
DataExchange overview



Table of contents

- DataExchange overview 1
 - Table of contents 2
- Introduction 3
 - Platform 3
- DataExchange Operation 4
 - Hiding the details 5
 - Connecting to data 6
- DataExchange Architecture 7
 - Application Components 8
 - Multi-protocol proxy 9
 - Protocols 10
 - Application interfaces 11
 - Event Management 12
 - Security 13
- Appendix A – Protocol Descriptions 14
 - File Protocol 15
 - FTP Protocol 15
 - JDBC Protocol 16
 - POP3/SMTP Protocol 17
 - Program Service 18
 - Java Message Service 19
- Appendix B – Product futures 20
 - Data Validation 20
 - Transformation 20
 - Self Describing Data Transfer 20

Introduction

Allowing users to access information over the Internet is an endeavour that must be undertaken in a measured manner. Considerations in doing so include :

- Ensuring users only access their data and no unauthorized disclosure of data takes place
- Access is monitored
- Access is easy and intuitive
- Data can be easily managed

Platform

DataExchange is built to execute in a J2EE Servlet (2.3) container under J2SE 5. This is the only pre-requisite for operation. DataExchange can be deployed into an existing container with no other requirements.

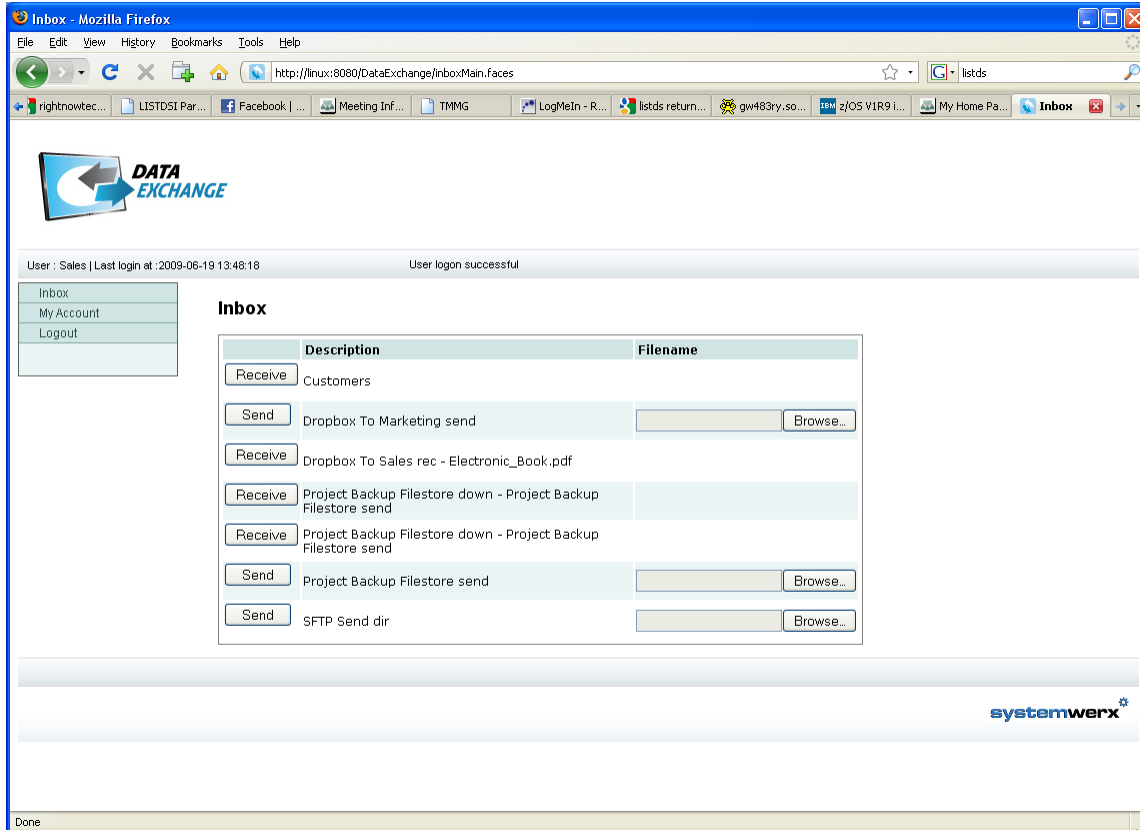
DataExchange also employs the following technology :

- Java Server Faces – Apache Myfaces and Tomahawk
- Apache Derby (Embedded SQL database for control info) – an external DB can be used also
- Richfaces components

DataExchange Operation

DataExchange provides a portal for accessing a variety of data in a uniform manner a HTTP session in this first version.

Users are presented with an Inbox :

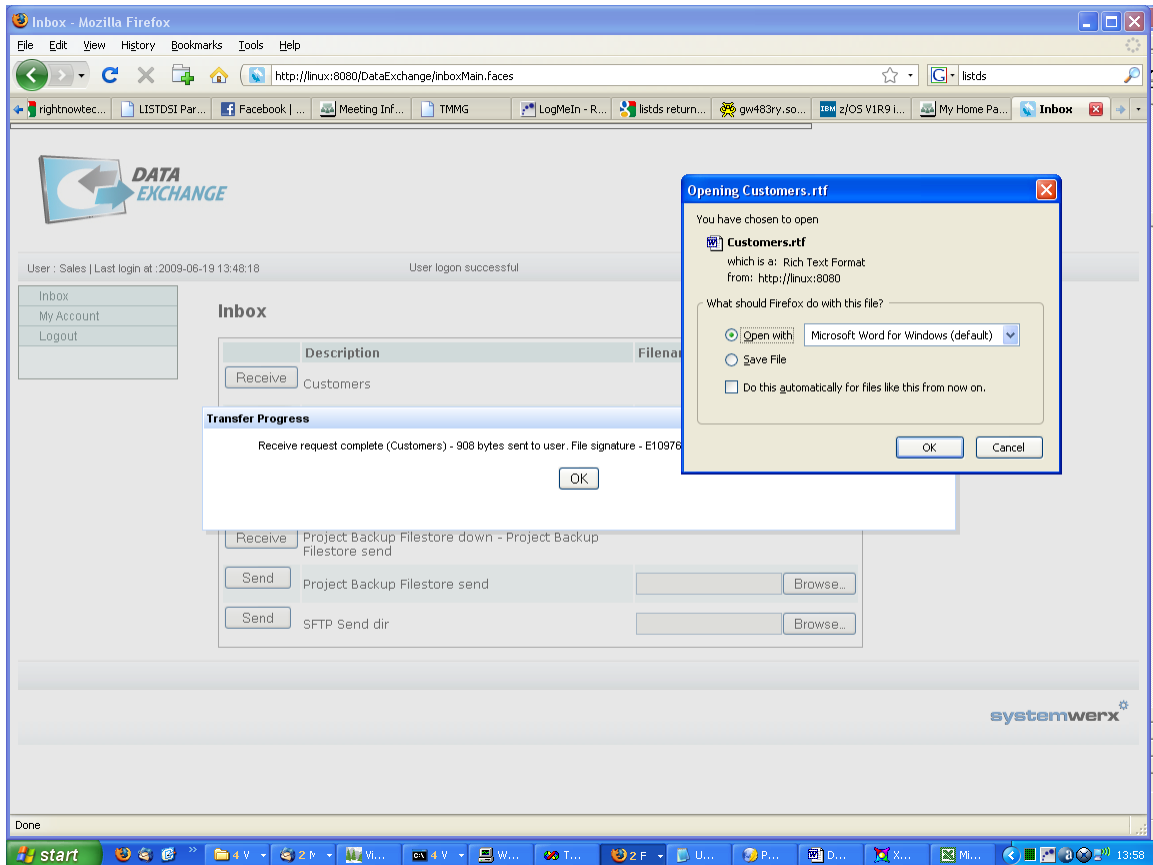


This limits exactly what the user can access. Users can only access data as described by the DataExchange manager. The user can only request an operation and select file used for upload or download.

Unlike conventional Peer-to-Peer Manager File Transfer programs users can only access prescribed data in a prescribed manner giving greater control (and administration) than MFT programs.

Hiding the details

DataExchange hides the details of the background file access that takes place, DataExchange delivers all data types in a uniform manner.

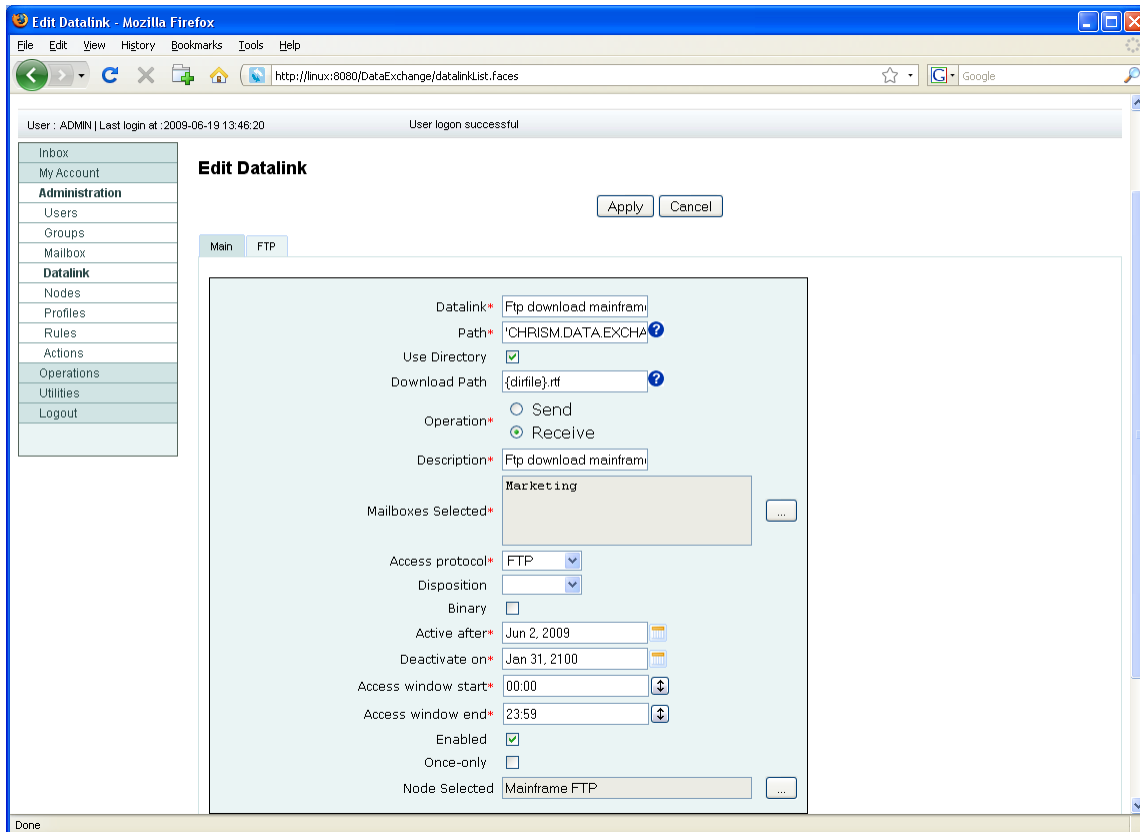


This allows users to gather a number of datasources behind a single presentation without regard to protocol. This also hides the physical details of the network from the outside user.

Connecting to data

Users are connected to data using Datalinks. These are used to build the users Inbox of data selections.

A Datalink is assigned to a user or group and can be defined with access windows to control the users view of available data.

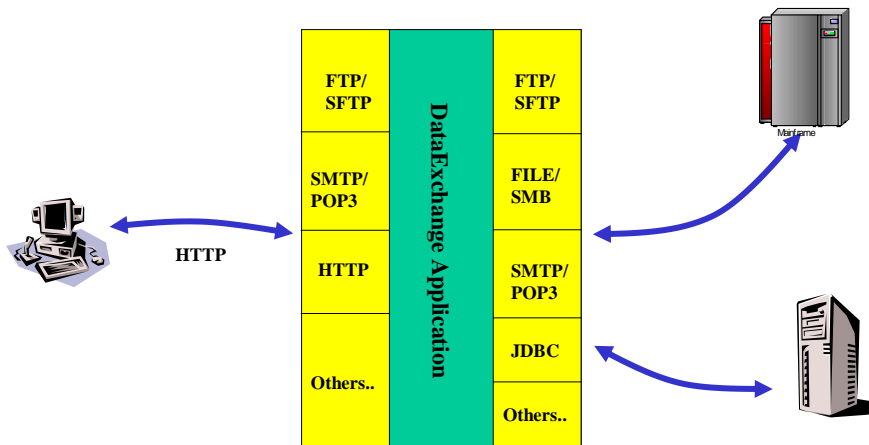


DataExchange Architecture

Data Exchange is an application to manage the Exchange of Data. DataExchange acts as a broker between Sources of data and a user to deliver data in a controlled uniform manner.

DataExchange is implemented as a J2EE Application executing in a Servlet container.

Architecture



DataExchange in Proxy mode

Users are seamlessly proxied to another data source

As DataExchange is Internet facing access control is tightly controlled, DataExchange uses the Concept of an inbox that contains DataLinks – A link to data. This link to data can be accessed in a variety of ways:

Application Components

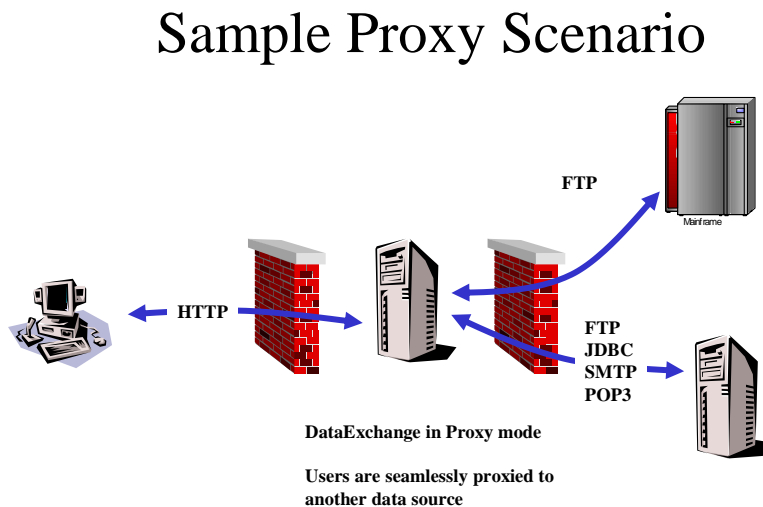
Component	Purpose
Multi-protocol Proxy	Servlets performing data access using a proxy operation – connects user to their data across multiple protocols
Protocol Management	Provides a plugable protocol interface for the proxy – allows abstraction of protocols from the proxy
Security management	Provides a security interface to manage logon and access rights
Audit management	Provides an local and/or remote audit of Dataexchange operations
Inbox management	The manager of links in a users inbox – also validates all requests to ensure they are valid.
FileStore facility	SQL based storage facility that stores files in a BLOB. This is analogous to the Connect:Enterprise Mailbox
API	Application interface to provide access to : <ul style="list-style-type: none">• Administration• File transfer• Filestore Management
Rules Facility	Event driven facility to drive application activity. This allows post-processing actions to be initiated. This is analogous the the Connect:Enterprise Rules feature
Batch Utilities	Utilities to allow Batch management and initiate transactions from a scheduler.

Multi-protocol proxy

Access to data is provided using a proxy technology. Users request for data is relayed to a remote system using the protocol chosen in the Datalink definition. This results in :

- The DataExchange system not storing any application data – this reduces risk as the DataExchange System does not have to store data (using the file protocol a local file can be used if desired).
- A single system image can be presented as all systems and protocols are hidden behind the proxy.

For example a user may request a file, this is passed to the user as a HTTP file request, DataExchange could then access the data from a remote system such as z/OS :



The exact nature of the system containing the data is hidden from the user, the data type is also hidden. Users can only access the Data that is provided via a Datalink – this controls access more than conventional P2P MFT products.

Protocols

Protocols supported by the proxy include :

Protocol	Purpose
FTP/FTPS	File transfer protocols – these allow the user to access files stored on a remote host. For example files on a z/OS system could be accessed directly.
SFTP	File transfer protocols – these allow the user to access files stored on a remote host.
POP3/SMTP	Email protocols – users can be supplied information using email – data suppliers can post data using a standard email client. DataExchange allows access to attachments in emails using a filtered criteria
JDBC	Database access – users can receive data from databases directly
JMS	Messages can be read and written to a message queue
Program	Programs can be called to provide data presented by application logic. In many cases data needs to be processed before it is available in a suitable format. In this case we retrieve data from or pass data to a program which performs the function of a presentation layer for an application system.
Filestore facility	SQL based mailbox facility that stores files in a BLOB. This is analogous to the Connect:Enterprise Mailbox

Protocols are added as plugins – this allows other protocols to be easily added .
Additional protocols to be added in the bear include :

- AS/2
- WebDAV

Application interfaces

Access to DataExchange is available for applications to :

- Perform administration functions
- Access file transfer facilities

Access is available via :

- DataExchange exposing it's functionality using Web Services – accessible via SOAP and WSDL. Applications platforms such as Java and .NET can access the data using exposed service wrappers generated by IDEs such as Visual Studio, Rational Application Developer and Eclipse
- Command line interface to provide access to scripting languages.

Event Management

A facility will be available to filter significant application events such as file send/receive and errors – these will be passed to a number of possible destinations :

- Application program
- JMS Queue

The expression of these filters will be using Rules :

```
SELECT GROUP=HEADOFFICE2,EVENT=DOWNLOADSUCCESS,ACTION=(SENDJOB1,SENCMD2)
SELECT GROUP=HEADOFFICE*,EVENT=UPLOADSUCCESS,ACTION=(SENDSTATUSJOB1)
SELECT USER=FRED,EVENT=LOGON,ACTION=(SENDLOGONNOTIFY)
SELECT DATALINK=COMISSION,EVENT=DOWNLOADSUCCESS,ACTION=SENDEMAIL2
```

Security

Security is a primary concern for this type of application. To facilitate this a number of features exist :

Feature	Purpose
Digital Signature of data	To verify Data content files can be optionally signed. Algorithms include : <ul style="list-style-type: none">• XML Signature• DSS Signature• RSA Signature Client software (Java Applet) will be available to allow generation and verification of signatures.
Simple hashing	Data fingerprints will be written to audit logs to allow confirmation of data integrity Examples include MD5, SHA, SHA/256
Public Key Encryption	Allows file to be only read by designated recipients
Remote logging	Logging of data to remote systems to protect against compromise
Support for Hardware Security Modules	Support for hardened Key storage and Crypto processors

Appendix A – Protocol Descriptions

File Protocol

The file protocol provides access directly to the local file system. Users read and write directly to O/S files. Under a Windows environment UNC could be used to access file across a network.

FTP Protocol

Users access files on a remote host using FTP. File names specified are accessed from the remote system directly using the FTP protocol.

JDBC Protocol

Users access data residing in an SQL Database. The datalink contains an SQL statement which described the data to be accessed.

For a GET operation an SQL SELECT command is used to generate data. The columns are concatenated to form a continuous data stream.

For a PUT operation processing an SQL INSERT command is used to add data. The SQL syntax is extended to allow selection of the data to be inserted into a column.

An example of the modified SQL syntax :

```
INSERT INTO Sample_table(Samplecol1, Samplcol2) VALUES ($Field1,  
$Field2)
```

POP3/SMTP Protocol

This protocol provides an implementation for GET and PUT that operates across email protocols.

This provides a number of useful functions :

- Data can be relayed inside a organization without opening ports on a firewall using SMTP.
- Users in an organization that wish to send information to DataExchange users require nothing more than an email client to do so.

For PUT operations the data is packaged as an email and the file attached to it. The data is sent to the user specified in the Datalink specification.

For GET operations a regular expression is used to match emails containing potential attachments. These attachments can also be filtered using Regular Expressions. All the attachments are presented as a directory to be downloaded.

Program Service

This protocol calls an application program either directly to access data.

In many environments the data required may only be available through an application presentation and not directly available by accessing underlying data.

When calling a program an external process will be called and data sent via stout or stdin.

Java Message Service

This protocol reads or writes data to a JMS queue.

Appendix B – Product futures

Detail of these will be expanded in future.

Data Validation

A feature to allow input data format to be validated for format and against data sources such as Database tables for acceptable value ranges

Transformation

A feature to allow data format to be translated on input or output, for example converting Packed or binary fields to and from Text for end users.

Self Describing Data Transfer

This feature will translate back-end data to/from an XML format to allow data to be transferred to a user in a manner that can easily be processed by database systems and applications.
