TABLE OF CONTENTS

INTRODUCTION TO ODATA ................................................................. 3
Advantages of using ODATA Services from database directly ........................................ 3

ODATA SUPPORT IN SYBASE ASE .................................................................. 3
Here is sample output in ATOM/Pub XML format in browser ........................................ 5
Sample Output in JSON format .................................................................................. 5
Enabling ODATA support for a schema in Sybase ASE .................................................. 6

RUNNING INCLUDED ODATA PRODUCER AND CONSUMER SAMPLE IN SYBASE ASE ...... 7
Step 1: ASE ODATA Setup .............................................................................. 7
Step 2. Confirm ODATA Producer configuration file “SecureViewConfig.properties” ........ 8
Step 3: Start ASE ODATA Server and verify connections ............................................ 8
Step 4: Create ODATA URL’s for QUERY/UPDATE .............................................. 9
Sample ODATA Query URLs for browser for AtomPub response ..................................... 12
Using cURL Utility on Windows for JSON response .................................................. 12
Step 5: Setup Java and ODATA dependent Libraries ................................................ 12
Step 6: Running ODATA JAVA Consumer Client .................................................... 12

CONCLUSION AND NEXT STEPS .................................................................. 13
Goal of this document is to give overview of ODATA protocol (http://www.odata.org) now supported by Sybase ASE and to give instructions for setting up and running bundled ODATA sample using both browser and Java ODATA consumer client. This is hands on guide and after this introduction, user should be able to setup own ASE database for ODATA support and learn new paradigm for developing applications using data over HTTP.

INTRODUCTION TO ODATA

The Open Data Protocol (ODATA) enables creation of REST like HTTP services for relational data thus exposing database model in a uniform manner for creating, querying and updating data over HTTP backed by enterprises such as SAP and Microsoft. One can control which OData operations are exposed, host multiple OData endpoints, alongside non-OData endpoints and exercise full control over your data model, back-end business logic, and data layer.

OData supports two formats for representing the resources (Collections, Entries, Links, etc.) it exposes: the XML-based AtomPub format and the JSON format which are standard in industry and applications can be written for ATOM or JSON using any modern programming language: Java, C#, .NET, PHP, Python, Javascript etc.

Latest version of Sybase ASE database (all editions) including publicly available free Developer Edition comes with built-in support for ODATA protocol.

One does not have to use any tools to consume the OData services but instead know how to build URI's for querying data, then retrieving the data with any of many options to get a web page content, and parse it using Gson or a similar library.

Advantages of using ODATA Services from database directly

- Easier for internet applications to provide API for 3rd party application: With ODATA support, application developers no longer to manually fetch data from database and do Tabular format to XML transformation for data over HTTP support themselves. For instance, Netflix/Facebook/Twitter are not only internet services but they also lets 3rd party application developers build application using their data in REST format.

- Rapidly develop web as well mobile application using same data: Without direct ODATA support from database, applications have to fetch data using columnar format and then transform it for appropriate view say JSON or other XML based. With ODATA, applications can continue using RDBMS as repository build business logic using JSON/AtomPub format. Native Android/iOS applications can be developed quickly by consuming data in JSON.

ODATA SUPPORT IN SYBASE ASE

Sybase ASE database server comes pre-bundled with OData Producer which automatically transforms database tables into ODATA format and a built in HTTP server lets application consume XML based oDATA over HTTP in a transparent manner. Sybase ASE has provided utilities which require bare minimum configuration for this data transformation to occur.

ODATA Producer is combination of Java application in form of Servlet developed using OData4j library which fetches data from Sybase ASE database and transforms requested data into JSON/AtomPub payload and a HTTP server (JettyServer) which lets a web client consume that payload. Here is architecture for Sybase ASE ODATA Producer.
Applications can not only retrieve data over HTTP directly from Sybase ASE but also perform edits. Method mapping between HTTP Requests and Sybase ASE database operations

<table>
<thead>
<tr>
<th>HTTP Methods</th>
<th>Sybase ASE Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>SELECT</td>
</tr>
<tr>
<td>POST</td>
<td>INSERT</td>
</tr>
<tr>
<td>PUT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

Figure 1: Sybase ASE ODATA Architecture
Here is sample output in ATOM/Pub XML format in browser

```xml
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:rights="http://purl.org/dc/elements/1.1/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <atom:link rel="edit" href="https://localhost:8090/odata/products(300)"/>
  <atom:title type="text">Book</atom:title>
  <atom:rights>...</atom:rights>
  <atom:rights>...</atom:rights>
  <atom:rights>...</atom:rights>
  <atom:rights>...</atom:rights>
  <atom:rights>...</atom:rights>
</atom:entry>
```

**Figure 2 AtomPub Example**

Sample Output in JSON format

Table name: "titles"; Columns: title_id, title, type, pub_id, price, advance, total_sales, notes, pubdate, contract

```json
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://localhost:8090/odata/titles('BU1032')",
          "type": "SAPSybaseOData.titles"
        },
        "title_id": "BU1032",
        "title": "The Busy Executive's Database Guide",
        "type": "business",
        "pub_id": "1389",
        "price": "19.99",
        "advance": "5000",
        "total_sales": 4095,
        "notes": "An overview of available database systems with emphasis on common business applications. Illustrated."
      },
      {
        "__metadata": {
          "uri": "https://localhost:8090/odata/titles('BU1111')",
          "type": "SAPSybaseOData.titles"
        }
      }
    ]
  }
}```
Enabling ODATA support for a schema in Sybase ASE

**ODATA producer packaged with ASE standard install**
Sybase ASE standard installation includes ODATA producer which can be started and stopped by command line tool called “dboserv” bundled with Sybase ASE. This utility requires a text based configuration properties file which includes information about database schema, JDBC connection, and embedded HTTP Server port. Running dboserv16 utility starts the embedded HTTP server while as well as enabling ODATA transformation while “dbostop” stops the server dboserv16.exe and dbostop.exe utility are located at $SYBASE/ODATA-16_0/bin64
%SYBASE% refers to root of Sybase ASE installation for example, c:\sybase then dboserv16.exe is located at %SYBASE%\ODATA-16_0\bin64\dboserv16.exe

**Start ODATA Producer:**
dboserv16.exe takes *.properties file as an argument, file name can be anything.
%SYBASE%\ODATA-16_0\bin64\dboserv16.exe server.properties

**Stop ODATA Server:**
%SYBASE%\ODATA-16_0\bin64\dbostop.exe server.properties

One cannot manually configure the embedded HTTP server, or use it to serve other non- OData content, such as HTML files

Sample server.properties typically contains four sections.

- **Embedded HTTP Server options:** In this section, one can specify HTTP server listening and shutdown port one would like HTTP server to start and listen to, HTTP server log file, SSL key store location and password for HTTPS support (As this is Java based web server). Important thing to note here that context root for ODATA is always “odata”. So, URI for ODATA will typically be “http://localhost:<http_port>/odata/(DatabaseEntity)

- **ODATA Producer Option:** Single property “DBConnectionString” specifies JDBC URL information about how ODATA servlet connects to Sybase ASE server, format is <ase_database_hostname>:<ase_database_port>/<database_user>. For example, localhost:5000/odata

- **Database Authentication:** Specify type of database with property “DBProduct”, and type of authentication with property “Authentication”. Authentication property can be set to “none” or
“database” in which case, browser prompts for Database credentials before presenting Atom/JSON data.

- **Model file**: OData Producer service model to expose specific tables and views by defining a namespace (the model) in a text file that complies to the OData Service Definition Language (OSDL) syntax.

Complete sample of ODATA producer configuration file

```plaintext
# Embedded HTTP server options
# -----------------------------
# Set the server listening port to 8090
ServerPort=8090
PageSize=30
# Enable Info level logging
LogVerbosity=4
# Write logs to a file
LogFile = SecureViewLog.txt
ShutdownListenerPort=8082
# Load the server certificate (in the form of a Java Keystore). This enables HTTPS traffic and
# disables HTTP traffic
# The path is relative to the location of the server executable
SSLKeyStore = keystore.jks
SSLKeyStorePassword = sample

# OData Producer options
# -----------------------
DbConnectionString=OAKN00527588A:5000/odata
# Database connection parameters
# ----------------------------------
# Connect to a ASE database. This property lets one connect to Sybase ASA database too
DbProduct=ASE
# Enable database authentication. This prompts for userID and password
Authentication=database

# Provide an optional connection string used to validate the OSDL file on startup
# (This option should be used during development only, and removed for production deployment)
# Model = SecureView.osdl
ModelConnectionString = servername:portNumber/databaseName?user=httpAuthUser2&password=password2
```

**RUNNING INCLUDED ODATA PRODUCER AND CONSUMER SAMPLE IN SYBASE ASE**

Following tutorial assumes basic familiarity with Sybase ASE installation on Windows, otherwise, these two getting started guides are highly recommended:

- [Sybase ASE Developer Edition Windows Installation Guide](#)
- [Getting Started with SAP Sybase ASE 15.7 and Eclipse IDE](#)

JAVA ODATA sample is located at $SYBASE/ODATA-16_0/samples/java and contains batch files to start the ODATA server, setup ASE database as well as compile and run Java program.

This sample illustrates how to use an OData4J Java client to send and consume OData requests over HTTPS to an OData Server that connects to an Adaptive Server sample database. This shows you how an OData Producer service model works with database authentication.

**Step 1: ASE ODATA Setup**
Sybase ASE comes pre-bundled with sample database “pubs2” created at time of installation. To ensure that our ODATA sample maintains integrity of original database, we are going to create another database user “httpAuthUser2” with password “password2” and another database schema “odata” (It can be anything, for this sample, these username and database name are selected). We will copy table “titles” under “pubs2” database to “odata” database under username “httpAuthUser2”.

Also, add primary key constraint on column title_id as pubs2 database doesn’t have those and ODATA implementation requires presence of Primary key constraint on a table init.sql located at $SYBASE/ODATA-16_0/samples/java contains above steps and included as part of run.bat One can run init.sql from Sybase isql commandline directly, or manually execute above steps on any other SQL editor.

%SYBASE%\OCS-15_0\bin\isql -Slocalhost -UhttpAuthUser2 -Ppassword2 -i init.sql -o init.out

Step 2. Confirm ODATA Producer configuration file “SecureViewConfig.properties”

```
# Embedded HTTP server options
# ------------------------
# Set the server listening port to 8090
ServerPort=8090
# Enable info level logging
LogVerbosity=4
# Write logs to a file
LogFile = SecureViewLog.txt
ShutdownListenerPort=8082
# Load the server certificate (in the form of a Java Keystore).
# This enables HTTPS traffic and disables HTTP traffic
SSLKeyStore = keystore.jks
SSLEncryptionPassword = sample
# OData Producer options
# ------------------------
dbProduct = ASE
# Database connection parameters
# ------------------------
dbConnection=OAR00527588A:5000/odata
# Enable database authentication. This prompts for userID and password
AuthenticationDatabase=database
PageSize=30
# Provide an optional connection string used to validate the OSDL file on startup
# (This option should be used during development only, and removed for production deployment)
Model = SecureView.osdl
#ModelConnectionString = OAR00527588A:5000/databaseName?user=httpAuthUser2&password=password2
```

Figure 3 Property Sample

Step 3: Start ASE ODATA Server and verify connections

Here assumption is that ASE server is already running, and if not, start ASE server using Windows service. On command prompt at %SYBASE%\ODATA-16_0\samples\java, execute following commands:

- %SYBASE%\sybase.bat to setup SYBASE related environment variable
- %SYBASE%\ODATA-16_0\samples\java>..\..\..\bin64\dbosrv16 SecureViewConfig.properties

Now ODATA Server should start
Verify ODATA Producer working correctly by going to URL https://localhost:8090/odata/$metadata on a browser. Browser should prompt from database username and password - Enter “httpAuthUser2” and “password2” and now AtomPub feed should appear for table “titles” and meta description.

Step 4: Create ODATA URL’s for QUERY/UPDATE

So far, we don’t need to run ODATA Java client and just perform direct URL operations on browser for default AtomPub ODATA response or use “curl” type utility to get JSON response. https://localhost:8090/odata/titles on browser will display all titles in browser.

Depending upon your browser security setting, you might see this security error, Click on “Continue to this website” and enter “httpAuthUser2” and “password2” in subsequent browser authentication dialog.
Figure 5: Database Authentication prompts over HTTPS

If database credentials are correct, then browser will display metadata about “Titles” table in AtomPub format.
Figure 6: Sample AtomPub ODATA Response

In addition to the value and the count, OData defines other URL options to customize the output of your query. For example, $orderby allows you to order a query result by one or more properties in ascending or descending order:

<table>
<thead>
<tr>
<th>Query Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$expand</td>
<td>Expand related data inline with the results, e.g. Categories/$expand=Products would expand Product data inline with each Category entry.</td>
</tr>
<tr>
<td>$filter</td>
<td>A Boolean expression for whether a particular entry should be included in the feed, e.g. Categories?$filter=CategoryName eq 'Produce'. The Query Expression section describes OData expressions.</td>
</tr>
<tr>
<td>$format</td>
<td>One of &quot;atom&quot; (the default), &quot;json&quot; or &quot;xml&quot; depending on how you'd like your data returned to you.</td>
</tr>
<tr>
<td>$inlinecount</td>
<td>Includes the number of entries without the filter applied as a count element on the feed itself, e.g. Categories? $stop=4&amp;$inlinecount=allpages will return 8 with the default Northwind database installed.</td>
</tr>
<tr>
<td>$orderby</td>
<td>One or more comma-separated expressions with an optional &quot;asc&quot; (the default) or &quot;desc&quot; depending on the order you'd like the values sorted, e.g. Categories?$orderby=CategoryName desc.</td>
</tr>
<tr>
<td>$select</td>
<td>Limit the properties on each entry to just those requested, e.g. Categories?$select=CategoryName, Description.</td>
</tr>
<tr>
<td>$skip</td>
<td>How many entries you'd like to skip, e.g. Categories?$skip=4.</td>
</tr>
</tbody>
</table>
Sample ODATA Query URLs for browser for AtomPub response

- https://localhost:8090/odata/titles: returns listing of all titles rows
- https://localhost:8090/odata/titles/$count: returns number of titles
- https://localhost:8090/odata/titles?$top=1: returns top 1 title

Using cURL Utility on Windows for JSON response

Instead of browser, one can use command line utility like cURL too which is more flexible for changing requested HTTP headers, type of data, submit POSTS etc. Default response from CURL is application/xml. Make sure curl with SSL supported is installed from given options at http://curl.haxx.se/download.html to run requests over HTTPS.

Sample cURL command to get number of records in our titles table:

  Verify that console response: 18

To get JSON response from ODATA server, http request needs to indicate output is required in JSON. In cURL we do that passing “-H” flag.


Step 5: Setup Java and ODATA dependent Libraries

After verifying both JSON and AtomPub responses are returned correctly from Sybase ASE ODATA, one can run ODATA Consumer sample by compiling and running SecureView.java file. Just make sure that odata4j consumer libraries are added in classpath.

Download OData4J Version 0.7 from http://code.google.com/p/odata4j/.

Setup JAVA_HOME environment variable to latest JDK installation on machine and modify CLASSPATH environment variable to include OData4J client bundle .jar file (odata4j-0.7.0-clientbundle.jar)

```
set ODATA4J_JAR=C:\odata4j\bundles\odata4j-0.7.0-bundle.jar

cd %SYBASE_HOME%\ODATA-16_0\samples\java

"%JAVA_HOME%\bin\javac -cp "%CLASSPATH%\;ODATA4J_JAR% *.java
"%JAVA_HOME%\bin\java -cp "%CLASSPATH%\;ODATA4J_JAR% SecureView
```

Optionally, one can run “build.bat” and “run.bat” under %SYBASE%\ODATA-16_0\samples\java to compile and run Java file respectively.

Step 6: Running ODATA JAVA Consumer Client

SecureView Java ODATA consumer client gets JSON view of ODATA for getting all titles of type "psychology". Equivalent SQL Query is:

```
select pub_id,total_sales,title_id,type,title from titles where type='psychology' OR
https://localhost:8090/odata/titles?$filter=type%20eq%20'psychology' in web browser or following cURL command
```
curl -k --user "httpAuthUser2:password2" https://localhost:8090/odata/titles?$filter=type%20eq%20'psychology'  

One can verify by running exact same query in iSQL or other SQL query tool that results from both select pub_id,total_sales,title_id,type,title from titles where type='psychology' and https://localhost:8090/odata/titles?$filter=type%20eq%20'psychology' are identical.

Database response from iSQL Query:

```
1> select pub_id,total_sales,title_id,type,title from titles where type='psychology'  
2> go
```

```
Msg 207, Level 16, State 4: 
Server '0AKNO527588A', Line 1: 
Invalid column name 'pubid'.
```

```
1> select pub_id,total_sales,title_id,type,title from titles where type='psychology'  
2> go
```

```
<table>
<thead>
<tr>
<th>pub_id</th>
<th>total_sales</th>
<th>title_id</th>
<th>type</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>0877</td>
<td>375</td>
<td>PS1372</td>
<td>psychology</td>
<td>Computer Phobic and Non-Phobic Individuals: Behavior Variations</td>
</tr>
<tr>
<td>0736</td>
<td>2045</td>
<td>PS2091</td>
<td>psychology</td>
<td>Is Anger the Enemy?</td>
</tr>
<tr>
<td>0736</td>
<td>111</td>
<td>PS2106</td>
<td>psychology</td>
<td>Life Without Fear</td>
</tr>
<tr>
<td>0736</td>
<td>4072</td>
<td>PS3333</td>
<td>psychology</td>
<td>Prolonged Data Deprivation: Four Case Studies</td>
</tr>
<tr>
<td>0736</td>
<td>3336</td>
<td>PS7777</td>
<td>psychology</td>
<td>Emotional Security: A New Algorithm</td>
</tr>
</tbody>
</table>

(5 rows affected)

Figure 7: iSQL output

Figure 8: ODATA Consumer output using Java Client

```
c:\Sybase\15\sql\samples\java\Z\JAVA_HOME\bin\java -cp "%CLASSPATH%;ODATAJ_HAR% SecureView
Warning: URL Host: '04K9N8527588A' does not equal '04K9N8527588A'
pubID Sales title id book title
0877 375 PS1372 Computer Phobic and Non-Phobic Individuals: Behavior Variations
0736 2045 PS2091 Is Anger the Enemy?
0736 111 PS2106 Life Without Fear
0736 4072 PS3333 Prolonged Data Deprivation: Four Case Studies
0736 3336 PS7777 Emotional Security: A New Algorithm
```

Figure 8: ODATA Consumer output

**CONCLUSION AND NEXT STEPS**

Running this built in ODATA Producer sample verifies that Sybase ASE is setup correctly for ODATA and now one expose own data and start building applications. We have used database authentication in this example, but that option need not be set, like in .NET sample, and one can view response directly over HTTP instead of HTTPS. 

**Advanced:** One can also use some other webserver instead of embedded HTTP server, details are provided at Sybase ASE documentation. Sybase ASE at this point, supports only v2 version of ODATA specification.