

INTERACTIVE EDGE

DataDefractor™ SSIS User's Guide

A guide to setting up and using DataDefractor SSIS
v1.1

This document is an introduction to DataDefractor SSIS 1.1 – an SQL Server 2005 Integration Services Data Flow Source Component designed to extract and normalize data from semi-structured data sources. It assumes you are familiar with SQL Server 2005 Integration Services. It also assumes you have working experience with Visual Studio® and have conceptual understanding of data warehousing.

Table of Contents

About DataDefractor.....	5
Introduction	5
Brief History	5
Product Overview	5
Terminology	6
Installing DataDefractor	8
Introduction	8
Before You Install	8
Operating System.....	8
SQL Server Integration Services	8
Microsoft .NET Framework	8
Hardware	8
Installing DataDefractor	9
Removing old versions of DataDefractor from SQL Server Business Intelligence Development Studio’s Toolbox.....	10
Adding DataDefractor to SQL Server Business Intelligence Development Studio’s Toolbox	11
Understanding DataDefractor Licensing.....	13
Using DataDefractor License Manager	14
Silent Installation	18
Silent Activation	19
DataDefractor Version	19
32-bit and 64-bit Platform Support	19
Uninstalling DataDefractor	20
Understanding Semi-structured Data	21
Introduction	21
Fact-based Semi-structured Data Sources.....	21
Annual Sales Data Report	22
Freddie Mac Mortgage Margins Report.....	29
Freddie Mac Mortgage Series Report (multi-page report).....	34
Conclusion.....	37

Using DataDefractor	38
Introduction	38
Processing Workflow	38
Normalization Procedure	38
Creating a Connection Manager	40
DataDefractor Excel File Connection Manager	40
DataDefractor Text File Connection Manager	42
ODBC Connection Manager	43
Starting the DataDefractor Wizard	43
Choosing a Connection Manager	44
Defining the Data Source Layout	45
Header/Footer Layout Feature	46
Vertical Sub-Pages Layout Feature	47
Horizontal Sub-Pages Layout Feature	49
Fixed Width Sub-Pages Layout Feature	50
Defining the Fact Area	51
Defining the First Fact Row	52
Defining the Fact Source Columns	53
Defining the Dimensional Model	55
Axes List.....	55
Name Sources List and Name Source Origin Definition.....	56
Output Format	57
Exclude Members.....	58
Trim White Space.....	58
Skip Empty Member Names.....	58
Name Source Properties	58
Setting up the Measures.....	61
SSIS Outputs.....	62
Facts Output.....	62
Dimension Outputs	64
Outputs Usage Example.....	65

Acknowledgements..... 65
Bibliography 66

About DataDefractor

Introduction

This chapter touches on the history of DataDefractor, its purpose and main usability patterns. In addition, it expands on the fundamental terminology used throughout this document.

Brief History

DataDefractor inherits its functionality from XP3[®] Dimensional Data Loader – another product by Interactive Edge. XP3 Dimensional Data Loader is part of a larger system – XP3 Data Warehouse – and its purpose is to extract fact data and contextual metadata from various enterprise semi-structured data sources and load it into XP3 multidimensional models (relational star schemas). These multidimensional models are exported to Microsoft[®] Analysis Services in the form of OLAP cubes; the data is then analyzed and presented through various OLAP front-end applications.

XP3 Dimensional Data Loader is still being developed in parallel with DataDefractor, but the two products have parted ways and aim at two very different audiences. DataDefractor targets the developer working on SQL Server-driven data integration projects, while XP3 Data Warehouse targets the business intelligence (BI) manager or data analyst in need of full-scale integrated BI solution.

DataDefractor does not target any market sector in particular. Its usability spans many different industries and government agencies. Any SSIS data integration project, which involves extracting fact-based data from semi-structured sources, may derive some benefit from it.

Product Overview

DataDefractor is a custom Microsoft SQL Server 2005 Integration Services (SSIS) data flow source component designed to normalize semi-structured data sources such as Microsoft Excel[®] or CSV reports and spreadsheets. Once the data is extracted, DataDefractor normalizes it and feeds it into the SSIS pipeline in the form of normalized output of factual data and contextual metadata.

With the help of its example-driven wizard user interface you can map-out the rules that define the model of your semi-structured spreadsheets. During runtime, DataDefractor will use these rules to extract the data, normalize it and send it to the SSIS pipeline. The extraction rules are stored with the metadata of your SSIS package and can be reused and modified via DataDefractor's wizard user interface. Or if you need to modify the behavior of DataDefractor on the fly, you can access its properties via SSIS scripting or SSIS expressions.

Typical use-case scenarios for DataDefractor include extracting and normalizing fact and dimensional data from semi-structured enterprise Excel spreadsheet reports, Excel PivotTable[®] reports, flat files, text files and CSV files and feeding it into an enterprise data warehouse, a data mart, directly into a Microsoft SQL Server Analysis Services OLAP cube or any destination or transformation supported by SQL Server 2005 Integration Services.

The main features of DataDefractor include:

- **Automatic spreadsheet header/footer discovery**
DataDefractor includes algorithms for automatic discovery of headers and footers with flexible number of rows.
- **Normalization of nested multi-paged data sources**
Complex and deeply de-normalized worksheets with multi-level cascading structures can be mapped out and normalized without writing one line of code.
- **Flexible fact region definitions**
In many cases semi-structured data sources with a common structure have subtle differences – for example two worksheets with the same structure may have different number of columns. DataDefractor provides the user with flexible rules that capture the variable structure of a semi-structured data source. These rules can be reused repetitively to normalize other data sources with the same flexible structure.
- **Regular-expression-based context extractions**
The product features powerful extraction mechanism based on regular expressions.
- **Multi-data source processing**
DataDefractor's normalization engine can process many data sources at once as long as they observe a common structure.
- **Multidimensional model**
The output of DataDefractor is a full-blown multidimensional model which can be loaded directly into a decision support systems such as a data mart, enterprise data warehouse or an OLAP cube.

Terminology

To be able to successfully use DataDefractor it is important to understand the terminology associated with the product and the problems it is designed to solve. The terms which are fundamental for understanding how DataDefractor works revolve around the type of data it is designed to extract. First and foremost, the data targeted by DataDefractor is *factual*.

According to the Compact Oxford English Dictionary *facts* are “*information used as evidence or as part of a report*”. This information may represent measurements taken while monitoring series of events. The sales transactions recorded by a cash register or the air temperatures measured by an atmospheric sensor at a weather station are both examples of factual data.

Factual data is usually numeric, but may also be of any other type like text or date. For example, the respondents' answer to an open-ended survey question is a textual fact.

Regardless of the data type, facts are usually *cardinal* – they can be compared against each other, i.e. the distance between any two fact values can be measured. In rare cases, the facts are just *ordinal* – they cannot be compared quantitatively, but can be positioned relatively to each other.

Factual information sometimes appears as an attribute of more than one entity. For example, the price of a product may vary based on the stores it's being sold at. The particular price of a product at a particular store is a fact associated with both the product and the store where it is being sold.

Be it a measurement or a varying attribute, the factual information is usually associated with some kind of *context*. For example, in the case of a sales transaction recorded by a cash register, the date and time of the sale, the cash register ID and the store where the cash register was located are all part of the context for this transaction. The exact point in time and the location of the weather station where the air temperature was measured both represent some context for the temperature measurement. In the case of a varying product price, the product and the store it's being sold at are both context for the price.

Another aspect of the data extracted from DataDefractor is its structure. The product can extract factual and contextual information from both *structured* and *semi-structured* data sources. The real strength of the product is in its ability to extract data from complex semi-structured data sources; however since structured data is simpler to define and process, we will start with it.

The term *structured data* refers to data which originates from a relational source. Structured data is contained within or represented by a relational data object – a table or a view. The formal definition of a relational data object includes a set of fields. These fields are ordered and named and each of them has an explicit data type. A numeric data field cannot contain textual information and vice versa. Each relational data object contains zero or more records of data. Each record contains the same exact number of fields ordered according to the formal definition of the relational data object. DataDefractor can extract factual and contextual information from structured relational data sources and could be useful in case you need to normalize a structured but de-normalized data source.

However, *semi-structured* data is really what DataDefractor is all about. The term semi-structured refers to data which is not stored in a relational database, but still maintains a certain level of structure. An example of a semi-structured data is a financial report stored in an Excel workbook. It may contain headers full of contextual information; then it may contain a set of nested sub-pages laid out so that they please the human eye or are easy to print; the sub-pages may contain nuggets of facts wrapped by shells of contextual information. This type of data does have structure to it, although not as simple as a relational data object with its set of strongly-typed named and ordered data fields.

Additionally, the rules that govern the structure of a complex semi-structured data source are usually external to it. As opposed to structured data sources, which contain their own schema and are fairly self-descriptive, the semi-structured data sources subject to schemas stored and described elsewhere. These schemas express flexible data sources whose form varies over time.

It is the purpose of DataDefractor's wizard-driven user interface to help you define and map-out the structural patterns of such a semi-structured data source. Once the structure is identified, its rules are stored and re-used by DataDefractor to extract and normalize the data captured within any data source that observes this same structure.

Installing DataDefractor

Introduction

In this chapter you will learn about the prerequisite software required by DataDefractor’s installation as well as how to install DataDefractor and how to activate it for use in both development and production environments. You will also be introduced to the variety of license models supported by DataDefractor.

Before You Install

Operating System

DataDefractor supports the following operating systems:

- Windows® XP SP2
- Windows Server® 2003 SP1
- Windows Server 2003 R2
- Windows 2000 SP4

SQL Server Integration Services

DataDefractor installation requires SQL Server 2005 Integration Services SP1 or later to be deployed on the target system prior to running the installation. The following SQL Server deployment configurations are supported by DataDefractor:

- SQL Server 2005 SP1 with SSIS and Business Intelligence Development Studio
- SQL Server 2005 SP1 Client Components with Business Intelligence Development Studio (client components only)
- SQL Server 2005 SP1 with SSS (server components only)

DataDefractor supports the following SQL Server 2005 editions:

- Enterprise
- Standard
- Developer

Note: The list of supported SQL Server editions also depends on the DataDefractor license activated on the target system (for more information see section “**Understanding DataDefractor Licensing**”).

DataDefractor supports both the 32-bit and 64-bit editions of SQL Server 2005.

Microsoft .NET Framework

DataDefractor is implemented with Microsoft .NET Framework and requires .NET 2.0 Runtime Environment to be deployed to the target machine in order to function correctly. There is no need to explicitly deploy .NET 2.0 – it is installed by the installation of SQL Server 2005.

Hardware

DataDefractor requires the following hardware configuration:

- 600 MHz Pentium III-compatible or faster processor; 1 GHz or faster processor is recommended.
- 512 MB of RAM or more.
- 50 MB of available hard disk space is recommended.
- Super VGA (1,024x768) or higher-resolution video adapter and monitor.

Installing DataDefractor

DataDefractor is distributed as a standard Microsoft Installer installation package (MSI file). The product must be installed on the target system before it can be used to develop SSIS packages.

You can download and install a fully functional version of DataDefractor for a trial period of 14 days. During and after the trial period the following dialog box will appear every time you start DataDefractor:



During the trial period you will be able to press “**Try DataDefractor**” and evaluate the features of the product. If you have activation key you can activate the product by pressing “**Activate DataDefractor**”. That will invoke the **DataDefractor License Manager** where you will be able to activate the product (see below). You can also navigate to the DataDefractor website to purchase an activation key by pressing the “**Purchase activation key**” button.

Follow these steps to install DataDefractor:

1. Run DataDefractor.msi on the target system; this will launch DataDefractor’s installation procedure.
2. Click “**Next**” on the Welcome page.
3. Read DataDefractor End User Agreement and if you agree with it, select “**I accept the terms in the license agreement**”; Click “**Next**”.

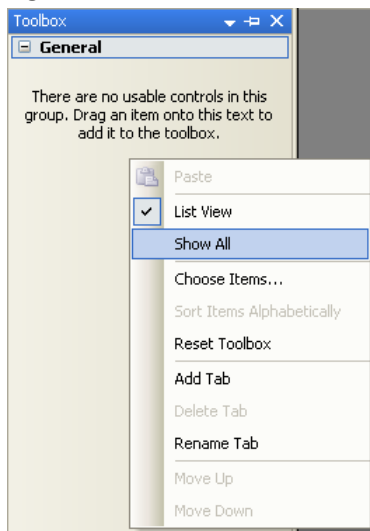
4. In case you need to deploy DataDefractor to a folder different than the default one, select “**Custom**” and click “**Next**”, then change the destination by clicking “**Change**”; In case you want to deploy DataDefractor to the default location, select “**Complete**”; Click “**Next**”.
5. Click “**Install**”; this will execute the installation script which will deploy DataDefractor to the target system.

Removing old versions of DataDefractor from SQL Server Business Intelligence Development Studio’s Toolbox

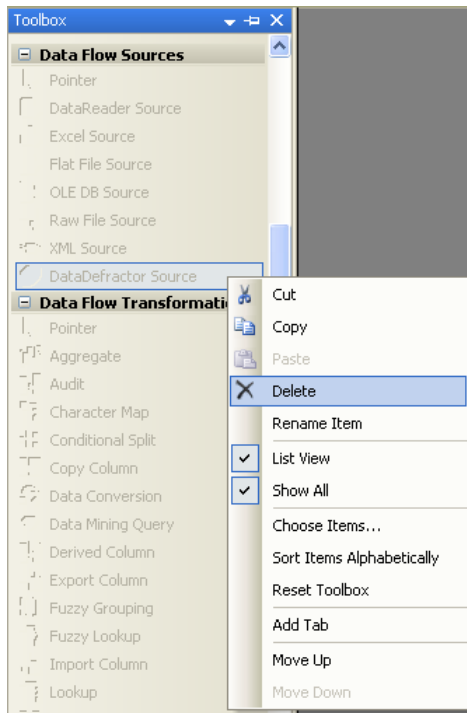
If you have installed previous versions of DataDefractor on your machine and if you have added DataDefractor to the SQL Server Business Intelligence Development Studio’s (BIDS) Toolbox, you have to manually remove the old version of DataDefractor from your toolbox before you add the new one. If this is the first time DataDefractor is being installed on the target machine, skip this section and move on to “**Adding DataDefractor to SQL Server Business Intelligence Development Studio’s Toolbox**”.

Follow these instructions:

1. Launch SQL Server Business Intelligence Development Studio.
2. Right-click on the toolbox and select “**Show All**”:



3. Locate “DataDefractor Source” in section “Data Flow Sources”, right-click on the component and select “Delete”:



4. Click “OK” to remove the previous version of DataDefractor from the toolbox.

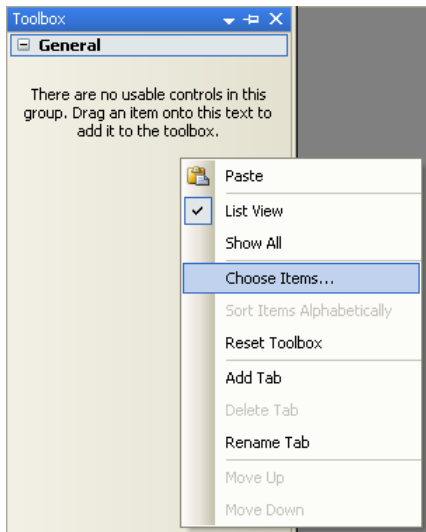
Adding DataDefractor to SQL Server Business Intelligence Development Studio’s Toolbox

After DataDefractor is installed, you need to add it to SQL Server Business Intelligence Development Studio’s (BIDS) Toolbox in order to use it while developing SSIS packages. At the time of writing, Microsoft has not provided a programmatic interface to automatically add components to BIDS’s toolbox, so you have to perform this procedure manually.

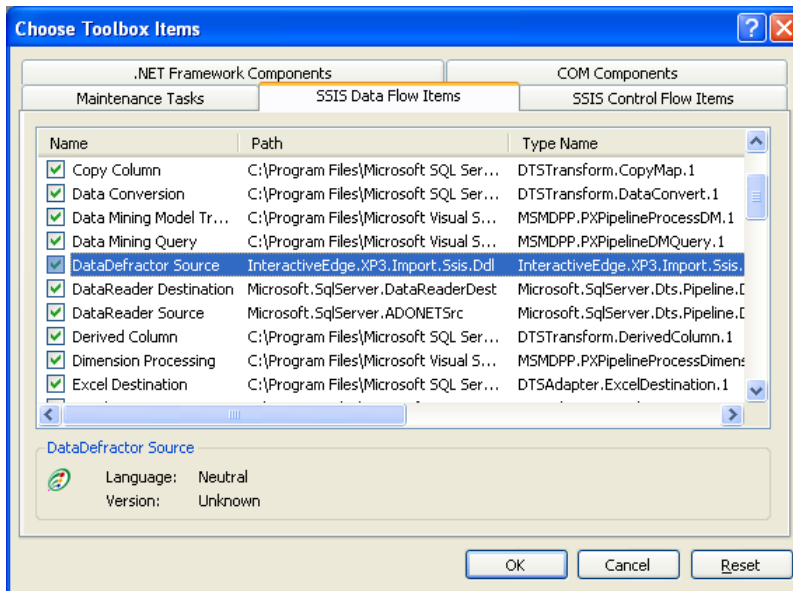
To do this, follow these steps:

1. Launch SQL Server Business Intelligence Development Studio.

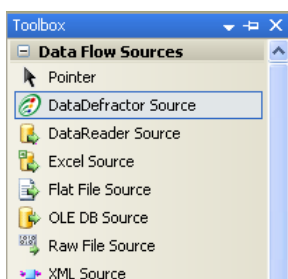
- Right-click on the toolbox and select **“Choose Items...”**



- Navigate to tab **“SSIS Data Flow Items”** and check **“DataDefractor Source”**:



- Click **“OK”**.
- You should see DataDefractor’s icon added to the **“Data Flow Sources”** category in the SSIS toolbox:



- You can start using DataDefractor by dragging and dropping it onto your SSIS Data Flow canvas.

Understanding DataDefractor Licensing

The license activation is a one-time procedure, during which a license certificate is downloaded from DataDefractor Activation Service and stored on the local machine. The activation procedure requires an Internet connection.

Activation

A license is activated with an activation key. You can obtain an activation key from www.datadefractor.com by purchasing a DataDefractor license.

You can activate DataDefractor either via Internet or e-mail. In both cases, you need to use a product activation key. When the product is activated, a DataDefractor license certificate is downloaded and stored on the target machine. Depending on the product activation key you used, the license certificate may enable/disable specific features; may apply to one user only or to all the users of the system; may be perpetual or time-limited.

Every license has a scope. A license could be either system-wide or user-specific. Once activated on a Windows machine, a system-wide license allows every user of this Windows system to use DataDefractor. A user-specific license on the other hand, is only valid for the Windows user account, which it was activated with. An activation key which unlocks a system-wide license is called a system-wide key; a key, which unlocks a user-specific license is called user-specific key.

Each activation key has a maximum number of available activations. Every time a system-wide key is used to activate a license on a new system, the number of available activations associated with this key is decremented. Every time a user-specific key is used to activate a license on a new combination of system and user account, the number of available activations associated with this key is decremented. Once the number of available activations for a key reaches zero, the activation key is exhausted and cannot be used to activate DataDefractor.

System-wide keys are system-locked. The number of available activations for a system-wide key is decremented only if the key has never been used on the target machine before. Once a system-wide key is used on a machine, it can be re-used forever on that machine without decrementing the count of available activations for that key. Even if the machine's hard-drive is reformatted and wiped out clean, an activation procedure with a key, which was used on this machine before, will not count against the maximum number of activations for that key.¹

User-specific keys are system-user-locked. The number of available activations for a user-specific key is decremented only if the key has never been used for this combination of machine and user account before. Once a user-specific key is used on a machine for a specific user account, it can be reactivated

¹ Disclaimer: No personal or identity information is transferred between the client machine and DataDefractor Activation Service. The licensing software collects negligible software and hardware footprint information from the target machine, scrambles it using a one-way hash algorithm and sends the hashed values to DataDefractor Activation Service. The information sent over the wire is mere shadow of the original information and due to the way one-way hashing works the original footprint information can never be recovered from the hashed values stored by DataDefractor Activation Service.

forever on that machine for that same user account without decrementing the count of available activations for that key.

Verification

After a license has been activated on a machine, DataDefractor will verify the license certificate every time it runs. DataDefractor does not need Internet connectivity while verifying your license.

There are two types of DataDefractor licenses – Developer and Production license. The following table lists the differences between the DataDefractor licenses:

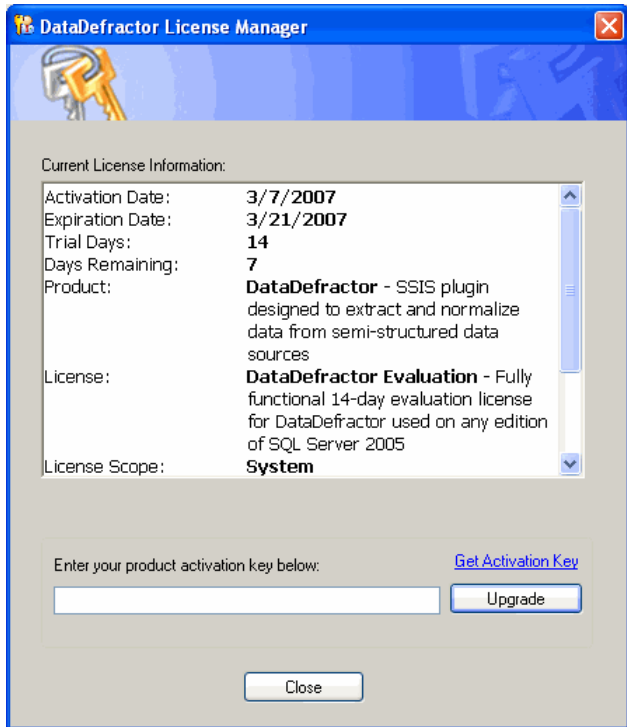
	License Scope	License Term	SSIS Edition
Developer Term	User-specific	1 year	Developer
Developer Perpetual	User-specific	Perpetual	Developer
Production Perpetual	System-wide	Perpetual	Any

A Developer license allows you to use DataDefractor only in the context of SSIS Developer Edition. In addition to this, a Developer license is user-specific. A “Term Developer license” expires 365 days after the activation. Once the term expires, the license needs to be renewed in order to enable DataDefractor. A “Perpetual Developer license” never expires.

A “Production license” allows you to use DataDefractor with any edition of SSIS – Developer, Enterprise or Standard. A Production license is system-wide. Once activated, a DataDefractor Production license can be used simultaneously by any number of users logged onto the target machine. A Production license is always perpetual – once activated, it will never expire.

Using DataDefractor License Manager

DataDefractor License Manager is a standalone application, which helps you manage your current DataDefractor licenses. You can launch DataDefractor License Manager by selecting “**Start ⇒ All Programs ⇒ Interactive Edge ⇒ DataDefractor ⇒ DataDefractor License Manager**”.



If you have an active DataDefractor license on your system, DataDefractor License Manager will display the terms of the license. In addition to this, DataDefractor License Manager will display all the license privileges associated with the license. The scope of the license is also displayed – you can verify if the license is system-wide or user-specific.

If you want to get a new license key, you can click on the link “**Get Activation Key**”.

Multiple License Certificates on One Machine

You can have only one system-wide license activated on a single machine at any given point in time. You can have many user-specific licenses activated on a single machine, but only one per user. System-wide and user-specific licenses can reside on the same machine.

When a user with a user-specific license is logged on to a machine, which also has a system-wide license, the user-specific license overrides the system-wide license for this user.

License Certificate

DataDefractor license certificates consist of the following fields:

Product: The name of the product the license certificate is associated with. This field is always “DataDefractor”.

Activation Date: For system-wide licenses this is the date of the first original activation of this certificate on this machine; for user-specific licenses this is the date of the first original activation of this certificate on this machine for this user.

Expiration Date: If the certificate expires, this field contains the date when the certificate will no longer be valid.

Trial Days: If the certificate expires, this field indicates the number of days between the activation and expiration date.

Days Remaining: If the certificate expires, this field indicates the number of days left before the license expires.

License: The name of the license.

License Scope: Indicates the scope of the license. “**System**” indicates system-wide license. “**User**” indicates user-specific license. Note that if a system-wide and a user-specific license for the currently logged on user are both available on the same machine, DataDefractor License Manager will display only the user-specific license.

Privileges: Indicates the privileges associated with the license. The privileges may be any one of the following:

- Developer – will allow DataDefractor to be executed on SQL Server Developer Edition
- Production – will allow DataDefractor to be executed on SQL Server Enterprise, Standard or Developer edition.

Activating a New DataDefractor License

Follow these steps to activate a DataDefractor license with DataDefractor License Manager:

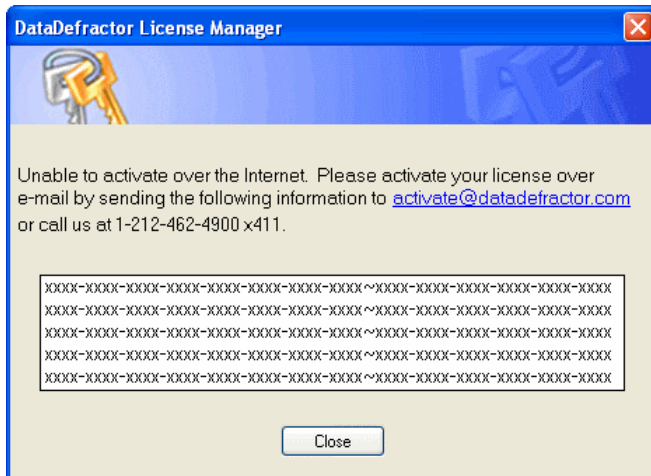
1. Launch DataDefractor License Manager.
2. Type your product activation key in the edit box labeled “**Enter your product activation key below**”.
3. Click “**Activate License**”.
4. DataDefractor License Manager will contact DataDefractor Activation Service – a web-service, which keeps track of all DataDefractor licenses.
5. DataDefractor Activation Service will search for the license associated with your product activation key.
6. In case the product activation key you provided is missing from DataDefractor Activation Service’s databank, or its maximum number of activations has been exhausted, DataDefractor Activation Service will deny activating the license on your machine.
7. If a valid license with unused activations is associated with your product activation key, DataDefractor Activation Service will send back a license certificate, which will be stored on your local system. The license will be either a system-wide (Production) or a user-specific license (Developer).

8. In case a Developer license is activated on a system, which already has an active Developer license for the currently logged-on user, the current Developer license is replaced by the new one. Once replaced a Developer license can always be re-activated on the same machine for the same user – this won't decrement the number of available activations for this license key.
9. In case a Developer license is activated on a system, which has a Production license, the Developer license is activated only for the currently logged on user. The Production license will continue to apply to the rest of the users.
10. In case a Production license is activated on a system which has Developer licenses, the Production license will only apply to the users who don't have Developer licenses. The Production license is a system-wide license that is available to all users of that machine provided they don't already have a Developer license. To upgrade a specific user that has a Developer license to the Production license, you must remove their developer license by deleting the `DataDefractorLicense.xml` file for that user. The license xml files are stored in the folder structure listed below (substitute the appropriate username):
`%Root%\Documents and Settings\username\Application Data\DataDefractor`

Internet Connectivity and DataDefractor License Manager

As stated earlier, DataDefractor License manager needs a live Internet connection to activate a license. It uses the Internet to send a web request to DataDefractor Activation Service.

If no Internet connection is available at the time of activation, DataDefractor License Manager will pop up the following message:



Please copy all the information from the message edit box and send it to activate@datadefractor.com or call us at 212-462-4900 x411. We will generate a license for you and will send the license with instructions where to store it in your system.

Silent Installation

In addition to the regular installation, DataDefractor features a silent installation procedure. It is invoked on the command line of the target machine and installs DataDefractor without user interface interruption. A silent installation is useful when DataDefractor must be deployed as part of another larger-scope deployment routine.

A silent installation procedure is invoked on the command prompt like this:

```
DataDefractor.msi /qn
```

The behavior of the silent installation can be modified through command line parameters. The values of these parameters are passed on the command line like this:

```
DataDefractor.msi /qn PARAM1=VALUE1 PARAM2=VALUE2
```

Here is the list of parameters used to control DataDefractor’s silent installation:

- **INSTALLDIR** – Specifies the full path to the target location where DataDefractor is to be deployed.

Example:

```
DataDefractor.msi /qn INSTALLDIR="c:\MyDataDefractor Target Location"
```

This will deploy DataDefractor to target folder `c:\MyDataDefractor Target Location`.

- **ADDLOCAL** – Overrides the default installation configuration. If **ADDLOCAL** is not specified, the silent installation deploys all DataDefractor’s features – the DataDefractor Source Component and the DataDefractor Documentation. In case you need to override it you need to pass **ADDLOCAL** with the list of features to be deployed.

The list of supported features is:

- **SourceComponent** – includes the DataDefractor SSIS Data Flow Source component and the DataDefractor connection managers.
- **Samples** – includes SSIS packages which demonstrate the use of DataDefractor with sample semi-structured data sources.

Examples:

```
DataDefractor.msi /qn ADDLOCAL="SourceComponent"
```

This will deploy only DataDefractor SSIS Source component and DataDefractor connection managers; will not deploy DataDefractor’s documentation.

```
DataDefractor.msi /qn ADDLOCAL="SourceComponent,Samples"
```

This will deploy DataDefractor SSIS Source component, DataDefractor connection managers and DataDefractor’s sample SSIS packages.

Note: There should be no spaces in the value of parameter ADDLOCAL.

You can perform a silent un-installation by invoking the following command:

```
Msiexec.exe /x DataDefractor.msi /qn
```

Silent Activation

DataDefractor’s silent installation does not activate the product. If you need to activate DataDefractor silently as part of a deployment procedure, you should use DataDefractor’s silent activation batch script `ActivateDataDefractor.bat`. You can find this script in DataDefractor’s program folder.

The activation batch script expects the product activation key to be passed as a parameter like this:

```
ActivateDataDefractor.bat "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX"
```

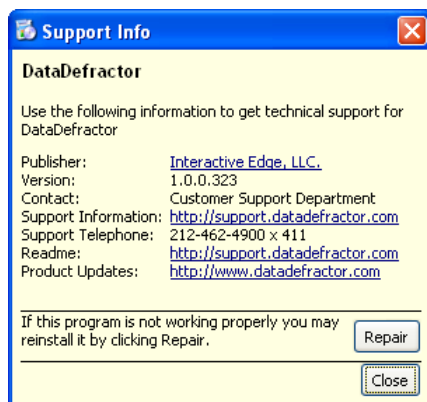
Similar to DataDefractor License Manager, the silent activation batch script requires a live Internet connection.

DataDefractor Version

New versions of DataDefractor are being released continuously. Major new versions are released rarely, but minor new versions and service packs are delivered more often. Occasionally you may need to know the exact version of DataDefractor currently installed on your system.

Follow these steps to find out what version of DataDefractor is installed on your machine:

1. Click **Start**⇒**Control Panel**.
2. Select **“Add or Remove Programs”**.
3. Navigate to item **“DataDefractor”** and select it.
4. Click the link **“Click here for support information”**.
5. You will see the following **“Support Info”** window with the exact version of DataDefractor.



32-bit and 64-bit Platform Support

DataDefractor SSIS Source features support for both 32-bit (x86) and 64-bit (x64) platforms. DataDefractor does not support the Itanium IA64 architecture.

Depending on the edition of SQL Server 20005, DataDefractor will run in either 32-bit or 64-bit mode.

One and the same DataDefractor installation is used for both 32-bit and 64-bit platforms. The installation automatically detects the platform of the target machine and deploys the appropriate components accordingly.

It is important to note that when deployed to a 64-bit machine, Business Intelligence Development Studio (BIDS) runs in 32-bit mode only, but when BIDS runs the SSIS package, it may run it in an 32-bit or a 64-bit SSIS Engine, depending on the value project setting `Run64BitRuntime`. In this case DataDefractor’s User Interface will run in 32-bit mode within the development environment, but when executed by BIDS it will run in either 32-bit or 64-bit mode depending on the mode which the package is being executed in.

Uninstalling DataDefractor

Follow these steps to uninstall DataDefractor:

1. Click **Start⇒Control Panel**.
2. Select **“Add or Remove Programs”**.
3. Navigate to item **“DataDefractor”** and select it.
4. Click **“Remove”**.

This will remove DataDefractor from your system. The SSIS packages built with the help of DataDefractor Data Flow Source component will be invalidated and won’t run on that machine anymore.

If a DataDefractor license was active on the machine at the time of uninstalling, the un-installation procedure will keep the license certificate file in its original place. If you install DataDefractor again, you do not need to activate it – the previous license certificate should still be active. You can verify your current license status by running DataDefractor License Manager.

Understanding Semi-structured Data

Introduction

In this chapter you will learn about the contents and layout of typical fact-based semi-structured data sources. Their classic fact-context model and multi-level composition are discussed in detail.

Fact-based Semi-structured Data Sources

Many enterprises today make decisions based on factual information stored in various forms and locations. Some of it is stored in analytical systems such as enterprise data warehouses or department data marts. These analytical systems provide many ways to pivot, aggregate and analyze the data using a mixture of end-user reporting and OLAP applications.

There is this other form of data, however, which is not easy to analyze and certainly not easy to pivot and aggregate; the de-normalized semi-structured data stored in tens and hundreds of spreadsheets, reports and data dumps scattered around the enterprise.

This type of data originates from various processes both internal and external to the enterprise. For example the process of collaboration and the business planning process both generate loads of semi-structured data. On the other hand, many external data providers present their data in semi-structured form.

On many occasions it is necessary to extract, normalize and load this data into an analytical system in order to involve it in the decision-making process. However, this kind of initiative is often hindered by numerous technical issues.

Most of all, the typical data layout of these data sources is very complex. They are either designed to be visually appealing to the human eye or to be easily interpreted by particular third party applications. These data sources usually observe some structure, but it is so intricate and multipart, with so many conditions and references, that they are close to being *unstructured*. We call these data sources *semi-structured*.

The structure of these semi-structured data sources is usually external to them. Therefore they are not self-descriptive, which makes it almost impossible to devise a process for automatic schema discovery, extraction and normalization.

Let's see some examples.

We will start with several simple worksheets, which demonstrate key aspects of typical semi-structured enterprise data sources.

Annual Sales Data Report

The following screenshot demonstrates a sales report stored in a comma-separated value (CSV) file. The contents of the report have been masked to protect the confidentiality of information:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD 60 OZ		AVALON AD 75 OZ		AVALON AD FS 45 OZ		AVALON AD LIQ 50 OZ	
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.348999226	14380	2.799443672	794	1.370277078
9	101	35	1.52148513	867	4.385061143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.327996113	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 1: Annual sales data report

Our report contains figures about the annual sales and average prices of a group of products sold in a chain of stores. The core of the report is characterized by the numeric values of two measures – **Dollar Sales** and **Average Price**.

Altogether, these numeric values represent the facts captured by the report. Each cell within the fact area contains a single numeric fact:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD 60 OZ		AVALON AD 75 OZ		AVALON AD FS 45 OZ		AVALON AD LIQ 50 OZ	
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.348999226	14380	2.799443672	794	1.370277078
9	101	35	1.52148513	867	4.385061143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.327996113	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 2: Fact area

Each fact is associated with a context. It is the context that assigns significance to the fact. Without it, the fact is just a meaningless number.

For instance, each fact-value in our report was recorded in the context of one of two measures – **Dollar Sales** or **Average Price**. The measure of a fact is part of its context.

In addition, each fact is associated with the store where it was recorded, the time period during which it was recorded and the product which it was recorded for. So, the context of each fact in the report consists of a measure, store, time period and a product.

Let’s take a closer look at our report and examine the links between the facts and their context.

Measure

The names of the measures are stored in row 7. If we take a fact cell and we project a vertical line from that cell to the top of the report, we will come across the fact’s measure at the intersection of its column and row 7. For example, the measure of fact cell **E14** is stored in cell **E7 – Avg Price**:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD 60 OZ		AVALON AD 75 OZ		AVALON AD FS 45 OZ		AVALON AD LIQ 50 OZ	
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.34009226	14380	2.799443672	794	1.370277078
9	101	35	1.52148513	867	4.30001143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.34008114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.30009613	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.34003542	1626	2.740137037	34	1.323667859
13	105	31	1.260069108	655	4.30000395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338436252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 3: Measure context

Store

If we take a fact cell and we project a horizontal line to the left, we will find the store at the intersection between the fact’s row and column A. For cell **E14**, the store is located in cell **A14** – store number **106**:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD 60 OZ	AVALON AD 75 OZ	AVALON AD FS 45 OZ	AVALON AD LIQ 50 OZ				
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.348999226	14380	2.799443672	794	1.370277078
9	101	35	1.52148513	867	4.385061143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.32799613	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14	106				4.36395660	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 4: Store context

Product

Identifying the product context of a fact cell is a bit trickier. The names of the products are located in row **6**, but unlike measures, they are not fully populated across all columns. Measures are nested within the product, which creates groups of consecutive data columns, associated with a single product. Only the first column of such a group will be populated with a product name at row **6**. The rest of the columns for this same product will have a blank cell at row **6** until the first column of the next product group.

In this case, the rule for identifying the product of a fact cell is based on the contents of the cell at the intersection of the fact’s column and row **6**. If the cell is not blank, it contains the name of the product the fact is associated with. If the cell is blank, the product is recorded in the first non-blank cell at row **6** to the left of the fact’s column. For example, the product for both fact cells **D14** and **E14** is stored in cell **D6**:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD	60 OZ	AVALON AD	75 OZ	AVALON AD FS	45 OZ	AVALON AD LIQ	50 OZ
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.34139226	14380	2.799443672	794	1.370277078
9	101	35	1.52148613	867	4.38131143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.34128114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.31399613	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.34133542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.39110395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 5: Product context

Time period

The time period is unique with that it is located in the header of the report. All the facts in the report are related to a single time period stored in cell **B3**:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD	60 OZ	AVALON AD	75 OZ	AVALON AD FS	45 OZ	AVALON AD LIQ	50 OZ
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.34899226	14380	2.799443672	794	1.370277078
9	101	35	1.52148613	867	4.385061143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.32799613	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 6: Time period context

Sub-context

In addition to the facts, parts of the context may have context of their own. For example, the stores in this report belong to a chain of stores, which is stored in cell **B2** in the report’s header. Every store number in this report exists in the context of chain **SPEND-LESS ATLANTA**:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD 60 OZ		AVALON AD 75 OZ		AVALON AD FS 45 OZ		AVALON AD LIQ 50 OZ	
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.348999226	14380	2.799443672	794	1.370277078
9		35	1.52148513	867	4.385061143	529	3.160024497	75	1.433420295
10		26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11		39	1.252738557	1150	4.32799613	1875	2.722134846	45	1.302076935
12		27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13		31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14		34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15		12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16		28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17		5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18		33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19		31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20		11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21		29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22		18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23		16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24		15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25		4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 7: Store chain

If we had data for more than one chain of stores and we had one report per chain, it would be beneficial from analytical point of view to merge the data from all those reports and load it into a single data mart. Now, what will happen if multiple chains numbered their stores sequentially starting from number 1? The store context of one chain’s report may start overlapping with the store context of another chain’s report. To avoid this, we need to qualify the store numbers with the name of their chain while we load the data into the data mart. For example we could construct the store context by concatenating the chain name located in cell **B3** with the appropriate store number located in column **A**.

We will call the context of another context *sub-context*.

This report contains another example of sub-context. Here the size of each product is contained within the last two words of the product’s name:

	A	B	C	D	E	F	G	H	I
1	Sample Store Level Data								
2	Chain	SPEND-LESS ATLANTA							
3	Period	12 Months							
4									
5		Name1	Data						
6		AVALON AD	60 OZ	AVALON AD	75 OZ	AVALON AD FS	45 OZ	AVALON AD LIQ	50 OZ
7		Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price	Dollar Sales	Avg Price
8	Store Number	507	1.337278107	9043	4.348999226	14380	2.799443672	794	1.370277078
9	101	35	1.52148513	867	4.385061143	529	3.160024497	75	1.433420295
10	102	26	1.346878559	494	4.343028114	810	2.829375972	50	1.398688965
11	103	39	1.252738557	1150	4.32799613	1875	2.722134846	45	1.302076935
12	104	27	1.368282943	9	4.343053542	1626	2.740137037	34	1.323567859
13	105	31	1.260069108	655	4.398400395	409	2.841214439	80	1.342183013
14	106	34	1.357948032	326	4.363956601	811	2.773863231	10	1.404804633
15	107	12	1.338438252	368	4.347759	0	2.796774114	64	1.397341197
16	108	28	1.305813918	572	4.302836143	359	2.825994682	54	1.364051223
17	109	5	1.401304908	206	4.318110589	1123	2.761977779	21	1.345564303
18	110	33	1.294078889	635	4.379378648	825	2.815390338	16	1.332298181
19	111	31	1.331955547	498	4.428042597	394	2.73739441	42	1.438006239
20	112	11	1.288533753	286	4.28987903	438	2.797316763	35	1.33156744
21	113	29	1.32362311	280	4.287392763	86	2.813211089	40	1.288112991
22	114	18	1.381522972	13	4.298262646	637	2.829057365	0	1.375131632
23	115	16	1.318818763	63	4.298463858	241	2.813862802	16	1.336228672
24	116	15	1.31573357	189	4.3757313	386	2.777619266	8	1.349523718
25	117	4	1.363323705	244	4.342806368	30	2.783532428	5	1.275135641

Figure 8: Product size

From an analytical point of view, it would be informative to be able to perform analysis on the product’s size as a separate attribute. This would require the size to be extracted from the name of the product as a separate attribute.

Relative and Absolute Context Locations

Each fact context is in a location either relative or absolute to the location of the fact cells. For example, the product, the store number and the measure in this report are all relative to the position of each fact cell. As the location of the fact cell changes, this context changes as well.

There are two types of relative context – horizontal and vertical. The product context for example is horizontal, because it changes with each fact column. The store context on the other hand changes with each fact row – this context is vertical.

In contrast with these relative contexts, the time period is in an absolute position as it relates to the facts. The time period of each fact cell in this report can be found in cell B3.

There are other kinds of absolute context. For example the name of the file or worksheet that contains the fact data is absolute as it relates to its position. Sometimes there is context, which is not stored in the data source at all, but is still implied and recognized by the analytical process. For example, our sales report contains sales figures for 12 months worth of sales, but just by looking at the report it is not clear which year is this report about.

Flexible Layout

Another aspect of this report is its flexible layout. It can grow both horizontally and vertically. When new products or measures are added, the report grows horizontally to the right, while adding stores grows the report vertically by adding new rows.

Changes to the number of stores, measures and products alter the matter of the report, but not its layout. The rules for mapping the fact and context data stay the same. In effect, these same rules could be applied to other reports with different contents as long as their layout is the same.

Normalized Form

Now, what if the data in this report was normalized and represented in a structured *first normal form*²? Here’s a portion of the same data, but this time normalized:

Store	Store Chain	Product	Product Size	Time Period	Avg Price	Dollar Sales	
SPEND-LESS ATLANTA 101	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.52148513	35
SPEND-LESS ATLANTA 101	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.385061143	867
SPEND-LESS ATLANTA 101	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	3.160024497	529
SPEND-LESS ATLANTA 101	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.433420295	75
SPEND-LESS ATLANTA 102	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.346878559	26
SPEND-LESS ATLANTA 102	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.343028114	494
SPEND-LESS ATLANTA 102	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	2.829375972	810
SPEND-LESS ATLANTA 102	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.398688965	50
SPEND-LESS ATLANTA 103	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.252738557	39
SPEND-LESS ATLANTA 103	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.32799613	1150
SPEND-LESS ATLANTA 103	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	2.722134846	1875
SPEND-LESS ATLANTA 103	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.302076935	45
SPEND-LESS ATLANTA 104	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.368282943	27
SPEND-LESS ATLANTA 104	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.343053542	9
SPEND-LESS ATLANTA 104	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	2.740137037	1626
SPEND-LESS ATLANTA 104	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.323567859	34
SPEND-LESS ATLANTA 105	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.260069108	31
SPEND-LESS ATLANTA 105	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.398400395	655
SPEND-LESS ATLANTA 105	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	2.841214439	409
SPEND-LESS ATLANTA 105	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.342183013	80
SPEND-LESS ATLANTA 106	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.357948032	34
SPEND-LESS ATLANTA 106	SPEND-LESS ATLANTA	AVALON AD	75 OZ	75 OZ	12 Months	4.363956601	326
SPEND-LESS ATLANTA 106	SPEND-LESS ATLANTA	AVALON AD FS	45 OZ	45 OZ	12 Months	2.773863231	811
SPEND-LESS ATLANTA 106	SPEND-LESS ATLANTA	AVALON AD LIQ	50 OZ	50 OZ	12 Months	1.404804633	10
SPEND-LESS ATLANTA 107	SPEND-LESS ATLANTA	AVALON AD	60 OZ	60 OZ	12 Months	1.338438252	12

Figure 9: Annual sales data report in first normal form

Each row would contain a unique combination of **Store**, **Product** and **Time Period**. This combination would be the primary key of the table and would be associated with the values stored in the two measure columns – **Avg Price** and **Dollar Sales**. In addition to this, columns **Product Size** and **Store Chain** would contain attribute information concerning the **Product** and **Store** respectively.

This form of normalization is also known as flattened star-schema³ recordset. It puts every component of the context in its own separate column, the only exception being the measures, which are distributed across the columns – every measure is in a separate column. A flattened star-schema recordset represents the full contents of a star schema in a single recordset.

If the raw data came in such a form in the first place, it wouldn’t be hard to extract it and load it into a data warehouse or to consume it directly with an analytical tool like Excel PivotTables for example.

² First normal form (1NF) is a normal form used in database normalization. First normal form excludes the possibility of repeating groups by requiring that each field in a database hold an atomic value, and that records be defined in such a way as to be uniquely identifiable by means of a primary key. (Wikipedia, First normal form)

³ The star schema (sometimes referenced as star join schema) is the simplest data warehouse schema, consisting of a single "fact table" with a compound primary key, with one segment for each "dimension" and with additional columns of additive, numeric facts. The name star schema is derived from the fact that the schema diagram is shaped like a star (Wikipedia, Star schema).

Unfortunately, in our work with enterprise data, we often come across data stored in semi-structured form, which makes it difficult to consume by automated non-manual processes.

Here are some of the major obstacles you will face if you attempt to extract this particular report with the conventional structured-data extraction mechanisms provided by typical ETL systems:

1. The report contains a header which contains parts of the context (e.g. **Time Period** and **Store Chain**).
2. Some columns contain both facts and context (see Figure 1, columns **B, D, F** and **H**).
3. Parts of the context run across the columns, which makes the report grow horizontally (e.g. **Product**). This makes the data source “pivoted” and “de-normalized”.
4. “Blank” context locations caused by context grouping and context carryover (see Figure 1, **Product** cells **C6, E6, G6** and **I6**).

Naturally, you would have to resort to programming an extraction script procedure or custom data source component, which will be specific to this report’s layout. A small change like nesting the time period with the store number, for example, could break the extraction procedure and force you to review and update your code to match the new layout.

Freddie Mac Mortgage Margins Report

Now let’s take a look at some other examples of fact-context associations you may come across in typical enterprise reports.

Consider the following report provided by Freddie Mac⁴. It is stored in an Excel workbook and contains public historical weekly mortgage survey data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	2006													
2	PRIMARY MORTGAGE MARKET SURVEY*													
3	1 yr ARM margin and 5/1 ARM margin													
4														
5	U.S.						U.S.							
6	5/1 ARM		Regional 5/1 ARM margins					1 yr ARM		Regional 1 yr ARM margins				
7	Week	margin	NE	SE	NC	SW	W	margin	NE	SE	NC	SW	W	
8														
9	1/5	2.78	2.83	2.76	2.81	2.78	2.74	2.77	2.79	2.76	2.79	2.80	2.74	
10	1/12	2.78	2.84	2.76	2.80	2.78	2.74	2.78	2.83	2.76	2.77	2.79	2.74	
11	1/19	2.78	2.84	2.75	2.81	2.78	2.74	2.78	2.84	2.75	2.81	2.79	2.72	
12	1/26	2.78	2.83	2.75	2.79	2.78	2.74	2.76	2.80	2.75	2.76	2.79	2.73	
13	2/2	2.78	2.83	2.76	2.82	2.78	2.74	2.78	2.83	2.77	2.80	2.81	2.73	
14	2/9	2.78	2.83	2.76	2.81	2.78	2.73	2.79	2.83	2.76	2.78	2.79	2.78	
15	2/16	2.78	2.84	2.76	2.81	2.78	2.74	2.79	2.88	2.76	2.78	2.79	2.73	
16	2/23	2.77	2.81	2.76	2.79	2.79	2.73	2.78	2.77	2.74	2.76	2.80	2.83	
17	3/2	2.78	2.82	2.74	2.79	2.78	2.74	2.77	2.80	2.74	2.78	2.80	2.74	
18	3/9	2.78	2.82	2.76	2.80	2.79	2.73	2.77	2.81	2.76	2.80	2.81	2.71	
19	3/16	2.79	2.84	2.76	2.80	2.78	2.76	2.77	2.78	2.76	2.78	2.79	2.76	
20	3/23	2.79	2.84	2.76	2.79	2.79	2.76	2.78	2.83	2.76	2.78	2.80	2.76	
21	3/30	2.78	2.82	2.76	2.78	2.78	2.76	2.77	2.76	2.77	2.79	2.80	2.76	
22	4/6	2.77	2.77	2.76	2.80	2.78	2.76	2.78	2.80	2.76	2.78	2.80	2.78	
23	4/13	2.78	2.83	2.76	2.80	2.78	2.76	2.78	2.81	2.76	2.80	2.79	2.76	
24	4/20	2.77	2.83	2.76	2.79	2.78	2.74	2.76	2.80	2.76	2.76	2.76	2.75	
25	4/27	2.78	2.83	2.76	2.80	2.78	2.74	2.77	2.79	2.76	2.77	2.80	2.74	

Figure 10: Freddie Mac mortgage margins report

The numeric facts in the report represent **1 year ARM** and **5/1 ARM** margins. These margins are captured for different geographical regions – **NE** (North East), **SE** (South East), **Total U.S.**, etc. Furthermore, the margins have been monitored every week in the course of year **2006**.

The context of any fact in this report consists of a measure, geographical region and time period. Now let’s identify the association paths between each fact and its context.

⁴ The Federal Home Loan Mortgage Corporation ("Freddie Mac"), is a stockholder-owned corporation chartered by the USA Congress in 1970 to keep money flowing to mortgage lenders in support of homeownership and rental housing. The report discussed here can be accessed at <http://www.freddiemac.com/corporate/pmms/2006/historicalweeklydata.xls>

Measures

The measure names are stored in row 6, but the content of the cells is not consistent. In some cases the measure cell contains just the measure name (cell B6); in other cases the measure cell contains the measure name plus some unnecessary information surrounding it (merged cells C6 through G6 and merged cells I6 through M6):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	2006												
2	PRIMARY MORTGAGE MARKET SURVEY*												
3	1 yr ARM margin and 5/1 ARM margin												
4													
5	U.S.						U.S.						
6	5/1 ARM		Regional 5/1 ARM margins				1 yr ARM		Regional 1 yr ARM margins				
7	Week	margin	NE	SE	NC	SW	W	margin	NE	SE	NC	SW	W
8													
9	1/5	2.78	2.83	2.76	2.81	2.78	2.74	2.77	2.79	2.76	2.79	2.80	2.74
10	1/12	2.78	2.84	2.76	2.80	2.78	2.74	2.78	2.83	2.76	2.77	2.79	2.74
11	1/19	2.78	2.84	2.75	2.81	2.78	2.74	2.78	2.84	2.75	2.81	2.79	2.72
12	1/26	2.78	2.83	2.75	2.79	2.78	2.74	2.76	2.80	2.75	2.76	2.79	2.73
13	2/2	2.78	2.83	2.76	2.82	2.78	2.74	2.78	2.83	2.77	2.80	2.81	2.73
14	2/9	2.78	2.83	2.76	2.81	2.78	2.73	2.79	2.83	2.76	2.78	2.79	2.78
15	2/16	2.78	2.84	2.76	2.81	2.78	2.74	2.79	2.88	2.76	2.78	2.79	2.73
16	2/23	2.77	2.81	2.76	2.79	2.79	2.73	2.78	2.77	2.74	2.76	2.80	2.83
17	3/2	2.78	2.82	2.74	2.79	2.78	2.74	2.77	2.80	2.74	2.78	2.80	2.74
18	3/9	2.78	2.82	2.76	2.80	2.79	2.73	2.77	2.81	2.76	2.80	2.81	2.71
19	3/16	2.79	2.84	2.76	2.80	2.78	2.76	2.77	2.78	2.76	2.78	2.79	2.76
20	3/23	2.79	2.84	2.76	2.79	2.79	2.76	2.78	2.83	2.76	2.78	2.80	2.76
21	3/30	2.78	2.82	2.76	2.78	2.78	2.76	2.77	2.76	2.77	2.79	2.80	2.76
22	4/6	2.77	2.77	2.76	2.80	2.78	2.76	2.78	2.80	2.76	2.78	2.80	2.78
23	4/13	2.78	2.83	2.76	2.80	2.78	2.76	2.78	2.81	2.76	2.80	2.79	2.76
24	4/20	2.77	2.83	2.76	2.79	2.78	2.74	2.76	2.80	2.76	2.76	2.76	2.75
25	4/27	2.78	2.83	2.76	2.80	2.78	2.74	2.77	2.79	2.76	2.77	2.80	2.74

Figure 11: Measure context

Let’s define a fact-to-measure association rule that would work for this report. The rule should correctly identify the measure context for any given fact in the worksheet. Here’s one:

If we project a vertical line from any fact cell up to row 6, we will come across a measure cell, which contains some text. If we parse this text and extract the sub-text beginning at the first occurrence of a digit symbol and ending at the first successive occurrence of the string **ARM**, we will get to the real measure. In this case cell B6 through G6 will yield measure **5/1 ARM**, while cells H6 through M6 will yield measure **1 yr ARM**.

Geographical Region

For some of the facts geographical regions are located in row 5 (e.g. region **U.S.** for columns **B** and **H**) and for some facts it is in row 7 (e.g. columns **C** through **G** and columns **I** through **M**):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	2006												
2	PRIMARY MORTGAGE MARKET SURVEY*												
3	1 yr ARM margin and 5/1 ARM margin												
4													
5		U.S.						U.S.					
6		5/1 ARM	Regional 5/1 ARM margins					1 yr ARM	Regional 1 yr ARM margins				
7	Week	margin	NE	SE	NC	SW	W	margin	NE	SE	NC	SW	W
8													
9	1/5	2.78	2.83	2.76	2.81	2.78	2.74	2.77	2.79	2.76	2.79	2.80	2.74
10	1/12	2.78	2.84	2.76	2.80	2.78	2.74	2.78	2.83	2.76	2.77	2.79	2.74
11	1/19	2.78	2.84	2.75	2.81	2.78	2.74	2.78	2.84	2.75	2.81	2.79	2.72
12	1/26	2.78	2.83	2.75	2.79	2.78	2.74	2.76	2.80	2.75	2.76	2.79	2.73
13	2/2	2.78	2.83	2.76	2.82	2.78	2.74	2.78	2.83	2.77	2.80	2.81	2.73
14	2/9	2.78	2.83	2.76	2.81	2.78	2.73	2.79	2.83	2.76	2.78	2.79	2.78
15	2/16	2.78	2.84	2.76	2.81	2.78	2.74	2.79	2.88	2.76	2.78	2.79	2.73
16	2/23	2.77	2.81	2.76	2.79	2.79	2.73	2.78	2.77	2.74	2.76	2.80	2.83
17	3/2	2.78	2.82	2.74	2.79	2.78	2.74	2.77	2.80	2.74	2.78	2.80	2.74
18	3/9	2.78	2.82	2.76	2.80	2.79	2.73	2.77	2.81	2.76	2.80	2.81	2.71
19	3/16	2.79	2.84	2.76	2.80	2.78	2.76	2.77	2.78	2.76	2.78	2.79	2.76
20	3/23	2.79	2.84	2.76	2.79	2.79	2.76	2.78	2.83	2.76	2.78	2.80	2.76
21	3/30	2.78	2.82	2.76	2.78	2.78	2.76	2.77	2.76	2.77	2.79	2.80	2.76
22	4/6	2.77	2.77	2.76	2.80	2.78	2.76	2.78	2.80	2.76	2.78	2.80	2.78
23	4/13	2.78	2.83	2.76	2.80	2.78	2.76	2.78	2.81	2.76	2.80	2.79	2.76
24	4/20	2.77	2.83	2.76	2.79	2.78	2.74	2.76	2.80	2.76	2.76	2.76	2.75
25	4/27	2.78	2.83	2.76	2.80	2.78	2.74	2.77	2.79	2.76	2.77	2.80	2.74

Figure 12: Geographical region

A fact-to-geography association rule may look like this:

Project a vertical line from the fact toward row 5. If the cell contains text, use this text as the geographical region. If the cell is empty, use the text from cell in row 7 as the geographical region.

Time period

The time period is straightforward. You just project a line from the fact to the left and you will find the time period at the intersection with column **A**.

Normalized Form

If we were to normalize the data in this report, here’s a portion of what it would look like:

Time Period	Geography Region	1 yr ARM	5/1 ARM
1/5	U.S.	2.77	2.78
1/5	NE	2.79	2.83
1/5	SE	2.76	2.76
1/5	NC	2.79	2.81
1/5	SW	2.8	2.78
1/5	W	2.74	2.74
1/12	U.S.	2.78	2.78
1/12	NE	2.83	2.84
1/12	SE	2.76	2.76
1/12	NC	2.77	2.8
1/12	SW	2.79	2.78
1/12	W	2.74	2.74
1/19	U.S.	2.78	2.78
1/19	NE	2.84	2.84
1/19	SE	2.75	2.75
1/19	NC	2.81	2.81
1/19	SW	2.79	2.78
1/19	W	2.72	2.74
1/26	U.S.	2.76	2.78
1/26	NE	2.8	2.83
1/26	SE	2.75	2.75
1/26	NC	2.76	2.79
1/26	SW	2.79	2.78
1/26	W	2.73	2.74

Figure 13: Freddie Mac mortgage margins report in first normal form

The measures and geographical regions are extracted out of their complex row header and pivoted in a clean structured form; the time period is aligned accordingly. If the report was normalized like this, its data would be ready to be loaded into an analytical system.

Freddie Mac Mortgage Series Report (multi-page report)

Now let’s take a look at another worksheet provided by Freddie Mac. It represents the monthly mortgage rates over the course of seventeen years⁵:


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		Primary Mortgage Market Survey®																			
2		CONVENTIONAL, CONFORMING 15-YEAR FIXED-RATE MORTGAGE SERIES SINCE 1991																			
3	We make home possible™																				
4																					
5		1991		1992		1993		1994		1995		1996		1997							
6		Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts				
7																					
8	January	NA	NA	8.01	1.7	7.51	1.7	6.57	1.7	8.80	1.8	6.55	1.7	7.33	1.7						
9	February	NA	NA	8.38	1.8	7.17	1.5	6.66	1.7	8.46	1.8	6.56	1.7	7.15	1.7						
10	March	NA	NA	8.58	1.9	7.00	1.6	7.18	1.7	8.06	1.8	7.11	1.8	7.41	1.7						
11	April	NA	NA	8.47	1.7	6.94	1.6	7.80	1.7	7.88	1.8	7.44	1.7	7.68	1.7						
12	May	NA	NA	8.29	1.7	6.93	1.8	8.08	1.7	7.51	1.7	7.58	1.7	7.47	1.6						
13	June	NA	NA	8.08	1.7	6.92	1.6	7.91	1.8	7.06	1.7	7.83	1.7	7.24	1.7						
14	October	8.49	1.8	7.55	1.7	6.37	1.5	8.39	1.8	7.01	1.8	7.43	1.7	6.85	1.7						
15	November	8.33	1.7	7.80	1.8	6.69	1.5	8.67	1.8	6.89	1.8	7.14	1.7	6.76	1.7						
16	December	8.07	1.7	7.74	1.6	6.68	1.5	8.80	1.8	6.74	1.7	7.10	1.7	6.66	1.8						
17	Annual Avgs:	NA	NA	8.10	1.7	6.91	1.6	7.86	1.8	7.48	1.8	7.32	1.7	7.13	1.7						
18																					
19		1998		1999		2000		2001		2002		2003		2004							
20		Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts						
21																					
22	January	6.58	1.4	6.43	1.0	7.80	1.0	6.64	0.9	6.48	0.7	5.30	0.6	5.02	0.7						
23	February	6.64	1.2	6.44	1.0	7.93	1.0	6.64	0.9	6.38	0.7	5.22	0.6	4.94	0.7						
24	March	6.74	1.2	6.68	0.9	7.83	0.9	6.51	1.0	6.52	0.7	5.07	0.6	4.74	0.7						
25	April	6.78	1.0	6.53	0.9	7.80	1.0	6.60	1.0	6.48	0.7	5.12	0.6	5.16	0.6						
26	May	6.78	1.0	6.75	1.0	8.18	1.0	6.68	1.0	6.28	0.7	4.86	0.7	5.64	0.7						
27	June	6.67	1.0	7.18	1.0	7.99	0.9	6.70	1.0	6.11	0.6	4.63	0.6	5.66	0.6						
28	October	6.36	0.9	7.47	1.0	7.47	1.0	6.10	0.9	5.50	0.6	5.27	0.6	5.12	0.6						
29	November	6.51	0.9	7.36	1.0	7.42	0.9	6.15	0.8	5.46	0.6	5.27	0.7	5.14	0.6						
30	December	6.39	0.9	7.52	1.0	7.06	0.9	6.54	0.8	5.45	0.6	5.20	0.6	5.18	0.6						
31	Annual Avgs:	6.59	1.1	7.06	0.7	7.72	1.0	6.50	0.9	5.98	0.6	5.17	0.6	5.21	0.6						

Figure 14: Freddie Mac mortgage series report

The facts of this report are associated with two measures. **Rate** represents the reported interest rate and **Points** represent the discount points which the rate applies to. The facts are also associated with a time period – a year and a month, and a product – in this case the product is “Conforming 15 year fixed-rate mortgage”.

⁵ The report discussed here can be accessed at http://www.freddiemac.com/pmms/docs/15yr_pmmsmth.xls

Let’s take a look at the layout of the report and the system of rules that relate the facts to their context:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Freddie Mac	Primary Mortgage Market Survey®																			
2	We make home possible™	CONVENTIONAL, CONFORMING 15-YEAR FIXED-RATE MORTGAGE SERIES SINCE 1991																			
3																					
4																					
5		(1991)		(1992)		(1993)		(1994)		(1995)		(1996)		(1997)							
6		Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts						
7																					
8	January	NA	NA	8.01	1.7	7.51	1.7	6.57	1.7	8.80	1.8	6.55	1.7	7.33	1.7						
9	February	NA	NA	8.38	1.8	7.17	1.5	6.66	1.7	8.46	1.8	6.56	1.7	7.15	1.7						
10	March	NA	NA	8.58	1.9	7.00	1.6	7.18	1.7	8.06	1.8	7.11	1.8	7.41	1.7						
11	April	NA	NA	8.47	1.7	6.94	1.6	7.80	1.7	7.88	1.8	7.44	1.7	7.68	1.7						
12	May	NA	NA	8.29	1.7	6.93	1.8	8.08	1.7	7.51	1.7	7.58	1.7	7.47	1.6						
13	June	NA	NA	8.08	1.7	6.92	1.6	7.91	1.8	7.06	1.7	7.83	1.7	7.24	1.7						
14	October	8.49	1.8	7.55	1.7	6.37	1.5	8.39	1.8	7.01	1.8	7.43	1.7	6.85	1.7						
15	November	8.33	1.7	7.80	1.8	6.69	1.5	8.67	1.8	6.89	1.8	7.14	1.7	6.76	1.7						
16	December	8.07	1.7	7.74	1.6	6.68	1.5	8.80	1.8	6.74	1.7	7.10	1.7	6.66	1.8						
17	Annual Avgs:	NA	NA	8.10	1.7	6.91	1.6	7.86	1.8	7.48	1.8	7.32	1.7	7.13	1.7						
18																					
19		(1998)		(1999)		(2000)		(2001)		(2002)		(2003)		(2004)							
20		Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts	Rate	Pts						
21																					
22	January	6.58	1.4	6.43	1.0	7.80	1.0	6.64	0.9	6.48	0.7	5.30	0.6	5.02	0.7						
23	February	6.64	1.2	6.44	1.0	7.93	1.0	6.64	0.9	6.38	0.7	5.22	0.6	4.94	0.7						
24	March	6.74	1.2	6.68	0.9	7.83	0.9	6.51	1.0	6.52	0.7	5.07	0.6	4.74	0.7						
25	April	6.78	1.0	6.53	0.9	7.80	1.0	6.60	1.0	6.48	0.7	5.12	0.6	5.16	0.6						
26	May	6.78	1.0	6.75	1.0	8.18	1.0	6.68	1.0	6.28	0.7	4.86	0.7	5.64	0.7						
27	June	6.67	1.0	7.18	1.0	7.99	0.9	6.70	1.0	6.11	0.6	4.63	0.6	5.66	0.6						
28	October	6.36	0.9	7.47	1.0	7.47	1.0	6.10	0.9	5.50	0.6	5.27	0.6	5.12	0.6						
29	November	6.51	0.9	7.36	1.0	7.42	0.9	6.15	0.8	5.46	0.6	5.27	0.7	5.14	0.6						
30	December	6.39	0.9	7.52	1.0	7.06	0.9	6.54	0.8	5.45	0.6	5.20	0.6	5.18	0.6						
31	Annual Avgs:	6.59	1.1	7.06	0.7	7.72	1.0	6.50	0.9	5.98	0.6	5.17	0.6	5.21	0.6						

Figure 15: Freddie Mac mortgage series report - facts and context

What’s new in this report is the fact that it is partitioned vertically into smaller sub-pages – each sub-page containing the facts for a group of seven years. Furthermore, each vertical page is partitioned horizontally into a series of sub-pages – one per year.

This cascading structure of multiple levels of sub-pages nested within each other can be defined by a set of patterns which indicate how each page is partitioned into a set of sub-pages.

For example, our report has a two-level cascading structure. The first level being the series of seven-year vertical sub-pages, the second level being the sub-series of one-year horizontal sub-pages nested within each page of the first level.

The first level can be defined with the following pattern applied to the contents of the report: Each row that contains a blank cell in column A, a four-digit integer number in column B and a blank cell in column C indicates the beginning of a new vertical sub-page.

If we drill into these vertical pages and we try to partition them further, the second level of partitioning can be defined with the following pattern: Each column that contains a four-digit integer number in the first row of each first-level sub-page indicates the beginning of a new second-level horizontal sub-sub-page.

The numeric facts are fully contained within these second-level fine-grained horizontal sub-sub-pages. Therefore, the lowest level of sub-pages is often called “the fact level”.

In contrast to the facts, the fact context is distributed throughout the report and its different sub-page levels. Let’s examine the context:

Product

The product is contained in the header of the report, in cell **B2**. The product contained in this single cell applies to all the facts contained in all sub-pages of the report.

Time Period

The time period consists of a year and a month.

The series of months is contained in the first column of each first-level sub-page. For example, the months for the first vertical page are contained in the range of cells **A8-A16**, while the months for the second vertical page are contained in range **A22-A30**. The series of months for a first-level vertical page applies to all its second-level horizontal sub-pages.

In addition, the year of the time period is contained within each second-level horizontal sub-page. It is located in the upper left cell relative to the borders of each of those fine-grained sub-pages. For example, the year of the first horizontal sub-page of the first vertical sub-page is contained in cell **B5**, while the year of the third horizontal sub-page of the second vertical sub-page is located in cell **H19**.

Measures

Similarly to the year, the measures are located in the second-level horizontal pages, this time in the cells of the second row of each second-level horizontal page.

Normalized Form

Here are the first rows of the same data normalized form:

Product	Year	Month	Rate	Pts
15-YEAR FIXED RATE MORTGAGE	1991	September	8.69	1.8
15-YEAR FIXED RATE MORTGAGE	1991	October	8.49	1.8
15-YEAR FIXED RATE MORTGAGE	1991	November	8.33	1.7
15-YEAR FIXED RATE MORTGAGE	1991	December	8.07	1.7
15-YEAR FIXED RATE MORTGAGE	1992	January	8.01	1.7
15-YEAR FIXED RATE MORTGAGE	1992	February	8.38	1.8
15-YEAR FIXED RATE MORTGAGE	1992	March	8.58	1.9
15-YEAR FIXED RATE MORTGAGE	1992	April	8.47	1.7
15-YEAR FIXED RATE MORTGAGE	1992	May	8.29	1.7
15-YEAR FIXED RATE MORTGAGE	1992	June	8.08	1.7
15-YEAR FIXED RATE MORTGAGE	1992	July	7.67	1.6
15-YEAR FIXED RATE MORTGAGE	1992	August	7.49	1.6
15-YEAR FIXED RATE MORTGAGE	1992	September	7.41	1.6
15-YEAR FIXED RATE MORTGAGE	1992	October	7.55	1.7
15-YEAR FIXED RATE MORTGAGE	1992	November	7.8	1.8
15-YEAR FIXED RATE MORTGAGE	1992	December	7.742	1.6
15-YEAR FIXED RATE MORTGAGE	1993	January	7.51	1.7
15-YEAR FIXED RATE MORTGAGE	1993	February	7.17	1.5
15-YEAR FIXED RATE MORTGAGE	1993	March	7	1.6
15-YEAR FIXED RATE MORTGAGE	1993	April	6.9425	1.6
15-YEAR FIXED RATE MORTGAGE	1993	May	6.93	1.8
15-YEAR FIXED RATE MORTGAGE	1993	June	6.92	1.6
15-YEAR FIXED RATE MORTGAGE	1993	July	6.72	1.6

Figure 16: Freddie Mac mortgage series report in first normal form

Conclusion

We analyzed several semi-structured data sources with layouts typical to the ones we meet in data which is often needed to support the analytical decision-making process within an enterprise. Normally semi-structured data sources are loaded into analytical systems via custom Extraction Transformation and Loading (ETL) scripts developed by data-integration and business-intelligence developers. The major problem with this approach is that it takes many hours to develop, test, deploy and document a solution based on custom scripts. Another problem is that such a solution is usually designed and developed with a narrow scope as it relates to the layout of the data source. This leads to maintenance issues when even the slightest change is introduced to the original layout of a data source.

In the next chapters we will explore how DataDefractor can help us resolve the problem of normalizing this kind of data quickly and efficiently within the environment of SQL Server Integration Services.

Using DataDefractor

Introduction

In this chapter you will learn how to normalize one or more semi-structured data sources with DataDefractor.

Processing Workflow

From a functional point of view DataDefractor consists of two parts – a wizard-based user interface and a data normalization engine. The user interface is designed to help you map out the schema of data sources with a common structure. Given the data sources and their mapping schema, the normalization engine processes the semi-structured contents of these data sources and outputs normalized data on the SSIS pipeline.

Mapping Schema

The mapping schema of data sources with a common structure includes the definition of their fact area. As you will see later, the rules that define the fact area are flexible enough to accommodate variable headers, footers and fact columns.

The mapping schema also contains the context model of the common data sources as well as the rules that map the separate components of this context model to different locations in the data sources.

The context model defined by the mapping schema is multidimensional in nature.

Each part of the fact context, with the exception of the Measures, is expressed as a dimension. For example, the fact context in our sales report example (see “**Annual Sales Data Report**” in “**Understanding Semi-structured Data**”) is expressed as a collection of three dimensions: product, store and time period. The Measures context is a special dimension, which always exists in the context model. It is used to group the facts by some special characteristics like data type.

In multidimensional terms, the sub-context of a context is expressed as an attribute of the appropriate dimension. For example, the chain of stores is expressed as an attribute of the store dimension while the product size is expressed as an attribute of the product dimension.

Normalization Procedure

The normalization procedure processes one or more data sources which observe the rules defined in a single mapping schema.

At the beginning of the normalization procedure the engine constructs a virtual normalized data model based on the fact-context model defined in the mapping schema. The composition of this normalized data model closely resembles that of a multidimensional star schema – a single data structure contains the facts and a separate data structure contains each dimension together with its attributes. Initially this data model is an empty shell, but as the normalization procedure progresses it is populated with data extracted from the data sources.

After the internal data model is constructed, the normalization procedure continues with the extraction process, which sequentially scans all data sources one by one collecting fact and context information.

Let’s take a closer look at this:

The extraction is driven by a cursor, which moves through the fact area of the currently processed data source. The cursor scans the fact cells following a certain path. It scans the first row of facts starting at the left-most fact cell and moving through all the fact cells to the right. Then it jumps at the beginning of the second fact row, scans its fact cells from left to right and so on until it scans the last fact row of the data source. Then it moves on to the next data source, scans its facts, and so on until it is done processing all the input data sources.

The following image illustrates an example of the scanning sequence:

	A	B	C	D	E	F	G	H	I												
1	Sample Store Level Data																				
2	Chain	SPEND-LESS ATLANTA																			
3	Period	12 Months																			
4																					
5		Name1		Data																	
6		AVALON AD	60 OZ	AVALON AD	75 OZ	AVALON AD FS	45 OZ	AVALON AD LIQ	50 OZ												
7		Dollar Sales		Avg Price		Dollar Sales		Avg Price													
8	Store Number	1	507	1.33	2	107	3	9043	4.34	4	226	5	14380	2.75	6	672	7	794	1.37	8	078
9	101	9	35	1.8	10	613	11	867	4.38	12	143	13	529	3.16	14	497	15	75	1.43	16	295
10	102	17	26	1.34	18	659	19	494	4.34	...	114	810	2.829375972				50	1.398688965			
11	103		39	1.252738657			1150	4.32799613				1875	2.722134846				45	1.302076935			
12	104		27	1.368282943			9	4.343063542				1626	2.740137037				34	1.323567859			
13	105		31	1.260069108			655	4.398400395				409	2.841214439				80	1.342183013			
14	106		34	1.357948032			326	4.363956601				811	2.773863231				10	1.404804633			
15	107		12	1.338438252			368	4.347759				0	2.796774114				64	1.397341197			
16	108		28	1.305813918			572	4.302836143				359	2.825994682				54	1.364051223			
17	109		5	1.401304908			206	4.318110589				1123	2.761977779				21	1.345564303			
18	110		33	1.294078889			635	4.379378648				825	2.815390338				16	1.332298181			
19	111		31	1.331955547			498	4.428042597				394	2.73739441				42	1.438006239			
20	112		11	1.288533753			286	4.28987903				438	2.797316763				35	1.33156744			
21	113		29	1.32362311			280	4.287392763				86	2.813211089				40	1.288112991			
22	114		18	1.381522972			13	4.298262646				637	2.829057365				0	1.375131632			
23	115		16	1.318818763			63	4.298463858				241	2.813862802				16	1.336228672			
24	116		15	1.31573357			189	4.3757313				386	2.777619266				8	1.349523718			
25	117		4	1.363323705			244	4.342806368				30	2.783532428				5	1.275135641			

Figure 17: Fact cursor scanning sequence

As it scans the fact cells, the engine determines the full context of each cell. It constructs the context by mapping the fact cell to the relative and absolute context locations defined in the mapping schema (see section **“Relative and Absolute Context Locations”** in **“Understanding Semi-structured Data”**).

The horizontal relative context is collected by projecting a vertical line from the fact cell toward the context rows above the cell. The vertical context on the other hand, is collected by projecting horizontal lines from the fact cell toward the columns containing context to the left and to the right of the cell. Depending on the type of absolute context used in the mapping schema, it is collected either from absolute context cells, from the name of the file (worksheet) or directly from the definition of the mapping schema in case the context is contained outside of the data source.

After the fact context of a fact cell is uncovered, the engine determines the context’s sub-context (see **“Sub-context”** in **“Understanding Semi-structured Data”**). This process is similar to the one we just examined. The engine projects vertical and horizontal lines from the fact cell to detect the sub-context. You may wonder why the sub-context is collected relatively from the fact cell and not from the context cell. If you think about it, this makes sense, because the sub-context of the context of a fact cell is

indirectly context of the fact cell itself. In addition to this, in most occasions the fact cell is the only link between the context and its sub-context.

Once the fact data, the fact context and the sub-context of a fact cell is collected, the engine inserts this information into the in-memory normalized data model and moves on to the next fact cell. As the normalized data model starts to fill up, the engine starts pumping the available normalized data into the SSIS pipeline. This process is asynchronous – depending on the number and size of the data sources, batches of normalized data may become available on the output before all the data on the input is processed. The size of the output batches is controlled by properties of the DataDefractor data flow source component and are accessible in design time.

The data on the SSIS output of DataDefractor can be transformed, loaded or processed in any way possible within the SSIS environment as soon as it becomes available.

Creating a Connection Manager

The first step in designing a mapping schema is to choose a data source. The data source consists of a connection manager and a set of tables found in that connection manager. Prior to selecting a data source you need to have a SSIS Connection Manager that defines a connection to your data.

DataDefractor supports the following connection managers:

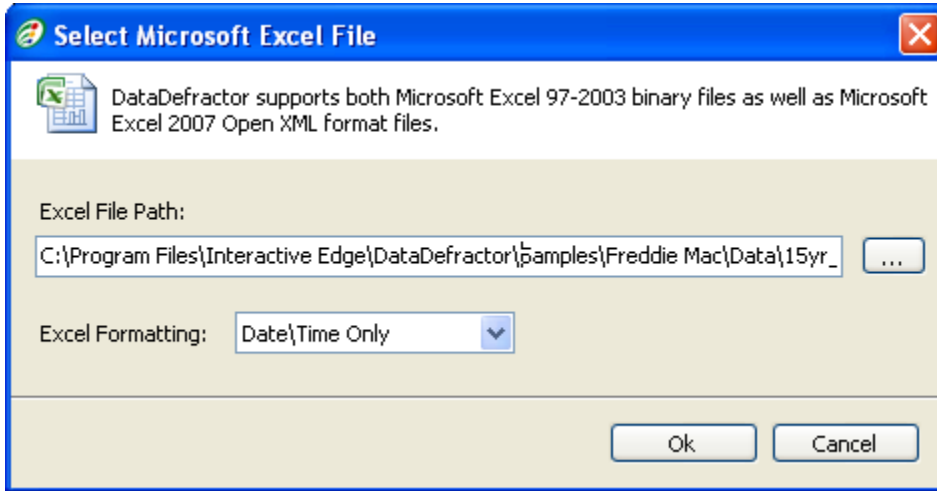
- DataDefractor Excel File Connection Manager (`DataDefractor.ExcelFile`)
- DataDefractor Text File Connection Manager (`DataDefractor.TextFiles`)
- ODBC Connection Manager (`ODBC`)

The DataDefractor connection managers are automatically installed during DataDefractor setup. The ODBC Connection Manager is installed with SQL Server Integration Services.

DataDefractor Excel File Connection Manager

DataDefractor Excel File Connection Manager is used to define a connection to an Excel file. It supports Excel 97/2000/XP/2003 XLS file format and the Excel 2007 XLSX file format.

To create a DataDefractor Excel File Connection Manager right-click on the Connection Managers design surface in the Business Intelligence Development Studio and choose the “**New Connection...**” command. In the “**Add SSIS Connection Manager**” dialog box choose to create a connection manager of type `DataDefractor.ExcelFile`.



In the “**Excel File Path**” text box enter the path to the Excel file. This could be an XLS or XLSX file.

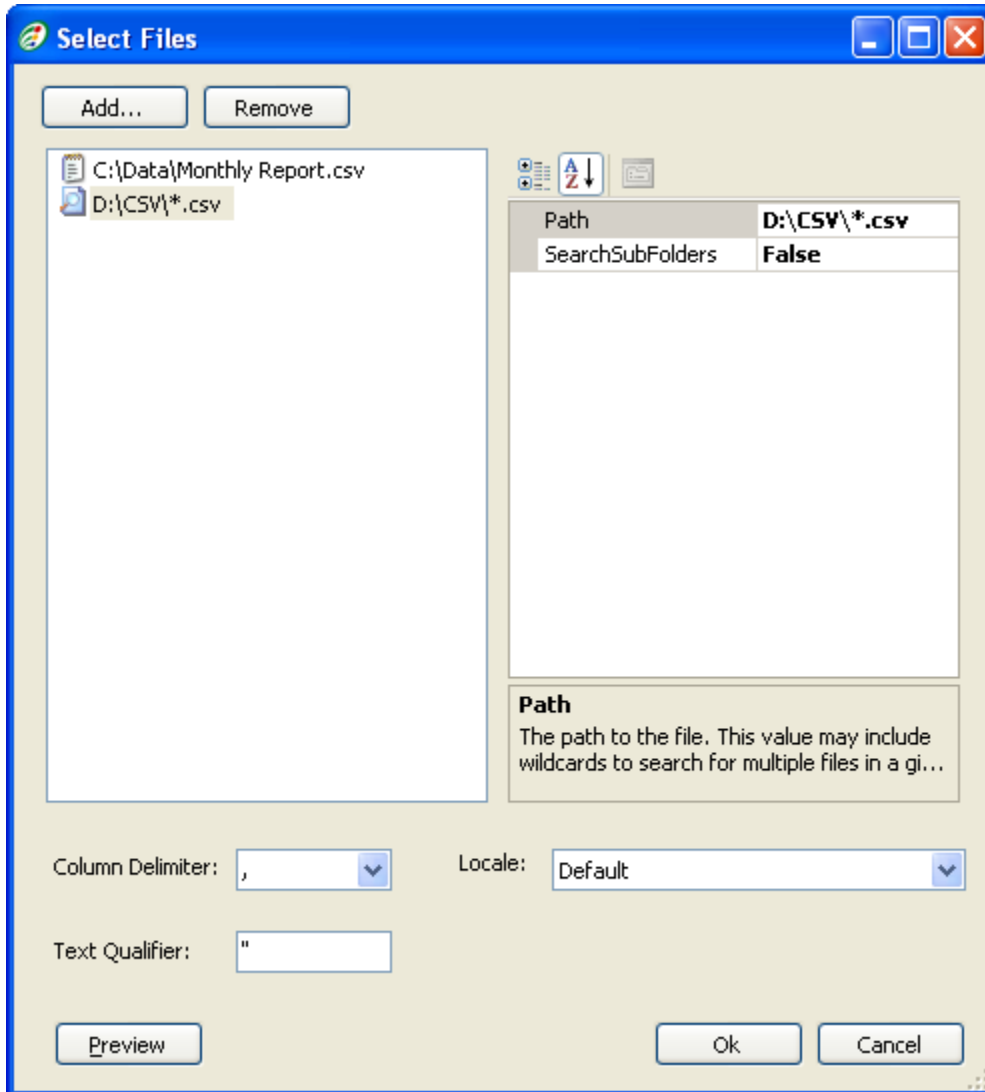
In the “**Excel Formatting**” combo box you can choose how DataDefractor should interpret the cell formatting information stored in the Excel file.

Date\Time Only	DataDefractor applies formatting only to cells that are formatted with a Date\Time format. For all other cells DataDefractor reads the raw cell values.
Full	DataDefractor applies all formatting information stored in the Excel file. The cells in the file are read as they are formatted by Excel.
None	DataDefractor does not use the cell formatting stored in the Excel file. It reads only the raw cell values.

DataDefractor Text File Connection Manager

DataDefractor Text File Connection Manager is used to define a connection to flat text files such as CSV files.

To create a DataDefractor Text File Connection Manager right-click on the Connection Managers design surface in the Business Intelligence Development Studio and choose “**New Connection...**”. In the “**Add SSIS Connection Manager**” dialog box choose to create a connection manager of type `DataDefractor.TextFiles`.



To add files to the connection manager use the “**Add...**” button. You can add as many files as you need. You can also use wildcards to match multiple files at once. If you use wildcards, you can optionally set the “**SearchSubFolders**” property of the item to `True`. This will instruct the connection manager to look in the subfolders of the selected folder for wildcard matches.

Use the “**Column Delimiter**” combo box to define the symbol that is used to delimit the columns in the flat file. You can enter any character or you can choose from the ones predefined in the dropdown list.

Use the “**Text Qualifier**” text box to enter the symbol that is used to enclose text items in the flat file.

Use the “**Locale**” combo box to define the locale for the flat file if it is different than the default.

You can use the “**Preview**” button to see how your flat file(s) will be interpreted.

ODBC Connection Manager

You can use the ODBC connection manager provided with Business Intelligence Development Studio to connect to any relational data source for which there is an ODBC driver.

Starting the DataDefractor Wizard

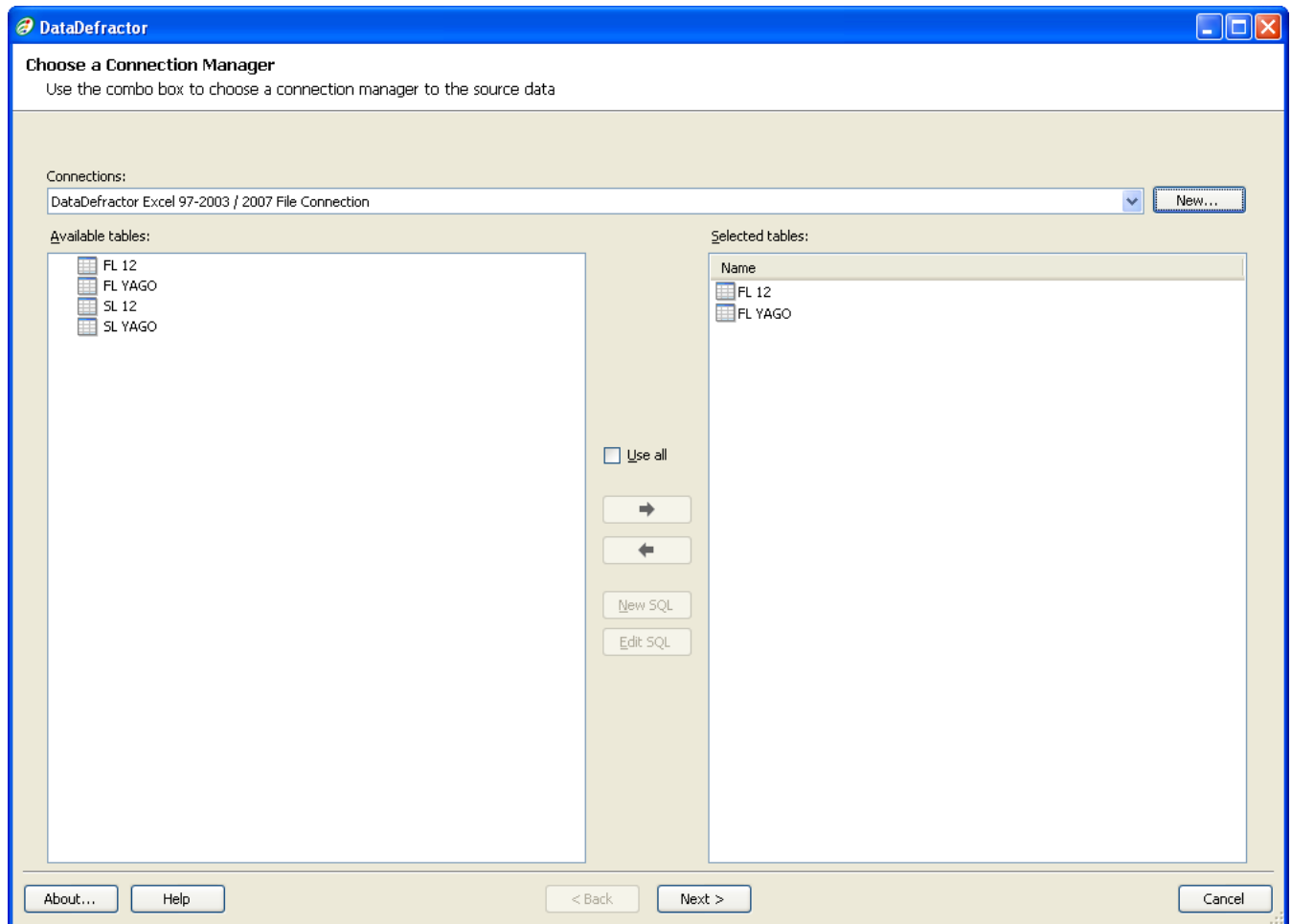
DataDefractor Wizard is the user interface that you will use to model your data source mapping schema. You will also use it to choose data sources for normalization.

There are three ways you can start DataDefractor Wizard:

- Drag a DataDefractor source component from the Business Intelligence Development Studio Toolbox to the design surface of a Data Flow Task. That automatically opens the DataDefractor Wizard.
- Double click on a DataDefractor object on the design surface of a Data Flow Task.
- Right click on a DataDefractor object on the design surface and then choose the “**Edit...**” item from the context menu.

Choosing a Connection Manager

On the first page of DataDefractor Wizard you define the data source.



In the “**Connections**” combo box choose a connection manager that defines a connection to your data. You can also use the “**New...**” button to define a new connection manager.

The left pane shows all the data objects available in the selected connection manager. Data objects can be database tables or views, Excel worksheets, or flat files, depending on the type of connection manager that you have chosen.

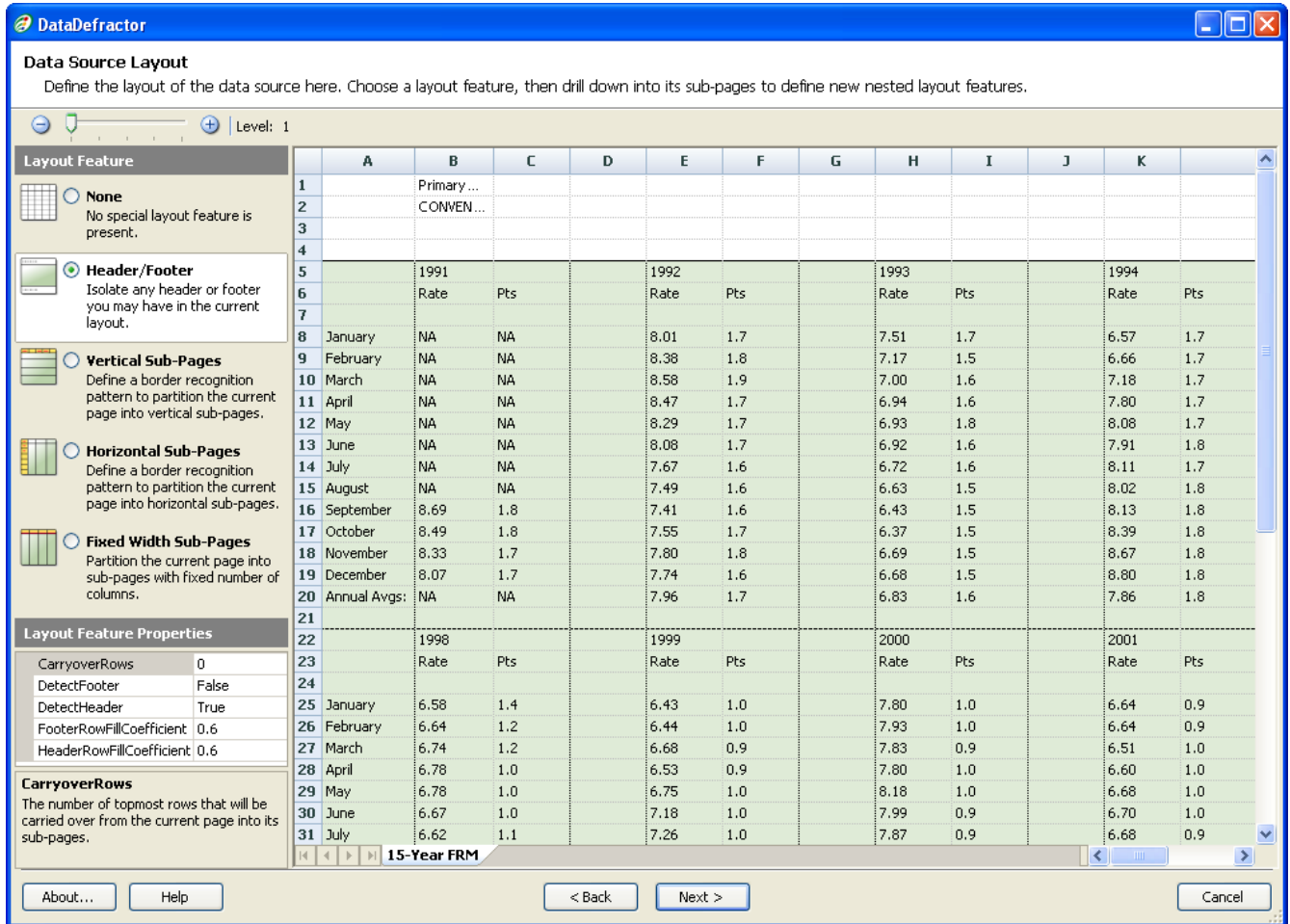
Use the **Right** arrow button to select those source objects that you want to normalize. Note that you will be defining a single mapping schema for each instance of DataDefractor Data Flow Source component used in an SSIS package. This is why the data sources you select on this page have to follow a common layout.

If you want to use all the data objects from the connection manager then check the “**Use all**” check box. This will ensure that when DataDefractor runs it will use all the objects available at runtime, even if the set of available objects have changed since defining the data source in the DataDefractor Wizard.

If the selected connection manager is of type ODBC then you can use the “**New SQL**” and “**Edit SQL**” buttons to define new queries into the relational database to be used by DataDefractor.

Defining the Data Source Layout


The Data Source Layout page is used to define the layout features of the data source, such as headers, footers, horizontal and vertical pages and nested sub-pages.



If your data source includes flexible headers, footers or sub-pages, you can define these features here. A layout feature defines a way to split the data source in sub-pages. There may be many levels of nested layout features and each feature applies to the data at the level for which it is defined.

Choose a layout feature in the left pane, set its properties, and then drill down into its sub-pages using the button, to define new nested layout features.

Keep drilling into the layout's sub-pages until you reach a basic sub-page whose layout contains a single continuous area of fact data surrounded, but unbroken by context information. That last layout level is said to be the **Facts Level**.

You can use the  button do drill up to previous levels.

It is important to note that by defining layout features, you are essentially breaking down the data source into virtual mini-data sources with similar structures. Once this is done, most of the data mapping rules defined in subsequent pages of the wizard will be applied to those lowest **Facts Level** virtual data sources.

Header/Footer Layout Feature

Choose this layout feature if you have a flexible header and/or footer in your data source. This layout feature results in one sub-page which contains the data portion of the data source, excluding any header and/or footer. You can control the resulting subpage by altering the feature’s properties. Following is a list of available properties and their meanings:

DetectHeader	Set this property to <code>True</code> if the data source contains a header. Default value is <code>True</code> .
DetectFooter	Set this property to <code>True</code> if the data source contains a footer. Default value is <code>False</code> .
CarryoverRows	The number of topmost rows to carry over from this sub-page level into the next. This is useful when these rows contain fact context that is relative to the fact position and will be needed at the Facts Level to determine fact context (see “ Relative and Absolute Context Locations ” in chapter “ Understanding Semi-structured Data ”). Default value is 0.
HeaderRowFillCoefficient	This property determines the percentage of non-empty cells required for a row to be considered a part of the data portion of the data source that follows the header. Set this property to a number between 0.0 and 1.0 where 0.0 represents 0% and 1.0 represents 100%. Default value is 0.6
FooterRowFillCoefficient	This property determines the percentage of non-empty cells required for a row to be considered a part of the data portion of the data source that precedes the footer. Set this property to a number between 0.0 and 1.0 where 0.0 represents 0% and 1.0 represents 100%. Default value is 0.6

Vertical Sub-Pages Layout Feature

This feature partitions the data source into sub-pages whose borders are determined by a pattern that appears at the beginning or the ending row of each sub-page. You can define the border pattern by using a special **pattern row** which appears at the top of the display grid. Just define a matching condition in each cell that participates in the border pattern and DataDefractor will partition the data source at the rows that match your pattern.

Consider the following example:

	A	B	C	D	E
	Empty	^\d{4}\$	Empty	No matc	No matc
1		Primary Mo...			
2		CONVENTI...			
3					
4					
5		1991			1992
6		Rate	Pts		Rate
7					
8	January	NA	NA		8.01
9	February	NA	NA		8.38
10	March	NA	NA		8.58
11	April	NA	NA		8.47
12	May	NA	NA		8.29
13	June	NA	NA		8.08
14	July	NA	NA		7.67
15	August	NA	NA		7.49
16	September	8.69	1.8		7.41
17	October	8.49	1.8		7.55
18	November	8.33	1.7		7.80
19	December	8.07	1.7		7.74
20	Annual Avgs:	NA	NA		7.96
21					
22		1998			1999
23		Rate	Pts		Rate
24					
25	January	6.58	1.4		6.43
26	February	6.64	1.2		6.44
27	March	6.74	1.2		6.68
28	April	6.78	1.0		6.53
29	May	6.78	1.0		6.75
30	June	6.67	1.0		7.18

The row pattern used here separates the data source into sub-pages at each row which has an empty first cell, and has exactly 4 digits in its second cell and has an empty third cell. The rest of the cells in the row pattern have no conditions defined – these cells are ignored by DataDefractor when it determines the sub- page borders. As you can see in this example rows **5** and **22** match the pattern defined in the **pattern row**.

Use the dropdown menus in the cells of the **pattern row** to define different matching conditions for each cell.

If you do not want to use a cell in the matching pattern, leave the “**No matching specified**” condition which is set for each cell by default. If you have defined a condition in a pattern cell and you want to remove it, click on the dropdown menu for that cell and select “**Clear Matching Pattern For This Cell**”.

If you want to match an empty cell, click on the dropdown menu and select “**Match Empty Cell**”.

If you want a cell to match a regular expression, type that regular expression in the corresponding cell of the **pattern row**.

Tip: If you want a cell to match one of several values, use an alternating regular expression by using the pipe symbol | to separate the different possible values. For example “cat|dog” will match a cell, which contains “cat” or “dog”.

You can control the way DataDefractor partitions the data source into sub-pages by altering the feature’s properties. Following is a list of available properties and their meanings:

PatternDirection	<p>This value indicates the direction in which DataDefractor scans the data source looking for sub-page borders..</p> <p>If this value is set to <code>TopToBottom</code>, DataDefractor scans the data source from top to bottom and each row that matches the row pattern is considered the first row of a sub-page.</p> <p>If this value is set to <code>BottomToTop</code>, DataDefractor scans the data source from bottom to top and each row that matches the row pattern is considered the last row of a sub-page.</p> <p>Default value is <code>TopToBottom</code>.</p>
SkipRows	<p>This value indicates the number of topmost rows to skip before the matching of the row pattern begins.</p> <p>Default value is 0.</p>
CarryoverRows	<p>The number of topmost rows to carry over from this layout level into the next. These rows will appear as the topmost rows of every sub-page. This is useful when these rows contain context that is relative to the position of the facts and will be needed at the Facts Level to determine fact context (see “Relative and Absolute Context Locations” in “Understanding Semi-structured Data”).</p> <p>Default value is 0.</p>
IncludeLeftoverSubpage	<p>If this value is set to <code>True</code> and a portion of the data does not appear in a sub-page then this portion will be included in the resulting set of sub-pages as a separate sub-page. Otherwise that portion of the data will not be included in the resulting sub-pages.</p> <p>Default value is <code>False</code>.</p>

MatchSingleSubpage	If this value is set to <code>True</code> , DataDefractor stops scanning for sub-page borders after the first border match. As a result, the data source is partitioned into a single sub-page, which either begins or ends at the matching row, depending on the value of PatternDirection . If the value of MatchSingleSubpage is set to <code>False</code> , DataDefractor scans the entire data source looking for sub-page borders. Default value is <code>False</code> .
---------------------------	--

Horizontal Sub-Pages Layout Feature

This feature divides the data source in sub-pages whose borders are determined by a pattern that appears at the beginning or the ending column of each sub-page. You can define the border pattern by using a special **pattern column** located in the left portion of the display grid. Just define a matching condition in each cell that participates in the border pattern and DataDefractor will partition the data source at the columns that match your pattern.

Consider the following example:

	⌘ Pattern		A	B	C	D	E	F	G	H	I
1	^\d{4}\$			1991			1992			1993	
2	No matching spe			Rate	Pts		Rate	Pts		Rate	Pts
3	Empty										
4	No matching spe		January	NA	NA		8.01	1.7		7.51	1.7
5	No matching spe		February	NA	NA		8.38	1.8		7.17	1.5
6	No matching spe		March	NA	NA		8.58	1.9		7.00	1.6
7	No matching spe		April	NA	NA		8.47	1.7		6.94	1.6
8	No matching spe		May	NA	NA		8.29	1.7		6.93	1.8
9	No matching spe		June	NA	NA		8.08	1.7		6.92	1.6
10	No matching spe		July	NA	NA		7.67	1.6		6.72	1.6
11	No matching spe		August	NA	NA		7.49	1.6		6.63	1.5

The column pattern used here splits the data source into sub-pages at every column, which has exactly 4 digits in its first cell and has an empty third cell. The rest of the cells in the column pattern have no conditions defined – these cells are ignored by DataDefractor when it determines the sub-page borders. As you can see in this example, columns **B**, **E** and **H** match the pattern defined in the **pattern column**.

Use the dropdown menus in the cells of the pattern column to define different matching conditions for each cell.

If you want to exclude a cell from the matching process, leave the “**No matching specified**” condition which is set for each cell by default. If you have defined a condition in a pattern cell and you want to remove it, click the dropdown menu and select “**Clear Matching Pattern For This Cell**”.

If you want to match an empty cell, click on the dropdown menu and select “**Match Empty Cell**”.

If you want a cell needs to match a regular expression, type that regular expression in the corresponding cell of the **pattern column**.

Tip: If you want a cell to match one of several values, use an alternating regular expression by using the pipe symbol | to separate the different possible values. For example “red|blue” will match a cell, which contains “red” or “blue”.

You can control the way DataDefractor partitions the data source into sub-pages by altering the feature’s properties. Following is a list of available properties and their meanings:

PatternDirection	This value indicates the direction in which DataDefractor scans the data source looking for sub-page borders. If this value is set to <code>LeftToRight</code> , DataDefractor scans the data source from left to right and each column that matches the column pattern is considered the first column of a sub-page. If this value is set to <code>BottomToTop</code> , DataDefractor scans the data source from right to left and each column that matches the column pattern is considered the last column of a sub-page. Default value is <code>LeftToRight</code> .
SkipColumns	This value indicates the number of leftmost columns to skip before the matching of the column pattern begins. Default value is 0.
CarryoverColumns	The number of leftmost columns to carry over from this layout level into the next. These columns will appear as the leftmost columns of every sub-page. This is useful when these columns contain context that is relative to the position of the facts and will be needed at the Facts Level to determine fact context (see “ Relative and Absolute Context Locations ” in “ Understanding Semi-structured Data ”). Default value is 0.
IncludeLeftoverSubpage	If this value is set to <code>True</code> and a portion of the data does not appear in a sub-page then this portion will be included in the resulting set of sub-pages as a separate sub-page. Otherwise that portion of the data will not be included in the resulting sub-pages. Default value is <code>False</code> .
MatchSingleSubpage	If this value is set to <code>True</code> , DataDefractor stops scanning for sub-page borders after the first border match. As a result, the data source is partitioned into a single sub-page, which either begins or ends at the matching column, depending on the value of PatternDirection . If the value of MatchSinglePage is set to <code>False</code> , DataDefractor scans the entire data source looking for sub-page borders. Default value is <code>False</code> .

Fixed Width Sub-Pages Layout Feature

Choose this layout feature if your data source contains sub-pages with a constant number of columns. You can control the way DataDefractor partitions the data source into sub-pages by altering the feature’s properties:

SubpageWidth	Defines the number of columns in a sub-page. Default value is 1.
---------------------	---

SkipColumns	This value indicates the number of columns to skip before the partitioning of the data source into sub-pages begins. Default value is 0.
CarryoverColumns	The number of leftmost columns to carry over from this layout level into the next. These columns will appear as the leftmost columns of every sub-page. This is useful when these columns contain context that is relative to the position of the facts and will be needed at the Facts Level to determine fact context (see “ Relative and Absolute Context Locations ” in “ Understanding Semi-structured Data ”). Default value is 0.
IncludeLeftoverSubpage	If this value is set to <code>True</code> and a portion of the data does not appear in a sub-page then this portion will be included in the resulting set of sub-pages as a separate sub-page. Otherwise that portion of the data will not be included in the resulting sub-pages. Default value is <code>False</code> .

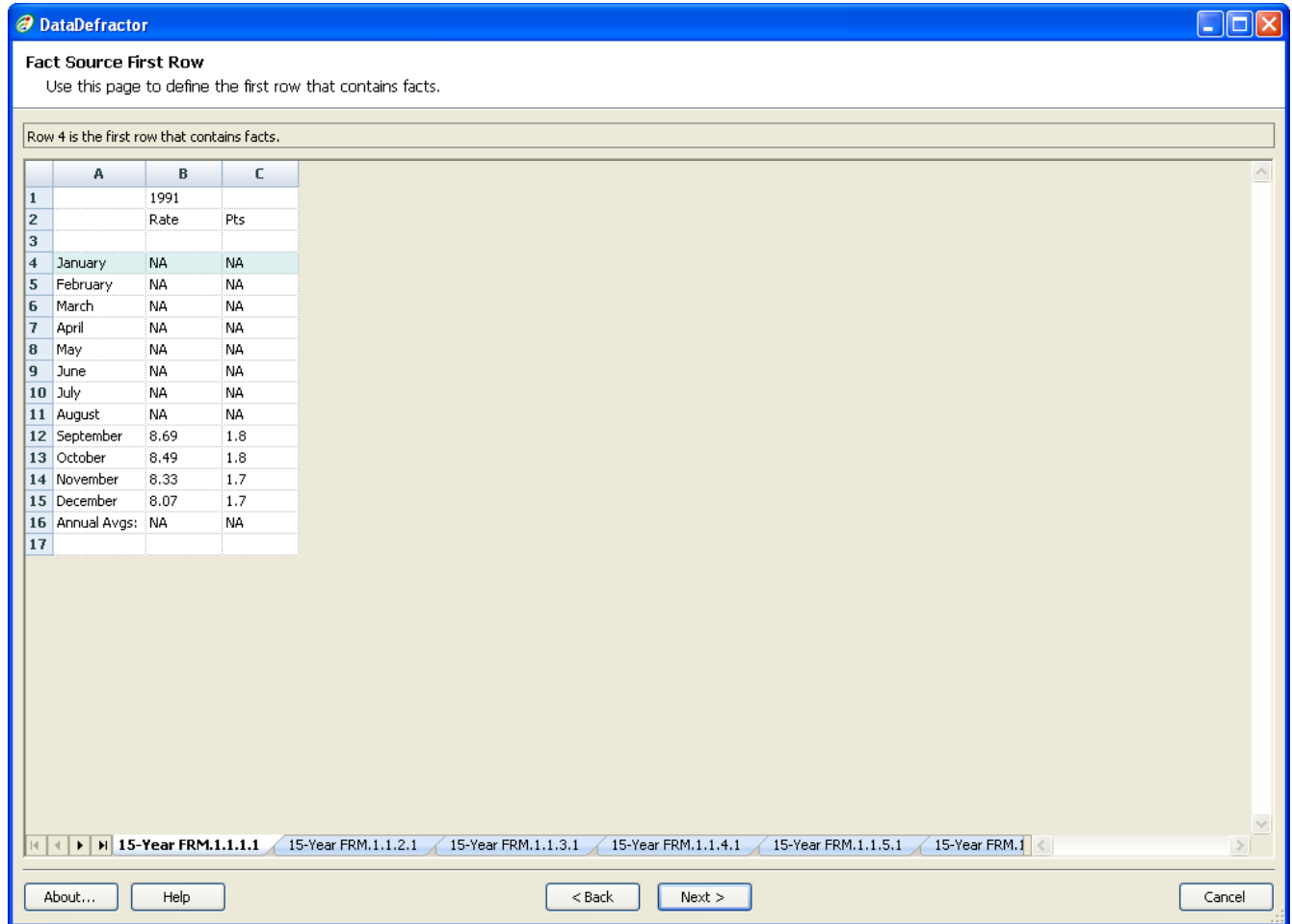
Defining the Fact Area

The fact area in a data source is characterized by the first row that contains facts and the ranges of columns that contain facts. If the data source has layout features defined, then the fact area is defined relative to the last layout level, i.e. the **Facts Level**.

Defining the First Fact Row

The **Fact Source First Row** page is used to select the first row of facts in your data source.

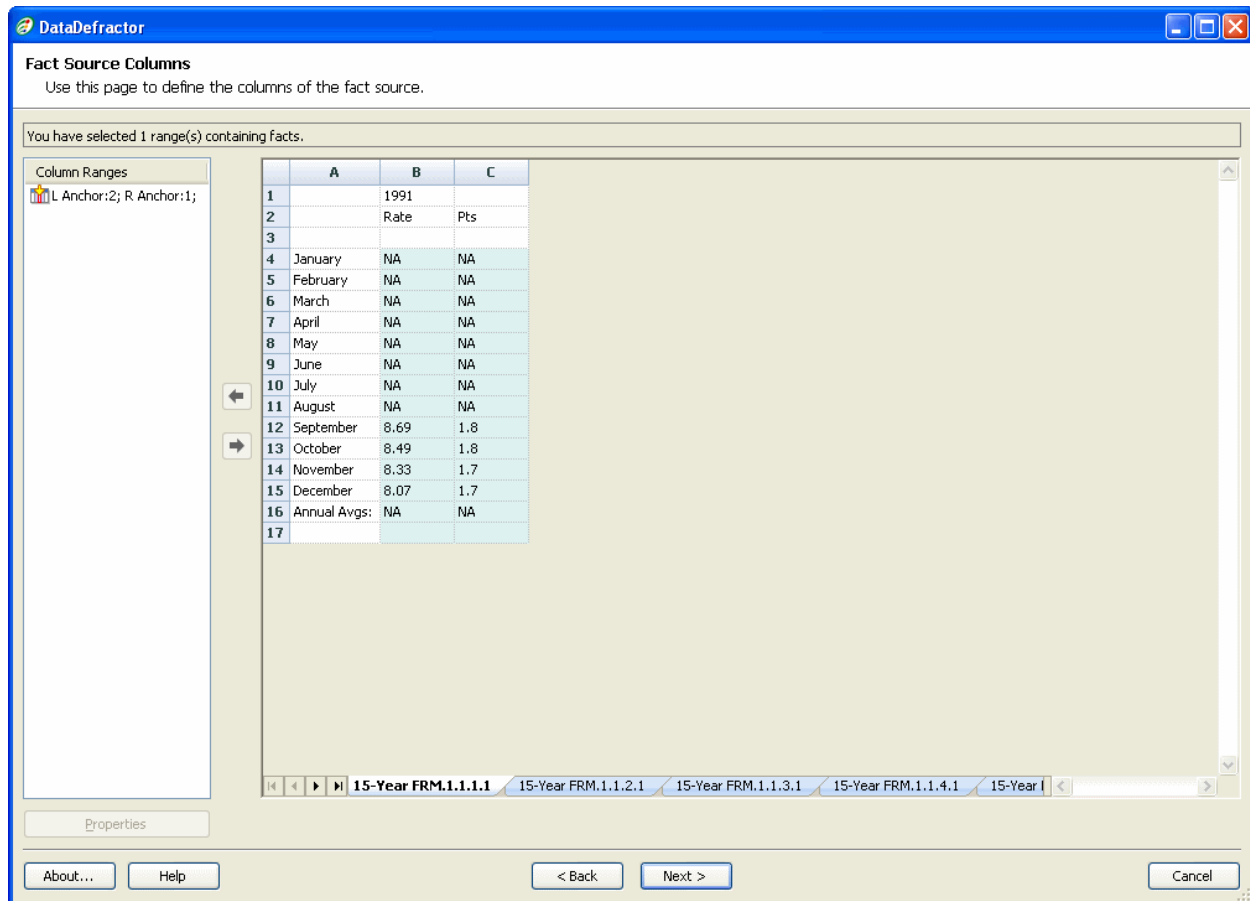
DataDefractor’s normalization procedure will begin reading fact data starting with this row and continue until it reaches the bottom of the **Facts Level** sub-page.



Select the first row that contains facts in your data source. If you have multiple sub-pages at the **Facts Level** then verify that the selected row is the first fact row for all of them, by selecting different sub-pages using the tabs at the bottom of the display grid. If the first row selection does not correctly apply to all sub-pages, then you may need to redefine the data source layout using the **Data Source Layout** page (see “**Defining the Data Source Layout**”).

Defining the Fact Source Columns

The **Fact Source Columns** page is used to define the fact columns in the data source. Because the fact data in a worksheet can often change position or size, these area selections will allow you to create flexible fact ranges that "adapt" to any additional columns that may appear in the **Facts Level** layout.



To define the data source fact columns:

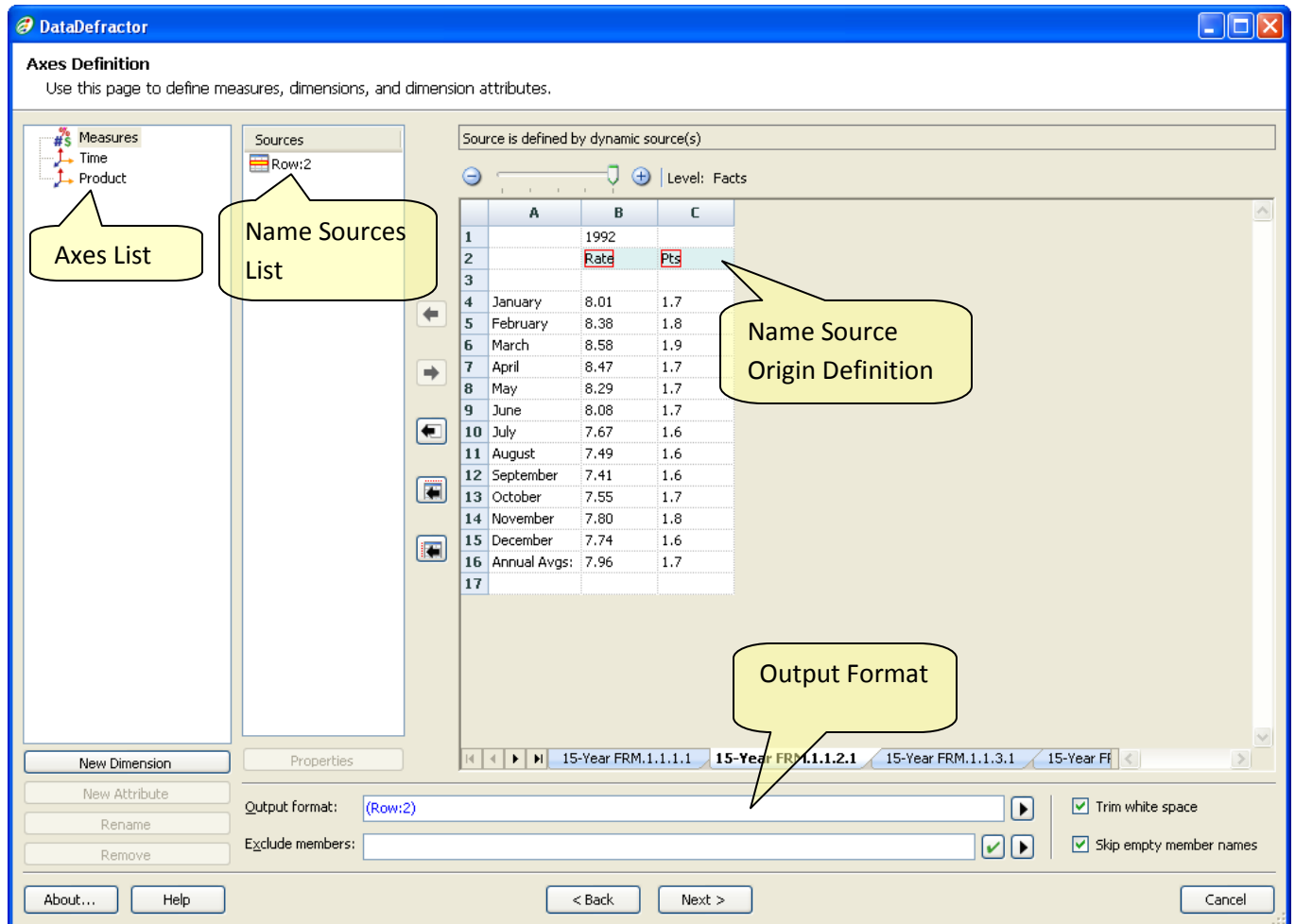
1. Select the columns that you would like to add to the list of fact columns.
2. Use the left arrow button to add the selected range to the fact columns ranges. The **Column Range Type** dialog will appear.
3. Select the method used to address the selected group of columns. The addressing method is important when you attempt to normalize new data sources using this same mapping schema. If fact columns are added, removed or renamed, the correct addressing method will allow the mapping schema to adapt automatically without any reconfiguration. The following options are available:

Flexible Width	Allows you to specify a group of columns that dynamically include new columns (e.g. include all columns from the fifth column to the last column in the data source). This type of column range is said to be anchored to both the left and the right side of the data source.
Fixed Width: Left Anchored	Allows you to define a column range with a fixed number of columns starting from a specific column counting from the left (e.g. include 2 columns starting at the 4 th column from the left of the data source). This type of column range is said to be anchored to the left side of the data source.
Fixed Width: Right Anchored	Allows you to define a column range with a fixed number of columns starting from a specific column counting from the right of the table (e.g. include 2 columns starting at the 4 th column from the right of the table). This type of column range is said to be anchored to the right side of the data source.
Named Column	Allows you to specify certain columns based on their name, regardless of position (e.g. include the columns named "Sales" and "Units").

You can add as many column ranges as you need. Keep repeating steps 1 through 3 until you have captured the entire fact area. If you have multiple sub-pages at the **Facts Level**, verify that the facts area is correct for all of them, by selecting different sub-pages using the tabs at the bottom of the display grid.

Defining the Dimensional Model

The **Axes Definition** page is used to define the dimensional model of your data. Here you map out the measures, dimensions, and their attributes by identifying the places where they appear in the data source.



Axes List

In this area you define the dimension axes of your data model. Here you also define the attributes of the dimensions. A special **Measures** axis is pre-created for you.

You can create new dimensions using the “**New Dimension**” button.

You can create attributes for a dimension by selecting the dimension in the **Axes List** and then pushing the “**New Attribute**” button.









You can rename dimension axes or attributes by selecting them in the **Axes List** and then pushing the **“Rename”** button or pressing the **F2** key on your keyboard.





You can delete dimension axes or attributes from the model by selecting them in the **Axes List** and then pushing the **“Remove”** button or pressing the **Del** keyboard key.

Name Sources List and Name Source Origin Definition

A name of a measure, a dimension member or an attribute is formed by the context information related to the facts in the data. These names can be formed by combining context coming from various name sources (see **“Relative and Absolute Context Locations”** in **“Understanding Semi-structured Data”**). You add these name sources to the **Name Sources List** of a given axis or attribute by making the appropriate selections in the **Name Source Origin Definition** grid and then using the buttons located between the **Name Sources List** and the **Name Source Origin Definition**.

Use the following table for instructions on how to add a name source based on its origin:

Name Source Origin	Instructions to Add to Name Sources List
Single Cell	<p>If a name source is located in a single cell then follow these steps to add it to the Name Sources List:</p> <ol style="list-style-type: none"> 1. Select the deepest layout level at which the cell appears by using the  and  buttons located above the Name Source Origin Definition grid. 2. Select the name source cell. 3. Push the  button or double-click the name source cell.
Relative Horizontal	<p>If a name source defines a relative horizontal context (i.e. names are coming from a row above the facts) then follow these steps to add it to the Name Sources List:</p> <ol style="list-style-type: none"> 1. Use the  button to drill down to the Facts Level. 2. Select the whole row by clicking on the row number in the Name Source Origin Definition grid. 3. Push the  button or double-click the row number.
Relative Vertical	<p>If a name source defines a relative vertical context (i.e. names are coming from a column) then follow these steps to add it to the Name Sources List:</p> <ol style="list-style-type: none"> 1. Use the  button to drill down to the Facts Level. 2. Select the whole column by clicking on the column header in the Name Source Origin Definition grid. 3. Push the  button or double-click the column header.
Data Source Name	<p>If a part or whole of the name is located in the data source name (i.e. worksheet name, flat file name or a database table name) then push the  button to add a Data Source Name source to the Name Sources List.</p>

Fact Column Position	<p>This type of naming source is useful when the naming context is horizontal but there is no row that contains the names of the axis member or attribute values. If a name is defined by the position of the column where the fact appears, follow these steps to add a Fact Column Position naming source to the Name Sources List:</p> <ol style="list-style-type: none"> 1. Use the  button to drill down to the Facts Level. 2. Push the  button to add a Fact Column Position naming source.
Fact Row Position	<p>This type of naming source is useful when the naming context is vertical but there is no column that contains the names of the axis member or attribute values. If a name is defined by the number of the row where the fact appears then follow these steps to add a Fact Row Position naming source to the Name Sources List:</p> <ol style="list-style-type: none"> 1. Use the  button to drill down to the Facts Level. 2. Push the  button to add a Fact Row Position naming source.


You can define as many naming sources for an axis or an attribute as you need. You can later combine them in the **Output Format** box to form a final axis member name or attribute value (see Output Format below).

You can define single cell sources at any sub-page level. For example, you can use a cell that is found in the header portion of a data source by drilling up to the sub-page level that contains that cell and then adding it to the **Name Source List**. However, you can add relative and fact position dependent sources only at the **Facts Level** since they are dependent on the positions of the facts.

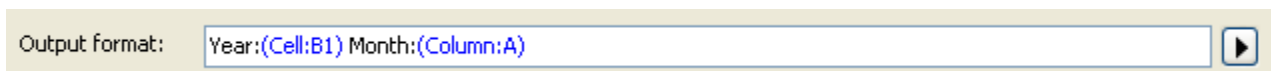
If your data source contains relative name sources like rows or columns that are left outside of the **Facts Level**, then go back to the **Data Source Layout** page and use the **CarryoverRows** or **CarryoverColumns** properties of the appropriate layout features, so that these rows or columns get carried over to the **Fact Level** (see “**Defining the Data Source Layout**”). Once at the fact level, these rows or columns can be used as relative name source origins.

Output Format

Use the Output Format box to define how to construct the axis member names out of the available name sources.

You can use the  button located to the right of the Output Format box to select from a list of available name sources to paste in the box.

You can arrange the name sources in any order, you can repeat name sources and you can add arbitrary text anywhere in the **Output Format** box. The following example constructs the name by taking the contents of two name sources and merging them with constant text.



Tip: if your axis has a single member with a constant name then do not add name sources to the **Name Sources List** but rather enter the member name in the Output Format box.

Exclude Members

Use this box to define a matching regular expression to filter out facts for selected axis members. Any axis member, whose name matches the regular expression will be excluded from the final SSIS output.

The following example demonstrates how to exclude members of an axis whose names contain either “1991” or “1992”:

Exclude members:

Trim White Space

Use the **Trim White Space** checkbox to trim any whitespace found at the beginning or the ending of a member name or attribute value. This box is checked by default.

Skip Empty Member Names

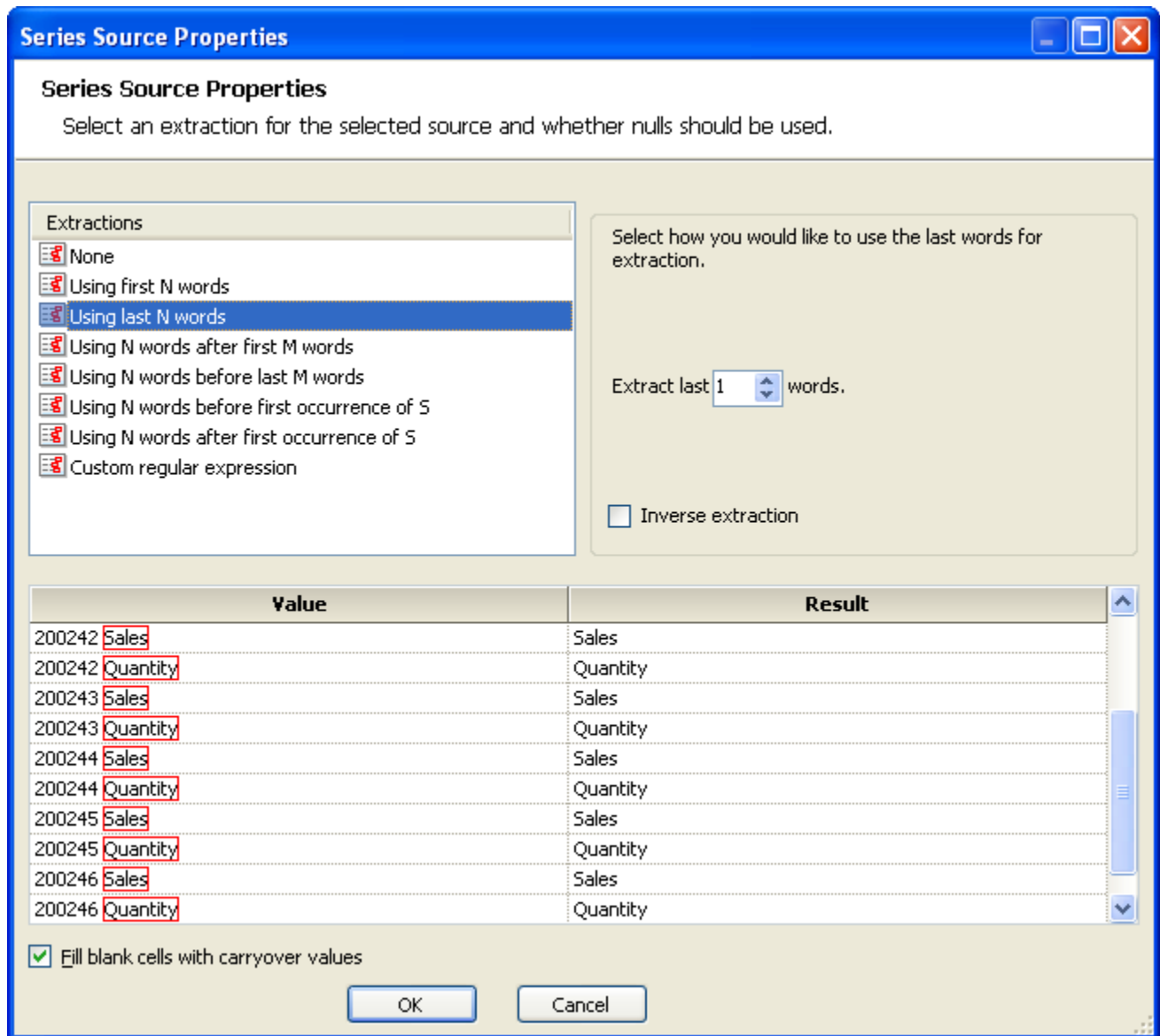
If this checkbox is checked, DataDefractor will skip member names that are empty. This box is checked by default and is available only for axis members, i.e. it is not available for attribute values.

Name Source Properties

Sometimes you need to fine-tune the output of a name source. For example you may need to extract just a part of the contents of a given cell. For this purpose you can use the **Name Source Properties** dialog box.

To open the **Name Source Properties** dialog box select the name source in the **Name Source List** and push the “**Properties**” button located below the **Name Source List**. Alternatively you can double-click the name source whose properties you want to edit.

Following is an example screen shot of the **Name Source Properties** dialog box:

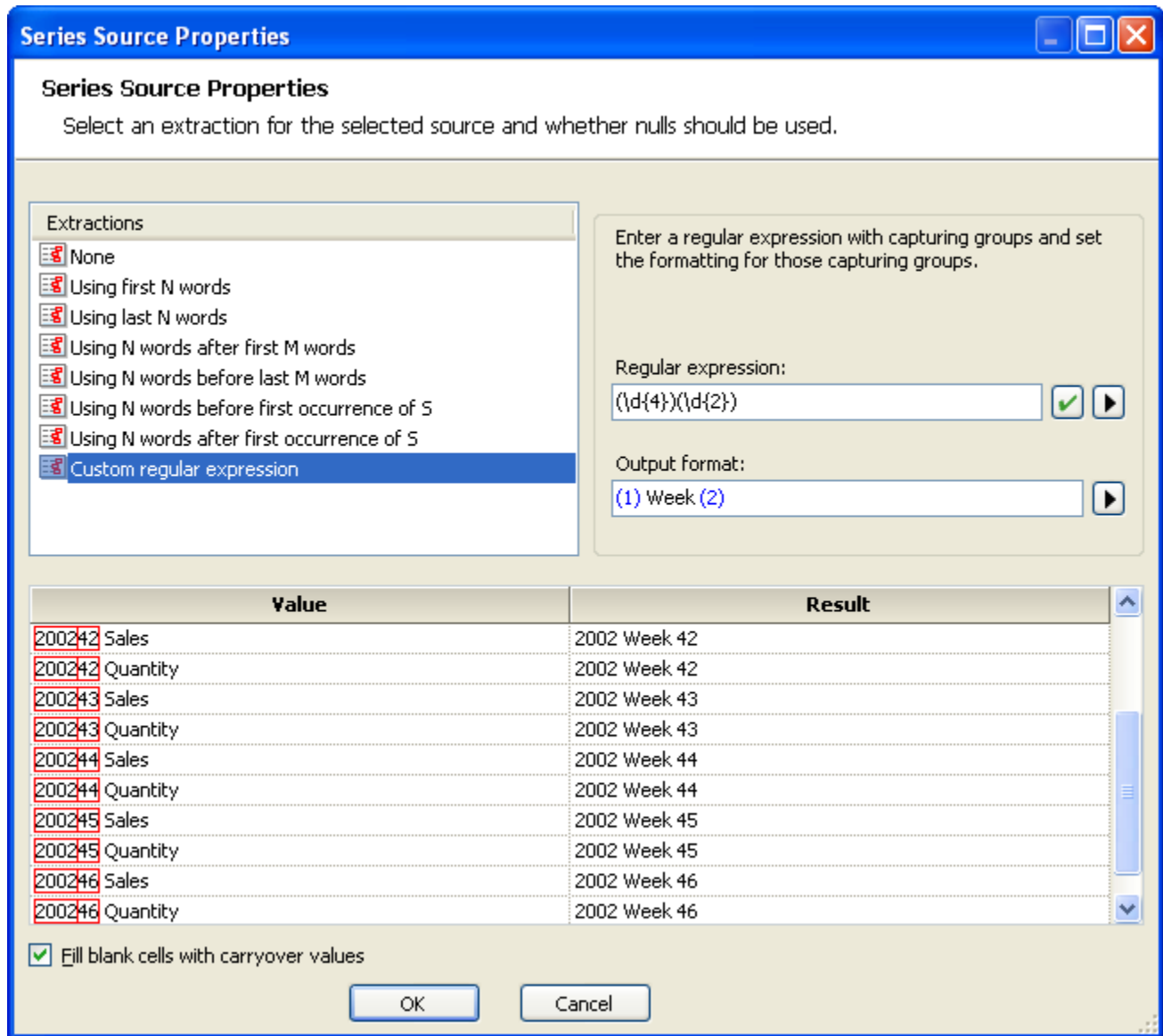


You can choose an extraction type from the ones listed in the **Extractions List**. You can control each extraction by modifying its properties, which appear in the right portion of the dialog box. The sample input values along with the results of the extraction are displayed in the bottom half of the dialog box.


The above example is taken from the POS BigMart sample SSIS package installed with DataDefractor. It demonstrates how you can construct the names of the POS measures by extracting the last word from the cells where they are stored together with the time period.

You may encounter extraction requirements that could not be met by any of the predefined extractions in the **Extractions List**. Then you can use the **Custom regular expression** extraction to define your own extraction rules.

Here’s an example screen shot of the **Custom regular expression** extraction from the same POS BigMart sample:



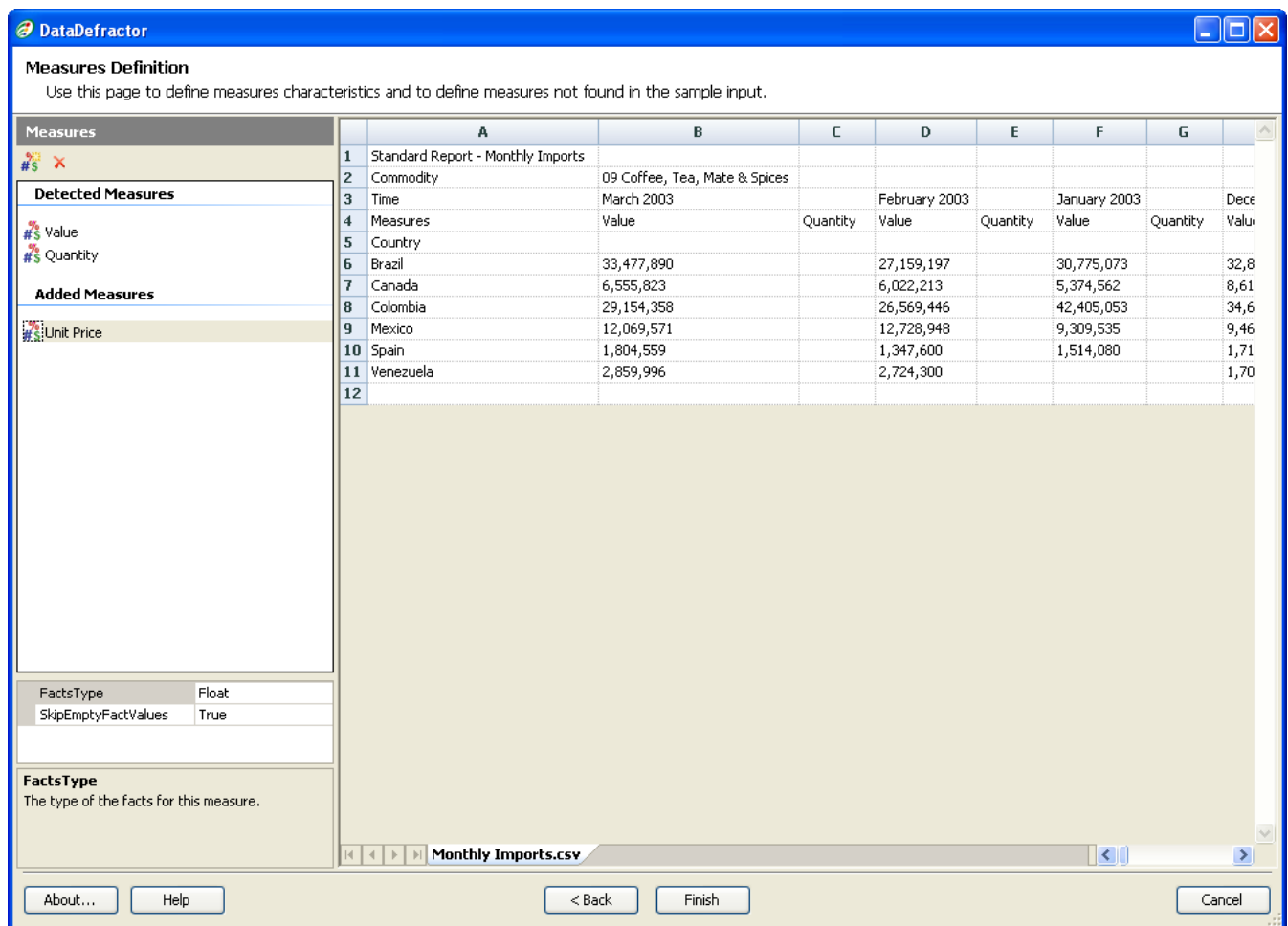
Here the extraction is defined to form the time dimension’s naming source by extracting the year and the week number from the cells. This is achieved by a regular expression that captures the first four digits and the next two digits in two separate capturing groups. Then these capturing groups, appearing as (1) and (2) in the **Output format box**, are used to construct a user friendly output.


To capture parts of the expression in capturing groups, put these parts in parenthesis. Then use the  button to the right of the **Output format box** to pick from the available capturing groups. You can combine these groups with freeform text in the **Output format box** to construct the desired result.

The “**Fill blank cells with carryover values**” checkbox is useful when selecting horizontal or vertical relative naming sources. It allows you to specify whether the normalization engine should consider blank cells as blank or should it fill them up with the previous non-empty value extracted from that name source. If it is checked, when a blank cell is encountered, DataDefractor will attempt to search the cells above (in a column selection) or to the left (in a row selection) until a non-empty value is found, and will use that value instead of an empty value. This box is checked by default for axis naming sources and is unchecked by default for attribute values naming sources.


Setting up the Measures

In the **Measures Definition** page you can setup different properties of the measures of your mapping schema.



DataDefractor wizard scans the top 500 rows of each table of the data source to detect measures names. In most cases that is enough to detect all the measures from the data sources. However DataDefractor may miss measures names, which appear below these top rows. In this case you should manually add these measures by pushing the  button located at the top of the **Measures List**. You

can rename an added measure by clicking on its name in the **Measures List** or by selecting it and then pressing the **F2** key on your keyboard.

You can also remove added measures by selecting them and pushing the  button or pressing the **Del** keyboard key.

You cannot rename or remove measures detected in the sample data sources.

Each measure has properties that you can modify in order to customize the way fact values are being processed. You can modify the properties by setting their values in the property grid located below the **Measures List**. Following is a list of available properties:

FactsType	<p>Determines the type of the fact values. Possible values are:</p> <ul style="list-style-type: none"> • Float – the facts are of floating point type. Values that cannot be converted to floating point type are considered “not a number” or NaN. • Text – the facts are of textual type. <p>This value is Float by default.</p>
SkipEmptyFactValues	<p>Determines if values that are empty (i.e. NaN for Float type or zero length strings for type Text) are omitted from the output. If this property is set to <code>True</code> then the facts with empty values will be skipped along with their corresponding context axis names. Otherwise the empty fact values will be present in the output.</p> <p>The default value is <code>True</code>.</p>
TrimWhiteSpace	<p>This property is only available for measures of type Text. It determines whether to trim any white space (like space or tab characters) found at the beginning or ending of the textual fact values.</p> <p>The default value is <code>True</code>.</p>

TrimWhiteSpace property is processed before **SkipEmptyFactValues**. This means that if both properties are set to `True` (the default settings) for a measure of type **Text** and a value is encountered that contains only white space characters, like spaces, or tabs, then that value is going to be skipped.

SSIS Outputs

When you push the “**Finish**” button at the **Measures Definition** page DataDefractor data flow source component creates one SSIS output for the facts data and one SSIS output for each dimension defined in the mapping schema. This structure of outputs is in the form of a data warehouse star schema. The following sections discuss the contents of each of these output types.

Facts Output

The facts output, as the name implies, is the output which contains the factual data.

For every dimension defined in the mapping schema DataDefractor creates two columns in the `Facts` Output. One of these columns contains the names of the members of the dimension and the other contains generated integer surrogate keys for these members. These two columns in effect contain foreign keys to the corresponding dimension outputs. Typically a subsequent data flow would use either one of these columns and they are provided both to accommodate a wide variety of data flow requirements.

For each measure defined in the mapping schema DataDefractor creates one column in the `Facts` Output. This column is of type “double-precision float [DT_R8]” or “Unicode string [DT_WSTR]” depending on the measure type selected in the **Measures Definition** page of the DataDefractor Wizard.

Each record in the `Facts` Output contains the facts for the corresponding measures found at the intersection of the dimensions defined by the dimension keys in the same record. The `Facts` Output contains only records for dimension intersections where facts were found.

Following is a sample `Facts` Output data viewer screenshot:

Commodity ID	Commodity	Country ID	Country	Time Period ID	Time Period	Quantity	Value
1	090111001...	1	Brazil	1	March 2003	20380554	18005890
1	090111001...	1	Brazil	2	February 2003	15936825	13794614
1	090111001...	1	Brazil	3	January 2003	16984535	15105011
1	090111001...	1	Brazil	4	December 2002	18111525	16186210
1	090111001...	1	Brazil	5	November 2002	15962192	12453320
1	090111001...	2	Canada	3	January 2003	1497	5280
1	090111001...	3	Colombia	1	March 2003	11348232	17194329
1	090111001...	3	Colombia	2	February 2003	11812765	18158445
1	090111001...	3	Colombia	3	January 2003	15641374	23845101
1	090111001...	3	Colombia	4	December 2002	16352135	24763226

Attached Total rows: 1215, buffers: 1 Rows displayed = 1215

In this example “Commodity ID”, “Country ID”, and “Time Period ID” are the columns that contain the surrogate keys for the dimensions’ members. “Commodity”, “Country”, and “Time Period” are the columns containing the member names of the corresponding dimensions. “Quantity” and “Value” columns contain the facts for the `Quantity` and `Value` measures.

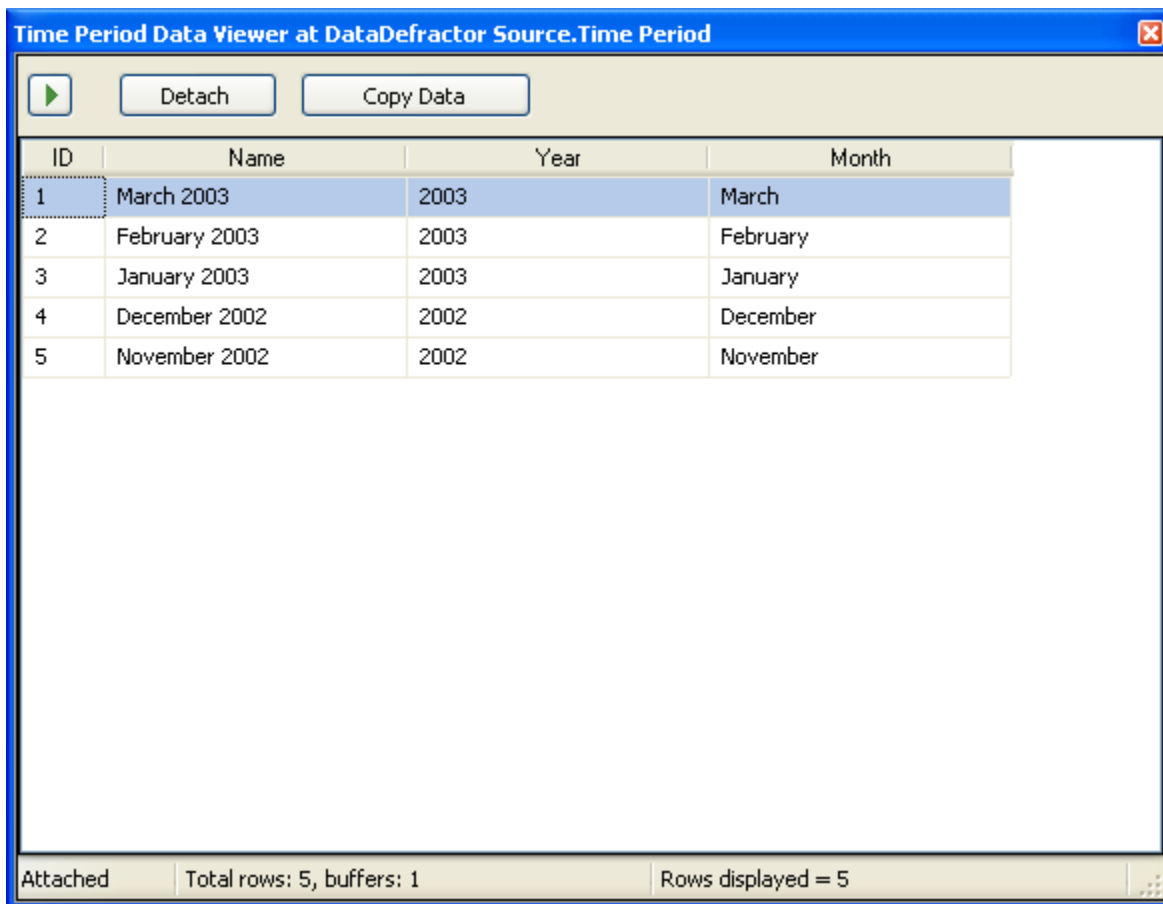
Dimension Outputs

Every dimension output has a `Name` column that contains the member names for that dimension. The contents of this column is unique, i.e. there are no duplicates in it. It could be used as the primary key of the dimension.

A dimension output also has an `ID` column that contains a generated surrogate key of integer type. This column is also unique and could be used as the primary key for the dimension.

The dimension output also contains one column for each attribute defined in the DataDefractor mapping schema for that dimension.

Following is a screenshot of the `Time Period` dimension’s output data viewer:



The screenshot shows a window titled "Time Period Data Viewer at DataDefractor Source.Time Period". The window contains a table with the following data:

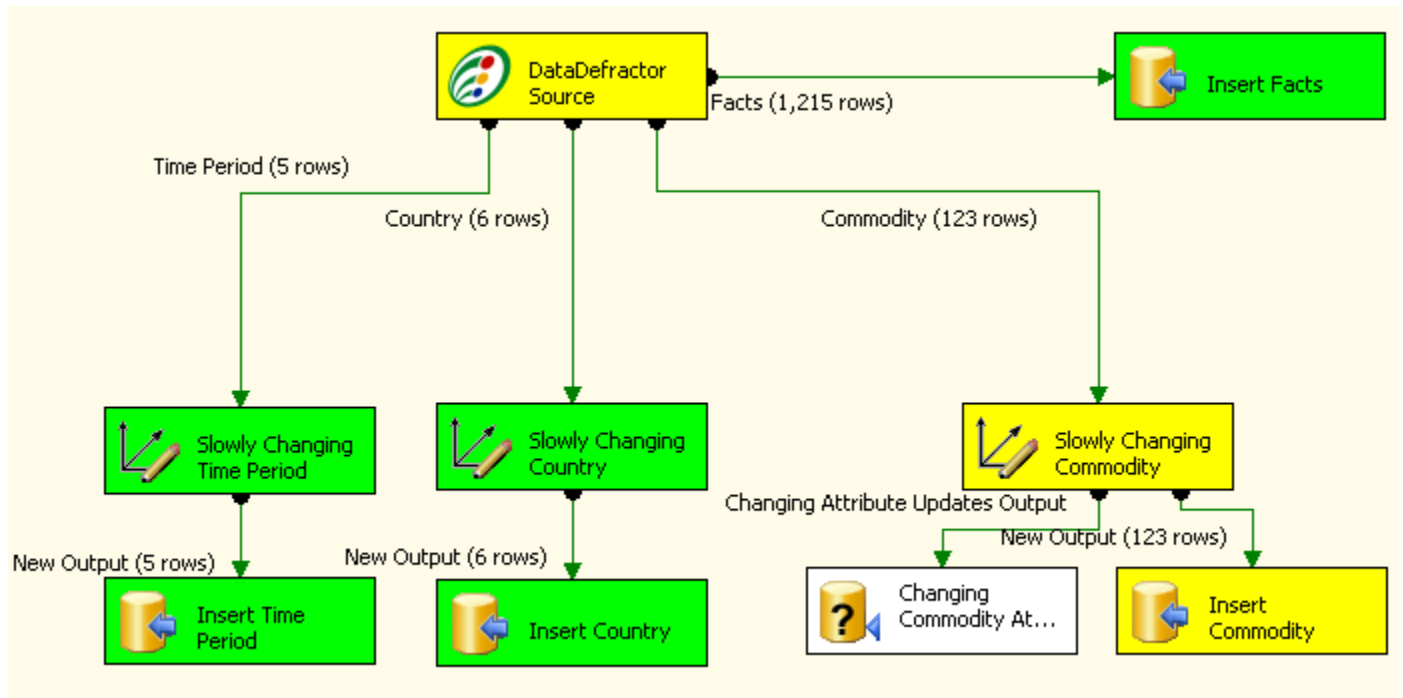
ID	Name	Year	Month
1	March 2003	2003	March
2	February 2003	2003	February
3	January 2003	2003	January
4	December 2002	2002	December
5	November 2002	2002	November

At the bottom of the window, there is a status bar with the following information: Attached, Total rows: 5, buffers: 1, Rows displayed = 5.

As you can see the `ID` column contains the surrogate keys for the dimension’s members while the `Name` column contains the member names. The `Year` and `Month` columns are attribute columns that contain the year and the month attribute for each member of the `Time Period` dimension.

Outputs Usage Example

The following screenshot demonstrates DataDefractor in action. It illustrates how DataDefractor’s SSIS outputs can be used to populate a data warehouse with the multidimensional star schema generated by the component:



The dimension outputs “Time Period”, “Country” and “Commodity” are used to populate the dimension tables of the data warehouse star schema using slowly changing dimension transformations. The “Facts” output is used to populate the fact table of the data warehouse star schema using an OLE DB Destination component.

Acknowledgements

XP3 is registered trademark of Interactive Edge LLC; DataDefractor is a trademark of Interactive Edge, LLC.

All rights to the marks, the content appearing on the Freddie Mac web sites, and the look and feel of the Freddie Mac web sites belong to Freddie Mac and/or its third party licensors.

Microsoft, Windows, Visual Studio, Excel, PivotTable and .NET logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other marks are properties of their respective owners.

Copyright© 2007 by Interactive Edge, LLC. All rights reserved.

Bibliography

Wikipedia. (n.d.). *First normal form*. Retrieved Jan 24, 2007, from Wikipedia:
http://en.wikipedia.org/wiki/First_normal_form

Wikipedia. (n.d.). *Star schema*. Retrieved March 12, 2007, from Wikipedia:
http://en.wikipedia.org/wiki/Star_schema