework for the Project is set to .NET Framework 4 or .NET Framework 3.5, and not the Client Profile version of either of these.

4. Code can now be added to the project to create a client that requests data.

2.2 **API for Java**

Tools You Will Need

1. Java SDK 1.6 or later version

2. A Java development environment is recommended to run the API. Available Java IDEs include:

   Eclipse (http://www.eclipse.org/)

   IntelliJ by JetBrains (http://www.jetbrains.com/index.html)

   Oracle JDeveloper (http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html)

   Apache Ant (http://ant.apache.org/)

Download the API for Java

1. Go to http://support.standardandpoors.com/gds/images/stories/Excel_Plug-in/SDK/SPCIQAPI_java_v2.0.0.46.zip

2. Your browser will prompt you to save the zip file.

3. Once saved, unzip the API into a folder of your choice.

Files Included in the Download

Included in the file download are **sdkjavaclient-2.0.0.46.jar** and **serviceschema-1.0.0.jar**, which comprise the primary API classes.

> **Note:** sdkjavaclient.jar and serviceschema.jar include several classes that are not used for data requests. In addition to data requests, the API libraries can be used to listen for Event Driven Alerts using a separate license; for more details, please request the Event Driven Alerts Developer's Guide.

Also included in the download are the following open source jar files that are required to run the API:

org.json-0.0.1.jar

Spring Source (http://www.springsource.org/):
 spring-ws-core-1.0.0.jar
 spring-xml-1.0.0.jar
 org.springframework.oxm-3.0.1.RELEASE-A.jar
 org.springframework.core-3.0.1.RELEASE-A.jar
 org.springframework.context-3.0.1.RELEASE-A.jar
 org.springframework.beans-3.0.1.RELEASE-A.jar

Apache Software Foundation (http://www.apache.org/)

    Logging Services (http://logging.apache.org/log4j/1.2/)

        log4j-1.2.14.jar

    Commons Project (http://commons.apache.org/)

        commons-logging.jar
        commons-lang.jar
        commons-io.jar
        commons-httpclient.jar
        commons-configuration-1.2.jar
        commons-collections.jar
        commons-codec.jar
        commons-beanutils.jar

            XMLBeans (http://xmlbeans.apache.org/)

    xmlbeans-2.3.0.jar

            Jackson - High-performance JSON processor (http://jackson.codehaus.org/)

    jackson-mapper-lgpl-1.2.1.jar
    jackson-core-lgpl-1.2.1.jar

            EHCache (http://ehcache.org/)

    ehcache-1.1.jar

EZMorph(http://ezmorph.sourceforge.net/)

    ezmorph-1.0.4.jar

JSON-lib (http://json-lib.sourceforge.net/)

    json-lib-2.4-jdk15.jar

Simple Logging Façade for Java (http://www.slf4j.org/dist/)

    slf4j-api-1.6.6.jar

    slf4j-jdkM-1.6.6.jar

Adding References to the Java API jar files

For development, references to the API jar files can be added to the Java build path, whereby API classes can be referenced in the code. The following example within the Eclipse IDE shows creating a new project and adding the references to the appropriate API jar files within the build path.

1. Within Eclipse, create a new Java project by clicking **New > Project** Assign a Project name (for example **SdkData**) and click **Finish**

2. To add the API jar files to the project build path, within Package Explorer

    a. Right-click the project (in this case **SdkData** > Build Path > Configure Build Path.

    b. Click the **Libraries** tab. Click the **Add External JARS** button.

    c. Navigate to the directory where the Java SDK was unzipped. All of the jar files shown are required to run the API. Select all jar files and click **Open**.

3. The API packages can now be referenced and used to create a client to request data.

3 Requesting Data on Demand

3.1 Retrieving S&P Global Data with Functions

Before coding, it is important to understand S&P Global data and how API Functions are used to retrieve the data.

Each category of data (for example, Company Data or Equity Research) includes various mnemonics pertaining to specific data, which can be queried using the **functions** provided in the API. Some functions return a single data value in time and others return a time series of data values. A description of each of the functions follows in Section 3.2.

Typical **input parameters** for the functions include a mnemonic, a Security/Entity Identifier (including CUSIP, CINS, ISIN, SEDOL, DUNS ID, GVKEY, Ticker, Capital IQ ID etc.), and a date range depending on the function type.

A complete listing of **mnemonics** that include applicable functions, security/entity identifier types, and additional parameters are available on the support site at http://support.standardandpoors.com (use the provided credentials to log in).

3.2 Available API Functions

Following is a description of each of the functions that can be used to retrieve S&P Global data.

GDSP

> This function retrieves a single data point for a point in time value for a mnemonic either current or historical. Default inputs include a Mnemonic and a Security/Entity Identifier.

> Example:

> Within Terms & Conditions data, this function can be used to retrieve the Coupon Rate (mnemonic COUPON_RATE) for a specific Security Identifier. A single data point is returned for the Security Identifier passed into the function.

GDSPV

> This function retrieves an array (a set of values) for the most current availability of content either end of day or intraday. The retrieved values are given in header and data value output classes, and typically represent a schedule, cash flows, or other items that involve multiple data values for the most current end of day. Default inputs include a Mnemonic and a Security/Entity Identifier.

> Example:

> Within Terms & Conditions data, this function can be used to retrieve the most recent Call Schedule (mnemonic CALL_SCHEDULE) for a specific Security Identifier. Headers and data values are returned for Call Type, Call Date, Call Schedule Price, and other data points that represent the current call schedule for that security.

GDSG

This function retrieves a set of GDSP mnemonics that belong to a specific group. This is a convenience function—instead of making multiple calls using GDSP, this allows you to grab several mnemonics that pertain to the same group of data. The most current end of day values are returned for the set of GDSP mnemonics. Inputs include a Group Mnemonic (there are specific mnemonics available for this function) and a Security/Entity Identifier.

Example:

Within Terms and Conditions data, there are approximately 27 mnemonics that pertain to the Basic Security Description, including COUPON_RATE, COUPON_TYPE, DATED_DATE, and others. Instead of making individual GDSP calls for each of these mnemonics, a GDSG function call can be made with the mnemonic BASIC_SECURITY_DESCRIPTION to retrieve all 27 mnemonics.

GDSHE

This function retrieves historical values for a mnemonic over a range of dates. Data values are returned according to the time series of dates. Default inputs include a Mnemonic, a Security/Entity Identifier, a start date, and an end date.

Example:

Within Pricing data, this function can be used to retrieve the time series of Bid Prices (BID_PRICE) for a given security over a range of dates. In the following example, a range from 08-01-2000 to 08-31-2011 was specified for a given security:

BID_PRICE	PRICE_DATE_D
100.228	05/27/2011
100.3	05/07/2011
100.3	05/06/2011
100.42	02/18/2011
100.67	08/25/2010
100.63	07/26/2010
100.63	07/02/2010
100.63	05/25/2010

GDSHV

This function retrieves an array or set of values over a historical range of dates. Data values are returned according to the time series of dates. Default inputs include a Mnemonic, a Security/Entity Identifier, a start date, and an end date.

Example:

Within S&P Ratings data, the Standard Instrument Credit Rating (INS_STD_CREDIT_RATING) can be retrieved for an entity based on a date range. A table of credit ratings is returned where one header specifies the date.

GDST

This function retrieves historical values for a mnemonic over a range of dates, where a specific frequency can be specified—Annual (A), Quarterly (Q), Monthly (M), Weekly (W), or Daily (D). Header rows for the returned values represent the dates of the periodic values. Default inputs include a Mnemonic, a Security/Entity Identifier, a start date, an end date, and a frequency.

Example:

Within Pricing data, this function can be used to retrieve the Ask Price (ASK_PRICE) of a security based on a date range and a specified frequency of Daily (D) or Weekly (W). The following data shows what is returned for Ask Price for a given security where the frequency is specified as Weekly and the date range is from 1/1/2010 to 1/31/2010:

01/01/2010	01/08/2010	01/15/2010	01/22/2010	01/29/2010
1.62	1.59	1.60	1.69	1.66

3.3 **Specialized Data Request Properties**

3.3.1 *Functions*

Following is a list of the Specialized Data Request properties that can be used to retrieve specific types of mnemonics:

	GDSP	GDST	GDSHE	GDSG	GDSPV	GDSHV
absolutePeriodEnd			✓			
absolutePeriodStart	✓		✓			
asOfDate	✓		✓			
compareTo	✓		✓			
consolidatedFlag	✓	✓	✓			
currencyConversionModeId	✓	✓	✓			
currencyId	✓	✓	✓			
enableMetadata					✓	✓
endDate	✓	✓	✓			✓
endRank			✓		✓	✓
filingMode	✓	✓	✓			
frequency	✓	✓				
limit		✓				
metaDataTag			✓			
modelId	✓		✓			
periodType	✓	✓	✓			
rank	✓					
restatementTypeId	✓		✓			
secondaryIdentifier	✓					
seniority	✓	✓		✓	✓	✓
snapType	✓	✓				
source	✓					
startDate	✓	✓	✓			✓
startRank			✓		✓	✓
tenor	✓					

3.3.2 *Descriptions*

Property type	Description
absolutePeriodEnd	Indicates the end period of the period range for which value is to be calculated for the metric
absolutePeriodStart	Indicates the start period of the period range for which value is to be calculated for the metric
asOfDate	Specifies the date for a metric
compareTo	Passes an identifier or list of identifiers to calculate the value of a given mnemonic so that comparisons can be made. This property is specific to Credit Related Data.
consolidatedFlag	The valid values for *consolidatedFlag* are CON – Consolidated [*default]* PAR – Parent Only UNC - Unconsolidated
currencyConversionModeId	If the currency is not the reported currency, this parameter controls how the data should be currency converted. The valid values for *currencyConversionModeId* are Historical and SpotRate.
currencyId	The currency in which the financial data should be displayed. Monetary data will be converted to this currency, if collected in a different currency. There are approximately 200 currency codes, for example: LOCAL, INR, USD etc.
enableMetaData	This property, when set to *true,* will output additional data elements for the mnemonic being requested. This property only applies to IQ_ mnemonics, and is only applicable with *GDSPV* and *GDSHV* functions.
endDate	This property should be used with startDate. It specifies the end date for which data is requested. If the endDate is not specified, the current date is considered to be the endDate by default. The date format is mm/dd/yyyy.
endRank	This property is used for range functions. The valid values are numbers.
filingMode	The valid values for filingMode are P – Period [default] F – Filing

Property type	Description
frequency	Specifies the Frequency of the requested data: FA = Fiscal Annual FS = Fiscal Semi-Annual FQ = Fiscal Quarter D = Daily W = Weekly M = Monthly Q = Quarterly S = Semi-Annual A = Annual Use the following frequencies for IQ_ mnemonics: Daily Weekly Monthly Quarterly Yearly
limit	Limit is only applicable when requesting Credit Stats Direct Public Finance and Financial Institution mnemonics using GDST. Limit should be used in conjunction with the Frequency property. If Limit is '8' and Frequency is 'FQ' (Fiscal Quarterly), the response will include time-series data values for the 8 most recent fiscal quarters. The value of Limit should be between 1 and 40.
metaDataTag	This property will influence the header row of a GDSHE request to a specific date or value depending on the user's preference and requested mnemonic. This property only applies to IQ_ mnemonics. The default value is AsOfDate and can be changed to any of the below fields (list not inclusive): • AsOfDate • CalendarQuarter • CalendarYear • EffectiveDate • FilingDate • FiscalQuarter • FiscalYear • Month • PeriodDate • PricingDate • Rank • ScoringDate • ScoreRank • ScoringYear • ValuationDate

Property type	Description
modelId	This property passes the requested Credit Analytics Model including: • CM – CreditModel • FN - Fundamental • MS – Market Signal Note: This property applies only to Credit Analytics Data
periodType	The following period types can be used in the API to specify the type of data being returned. Relative: For retrieving data relative to a current period the functions accepts IQ_FY: Fiscal Year IQ_CY: Calendar Year IQ_FH: Fiscal Half IQ_CH: Calendar Half IQ_FQ: Fiscal Quarter IQ_CQ: Calendar Quarter IQ_LTM: Last 12 Months IQ_NTM: Next 12 Months IQ_YTD: Year-to-Date Any of the above period types can be used with an offset (+/- #) in order to retrieve historical or future periods. Absolute: For retrieving data within a specific date range the functions accepts StartPeriodDate/EndPeriodDate as inputs. FY2011 = Fiscal Year 2011 CY2011 = Calendar Year 2011 FH2011 = Fiscal Half 1 2011 CH2011 = Fiscal Half 1 2011 FQ12011 = Fiscal Quarter 1 2011 CQ12011 = Calendar Quarter 1 2011 Any of the expressions above using Absolute referencing can be modified to pull a different period by changing the existing period reference. For IQ Financials, IQ_LTM is the PeriodType default; for IQ Estimates, IQ_NTM is the PeriodType default For additional PeriodType examples, see the Appendix
rank	This property is used for range functions. The valid values are numbers.
restatementTypeId	The valid values for *restatementTypeId* are None O – Original version P – Preliminary Version LC – Latest on Capital IQ [*default]* LFR – Latest fully reported version LRI – Latest reported instance LRP – Latest reported period

Property type	Description
secondaryIdentifier	Represents any identifier (e.g., as company, trading item, etc.)
seniority	Seniority is applicable only when requesting CDS Pricing mnemonics. It is mnemonic derived and not function derived. 'Seniority' should be set to 'SEN' (Seniority) or 'SUB' (Subordinated). If not defined, 'SEN' is used as the default for these mnemonics.
snapType	
source	Identifies the source of the data (i.e. S&P Capital IQ data or Proprietary data). Below are the supported values. • P = Proprietary data only • C = S&P Capital IQ data only • PC = Proprietary data initially,then S&P Capital IQ data
startDate	When used with the point in time function (GDSP), this property can be used to request data for a particular date. When used with the Range (GDSHE) or time series (GDST) functions, this property can be used to request the data for a specified date. The date format is mm/dd/yyyy.
startRank	This property is used for range functions. The valid values are numbers.
tenor	This property is specifically for Credit Analytics Data. Below are the supported Tenor values that represent the period for which data is required: 1M, 3M, 6M, 9M, 1Y, 2Y, 3Y, 4Y, 5Y, 6Y, 7Y, 8Y, 9Y, 10Y, 11Y, 12Y, 13Y, 14Y, 15Y, 16Y, 17Y, 18Y, 19Y, 20Y, 21Y, 22Y, 23Y, 24Y, 25Y, 26Y, 27Y, 28Y, 29Y, 30Y

3.4 **Requesting Data with Identifiers**

With the S&P Global Market Intelligence API, you may use the following identifiers in your requests:

3.4.1 *Securities and Instruments*

A **Security** is a financial instrument that may be traded on one or on multiple exchanges. You can designate the securities you want in your portfolio. Since API Drive can match all identifiers related to a security, you may submit any identifier including:

CUSIP9	ISIN
CINS9	RatingsXpress Instrument ID
SEDOL	

3.4.2 *Trading Items*

A **Trading Item** represents the equivalent of a security + the exchange on which it trades. For example, the common stock for Teva Pharmaceuticals trades on both the NYSE and the Tel Aviv exchanges. A tradingItem differentiates between the two exchanges on which the security trades.

CIQ Ticker:Exchange

CIQ Trading Item ID

3.4.3 *Entities and Issuers*

An **Entity** uniquely identifies a legal corporate structure or legal object. An Issuer is a legal organization that issues any type of security. With API Drive, you define the entities and/or issuers you want in your portfolio. Since API Drive can match all identifiers related to an entity or issuer, you may submit any identifier including:

CIQ Company ID	CUSIP6
Compustat GVKEY	CINS6
RatingsXpress Entity ID	CABRE
D&B DUNS ID	CMA Entity ID

3.4.4 *Other Identifiers*

Other identifiers you may select include:

CIQ Professional ID – The S&P Global identifier for a professional

CIQ Person ID – The S&P Global identifier for a person

CIQ Transaction ID – The S&P Global identifier for a transaction

CIQ KeyDev ID – The S&P Global identifier for a key development or future event

CIQ Contributor ID – The S&P Global identifier for a research contributor

CIQ Currency ID – The S&P Global identifier for a currency

CIQ CountryCode – The S&P Global identifier for a country

Ticker – A symbol for a stock ticker

Universe – This input identifier can be used with applicable mnemonics when you would like to retrieve a universe of entities and/or securities.

Company Professional Search String – This input identifier is applicable for Company and Professional quick search mnemonics. You can input a search string to look for a company or a professional.

4 Responses from API Function Call Requests

4.1 Response Types: OBJECT, JSON and XML

Query responses are returned in native OBJECT format (the default) which returns a List of SDKDataOutput objects that can be interrogated for the response data, as well as in XML format. There is one other response type—JSON—that can be retrieved by invoking the service with the following code:

.NET Client Library

OBJECT

```
var response = _API.InvokeDataService(sdkInput) as List<SDKDataOutput>;
```

XML

```
var response = _API.InvokeDataService(sdkInput,"XML") as XDocument;
```

JSON

```
string response = (string) _API.InvokeDataService service(sdkInput,"JSON");
```

Java Client Library

JSON

```
String responseJson = (String)serviceImpl.invokeDataService(input_request,"JSON");
```

XML

```
String responseXml = (String)serviceImpl.invokeDataService(input_request,"XML");
```

4.2 **Common Error Responses**

It is possible to receive the following error responses when submitting a request:

Data Unavailable

This means that there is no coverage for the data request (the identifier, mnemonic, and function are correct for input).

Not Applicable

This means that an invalid identifier has been submitted that doesn't match up with any of our identifier formats (at the entity or security level). It could mean you have used an entity level identifier as an input, but made a request for a security level mnemonic (or vice versa).

Invalid Identifier

This means that an invalid identifier was passed into a data request.

Invalid Mnemonic

This means that an incorrect mnemonic was passed into a data request.

Input Arguments Missing

This means that there are one or more input arguments missing.

Function Mismatch

This means that a mnemonic has been passed in that does not work with the specific function it was assigned to (it works with another function).

Not Entitled

This means that the user is not entitled to the specific mnemonic passed in to the data request.

Error Processing Function / Error Processing Request / Invalid Request

These are general errors meaning there is an issue with the input parameters in the request – an input parameter might be missing or some of the input parameters may be invalid.

Full List of Error Responses

For a full list of responses, please refer to the Error Messages Guide under Application Programming Interface (API) on the Support Center website http://support.standardandpoors.com.

5 Web Service Direct

5.1 Understanding Web Service Direct

The Web Service Direct service is a HTTPS Web-based REST/JSON (Representational State Transfer / Java Script Object Notation) service. This allows a client or application to integrate directly with the API without deploying any client libraries within the development environment.

The REST/JSON Web Service Direct (POST method) endpoint (versioned) is:

> https://sdk.gds.standardandpoors.com/gdssdk/rest/v2/clientservice.json

BASIC Authentication is required and requires the following:

> API Username / API Password

The JSON request structure and arguments are very similar to the API arguments currently used within the API Client libraries. The actual JSON request should be passed in the request BODY rather than as part of the POST command. This will enable a larger request size to be passed to the platform.

6 Code Examples

For a more detailed discussion of code samples for testing user credentials and requesting data, please refer to Downloads under Application Programming Interface (API) on the Support Center website http://support.standardandpoors.com.

6.1 Connecting Through a Proxy

Set Proxy Example for .NET

To connect through the SDK proxy, use the following code:

```
using SDKServices.Model;

// Create Data Input
SDKDataInput dataInput = new SDKDataInput();
// proxy required
dataInput._Proxy = new SDKProxy( "proxyUser",        // proxyUser - optional (proxy dependent) or null
                                 "proxyPass",        // proxyPass - optional (proxy dependent) or null
                                 "proxy.corp.com",   // proxyHost - required (uri or ip address)
                                 1234,               // proxyPort - required (port #)
                                 ""                  ); // proxyDomain - optional (proxy dependent) or null
```

Simple Request Example for .NET

To perform a simple request, use the following code:

```csharp
using System.Xml.Linq;
using SDKServices.ServicesImpl;
using SDKServices.Model;
using SDKServices.Util;

// Create Service
SDKDataServicesWrapperImpl service = new SDKDataServicesWrapperImpl();
// Create Data Request Container
List<SDKDataRequest> reqs = new List<SDKDataRequest>();
// Create Data Request
SDKDataRequest req;
// Set Data Request
req = new SDKDataRequest(SDKEnumerators.Functions.GDSP.ToString(),
        new List<string> { "IBM", "IQ313055" },
        new List<string> { "IQ_TOTAL_REV", "IQ_COST_REV" },
        new Dictionary<string, string> {    { "periodType",                          "IQ_FY"       },
                                            { "restatementTypeId",            "LFR"           },
                                            { "filingMode",                                 "P"             },
                                            { "currencyId",                                 "USD"           },
                                            { "currencyConversionModeId",        "HISTORICAL" } });
// Add Data Request to Data Request Container
reqs.Add(req);
// Create Data Request
req = new SDKDataRequest(SDKEnumerators.Functions.GDSP.ToString(),
        new List<string> { "00083BAB1", "001055102"          },
        new List<string> { "SECURITY_TO_ISSUER_NAME"          },
        new Dictionary<string, string>()                            );
// Add Data Request to Data Request Container
reqs.Add(req);

// Create Data Request
req = new SDKDataRequest(SDKEnumerators.Functions.GDST.ToString(),
        new List<string> { "00083BAB1", "001055102" },
        new List<string> { "ASK_PRICE", "BID_PRICE" },
        new Dictionary<string, string> { { "startDate", "01/01/2014" },
                                         { "endDate",    "" },
                                         { "frequency", "D"          } });
// Add Data Request to Data Request Container
reqs.Add(req);
// Create Data Input
SDKDataInput dataInput = new SDKDataInput(  "apiUser",        // apiUser - required
                                           "apiPass",        // apiPass - required
                                           true,             // default is true
                                           null,             // null or SDKProxy
                                           new List<SDKDataRequest>(reqs));
// Make Data Request - Native Object (List<SDKDataOutput>) Output
var respOBJECT = service.InvokeDataService(dataInput) as List<SDKDataOutput>;
// Make Data Request - XML Output
var respXML = service.InvokeDataService(dataInput, SDKEnumerators.OutputTypes.XML.ToString()) as XDocument;

... Process the Results
```

JSON Direct (JSON/REST) API Access Method – HTTP Post

Always use an HTTP POST (vs. HTTP GET) for API JSON Direct requests targeting the REST service URI endpoint. Use the POST body for passing the JSON API request (inputRequests=<JSON>) information.

Note: Do not issue the POST with the JSON request information as a parameter on the URI since there are restrictions on the URI size.

```
Simple JSON Request Examples for .NET:
//// Variables
string jReqJSON = "";
string jReqBody = "";
string jResp = "";
byte[] jReqBytes = null;
Stream dStream = null;
StreamReader sReader = null;
HttpWebRequest jDirReq = null;
HttpWebResponse jDirResp = null;

//// Set JSON POST Body
jReqBody += "inputRequests=" + jReqJSON;
//// Convert POST Body
jReqBytes = Encoding.UTF8.GetBytes(jReqBody);
//// Create JSON DIRECT Req Other
jDirReq = (HttpWebRequest)WebRequest.Create("apiRESTServiceURL");
jDirReq.Method = "POST";
jDirReq.Credentials = new NetworkCredential("apiUser", "apiPass");
jDirReq.ContentType = "application/x-www-form-urlencoded";
jDirReq.Headers.Add("Accept-Encoding:gzip,deflate");
jDirReq.KeepAlive = true;
jDirReq.Timeout = 15 * 60 * 1000;
jDirReq.ContentLength = jReqBytes.Length;
//// TLS security protocol required
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;
//////////////////////////////////////
//// MAKE API DIRECT (JSON) REQUEST ////
//////////////////////////////////////
//// Get JSON DIRECT Req Stream
dStream = jDirReq.GetRequestStream();
//// Make JSON DIRECT Req via Req Stream
dStream.Write(jReqBytes, 0, jReqBytes.Length);
//// Close JSON DIRECT Req Stream
dStream.Close();
//////////////////////////////////////
//// GET API DIRECT (JSON) RESPONSE ////
//////////////////////////////////////
jDirResp = (HttpWebResponse)jDirReq.GetResponse();
```

```
The above ContentType is required.
```

7 API Best Practices

This section outlines best practices to follow when using the S&P Global Market Intelligence API with the API Client Libraries or the JSON Direct (JSON/REST) access method.

Caching Data (Locally)

Caching data locally improves overall performance. If your application uses the same data point(s) frequently and you do not need to refresh data via the API every time, caching locally avoids the overhead of redundant requests. Additionally, if your workflow requires historical data points, local caching is highly recommended as most historical data is static.

Validate API Inputs

Validate all API request input parameters (i.e. functions, identifiers, mnemonics, and property key/value pairs) prior to making any API request. This avoids the most common API errors.

Avoid Duplicate API Requests

When an API request is in progress, wait for the API response before making the same request. Multiple requests for the same identifier/mnemonic combinations can cause temporary blocking of API request processing leading to delays in API responses.

Multiple API Requests

Although concurrent API requests per IP address are supported, we advise limiting the number of concurrent API request threads to 5. Too many concurrent requests can cause processing time delays and, depending on your subscription, may even block API access.

API Request Sizes (Identifiers/Mnemonics per Request)

It is good practice to group multiple functions, identifiers, and mnemonics in a single API request, although we recommend avoiding large requests. The API platform connection timeout is 15 minutes for a single API request.

When coding against the API, it's helpful to size the number of data points being requested. We suggest tuning individual API requests to yield approximately 1000 data points, where each request is defined as one function, mnemonic, identifier (plus other property combinations as applicable).

Time Series, Historical, and Vector Information

The Time Series, Historical, and Vector functions (GDST, GDSPV, GDSHE and GDSHV) require more processing time and return more data points than the Point-in-Time functions (GDSP and GDSG).

In addition to sizing the number of requests, when making requests for time series, historical, or vector API requests, limit the input ranges (data, rank, etc.) to return smaller API requests with better response times.

API Response Errors

It is good practice to resolve all known API errors. This avoids duplicate API requests with the same exception such as making multiple requests with an incorrect API user name and password, invalid inputs, etc.

Update to the Latest API Client Libraries

If you are using the API Client Libraries and are notified of a new library release, update the libraries to the latest version in order to leverage the latest features of the API.

Proxy Connection Issues

If your API application is behind a proxy, ensure you have necessary proxy credentials to access the API platform. If the proxy credentials change or expire, make the necessary adjustments prior to making API requests.

8 FAQ

1) **I'm getting error responses such as Not Applicable, Invalid Mnemonic, etc. What do these mean?**

   See the section on Error Responses.

2) **How do I handle an HTTP Proxy?**

   For the API client libraries:

   If a proxy is not required, it can be set to" null".

   If a proxy is required, the ProxyUser, ProxyPassword, ProxyHost and ProxyPort are required. ProxyDomain is optional and can be "null" if not required.

   For direct services access, please use the appropriate means in the language to set an HTTP Proxy while making web requests.

3) **Are there any limitations on the underlying HTTP commands (GET, POST, PUT, etc.)?**

   Only the HTTP POST command is supported for now.

4) **How do I handle connection persistence?**

   HTTP is a stateless protocol. However HTTP keepalive should be used to keep the connection persistent for a longer period of time. For most modern browsers, this is a built in feature.

   For direct services access, please use the appropriate means in the language to set keepalive property in HTTP Header.

   The client library already implements this and sets keepalive to 30 seconds.

5) **Are there any restrictions on the number of connections per user?**

   Please refer to Section 9 Standard Account Attributes for details.

6) **Are there request or response size limitations?**

   Please refer to Section 9 Standard Account Attributes for details.

7) **Do you support HTTP compression?**

   To make better use of available network bandwidth and faster transmission speeds between the client and the On Demand server, we support HTTP Compression in gzip format.

   http://en.wikipedia.org/wiki/HTTP_compression

8) **What is required for multithreading using the client libraries?**

   Each thread must create a separate instance of the Data Services Wrapper class.

9 Standard Account Attributes

Trial Account	
Concurrent Mnemonic Function Limit	100
Concurrent Request Limit Per Server	2
Daily Limit	24000
Hourly Limit	1000

Production Account	
Concurrent Mnemonic Function Limit	500
Concurrent Request Limit Per Server	5
Daily Limit	-
Hourly Limit	-

10 Appendix

ALL	Absolute Date	Calendar Date	mm/dd/yyyy		06/30/2010	Current Date
n/a	Absolute Date	Calendar Date	mm-dd-yyyy		06-30-2010	n/a
n/a	Absolute Date	Calendar Date	dd-MMM-yyyy	where MMM is month abbreviation	30-JUN-2010	n/a
CSD	Absolute Y	Fiscal Year	Yyyyy		Y2010	Current Y
CSD	Absolute SY	Fiscal Semi	SsYyyyy	where s is 1 or 2	S2Y2010	Current SY
CSD	Absolute QY	Fiscal Quarter	QqYyyyy	where q is 1 thru 4	Q4Y2010	Current QY
IQ	Absolute Month	Calendar Month	MMMyyyy	where MMM is month abbreviation	JUN2010	Current Month
IQ	IQ Absolute FY	Fiscal Year	FYyyyy		FY2010	Current FY
IQ	IQ Absolute CY	Calendar Year	CYyyyy		CY2010	Current CY
IQ	IQ Absolute FH	Fiscal Half	FHhyyyy	where h is 1 or 2	FH12010	Current FH
IQ	IQ Absolute CH	Calendar Half	CHhyyyy	where h is 1 or 2	CH22010	Current CH
IQ	IQ Absolute FQ	Fiscal Quarter	FQqyyyy	where q is 1 thru 4	FQ12010	Current FQ
IQ	IQ Absolute CQ	Calendar Quarter	CQqyyyy	where q is 1 thru 4	CQ42010	Current CQ
IQ	IQ Absolute LTM	Last 12 Months	LTMqyyyy	where q is 1 thru 4	LTM42010	Current LTM
IQ	IQ Absolute NTM	Next 12 Months	NTMqyyyy	where q is 1 thru 4	NTM42010	Current NTM
IQ	IQ Absolute YTD	Year-to-Date	YTDqyyyy	where q is 1 thru 4	YTD42010	Current YTD
IQ	IQ Relative FY	Fiscal Year	IQ_FY	current FY period	IQ_FY	Current FY
IQ	IQ Relative FY	Fiscal Year –	IQ_FY–n	where n is FY periods back	IQ_FY–3	Current FY–n
IQ	IQ Relative FY	Fiscal Year +	IQ_FY+n	where n is FY periods forward	IQ_FY+3	Current FY+n
IQ	IQ Relative CY	Calendar Year	IQ_CY	current CY period	IQ_CY	Current CY
IQ	IQ Relative CY	Calendar Year –	IQ_CY–n	where n is CY periods back	IQ_CY–7	Current CY–n
IQ	IQ Relative CY	Calendar Year +	IQ_CY+n	where n is CY periods forward	IQ_CY+7	Current CY+n
IQ	IQ Relative FH	Fiscal Half	IQ_FH	current FH period	IQ_FH	Current FH
IQ	IQ Relative FH	Fiscal Half –	IQ_FH–n	where n is FH periods back	IQ_FH–5	Current FH–n
IQ	IQ Relative FH	Fiscal Half +	IQ_FH+n	where n is FH periods forward	IQ_FH+5	Current FH+n
IQ	IQ Relative CH	Calendar Half	IQ_CH	current CH period	IQ_CH	Current CH
IQ	IQ Relative CH	Calendar Half –	IQ_CH–n	where n is CH periods back	IQ_CH–3	Current CH–n
IQ	IQ Relative CH	Calendar Half +	IQ_CH+n	where n is CH periods forward	IQ_CH+3	Current CH+n
IQ	IQ Relative FQ	Fiscal Quarter	IQ_FQ	current FQ period	IQ_FQ	Current FQ
IQ	IQ Relative FQ	Fiscal Quarter –	IQ_FQ–n	where n is FQ periods back	IQ_FQ–11	Current FQ–n
IQ	IQ Relative FQ	Fiscal Quarter +	IQ_FQ+n	where n is FQ periods forward	IQ_FQ+11	Current FQ+n
IQ	IQ Relative CQ	Calendar Quarter	IQ_CQ	current CQ period	IQ_CQ	Current CY
IQ	IQ Relative CQ	Calendar Quarter –	IQ_CQ–n	where n is CQ periods back	IQ_CQ–15	Current CY–n

IQ	IQ Relative CQ	Calendar Quarter +	IQ_CQ+n	where n is CQ periods forward	IQ_CQ+15	Current CY+n
IQ	IQ Relative LTM	Last 12 Months	IQ_LTM	current LTM period	IQ_LTM	Current LTM
IQ	IQ Relative LTM	Last 12 Months –	IQ_LTM–n	where n is LTM periods back	IQ_LTM–19	Current LTM–n
IQ	IQ Relative LTM	Last 12 Months +	IQ_LTM+n	where n is LTM periods forward	IQ_LTM+19	Current LTM+n
IQ	IQ Relative NTM	Next 12 Months	IQ_NTM	current NTM period	IQ_NTM	Current NTM
IQ	IQ Relative NTM	Next 12 Months –	IQ_NTM–n	where n is NTM periods back	IQ_NTM–7	Current NTM–n
IQ	IQ Relative NTM	Next 12 Months +	IQ_NTM+n	where n is NTM periods forward	IQ_NTM+7	Current NTM+n
IQ	IQ Relative YTD	Year-to-Date	IQ_YTD	current YTD period	IQ_YTD	Current YTD
IQ	IQ Relative YTD	Year-to-Date –	IQ_YTD–n	where n is YTD periods back	IQ_YTD–9	Current YTD–n
IQ	IQ Relative YTD	Year-to-Date +	IQ_YTD+n	where n is YTD periods forward	IQ_YTD+9	Current YTD+n

11 Revision History

The changes made to this document include the following:

Revision	Date	Changes
2.0.0	December 2011	Beta Release Version 2.0.0
2.0.0.7	March 2012	Beta Release Version 2.0.0.7
2.0.0.12	June 2012	Beta Release Version 2.0.0.12
2.0.0.15	August 2012	Beta Release Version 2.0.0.15
2.0.0.16	August 2012	Beta Release Version 2.0.0.16
2.0.0.19	January 2013	Beta Release Version 2.0.0.19
2.0.0.29	May 2013	Beta Release Version 2.0.0.29
2.0.0.31	July 2013	Version 2.0.0.31
2.0.0.40	November 2013	Version 2.0.0.40
2.0.0.42	June 2014	Version 2.0.0.42
2.0.0.45	August 2014	Version 2.0.0.45
2.0.0.46	December 2014 January 2015 May 2015	Version 2.0.0.46
	June 2016	Added the Appendix and Revision History sections
	September 2016	• Reflects new company branding • Section 1.1 now includes detailed descriptions of Data Sets • Section 3.3 Specialized Data Request Properties has been updated to include additional Functions and detailed Descriptions • Added new content: Section 3.4 Requesting Data with Identifiers
	November 2016	• Added definition for tenor in Section 3.3.2 Descriptions • Minor revisions to Section 8 FAQ • Added Section 1.1.18 Cross Reference Services
	February 2017	• Updated URLs for client libraries