

DS4DB 4.4

DataSource for Databases Overview

March 2009

Contents

1	Preface.....	1
1.1	What this document contains.....	1
	About Caplin document formats	1
1.2	Who should read this document.....	2
1.3	Related documents.....	2
1.4	Feedback.....	2
1.5	Acknowledgments.....	2
2	What is DataSource for Databases?.....	3
2.1	Features of DataSource for Databases.....	3
2.2	What can you use DataSource for Databases for?	4
3	Architecture.....	5
4	Concepts.....	7
4.1	Events.....	7
4.2	Database accessing.....	7
4.3	Result set processing.....	8
4.4	Processing flow.....	9
4.5	Parallel processing.....	10
5	Examples of use.....	12
5.1	Timed Update example.....	12
5.2	Index Distribution example	13
6	Glossary of terms and acronyms.....	14

1 Preface

1.1 What this document contains

This document provides an overview of the Caplin DataSource for Databases product, version 4.4.

It explains:

- ◆ what DataSource for Databases is and what it can be used for,
- ◆ how it fits into the overall Caplin product architecture,
- ◆ key concepts relating to the product.

It also gives some examples of how the product can be used in real business situations.

Note: In this document the DataSource for Databases product is also referred to by the acronym "DS4DB".

About Caplin document formats

This document is supplied in three formats:

- ◆ Portable document format (*.PDF* file), which you can read on-line using a suitable PDF reader such as Adobe Reader®. This version of the document is formatted as a printable manual; you can print it from the PDF reader.
- ◆ Web pages (*.HTML* files), which you can read on-line using a web browser. To read the web version of the document navigate to the *HTMLDoc_m_n* folder and open the file *index.html*.
- ◆ Microsoft HTML Help (*.CHM* file), which is an HTML format contained in a single file. To read a *.CHM* file just open it – no web browser is needed.

For the best reading experience

On the machine where your browser or PDF reader runs, install the following Microsoft Windows® fonts: Arial, Courier New, Times New Roman, Tahoma. You must have a suitable Microsoft license to use these fonts.

Restrictions on viewing .CHM files

You can only read *.CHM* files from Microsoft Windows.

Microsoft Windows security restrictions may prevent you from viewing the content of *.CHM* files that are located on network drives. To fix this either copy the file to a local hard drive on your PC (for example the Desktop), or ask your System Administrator to grant access to the file across the network. For more information see the Microsoft knowledge base article at <http://support.microsoft.com/kb/896054/>.

1.2 Who should read this document

This document is intended for anyone who requires an introduction to DataSource for Databases. Typical readers include:

- ◆ Technical Managers
- ◆ System Architects
- ◆ System Administrators
- ◆ Software Developers

1.3 Related documents

- ◆ **DataSource Overview**
Describes the DataSource concept and how it fits into the Caplin platform.
- ◆ **DataSource for Databases Administration Guide 4.4**
Describes how to install, configure and run a DataSource for Databases enabled application.
- ◆ **DataSource for Databases API Reference**
Defines the Java™ APIs relating to DataSource for Databases.
Explains how to implement custom result set processing.

1.4 Feedback

Customer feedback can only improve the quality of our product documentation, and we would welcome any comments, criticisms or suggestions you may have regarding this document.

Please email your feedback to documentation@caplin.com.

1.5 Acknowledgments

Adobe® Reader is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Sun, Solaris, Java and *JDBC* are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

2 What is DataSource for Databases?

DataSource for Databases is a Caplin DataSource that provides a mechanism for moving data between databases and the Caplin Platform. It can read data from databases and forward it to other DataSource-enabled applications as structured records with fields; conversely it can accept data from other DataSource-enabled applications and persist it on a database. A typical database could be an existing repository of data, for example a database of financial market data or reference information.

DS4DB can interact with databases on a timed basis, or when connections are made with other DataSources, or on receipt of DataSource messages.

As with other types of DataSource component, DS4DB can communicate with any other suitably configured DataSources peers including Caplin's Liberator, Transformer, and PriceMaster products, and any user defined DataSources that have been implemented using the DataSource API.

For an explanation of the DataSource concept refer to the document **DataSource Overview**.

2.1 Features of DataSource for Databases

- ◆ DataSource for Databases is built on the standard DataSource SDK, so it can communicate with a network of other DataSource peers.
- ◆ Written in Java™ and will run on any Java 1.5 enabled operating system platform.
- ◆ Will work with any database management systems that support the [JDBC](#) 1.4™ API.
- ◆ Can access data in multiple databases, with queries spread across them.
- ◆ Works with existing databases.
- ◆ Implements parallel processing for optimal throughput and responsiveness:
 - Simultaneous handling of multiple events.
 - Events can issue and respond to multiple database reads across multiple databases simultaneously.
 - Configuration settings control the degree of parallel processing.
- ◆ Includes a comprehensive XML-based configuration facility.
 - Allows you to implement fully functioned applications with no need to write code.
 - Configuration changes can be dynamically applied to a running system.
- ◆ Includes a Java API that allows you to implement customized result set processing (requires Java programming expertise).

2.2 What can you use DataSource for Databases for?

You can use DataSource for Databases to implement various business requirements for managing and distributing database data.

For example:

- ◆ **Distribute price information**

DS4DB is connected to a database containing equity price information. An external system updates the prices. DS4DB retrieves price data from the database once a second and passes it to a Caplin Liberator server. The Liberator pushes the price updates to end users, via the RTTP protocol across the internet.

- ◆ **Distribute Index information**

DS4DB is connected to a database holding information about the set of financial Instruments comprising a market index. At the start of the day an external administration system sends the DS4DB application a start up message. This message triggers DS4DB to read the latest index data from the database and distribute it to other systems, such as a Liberator server (so it can be viewed by end users), and Caplin's Market Data Platform.

- ◆ **Provide historical data on-demand**

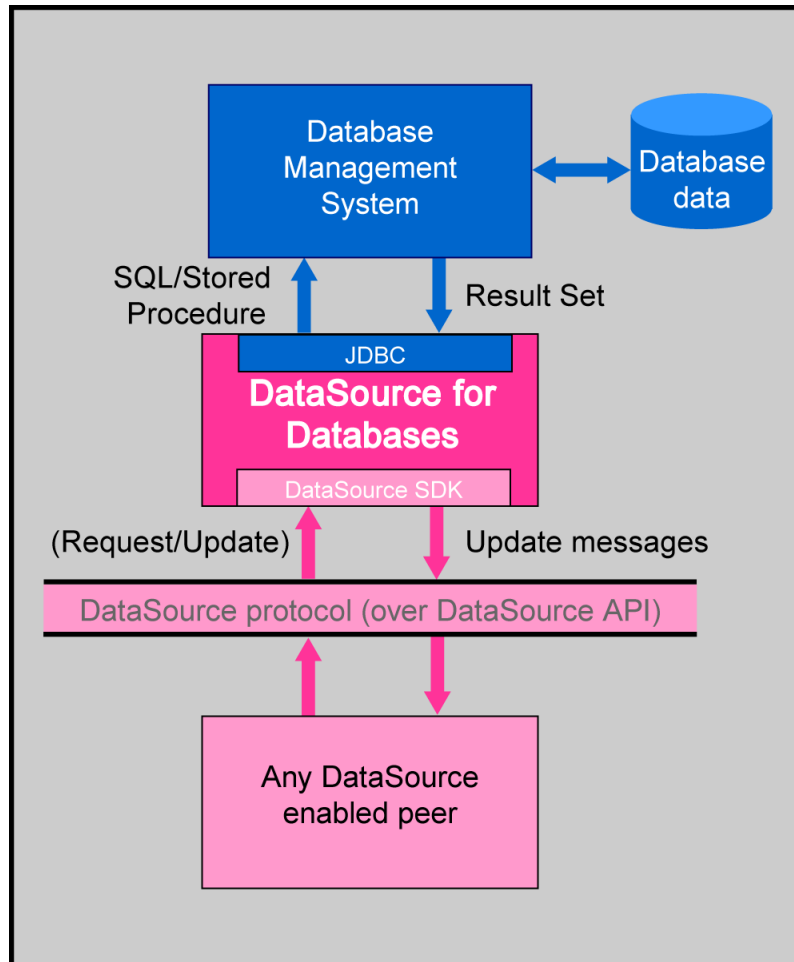
DS4DB is connected to a database of historical financial data. Users can make ad-hoc requests for this data over the internet via an RTTP connection to a Liberator server. The server passes on the requests to DataSource for Databases, which obtains the relevant records from the database and forwards them to the users via the Liberator.

- ◆ **Enrich real-time data with static reference data**

DS4DB is connected to an instrument repository database. When a client requires instrument data, a request is sent both to a live pricing DataSource *and* to DS4DB. The live pricing DataSource provides the changing data and DS4DB provides the static data by querying the database.

3 Architecture

DataSource for Databases is a component of the Caplin platform. The following diagram shows how it fits into the platform architecture.



DataSource for Databases in Caplin's platform architecture

The diagram shows that DS4DB communicates with other DataSource peers. Because DS4DB is itself a true DataSource, it can be configured to interact with the other DataSource peers as a [broadcast peer](#)^[14] or as an [active peer](#)^[14]. When configured as a broadcast peer it will respond to Update messages, but not Request messages. In contrast, when it is an [active peer](#)^[14] it will respond to a Request message by sending an acknowledgement back to the requesting peer.

DS4DB also communicates with one or more external database management systems (DBMS), using the standard [JDBC](#)^[14] API. The DBMS are provided by the user and can be any database products that have a JDBC technology-based driver. The driver mediates between the JDBC API and the database management system.

DS4DB sends database queries to the DBMS; these queries can be [SQL](#)¹⁴, or if the DBMS supports them, [stored procedure](#)¹⁴ calls. A query can read data from the database or write data to the database, including updating the values of existing data items.

Note: In the rest of this document "query" means "SQL query" or "stored procedure call".

DS4DB is highly configurable. You use XML-based configuration data to define:

- ◆ The actions DS4DB takes when it receives DataSource messages.
- ◆ The databases it can access.
- ◆ The queries it executes.
- ◆ How it reformats database data into messages for transmission to its DataSource peers.
- ◆ The amount of resource allocated to parallel processing.

4 Concepts

The following concepts are key to understanding what DataSource for Databases does and how it can be used:

- ◆ [Events](#) ⁷
- ◆ [Database accessing](#) ⁷
- ◆ [Results set processing](#) ⁸
- ◆ [Parallel processing](#) ¹⁰

These concepts are explained in the following sections.

4.1 Events

DataSource for Databases accesses databases in response to events, which can be incoming DataSource events or internal timed events.

DS4DB can support multiple events. There is no limit on the number of events (of any type) that can be configured, other than system resources such as available memory. When several events occur at the same time, DS4DB will process them in parallel, up to a configurable maximum number of simultaneous events.

DataSource Event

A DataSource Event fires when DS4DB receives a DataSource message that matches a (configured) set of criteria; typically a match against the object (subject) name and/or the values of one or more fields of the message. When the event fires, DS4DB performs a configured set of actions on the database(s).

There are three types of DataSource event:

- ◆ Peer Connection Event
- ◆ Request Event
- ◆ Update Event

For more information about these event types see the **DataSource for Databases Administration Guide**.

Timed Event

A Timed Event is an event that fires at configured intervals. When the event fires, DS4DB performs a configured set of actions on the database(s).

4.2 Database accessing

DataSource for Databases accesses one or more databases in response to an event firing; to do this it executes one or more [database queries](#) ¹⁴. A query can read data from a database or it can write data to a database.

Queries can take parameters whose values are supplied by the event that fired. For example, if an event fires in response to a DataSource Update message arriving, the subject name of the DataSource message, and/or field values from the message, might be passed into the database query as a parameter.

If an event needs to issue several database read queries, it can make a separate database connection for each query and then execute the queries in parallel in order to reduce processing time. The maximum number of simultaneous connections to each database is defined through a configuration parameter.

4.3 Result set processing

When a database read query is executed, the DBMS returns a [result set](#)¹⁴. This is a set of rows and columns containing the data selected by the query. DataSource for Databases processes the result set, converts it into one or more DataSource Update messages, and broadcasts the messages to all the connected peers.

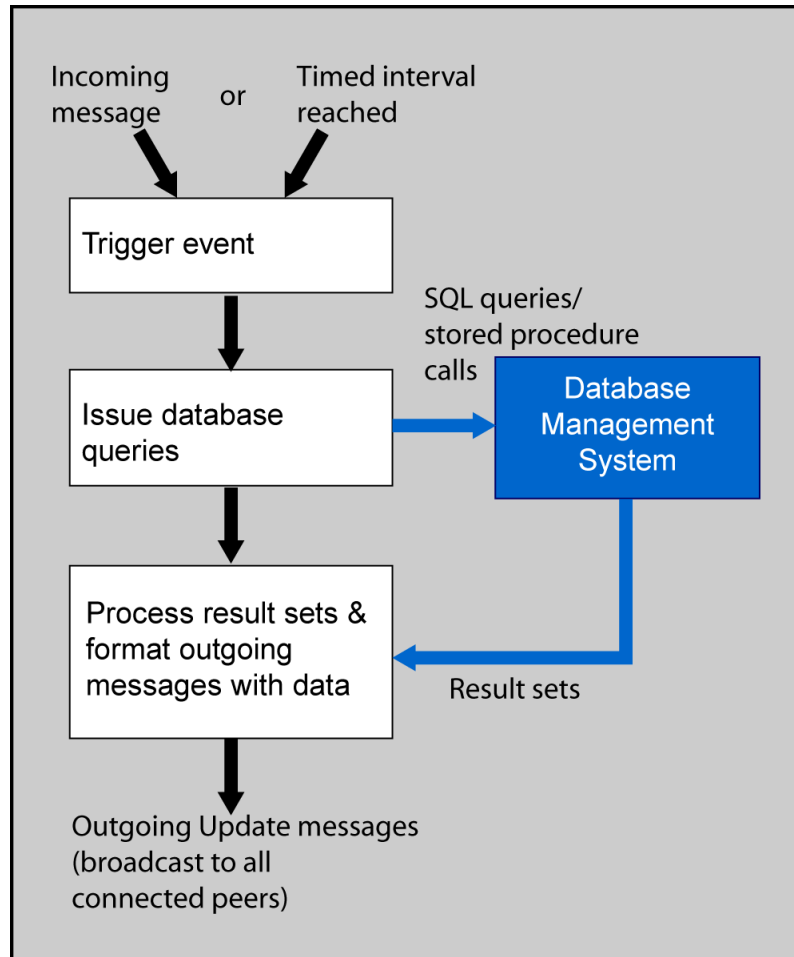
DS4DB provides several configurable ways to process a result set:

- ◆ Standard result set processing
- ◆ Compound result set processing
- ◆ Collapsing result set processing
- ◆ Custom result set processing

For more information on these types of result set processing see the **DataSource for Databases Administration Guide**.

4.4 Processing flow

The following diagram shows the basic flow of processing when an event occurs.



DataSource for Databases - processing flow

How DataSource for Databases responds to a particular incoming DataSource message, or even whether it responds at all, depends on how the configuration is defined. If an incoming message does not match any of the configured events then DS4DB ignores it.

An event causes DataSource for Databases to execute one or more database queries or stored procedure calls, which can read data from one or more databases and also write data to a database. When a database read query is executed, the DBMS returns the data as a result set. DS4DB processes this result set, converts it into one or more DataSource Update messages and broadcasts the messages to all the connected peers.

Events can be configured to just write data to a database, with no result set being produced; in this situation no messages are broadcast.

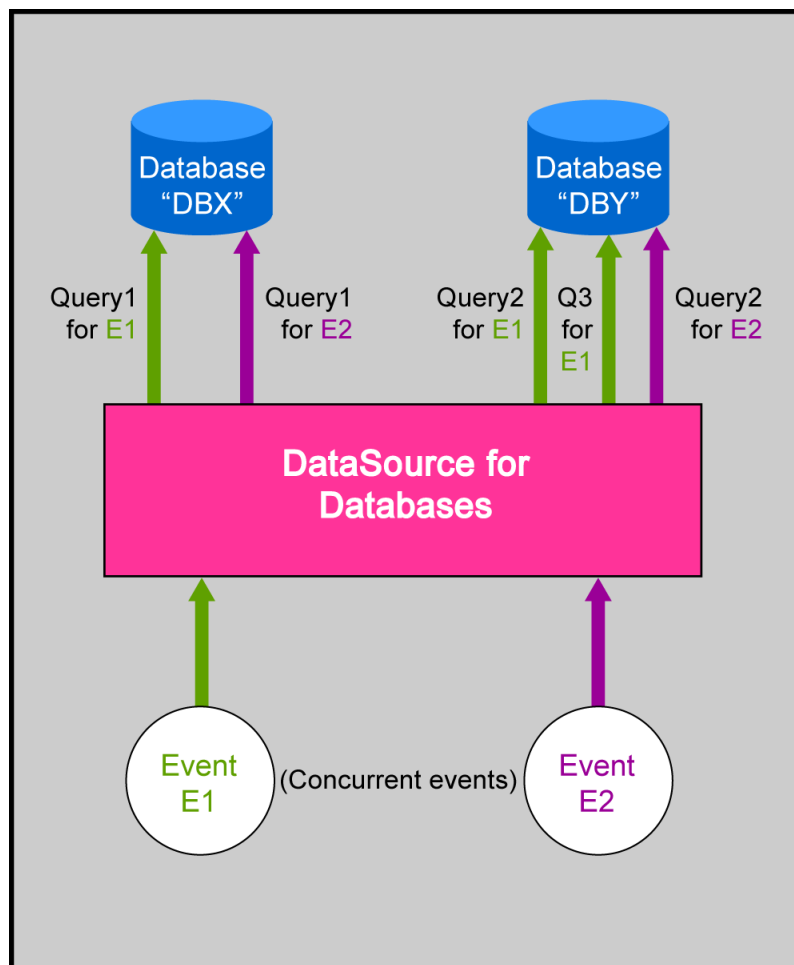
4.5 Parallel processing

DataSource for Databases has three features that enable you to improve performance through parallel processing.

- ◆ When several events occur at the same time, DS4DB can process them in parallel.
- ◆ Within an event the database read queries can be executed in parallel.
- ◆ DS4DB can access several databases simultaneously.

These features are controlled through configuration.

The following diagram shows how DS4DB's parallel processing features work.



Parallel event and query processing

This diagram shows two events E1 and E2 arriving at DS4DB at the same time. DS4DB is configured for parallel processing and so processes the two events concurrently.

The processing for Event E1 specifies that it runs three database queries; "Query1" is to be submitted to the database called "DBX", "Query2" and "Query3" are to be submitted to a different database, "DBY". The processing for Event E2 runs two queries, "Query1" on "DBX" and "Query2" on "DBY".

DS4DB and the DBMS are configured to run multiple queries in parallel, so DS4DB submits all the required queries to the databases at the same time.

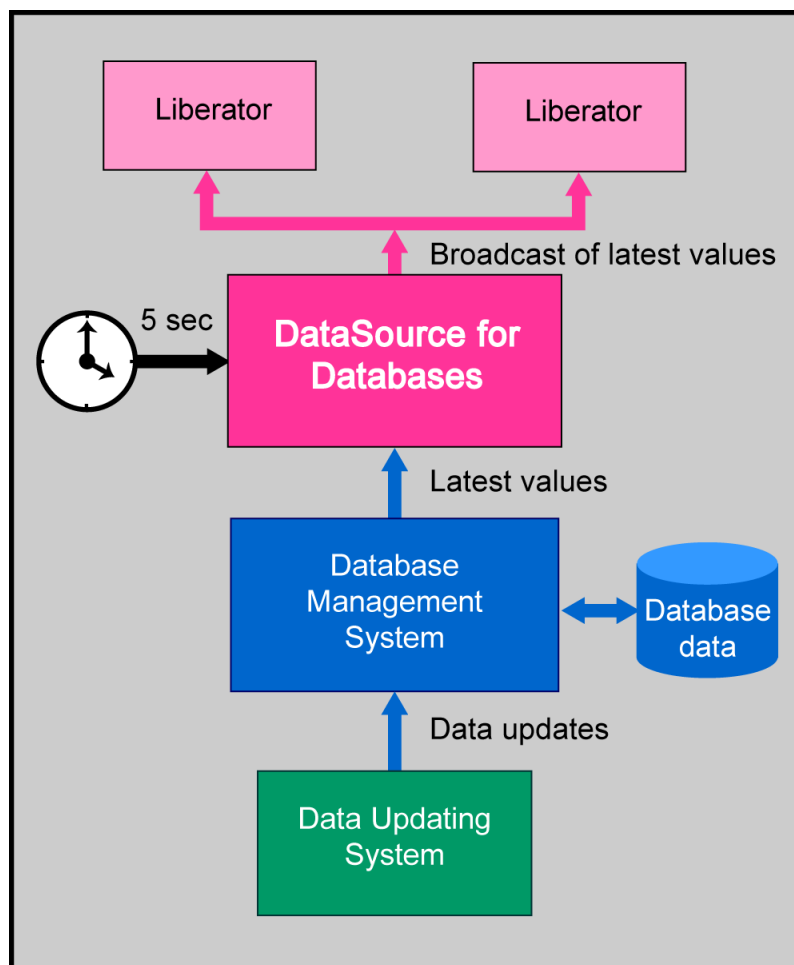
5 Examples of use

Here are two simple examples of how DataSource for Databases could be deployed.

- ◆ [Timed Updates](#) ¹²
- ◆ [Index Distribution](#) ¹³

5.1 Timed Update example

In this example DataSource for Databases is used to broadcast Updates at regular intervals.

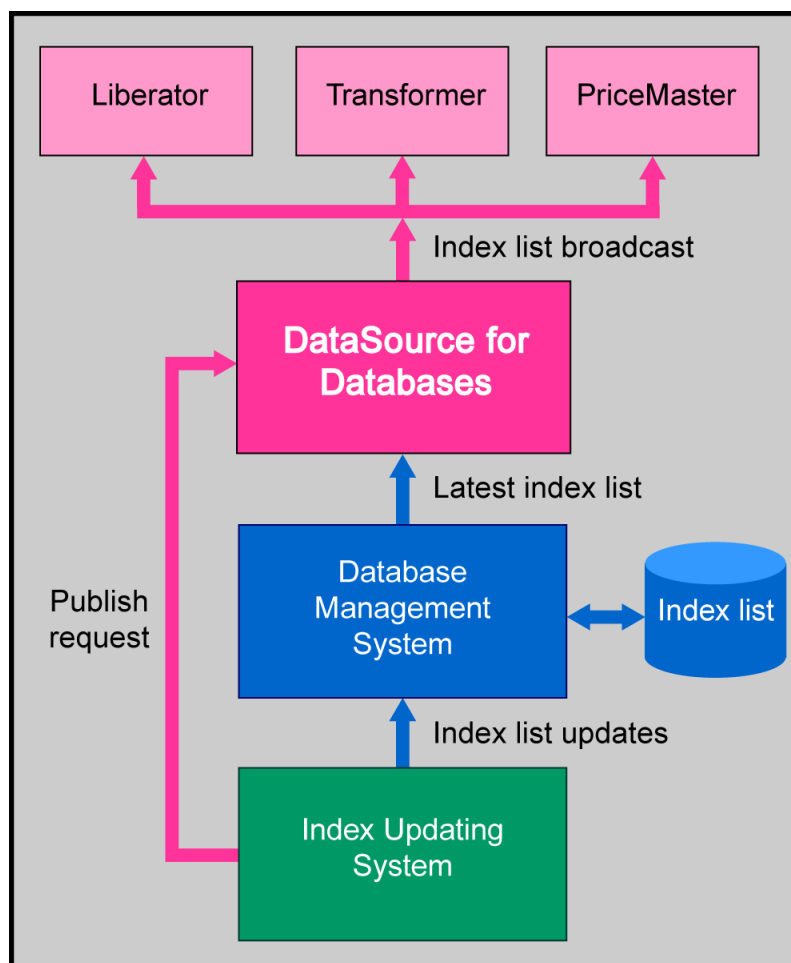


Timed updates using DataSource for Databases

Data on the database is continuously updated by an external Data Updating System. DS4DB is configured with a timed event, so that it reads the latest values from the database every 5 seconds and broadcasts them to a set of Liberator servers.

5.2 Index Distribution example

In this example DataSource for Databases is used to distribute an index list.



Distribution of financial index list using DataSource for Databases

The database contains a list of the financial instruments comprising an index, such as the shares in a share index. Instruments join and leave the index over time. The index list is held on a database because it needs to be persistent and secured against inadvertent loss or corruption. DataSource enabled components such as Liberator, Transformer and PriceMaster make use of the index list, so they need to be informed of changes to it.

Items are added to the list and removed from it by an Index Updating System. The Index Updating System contains a management application, which is a DataSource peer communicating with an instance of DataSource for Databases. After a change is made to the database the management application sends a DataSource message to DS4DB, requesting that the index list be re-published, so that the latest changes are visible. This request triggers a DS4DB event that reads the index list from the database and broadcasts it to the Liberator, Transformer and PriceMaster.

6 Glossary of terms and acronyms

This section contains a glossary of terms and acronyms relating to the DataSource for Databases product.

Term	Definition
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
Active peer	<p>A DataSource application that is configured to respond to DataSource Request messages received from one or more of its DataSource peers, by keeping track of which objects have been requested, and sending to the requesting peers updates just for those objects. This improves performance by reducing network bandwidth.</p> <p>See also broadcast peer.</p>
Broadcast peer	<p>A DataSource application that is configured to broadcast (push) DataSource messages to all the peers connected to it.</p> <p>See also active peer.</p>
DataSource peer	An application that can communicate with another application using the DataSource protocol. DataSource for Databases is a DataSource peer.
Database query	<p>A programmatic request made to a DBMS in order to retrieve data from the database, or insert or modify values in the database. In many DBMS implementations database queries can be packaged as stored procedures.</p>
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DS4DB	<u>D</u> ata <u>S</u> ource for (4) <u>D</u> ata <u>b</u> ases.
JDBC	<u>J</u> ava <u>D</u> ata <u>b</u> ase <u>C</u> onnectivity
	A Sun® Microsystems standard defining how Java applications access data in a database.
Peer	See DataSource Peer .
Result set	The logical database table (set of rows and columns) returned by a query on a database.
SDK	<u>S</u> oftware <u>D</u> evelopment <u>K</u> it
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
Stored procedure	An encapsulation of a database query or queries that runs inside the database server. Because it runs on the server a stored procedure usually executes faster than when the client application submits the equivalent query to the server.

Contact Us

Caplin Systems Ltd
Triton Court
14 Finsbury Square
London EC2A 1BR
Telephone: +44 20 7826 9600
Fax: +44 20 7826 9610
www.caplin.com

The information contained in this publication is subject to UK, US and international copyright laws and treaties and all rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the written authorization of an Officer of Caplin Systems Limited.

Various Caplin technologies described in this document are the subject of patent applications. All trademarks, company names, logos and service marks/names ("Marks") displayed in this publication are the property of Caplin or other third parties and may be registered trademarks. You are not permitted to use any Mark without the prior written consent of Caplin or the owner of that Mark.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors and is subject to change without notice. Changes are periodically added to the information herein; these changes will be incorporated in new editions of this publication.

Caplin Systems Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

This publication may contain links to third-party web sites; Caplin Systems Limited is not responsible for the content of such sites.