SJS Solutions

Optymyse

ODBC Connector for Optymyse 5.1.x

Contents

1 Introduction

The ODBC Connector for Optymyse 5.1.x allows you to connect *any* data source with an ODBC driver to an Optymyse 5.1.x installation (Cloud or On Premise).

For example, you may have data in any one of the following:

- SQL Server, Oracle or MySQL database
- MS Excel spreadsheet
- SalesForce.com
- etc.

The ODBC connector allows you to specify an ODBC connection using either a defined DSN, or by specifying the full connection details in the configuration. You may also specify the query to execute on that ODBC connection - this will usually be a SELECT query of some kind; the actual content and format of the query will be dependent on the data source.

A full description of creating ODBC DSNs and queries is beyond the scope of this article; we will use, as an example, a SQL Server database, and a Microsoft Excel spreadsheet as our data sources.

2 Installation

Installing the ODBC Connector is simple; download the connector setup executable by clicking the link provided in the e-mail you were sent when you purchased this connector.

Start the installation by double-clicking the executable file. After accepting the license terms & conditions, you will be asked some questions about your Optymyse system. In order to answer these questions, you will need to log in to the Optymyse director:

Value	Description
Connector Name	You should provide a unique, and ideally short, name for this connector. For example, "ODBC" is usually fine. If you wish to run multiple copies of the same connector (e.g. you have a source on a different site), each one should have its own distinct name, so you can identify the true source of the data.
API KEY	The API Key can be found on the Configuration page of Optymyse Director. Copy/paste or re-type the key shown on that page into this box.
Secret Key	This can be found in the same place. Note that it is case-sensitive and must not contain any spaces, punctuation or any other special character. As it's quite long and random, I would strongly suggest cut & pasting this value.

Value	Description
Datapool Address	The Optymyse Datapool is the interface which stores the real-time data items which Optymyse can report. For an on-premise system, the default port for direct access to the Datapool is 8080. So, for example, if you are installing the connector on the Optymyse server, the Datapool address will be: http://localhost:8080/. If you are installing the connector on a remote machine, and port 8080 is accessible, then replace "localhost" with the DNS name or IP address of the Optymyse server (e.g. http://10.0.0.5:8080/) If port 8080 is not available; or you are installing the connector to send data to a cloud instance of Optymyse, then you will use the EDGE service instead. This is listening to port 80, and will act as a "middle man" between the connector and the Datapool services. e.g. to install on-premise where Optymyse is on a different machine, you might use http://optymyse_server.corporatenetwork.local/) To install the connector to a Cloud Optymyse instance, use either of the following URLs (substituting <my_company_name> as require http://<my_company_name>.optymyse.com https://<my_company_name>.optymyse.com

If you already have a PBX connector from SJS, you will note that the installation procedure is very similar.

Once you've answered the basic questions, the connector will install itself. You may be prompted to re-boot the server at the end of the process; this step is optional - however, if you click "No" (or don't get asked), the connector will have to be started manually.

Upon starting the connector service, it will attempt to register with the Optymyse details you have provided; it will then create a default configuration which you must edit in order to start using the connector.

3 Configuration

Configuration is handled via the Optymyse Director. Login to Optymyse, navigate to Configuration, and then to the Connectors sub-page. Find the ODBC connector on the list and click it's name (or hover over it and select Edit). This will open up the configuration "file" for viewing and editing. At this point, I would suggest copying the entire text, paste into NotePad (or a 3rd party equivalent such as NotePad++ or TextPad), and save it.

The configuration file has two main areas: The main connector configuration information, and a section for each feed that is configured. Note that each defined feed consumes one licence from your Data Sources licence count. The configuration file is a JSON format text file. JSON has a specific structure (you can read about it here, for a more formal definition see json.org), the important bit to remember being that each configuration setting name is followed by a colon, and the setting value.

ⓘ JSON particulars

All JSON labels (configuration names in this case) are surrounded with double-quote characters - "

e.g. "ConnectorName", "Version", etc.

A label (e.g. "Feeds") which can contain many items, has square brackets around its value(s), these are arrays in JSON terminology.

Each value in an array is, normally, surrounded by curly brackets - { and }

Each "value" which is defined with curly brackets may have several actual settings inside it - including more arrays (surrounded with [and] as before), and possibly more curly-bracket objects.

The default config file is pre-formatted in a way which makes the file fairly easy to read. However, if you are not confident with JSON, you should ask your IT department for assistance with the configuration.

3.1 Connector Settings

These settings apply to the connector as a whole:

Setting Name	Default Value	Description
Version	1	The version number of the configuration file. This may be incremented in future versions. It is not currently used, except to display in Optymyse.
ServiceID	"" (blank string)	Not currently used - this setting is reserved for future use and should be left blank
TransmitEnable	true	Set to false to turn transmission off. Only useful for debugging purposes. Note "true" or "false" are not in quotes, and are all lower case.
TransmitInterval	1000	No. of milliseconds between transmissions to Optymyse. Set this value higher if you have a high-volume call centre generating lots of traffic. Note that the default configuration has this setting right at the end of the file.
Feeds	See below	An array of feed objects. To add a new feed, duplicate an existing section and alter the settings as needed.

3.2 Feed Settings

These settings apply to each feed you wish to register.

Setting Name	Default Value	Description
FeedName		This is the name the feed will be given. It must be unique within this connector, and it should be as short as is practical. We do not recommend using spaces in the feed name.
FeedActive	true	If true, this feed will be enabled and run; otherwise it will not.

Setting Name	Default Value	Description
ODBCConnectionString		The code required to connect to the ODBC data source. This might be as simple as DSN=pre-saved-dsn-file.
ODBCQuery		The query that will be run to retrieve data to this feed.
QueryIntervalMsec	2500	The time to wait between query runs, in milliseconds. e.g. 2500 will run the query every 2,5 seoonds
CloseConnectionBetweenReads	false	Designed for file-based ODBC data sources (e.g. MS Excel spreadsheets), setting this value to true will close the file - thus allowing updates - between queries. If a query cannot proceed due to file access errors (e.g. the file is in use by another application), the update is missed and the previous figures continue to be reported.
Fields	See below	An array of fields. The number of fields defined in the feed should match the number of fields returned in the feed query.

3.3 Field Settings

Each field in an Optymyse feed, is populated from the data source whenever the query is run. Fields have a number of settings of their own which affect their behaviour:

Setting Name	Default Value	Description
FieldID		A numeric ID to uniquely identify this field in the array. For some connectors, this corresponds to the field's position in the source data; the ODBC connector uses SourceFieldName instead

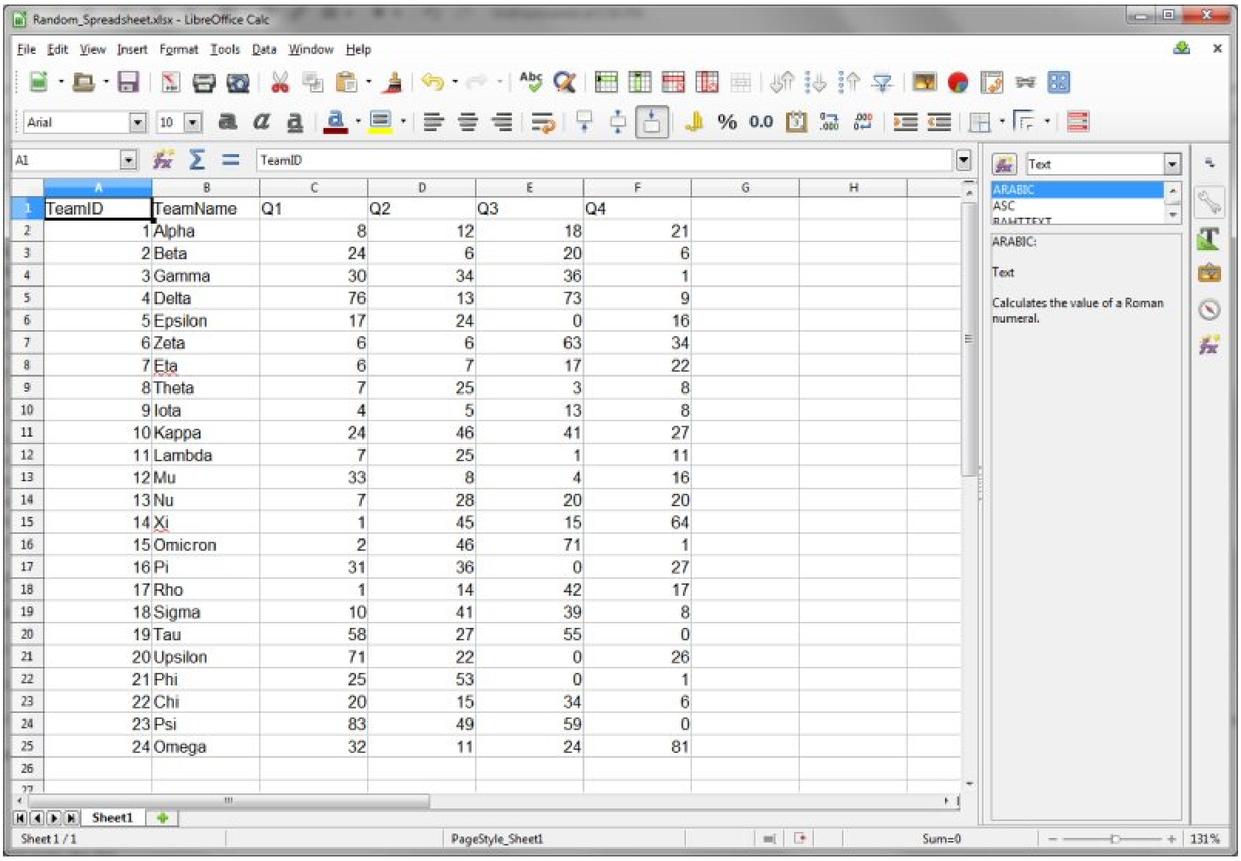
Setting Name	Default Value	Description
SourceFieldName		The name of this field in the data source. Since all queries will return named fields, this should correspond to the name of one of the selected data source fields
OutputName		The name of this field as presented to Optymyse; e.g. if you wish to change its name to use a more business-friendly term.
FieldType		One of: "KEY", "NAME", "DATA": A KEY field provides the unique identifier for this data set. It is usually the primary key field in a database. More than one key field may exist, in which case they are combined in the appropriate sort order - see below for more info A NAME field provides the visible value which will appear in a drop-down field. If no NAME field is present, the connector will use KEY as NAME instead. One or both of KEY and NAME fields must be present in the field list A DATA field, as the name suggests, contains data.
KeyPriority	0	A simple number - if multiple KEY fields are provided, the order in which they appear in the primary key can be set here.
DataType	INT	One of the following: • INT - An integer number between 0 and 2.4bn (approx) • FLOAT - A floating point number between approx. +/- 1x10e127 (many many trillions) • PCT - A percentage-formatted floating point number. • INTTIME - An integer value in the data source, which should be displayed in Optymyse as a time span. e.g. 97 becomes 1:37. Ideal for state times, etc. • STARTTIME - A date/time value in the data source, which represents the start of a date range (e.g. state start time). Optymyse converts this to a number of seconds, then treats like INTTIME above • STRING - A plain text value

Setting Name	Default Value	Description
		- TIMESTAMP - Displayed as a time on the wallboard
AggregateType		AggregateType is used when building supergroups, and determines how the data item(s) are aggregated together. Note that KEY and NAME fields should be given the type KEY: - "KEY" - Indicates this is a KEY or NAME field. Any supergroup will be constructed using these fields to locate values in the main list. - "AVG" - Take the average of the field values, and return it. See notes, below - "MIN" - Takes the smallest field value from the aggregated grouo - "MAX" - As above for the largest value. - "FIRST","LAST - Take either the first item in the list, or the last. - "SUM" - Add all the values together and return that total.
SetFunction		Reserved for future use
GetFunction		Reserved for future use
Cumulative		Not used in this connector. Set to true or false (all lower case, no quotes)
ValueIfMissing		If the data source returned a NULL; or there is no value for a given item. Values must be supplied quoted, even if they are numbers (e.g. "1.5").
IncrementTime		For time-based fields, tell Optymyse to automatically increment the value in the absence of new or changed information. e.g. agent state times where the state time is supplied as an INT value rather than a state start time value.
Format		

Setting Name	Default Value	Description
		For any field, when it is passed to Optymyse, it can be formatted using this value. See below for more information.

3.4 Sample Configuration

As an example, lets say we have an Excel spreadsheet that looks like this:


Random_Spreadsheet.xlsx - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Arial
10
A1
TeamID
Text
ARABIC
ASC
Sheet1
Sheet 1 / 1
PageStyle_Sheet1
Sum=0
131%

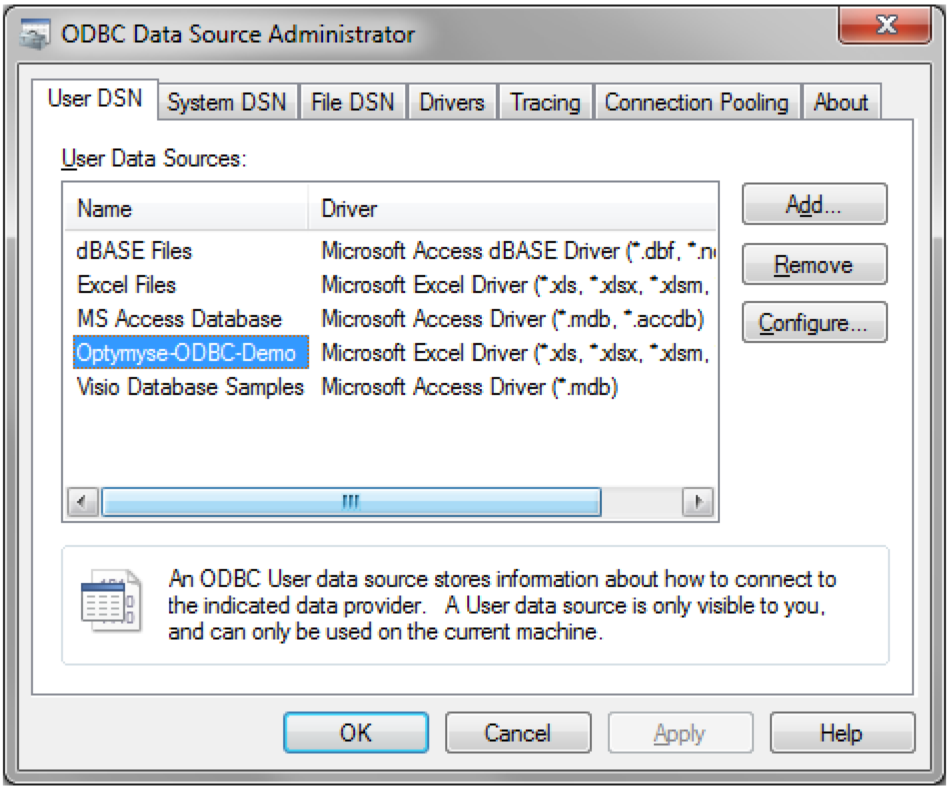
TeamID	TeamName	Q1	Q2	Q3	Q4
1	Alpha	8	12	18	21
2	Beta	24	6	20	6
3	Gamma	30	34	36	1
4	Delta	76	13	73	9
5	Epsilon	17	24	0	16
6	Zeta	6	6	63	34
7	Eta	6	7	17	22
8	Theta	7	25	3	8
9	Iota	4	5	13	8
10	Kappa	24	46	41	27
11	Lambda	7	25	1	11
12	Mu	33	8	4	16
13	Nu	7	28	20	20
14	Xi	1	45	15	64
15	Omicron	2	46	71	1
16	Pi	31	36	0	27
17	Rho	1	14	42	17
18	Sigma	10	41	39	8
19	Tau	58	27	55	0
20	Upsilon	71	22	0	26
21	Phi	25	53	0	1
22	Chi	20	15	34	6
23	Psi	83	49	59	0
24	Omega	32	11	24	81

ARABIC:

Text

Calculates the value of a Roman numeral.

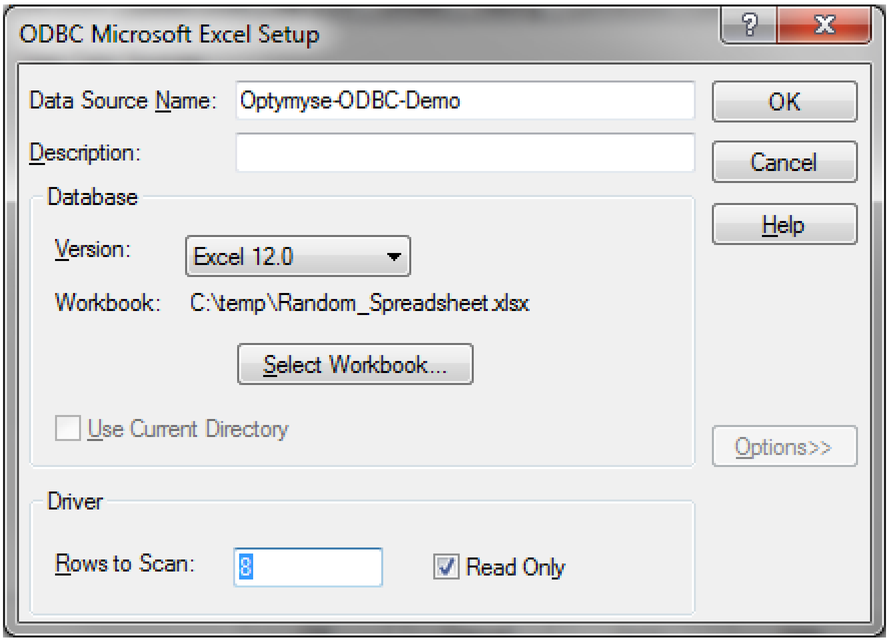
We use the ODBC Driver tool to create an ODBC DSN to this spreadsheet:



User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.n
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm,
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
Optymyse-ODBC-Demo	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm,
Visio Database Samples	Microsoft Access Driver (*.mdb)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.



Data Source Name: Optymyse-ODBC-Demo

Description:

Database

Version: Excel 12.0

Workbook: C:\temp\Random_Spreadsheet.xlsx

□ Use Current Directory

Driver

Rows to Scan: 8

☑ Read Only

The configuration file used to read this spreadsheet (or, rather, Sheet1 within the workbook) is as follows:

Setting	Notes
{	All configuration files must start with a { character
"Version": 1,	Configuration format (version 1 in this case)
"ServiceID": "",	Reserved for future use
"TransmitEnable": true,	Debugging setting - ignore.
"TransmitInterval": 1000,	Number of mSec between data transmissions to Optymyse
"Feeds": [	Start of feeds array
{	Start of first feed object in feeds array
"FeedName": "XL_TeamSales",	Name of this feed
"FeedActive": true,	true = feed will be registered and connected; false = feed is inactive
"ODBCConnectionString": "DSN=Optymyse-ODBC-Demo",	The properly formatted ODBC connection parameters; in this case, just the DSN name is needed
"ODBCQuery": "SELECT * FROM [Sheet1$]",	The query to run on the ODBC data source (formatted according to the data source requirements)
"QueryIntervalMsec": 10000	The frequency with which to run the query in milliseconds. 1,000 is the minimum, 86,400,000 is the maximum (corresponding to 1 second to 24 hours)
"CloseConnectionBetweenReads": true,	For file-based data sources, close the connection between reads. This will allow the file to be updated (otherwise a read/write lock is held) This setting can be set to false for database-based data sources, which will reduce resource use on the client computer.

Setting	Notes
"Fields": [	Start of the array of fields we expect to retrieve from this data source
{	Each field is an object, so must start with a { character.
"FieldID": 1,	The ID of this field
"SourceFieldName": "TeamID",	The name of this field, as exposed by the data source (in this case, the contents of cell A1)
"OutputName": "TeamID",	The name of this field, as Optymyse will see it. Leave blank ("OutputName":"",) to use the Source Field Name.
"FieldType": "KEY",	Type of field
"KeyPriority": 0,	As this is the only key field, it has a priority of zero.
"DataType": "INT",	Integer data in this field
"AggregateType": "KEY",	This field is the primary key
"SetFunction": "",	Reserved for future use
"GetFunction": "",	Reserved for future use
"Cumulative": false,	Cumulative stats are not supported by the ODBC driver
"ValueIfMissing": "",	If the database returns NULL, or this column can't be found in a spreadsheet row, substitute the value in this field. Note: For KEY or NAME field types, a missing value will cause the row to be ignored.
"IncrementTime": false,	For INTTIME fields, the connector will automatically increment the duration, if this is set to true
"Format": null	

Setting	Notes
	Use this setting to format the output of the field; e.g. for percentages, the format "#0.00%" will output numbers like 93.20%
},	End of this field object. The comma indicates another field will follow it.
{	Start of the next field...
"FieldID": 2,	
"SourceFieldName": "TeamName",	
"OutputName": "TeamName",	
"FieldType": "NAME",	This field is a NAME type; the contents will be used to select the data item in Optymyse.
"KeyPriority": 0,	KeyPriority is ignored for all non-KEY fields.
"DataType": "STRING",	
"AggregateType": "FIRST",	
"SetFunction": "",	
"GetFunction": "",	
"Cumulative": false,	
"ValueIfMissing": "",	
"IncrementTime": false,	
"Format": null	

Setting	Notes
},	
{	
"FieldID": 3,	
"SourceFieldName": "Q1",	
"OutputName": "Q1",	
"FieldType": "DATA",	This is our first DATA field, i.e. field containing actual figures for the wallboard.
"KeyPriority": 0,	
"DataType": "INT",	
"AggregateType": "SUM",	
"SetFunction": "",	
"GetFunction": "",	
"Cumulative": false,	
"ValueIfMissing": "",	
"IncrementTime": false,	
"Format": null	
},	
{	
"FieldID": 4,	

Setting	Notes
"SourceFieldName": "Q2",	
"OutputName": "Q2",	
"FieldType": "DATA",	
"KeyPriority": 0,	
"DataType": "INT",	
"AggregateType": "SUM",	
"SetFunction": "",	
"GetFunction": "",	
"Cumulative": false,	
"ValueIfMissing": "",	
"IncrementTime": false,	
"Format": null	
},	
{	
"FieldID": 5,	
"SourceFieldName": "Q3",	
"OutputName": "Q3",	
"FieldType": "DATA",	
"KeyPriority": 0,	

Setting	Notes
"DataType": "INT",	
"AggregateType": "SUM",	
"SetFunction": "",	
"GetFunction": "",	
"Cumulative": false,	
"ValueIfMissing": "",	
"IncrementTime": false,	
"Format": null	
},	
{	
"FieldID": 6,	
"SourceFieldName": "Q4",	
"OutputName": "Q4",	
"FieldType": "DATA",	
"KeyPriority": 0,	
"DataType": "INT",	
"AggregateType": "SUM",	
"SetFunction": "",	
"GetFunction": "",	

Setting	Notes
"Cumulative": false,	
"ValueIfMissing": "",	
"IncrementTime": false,	
"Format": null	
}	End of object; note, no comma means this is the last defined field in the list
]	The square bracket closes the list off.
}	End of feed object. No comma means this is the last (only) defined feed in this configuration file.
]	The square bracket indicates the end of the array of feeds.
}	The file must end with a close curly bracket; This completes the configuration object

The feed has two settings of great interest:

"ODBCConnectionString": "DSN=Optymyse-ODBC-Demo"

There are numerous ways of creating an ODBC DSN (Data Source Name), too many to cover in this document. The pictures above show one way of creating a "User DSN". It is also possible to create a DSN without any specific configuration. For example, the following string connects to a SQL Server database:

> *"ODBCConnectionString": "Driver=SQL Server;Server=(local)\\Optymyse; Database=odbctest;UID=odbcuser;PWD=odbcpass"*

ⓘ Important Note

> Within JSON strings, there are two characters which must be "escaped" - i.e. have a backslash added in front of them. The first is the double-quote symbol; so if you needed to put the string "test" inside of a JSON variable, you would actually write \" test\"
>
> The second character that must be escaped is the backslash itself: \ becomes \\. In the string above, I connect to a named instance of SQL Server on the local machine - (local)\Optymyse. In order to correctly set this, I must type (local)\\Optymyse. The "extra" backslash is automatically removed when the JSON is parsed.
>
> If you get this wrong - or, indeed, you miss any of the required punctuation - the connector will fail to start with an invalid configuration error. This will be recorded in the log file, so long as logging is not set to NONE.

The ODBCQuery value is now a straightforward TransactSQL query:

"ODBCQuery": "SELECT * FROM teamSales"

Note that I have to include the database name (odbctest); the double-dot in between is shorthand for .dbo., a SQL Server idiosyncrasy.

Assuming I have a table called teamSales, in my odbctest database, with the fieldnames TeamId, TeamName, Q1, Q2, Q3 and Q4; that configuration will now return data from the database.

4 Other ODBC data sources

You can connect to any ODBC data source for which you have a driver. A typical Windows 7 installation will include MS Access, Paradox, FoxPro, MS Excel, SQL Server and Oracle. MySQL drivers can be downloaded for free, and an every increasing number of Cloud information providers (e.g. Salesforce.com) offer either free or paid-for ODBC drivers.

Covering the connectivity and querying of all of the ODBC drivers available is beyond the scope of this document, please contact your ODBC vendor for assistance; show them this document and they should be able to guide you towards creating a suitable DSN or connection string, and query.