Aggregation Development Quick Start Guide

DataConnect Product Version: V4.0

Document Version: 2

Document Issue Date: 1/5/2017

Technical Support: (866) 856-4951

Telephone: (781) 376-0801

Fax: (781) 376-8040

Web: byallaccounts.morningstar.com

Table of Contents

OVERVIEW

The Morningstar® ByAllAccounts^{SM} service provides normalized aggregation of client account data from more than 15,000 financial institutions using sophisticated artificial intelligence to automatically gather and transform financial data. Incorporating APIs and UI elements, ByAllAccounts is a complete solution ready for you to quickly integrate.

Using the ByAllAccounts DataConnect API you can create users, then give them access to the ByAllAccounts AccountView interface so they can set up their accounts, then use the API to retrieve their aggregated account data which includes accounts, positions (stock, bonds, cash, etc.), and transactions.

Why use data aggregation?

Data aggregation is automated gathering, transforming and delivering of investors' financial account data. Get a unified view of portfolios by outsourcing aggregation of securities and alternative investments from multiple custodians. Improve productivity, reduce friction, and focus on client relationships by eliminating manual entry of hard-to-get data.

About this document

This document explains how:

- you can use the ByAllAccounts DataConnect API to create users.

- your users can access the ByAllAccounts AccountView ConsumerUI to set up their financial accounts.

- you use the API to retrieve their aggregated account data which includes accounts, positions, balances, and transactions.

This document reiterates and expands upon the content of https://developer.morningstar.com/apis/data-aggregation.

Where to get more information

- For descriptions of the export files listed in this guide, refer to the *AccountView and DataConnect Export* guide posted at http://www.byallaccounts.net/ manuals/accountview/BAA_Export.PDF.

- The DataConnect API provides additional operations that you may wish to use depending on your needs. For more information, refer to *DataConnect V4 Ultra User Guide posted at* http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF.

- For details on managing single sign on (SSO) users refer to *AccountView Single Sign-on Guide* http://www.byallaccounts.net/Manuals/Accountview/AccountView_SingleSignOn.pdf

GETTING STARTED

This section explains how to get going quickly with Morningstar ByAllAccounts aggregation.

1. **Get Developer Access** - Request a developer login for the service.

   - Send an email to byallaccounts-support@morningstar.com

   - Include your name, company name, and email address.

   - Use the developer login with <LOGINRQ> to authenticate the user as having sufficient privilege to perform DataConnect functions.

2. **Create Users** - Use <USERADDRQ> in the API to users, giving them a service login and password.   The sample code in this guide shows how to add an investor user.

3. **Send Users info to log in to AccountView** - Send the following to each of your investor users that you created in step #2.

    a. For direct login users, send the following to each of your investor users that you created in step #2.

        ▪   The AccountView URL:https://www.byallaccounts.net/BAAWebApp/BAALogin.html

        ▪   Their login ID and password for AccountView

        ▪   Provide instructions telling your users to log into AccountView and set up their accounts.
            The user interface should be intuitive to the investors. You can learn more here:
            http://www.byallaccounts.net/Manuals/Accountview/TrainingVideos/ConsumerUI.mp4

    b. For details on managing single sign on (SSO) users refer to
        http://www.byallaccounts.net/Manuals/Accountview/AccountView_SingleSignOn.pdf

4. **Retrieve the aggregated data**
    The service automatically aggregates the data for an account when that account is first set up and then each night after that. An account must have valid credentials to the institution to aggregate. When you retrieve aggregated data from the API you are getting the data from the most recent successful aggregation. The API provides additional operations if you want to force an aggregation of an account on demand.

    Use the API to retrieve the aggregated content in .CSV files. For details about individual calls listed below refer to Operation Summary.

        ▪   <DATAEXPORTRQ_A>

        ▪   <DATACLAIMRQ>

        ▪   <FILECLAIMRQ>

    For a sample of how these go together with the Create User step, refer to Sample Code.

The API provides additional operations that you may wish to use depending on your needs. For more information, refer to
http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF.

SAMPLE CODE

The following sample code shows requests and responses for:

- Add an Investor User
- Export the Data: sample request that gets all three files (accounts, holdings, and transactions) for two accounts, with an historic date and a transaction date range.
- Claim the Data
- Claim the CSV Files

The result should yield a ZIP file containing a README.txt file and CSV files depending on what you requested and what is available.

Add an investor user using <USERADDRQ> request

```
<DATACONNECTRQ>
    <VERSION>VERSION4.0</VERSION>
      <LOGINRQ>
          <LOGIN_NAME>AVAdvisor1</LOGIN_NAME>
          <LOGIN_PW>AVAdvisor1</LOGIN_PW>
      </LOGINRQ>
      <USERADDRQ>
        <PERSON>
            <ROLE>INVESTOR</ROLE>
            <FIRST_NAME>Chris</FIRST_NAME>
            <MIDDLE_NAME></MIDDLE_NAME>
            <LAST_NAME>RWInvestor</LAST_NAME>
            <EMAIL_ADDRESS>Chris_RWInvestor@CRW.com</EMAIL_ADDRESS>
            <PHONE></PHONE>
        </PERSON>
         <LOGIN>
            <LOGIN_NAME>RWInvestor</LOGIN_NAME>
            <LOGIN_PW>RWInvestor</LOGIN_PW>
        </LOGIN>
        <OWN_FP_ACCESS>READWRITE</OWN_FP_ACCESS>
      </USERADDRQ>
  </DATACONNECTRQ>
```

```
<DATACONNECTRS>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRS>
      <STATUS>
      <ERRCODE>0</ERRCODE>
      <ERRMSG>Success</ERRMSG>
      </STATUS>
   </LOGINRS>
   <USERADDRS>
      <STATUS>
         <ERRCODE>0</ERRCODE>
         <ERRMSG>Success</ERRMSG>
      </STATUS>
      <ID>19118</ID>
      <FP_ID>13565</FP_ID>
   </USERADDRS>
</DATACONNECTRS>
```

Export the Data using <DATAEXPORTRQ_A>

```
<DATACONNECTRQ>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRQ>
      <LOGIN_NAME>RWInvestor</LOGIN_NAME>
      <LOGIN_PW>RWInvestor</LOGIN_PW>
   </LOGINRQ>
   <DATAEXPORTRQ_A>
      <EXPORT_DATA_QUERY>
         <ACCOUNT_QUERY>
            <FP_ID>13565</FP_ID>
            <ID>360744</ID>
            <FP_ID>13565</FP_ID>
            <ID>360747</ID>
         </ACCOUNT_QUERY>
      </EXPORT_DATA_QUERY>
```

```
        <EXPORT_DATA_AS_OF_DATE>20161011</EXPORT_DATA_AS_OF_DATE>
        <INCACCOUNTFILE/>
        <INCHOLDINGFILE/>
        <INCTXFILE>
            <TX_START_DATE>20161005</TX_START_DATE>
            <TX_END_DATE>20161011</TX_END_DATE>
        </INCTXFILE>
    </DATAEXPORTRQ_A>
</DATACONNECTRQ>

<DATACONNECTRS>
    <VERSION>VERSION4.0</VERSION>
    <LOGINRS>
        <STATUS>
            <ERRCODE>0</ERRCODE>
            <ERRMSG>Success</ERRMSG>
        </STATUS>
    </LOGINRS>
    <DATAEXPORTRS_A>
        <STATUS>
            <ERRCODE>0</ERRCODE>
            <ERRMSG>Operation started</ERRMSG>
        </STATUS>
        <RECEIPT>4768946401097394200</RECEIPT>
        <RECEIPT_EXP>20161228121229 [-5:EST]</RECEIPT_EXP>
        <CLAIM_WAIT>1000</CLAIM_WAIT>
    </DATAEXPORTRS_A>
</DATACONNECTRS>
```

Claim the Data using<DATACLAIMRQ>

```
<DATACONNECTRQ>
    <VERSION>VERSION4.0</VERSION>
    <LOGINRQ>
        <LOGIN_NAME>AVAdvisor1</LOGIN_NAME>
        <LOGIN_PW>AVAdvisor1</LOGIN_PW>
    </LOGINRQ>
    <DATACLAIMRQ>
        <RECEIPT>4768946401097394200</RECEIPT>
    </DATACLAIMRQ>
</DATACONNECTRQ>


<DATACONNECTRS>
    <VERSION>VERSION4.0</VERSION>
    <LOGINRS>
        <STATUS>
        <ERRCODE>0</ERRCODE>
        <ERRMSG>Success</ERRMSG>
        </STATUS>
    </LOGINRS>
    <DATAEXPORTRS_A>
        <STATUS>
            <ERRCODE>0</ERRCODE>
            <ERRMSG>Success</ERRMSG>
        </STATUS>
    </DATAEXPORTRS_A>
</DATACONNECTRS>
```

Claim the CSV files using <FILECLAIMRQ>

```
<DATACONNECTRQ>
    <VERSION>VERSION4.0</VERSION>
    <LOGINRQ>
        <LOGIN_NAME>RWInvestor</LOGIN_NAME>
        <LOGIN_PW>RWInvestor </LOGIN_PW>
    </LOGINRQ>
    <FILECLAIMRQ>
        <RECEIPT>4768946401097394200</RECEIPT>
    </FILECLAIMRQ>
</DATACONNECTRQ>
```

BASIC DATACONNECT OPERATIONS

Overview

This section describes the basics of what you need to know to effectively use DataConnect.

DataConnect URL

The DataConnect API for V4 consists of a single URL invocation point:

https://www.byallaccounts.net/dataconnect/WPServlet?RequestType=DataConnectV4

This URL must be specified exactly (including case). Your request is routed to the appropriate component on ByAllAccounts' servers. You must:

- Use https (http requests are rejected).

- Use a request method type of **POST** (**GET** type requests are rejected).

The data you provide in the **POST:**

- Is the Input Request Document.

- Contains your Login request (with credentials that authenticate you as a valid user of DataConnect).

- Is your data retrieval or data update request.

The DataConnect web server returns to you a DataConnect response file that is compressed using the ZIP compression format.

DataConnect complies with the HTTP/V1.1 protocol and requires use of SSL (https). All input request documents must be sent via **POST** over https.

DataConnect Response documents are always compressed in ZIP format.

General XML Document Information

All DataConnect Request and Response documents must adhere to the following:

- All XML element names must be in upper case. For example, the element **<USER_GROUP>** is correct, while the element **<user_group>** is incorrect.

- Values of elements are not case-sensitive unless specifically noted.

- Each document must include a DTD reference to a publicly available DataConnect DTD. The DTD to use is defined in subsequent sections. DataConnect validates your request document against the DTD referenced in your document. If the document cannot be validated, an error is returned to you.

- UTF-8, a compressed version of Unicode that uses only a single byte for most common characters, is the character set used for all documents.

Overall Document Structure

Input Request

The input request contains the following:

- **DOCTYPE**

- **DataConnect Version Specification**

- **Login Request** providing credentials to authenticate the caller

- **Operation Request.**

An input request document has the following overall structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE DATACONNECTRQ PUBLIC

'-//DataConnect DTD//DataConnect//EN'
'http://www.byallaccounts.net/WebPortfolio/com/baa/dtd/v4/
DataConnectRQ.dtd'>
<DATACONNECTRQ>

    <VERSION>Version4.0</VERSION>
    { One login request }
    { One operation request }

</DATACONNECTRQ>
```

Output Response

A response document has the following overall structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE DATACONNECTRS PUBLIC

'-//DataConnect DTD//DataConnect//EN'

'http://www.byallaccounts.net/WebPortfolio/com/baa/dtd/v4/
DataConnectRS.dtd'>
```

```
<DATACONNECTRS>
   <VERSION>Version4.0</VERSION>
   { One login response }
   { One operation response }

</DATACONNECTRS>
```

Asynchronous Operations

By nature, some DataConnect operations are lengthy. For these operations, an http connection cannot be maintained for the duration of the operation because it consumes resources unnecessarily and is subject to interruption by several factors, including various client and server timeout settings. For example, this asynchronous operation is potentially lengthy:

- **Export Data:** Retrieves accounts, holdings, and transactions in CSV-delimited format.

Asynchronous operations are distinguished by an **_A** at the end of the request and response tags (e.g., **<DATAEXPORTRQ_A>** is asynchronous).

For asynchronous operations, the simple request/response communication sequence used by synchronous operations is replaced with the following steps:

1. DataConnect client sends initial request (**<DATAEXPORTRQ_A>** for example).

2. DataConnect server reads and parses initial request. If the request is not valid, an error is returned. If the request is valid, an **Operation Started** status along with a receipt (to retrieve the data later) and an expiration timestamp (point after which the data is no longer retained) are returned. A suggested wait time in milliseconds before issuing the **Claim Data** request is also included in the response.

3. DataConnect server begins processing the request.

4. DataConnect client waits for the number of milliseconds specified in the **<CLAIM_WAIT>** and then sends the subsequent request (<**DATACLAIMRQ>**), including the receipt, to determine if the operation is completed. DataConnect server responds with one of the following:

   - Invalid/expired receipt

   - Operation in progress

   - Status and data from the operation

5. Once the client receives data, he should call back the DataConnect server with an Acknowledge Data Receipt request. Upon receipt of this acknowledgement, the DataConnect server removes the data from the temporary data store.

DataConnect clients must accommodate the return of the **Operation Started** status and code to the above scheme in those cases.

Once the operation completes, the data resulting from the operation is retained until the expiration time of the receipt. Expiration time is determined on a per operation basis.

The DataConnect client must submit subsequent requests for data resulting from an asynchronous operation from the same IP address as the initial request.

Operation Summary

The following table summarizes the DataConnect operations used in Sample Code, page 3.

General Operations		
Login	Login authentication. It authenticates the caller as a user with sufficient privilege to perform DataConnect functions. There must be only one LOGINRQ in an input request document and it must be the first request in the document.	Request: <LOGINRQ> Response: <LOGINRS>
Modification Operations		
Add User	Creates a new user. The user types include Investor, Advisor, Assistant, and Consultant. The code example in this guide created an Investor user.	Request: <USERADDRQ> Response: <USERADDRS>
Retrieval Operations		
Export Data (asynchronous)	Export the data for accounts, holdings, transactions, and clients in CSV-delimited format.	Request: <DATAEXPORTRQ_A> Response: <DATAEXPORTRS_A>
Asynchronous Helper Operations		
Claim Data	Retrieves data resulting from a previously requested asynchronous operation.	Request: <DATACLAIMRQ> Response: <DATACLAIMRS>
Claim File	Used in conjunction with the Export Data (Asynchronous) operation to download the ZIP file. For descriptions of the export files within the ZIP file, refer to the *ByAllAccounts Export* guide posted at http://www.byallaccounts.net/manuals/account view/BAA_Export.PDF.	Request: <FILECLAIMRQ> Response: ZIP file containing CSV exports, or <FILECLAIMRS>

For details on all of these operations, refer to the *DataConnect V4 Ultra User Guide* posted at http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF.

That guide describes the operations of general utility, including authentication and general error response. The following items are defined for each DataConnect operation where that information is relevant:

- **Purpose:** Why or for what the operation is used

- **Restrictions:** Limitations or notes on the use of the operation

- **Behavior (optional):** For complex operations, explains in detail the behavior of the operation under different circumstances

- **User Notifications (optional):** User notifications resulting from the function (if any)

- **Request:** Form of the function request
- **Response:** Form of the function response
- **Errors:** Status codes that may be returned in the response
- **Sample XML:** Examples of the operation request and response

DATACONNECT ACCESS AND TYPE DEFINITIONS

This section provides a quick reference to the DataConnect access point and DataConnect document type definitions.

The main DataConnect URL (case-sensitive):

https://www.byallaccounts.net/dataconnect/WPServlet?RequestType=DataConnectV4

The DataConnect Document Type (DTD) definitions are described in two separate documents.

This document describes the inbound requests:

http://www.byallaccounts.net/WebPortfolio/com/baa/dtd/v4/DataConnectRQ.dtd

This document describes the responses that come back:

http://www.byallaccounts.net/WebPortfolio/com/baa/dtd/v4/DataConnectRS.dtd

SAMPLE INTEGRATION CODE

The following sample call to the API is a program fragment written in the Java programming language. It uses the capabilities available in the Java 2 programming environment (Java 1.3 Runtime). This fragment demonstrates the invocation of the DataConnect URL, posting of a request document to that URL, receipt of a response stream, and the writing of that compressed data (ZIP) to a file.

The compressed file produced by the fragment:

- Can be read using a tool capable of decompressing data in ZIP compression format (such as WinZip or GNU gzip)
- Should be given a file extension of .zip
- Will contain a single XML file with the response data or an error response document
- Will be named **WpGetDatammddhhmi.xml** where mmddhhmi is a date and time (mm=month, dd=day, hh=hour, mi=minute)

```
…
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.net.HttpURLConnection;
```

```java
import java.net.URL;
…
 try {
// Configure for use of https
System.setProperty( "java.protocol.handler.pkgs","com.sun.net.ssl.internal.www.protocol");
   java.security.Security.addProvider( new com.sun.net.ssl.internal.ssl.Provider());<br />
 // open the URL to DataConnect
 //
String contactUrl = "https://www.byallaccounts.net/dataconnect/WPServlet?RequestType=DataConnectV4";
String inputFileName = "myRequest.xml";  // my request document
URL url = new URL(contactUrl);
     HttpURLConnection urlc = (HttpURLConnection)url.openConnection();
   urlc.setDoOutput (true);
     urlc.setUseCaches (false);
urlc.setRequestProperty ("Content-Type", "text/xml; charset=UTF-8");
     urlc.setRequestProperty ("Accept-Encoding", "zip");
     urlc.setRequestMethod ("POST");
     ByteArrayOutputStream byteStream =<br />
(ByteArrayOutputStream)urlc.getOutputStream();
// write the request document (POST it) to the DataConnect URL
     FileReader fileReader = new FileReader(new File(inputFileName));
     int next=fileReader.read();
     while ( next != (-1) )
     {
         byteStream.write(next );
       next=fileReader.read();
     }
     byteStream.flush();</p>

<p>// create the file to store the response - response data will be compressed in ZIP format
   String outputFileName = "responseFile";
   FileOutputStream reply = new FileOutputStream(File.createTempFile(outputFileName, ".zip"));
   InputStream input = urlc.getInputStream();
// read compressed bytes from the response write them to our file
   int received = input.read();
   while (received != (-1))
   {
       reply.write(received);
       received=input.read();
}
   input.close();
   reply.close();
} catch (Exception e) {
```

```
    // report exception
  }
  …
```

SAMPLE AUTHENTICATION CODE

Sample Authentication Code

The Login Request is always the first request in an input request document. An additional request to retrieve or update data follows the Login Request.

The following is a sample Login Request:

```xml
<DATACONNECTRQ>

  <VERSION>Version4.0</VERSION>

  <LOGINRQ>

    <LOGIN_NAME>joesmith</LOGIN_NAME>

    <LOGIN_PW>xyz123</LOGIN_PW>

  </LOGINRQ>

  …

</DATACONNECTRQ>
```

The following is a sample Login Response:

```xml
<DATACONNECTRS>

  <VERSION>VERSION4.0</VERSION>

  <LOGINRS>

    <STATUS>

      <ERRCODE>0</ERRCODE>

      <ERRMSG>Success</ERRMSG>

    </STATUS>

  </LOGINRS>

  …

</DATACONNECTRS>
```