

InfoLease<sup>®</sup>

---

**Streamline** data sharing

with **Integration Manager<sup>®</sup>**



**White Paper**



INTERNATIONAL  
DECISION SYSTEMS<sup>®</sup>



## Streamline Data Sharing with *Integration Manager*®

### Executive Summary

Enterprise-wide data sharing is critical to staying competitive in today's markets. This type of integration can simplify your systems and support; with the right tools, your systems can be integrated both easily and effectively. IDS has developed *Integration Manager* to seamlessly integrate *InfoLease*® with other systems enterprise-wide. *Integration Manager* streamlines data sharing using state-of-the-art technology that conforms to industry standards. The following describes in detail the concepts behind *Integration Manager*, its strategic implementation, and real-world examples of its use. By simplifying integration of *InfoLease* in an enterprise-wide system and management of the resulting environment, *Integration Manager* reduces overall costs, improves effectiveness of systems and the organization, and improves service to customers.

Data and information isn't like other organizational resources. Unlike goods, money or labor, you don't have to allocate it. You can use it again and again and still have it to use elsewhere. Properly managed, it can be put to work in many places at once. In aggregate, it is worth more than the sum of its parts. And, made available for use in decision-making, it becomes something more than mere information. It becomes knowledge.

With the right infrastructure, information can be conveyed at light speed and at a low cost; hence, the growing passion for integrated enterprise-wide IT solutions. In manufacturing, enterprise-wide architecture may mean that order entry automatically generates an invoice, triggers an order for raw materials, and becomes part of next year's planning. In leasing, it might allow an address change in the customer relationship management (CRM) application to trigger an update in the leasing database. Infrastructure is the key to all this efficiency, but there is no single "right" infrastructure. Every organization's needs are unique, and effective systems are the ones that fit the need.

### The Right Tools for the Job

When your needs are casual, a Swiss Army knife may be a perfectly adequate toolkit. But when life or livelihood is at stake, you want the best and most specialized equipment you can find. In critical applications, the right tools can make the difference between success and failure. This is certainly the case in today's competitive markets, where second place rarely wins bids, and "adequate" service won't keep customers coming back.

Business systems are a company's tools, and when you're serious about the job, you don't cut corners in selecting tools. That's why IT managers buy from specialists – computers from computing pros, storage systems from storage specialists, and communications systems from telecom experts. The same is true of software, especially the systems at the heart of your business. Most companies choose database software from one of a handful of leading companies. Similarly, there are specialized packages for CRM and other functional areas. In lease management, the leading choice is *InfoLease* from IDS. These are the "best of breed" packages organizations use to achieve top performance in key areas. The challenge is melding these specialized functions into a true enterprise resource.

## Sharing Information

The simple way to join operational areas is with the “Swiss Army” approach, in which all functions exist in a single package. Unfortunately, as with any pocket tool kit, the price for simplicity is compromised functionality. This is particularly risky when the provider leaves its area of expertise to add new functions. This might be acceptable if the added-on operations were of secondary importance to your business. But when they are critical, the cost of turning them over to “dabblers” is too high.

## Bringing the Best Together

When functionality counts, the solution is true integration of best-of-breed systems. This lets each package do the job for which it is designed, with full and open sharing of information between systems. For *InfoLease* users, the key to this level of integration is *Integration Manager*, *InfoLease* is already the top choice among lease management system (used by half of all high-end leasing companies according to The Monitor). *Integration Manager* turns *InfoLease* into an enterprise-friendly package, able to freely share data with database, finance, ERP and CRM packages, auto-dialers, interactive voice response systems, and a variety of web applications. It communicates easily with modern XML-compliant systems and, with “data transformation,” with legacy non-XML-compliant systems as well.

## Doing it Your Way

*Integration Manager* supports two-way data sharing with a wide variety of applications in any combination of real-time, asynchronous, or batch communication. The system is highly scaleable to grow as you do. And, unlike some forms of multi-application communication, it reduces, rather than increases, the effort and cost of support. Information can be exported from, or imported into, *InfoLease*. Data can be shared either within or beyond your company’s firewall, allowing you to communicate freely, yet securely, with customers, partners and vendors. With the use of message-oriented middleware, applications need not be running simultaneously or even directly connected to participate. All this flexibility provides *InfoLease* customers with enormous capacity for growth, both in size and functionality.

## Customer-directed Development

The initial release of *Integration Manager* allowed access to some 14,000 fields within the *InfoLease* system. Since then, IDS has continued to work with customers to identify additional business rule related areas they want included in a read/write enterprise application framework. These business rules are then extracted, allowing interface with other enterprise applications. Ultimately, *Integration Manager* will become the proverbial “black box,” linking *InfoLease* with other systems according to customer-defined rules, with complete transparency to users.

## Benefits of Integration

Application integration can pay for itself both quickly and generously.

- ◆ Integration lets you leverage what you already have. With *Integration Manager* you can continue to reap the benefits of your *InfoLease* investment while expanding enterprise wide systems.
- ◆ You can avoid much of the cost of e-business development. As the “information superhighway” becomes the business superhighway, organizations have spent hundreds of billions of dollars integrating existing systems with e-business applications for use over the Internet. Much of this integration has been custom development, which is both expensive to develop and expensive to maintain. *Integration Manager* helps you get online quickly and affordably.
- ◆ Integration reduces data entry cost and the risk of error. Error is a particular problem, as conflicting information cannot be trusted. Integration eliminates the conflict, and better information means better decisions.
- ◆ Integration supports what the META Group calls a “zero-latency” enterprise, and helps make an organization “customer-oriented and relationship focused.”

## Integration at Work

Consider some of the many possible uses of *Integration Manager*. A customer support representative receives a call from a delinquent customer requesting an extension. The rep approves the request and enters a message into their support system confirming the two-day extension. An hour later, a collections agent working from data on *InfoLease* comes across this same customer scheduled for contact. Before calling to check on the past due payment, the agent sees the real-time message that was transferred from the support system and cancels the call, eliminating both wasted effort and customer frustration. Similarly, posting of a received payment could automatically generate notice from *InfoLease* to a collections system, deleting a scheduled collection call without manual intervention.

A credit analyst working on a front-end credit application system wants to view the applicant’s past credit history with the company for a more complete picture of the applicant’s creditworthiness. Without *Integration Manager*, the analyst would have to log onto *InfoLease* and locate the customer to view past leases. With *Integration Manager*, that information can be shown in real time within the front-end system, saving time and providing a more complete credit picture.

Voice response systems and a variety of web-enabled applications could be given real-time access to *InfoLease* using *Integration Manager*. This would allow customers, outside sales representatives, vendors, or other business partners to easily access the same kind of real-time information over the Internet or telephone. The use of *Integration Manager* would simplify configuration and eliminate the need for implementation and maintenance of multiple interfaces.

## Out of the Warehouse and Into the Bank

Organizations today have used the combined power of data processing and communications to build enormous “data warehouses.” But data in a warehouse isn’t necessarily data at work. Integration puts information where you need it to make better decisions and compete in today’s competitive global markets. With *Integration Manager*, critical information moves easily in and out of the *InfoLease* system – up to senior management, out to a wider range of employees, and beyond, to partners and customers who need it. In the process *Integration 4 Manager* turns information you have into knowledge you can use.

## IM Technical Architecture

*Integration Manager* simplifies and streamlines communication between *InfoLease* and other systems. It allows retrieval and update of information into *InfoLease* and propagation of data from *InfoLease* to other systems. It eliminates the complexities of previous access methods that used UniData technologies – UniObjects and ObjectCall – as well as offerings like ODBC.

*Integration Manager* is designed to be both flexible and data-neutral for easy integration with a wide range of other systems. As an XML-based solution, *Integration Manager* provides a data format that is easily manipulated by standard tools and transformed by the customer.

Middleware solutions (included with the system) allow easy access to *InfoLease* data via standard protocols such as HTTP and message-oriented middleware (MOM). Besides encapsulating the details of the database, the middleware provides a scalable interface to *InfoLease* via a connection pooling architecture.

An included administration utility allows the user to define and create APIs providing retrieval access to over 14,000 *InfoLease* data fields. Numerous areas of *InfoLease* can be updated, with new areas of the system continually being rolled out for update access. At the same time, *Integration Manager* also provides a powerful interface to push XML-based data from *InfoLease* out to other systems.

*Integration Manager* consists of two modules: the Access Framework and the Broadcast Framework. Access Framework allows data within *InfoLease* to be retrieved and updated; the Broadcast Framework facilitates outbound transactions from *InfoLease*.

## Access Framework

The Access Framework module gives third-party applications a flexible interface with which to integrate with *InfoLease*. Three architectural functions allow update and retrieval of *InfoLease* data in a manner not before possible. These are:

- ◆ XML-based transactions.
- ◆ Fine-grained user control to create and modify APIs.
- ◆ Standard middleware interfaces.

Combined, these three create a powerful tool for interaction with *InfoLease*.XML

The Extensible Markup Language (XML) provides a standard format that allows businesses to exchange data between systems. As a meta-language, it can be used to describe and structure data. The unique property of being application-, platform-, and presentation-independent makes it an effective tool for interoperability. Technologies such as the XML Stylesheet Transformation Language (XSLT) allow mapping and transformation of non-XML and XML files for application integration. The meta-language capabilities of XML suit it well for transport of *InfoLease* data. For consistency in invoking transactions, integration with intricate *InfoLease* data structures, and effective fault handling, *InfoLease* uses an XML standard for its interface. This standard defines the following:

- ◆ Formatting of data types.
- ◆ Invoking of APIs.
- ◆ Structure of request transactions and resulting reply structures.
- ◆ Manner of handling repeated data structures (multi-values).
- ◆ Fault structures.

The *Integration Manager* XML standards provide a guide for dealing with *InfoLease*. In some environments an “integration broker” may be required to handle transformations and data mapping between systems. If multiple systems are being integrated, it is very likely that each will want data represented in a different format. Some will not even deal with XML (fixed length, etc.). IDS has associated an XML “tag name” with each of over 14,000 fields, allowing *InfoLease* data to be clearly rendered in XML format. Illustrating the use of XML within *Integration Manager*, the following figure shows an XML document used to add a customer within *InfoLease*:

Figure 1 – Sample XML

```
<Envelope>
  <Header>
    <ApiName>IICreateCustomer</ApiName>
    <ChLogUserDefinedText>webID</ChLogUserDefinedText>
  </Header>
  <Body>
    <Request>
      <Customer process="add">
        <CsDefaultName>Joe Smith</CsDefaultName>
        <CsDefaultShortName>Joe</CsDefaultShortName>
        <CsDefaultAddr1>123 West Pike Lane</CsDefaultAddr1>
        <CsDefaultCity>Madison</CsDefaultCity>
        <CsDefaultState> WI</CsDefaultState>
        <CsDefaultZip>54703</CsDefaultZip>
        <CsDefaultContactPhone>555-555-9087</CsDefaultContactPhone>
        <CsDefaultFederalTaxId>4224</CsDefaultFederalTaxId>
        <CsDefaultArName>Joe Smith</CsDefaultArName>
        <CsDefaultArAddr1>123 West Pike Lane</CsDefaultArAddr1>
        <CsDefaultArCity>Madison</CsDefaultArCity>
        <CsDefaultArState>WI</CsDefaultArState>
        <CsDefaultArCountry>001</CsDefaultArCountry>
        <CsDefaultArZip>54703</CsDefaultArZip>
        <CsUdAlphaNumeric2>B</CsUdAlphaNumeric2>
        <CsVendor>0008480009</CsVendor>
      </Customer>
    </Request>
  </Body>
</Envelope>
```

## API Administration

As business needs change, so will the integration requirements. For increased responsiveness, *Integration Manager* provides the flexibility to create and modify APIs on a customer’s system without IDS intervention. An API administration tool allows any combination of retrieval APIs to be created and deployed by the customer. Complexities such as multi-values, bridging, field descriptions, etc. are also accounted for. The functionality of this tool includes:

- ◆ Customer creation and modification of retrieval APIs based on customer requirements
- ◆ Creation and modification of update APIs
- ◆ Generation of XML schemas
- ◆ Storage of information about created APIs and access to *InfoLease* field-level documentation
- ◆ Copying of APIs between test and production environments
- ◆ Creation of APIs that bridge between *InfoLease* files

In short, the administration tool provides all the functionality needed to maintain APIs in interacting with *InfoLease*. *Integration Manager* replaces the use of UniBasic programs and the need for compilation with an interpreter that provides fast, configurable access to data. For example, if an API needs to return six fields representing a customer address, the API is constructed as a logging definition within the database. When the API is invoked, the six fields are pulled and formatted into an XML document. Most APIs require no new UniBasic code.

For major integration projects, some areas of *InfoLease* that need to be updated may not be immediately available. In order to address this, new areas of the system are continually being opened in response to customer requests. Some more complex business needs may require custom development with IDS in order to make specific areas available through the Access Framework.

## Middleware

The last major component of the Access Framework is the middleware. Dealing directly with in-house developed, proprietary UniData interfaces can be costly, both in resources and time. Scaling these interfaces to handle high transaction volumes with acceptable response times is even more challenging. The Access Framework addresses this with two middleware solutions, allowing integrators to focus on the business requirements and not the technical details of interfacing with UniData and *InfoLease*. These two options, MessageConnect and WebConnect, interact with *InfoLease* through a pool of UniObjects for Java (UOJ) connections. UOJ is a UniData middleware tool that provides client access to the UniData database. Because they are Java-based, they provide portability across a range of operating system environments. Where they differ is in the interfaces they provide: WebConnect uses a HTTP based web services approach, while MessageConnect interfaces with a MOM environment.

## Connection Pooling

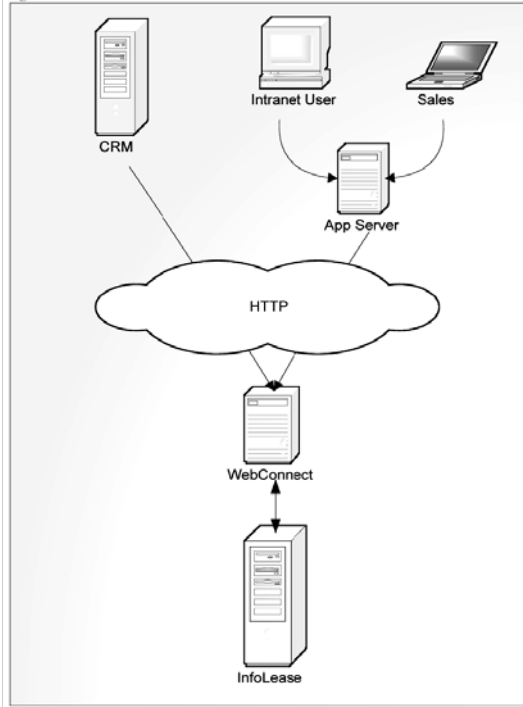
Using UOJ as a base, IDS has developed a thread-safe connection pool to *InfoLease*, providing the following functionality:

- ◆ Access to the pool requires knowing only the unique identifier of the pool. The connection pool hides the details of the database location, server name, etc. from the interfacing application. This enhances loose coupling between systems, as databases and servers can move almost anywhere in the network without other applications needing modification.
- ◆ Ability to define a minimum and maximum pool size. On startup, the pool is initialized to the minimum, growing during periods of high demand.
- ◆ Ability to throttle the connection pool down during periods of lower volume. For example, if the maximum pool size is reached, the pool will shrink based on a configurable 'timeout' to conserve resources on the *InfoLease* server.
- ◆ An interface to multiple *InfoLease* databases and servers. Distinct pools pointing to an unlimited combination of databases allow for flexible installation and administration.
- ◆ High level of transactional control. If UniData is unavailable, the pool will generate the appropriate error. The pool can be shut down for *InfoLease* maintenance activities like software updates or file resizing.
- ◆ Concurrent transactions handled via a multi-threaded interface to *InfoLease*. Both WebConnect and MessageConnect use this connection pool in interfacing with InfoLease.

## WebConnect

Using a simple XML-over-HTTP interface, WebConnect allows integration with the growing number of systems that are taking advantage of web technologies. Very simply, an XML document is posted to a Java servlet, routed through *InfoLease*, and a reply returned to the requestor. WebConnect handles the details of interacting with UniData, and in combination with the XML standards, returns appropriate transaction failures. WebConnect is a synchronous interface that, with a scalable connection to the *InfoLease* backend, provides high transaction volume with low latency. Another interface, SOAP-over-HTTP, is under development and will soon be available. With this interface, *InfoLease* functionality will be accessible as a true web service. The following is an example of a configuration using WebConnect. Note that client applications will need to use the HTTP protocol. WebConnect resides in a Java application server.

Figure 2 – WebConnect Overview



## MessageConnect

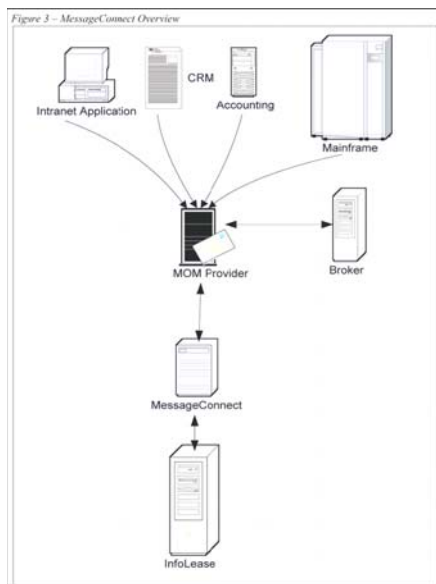
Messaging provides a key functionality: asynchronous communication. When systems are communicating, there is no guarantee that every piece of the environment will be functioning. Servers offline, databases down, or network issues can all contribute to synchronous “headaches,” impacting transactions that do not require a response. As asynchronous transactions function in a ‘send-and-forget’ mode, it is often desirable to initiate the

transaction and continue processing. MOM provides this ability, and MessageConnect supports both asynchronous and near real-time synchronous transactions. With initial support for WebSphere MQ, MessageConnect provides the middleware between the MOM provider and *InfoLease*. Instead of XML documents over HTTP, the same XML document placed in queue can be processed through *InfoLease*. MessageConnect is Java-based, and utilizes the Java Message Service (JMS) for interaction with the MOM provider. MessageConnect provides three basic means of defining a transaction:

- Synchronous - While MOM is typically asynchronous, it can easily be used for synchronous data transfers. WebSphere MQ has interfaces for dozens of languages and, as a language neutral platform, provides an excellent platform for synchronous communication. For this model, a reply is generated immediately for all successes and failures.
- Synchronous Batch - When a time-insensitive transaction is invoked, MessageConnect can be configured to treat a transaction as asynchronous, with the exception that a reply will be generated and returned to the user. Normal processing is to handle a transaction in a synchronous fashion.
- Asynchronous - For datagram processing, MessageConnect supports a transaction model in which messages are processed in a time-insensitive fashion and no reply is generated. If *InfoLease* is down for maintenance, a transaction that updates a customer address can be queued by the MOM environment and MessageConnect and processed when the system is again available. No reply to the user is needed; if the transaction cannot proceed it is desirable to attempt the transaction at a later time.

MessageConnect supports these transaction models through a highly configurable interface. A Java thread, termed a “listener,” is configured to register with the MOM provider for a specific queue. As messages are placed on the queue, MessageConnect receives the message and invokes the transaction with *InfoLease*. Listeners can monitor multiple queues across multiple queue managers. A listener is given a synchronous, synchronous batch, or asynchronous label by the administrator and handles transactions accordingly. Multiple listeners can be defined per queue, and multiple Java virtual machines (JVMs) can be configured to utilize resources of multi-processor servers.

Support is currently available only for WebSphere MQ. However, use of the JMS interface will allow interfaces to other MOM providers in the future. IDS is currently evaluating other MOM providers to determine where support should be offered. The following is a sample MessageConnect environment:



## Broadcast Framework

Very simply, the Broadcast Framework allows the propagation of data out of *InfoLease*. The Broadcast Framework is XML-based and utilizes the same XML standards as the Access Framework. Current support is for datagram transactions, in which data is pushed out of *InfoLease*. As data changes within *InfoLease*, broadcast triggers can be activated, initiating a broadcast event that sends data out of *InfoLease* to designated systems. Planned development will support a request-reply scenario, in which events can prompt *InfoLease* to interact with

other systems in gathering data or making processing decisions. As with the Access Framework, areas of the *InfoLease* system are being evaluated and prioritized for broadcast functionality.

Using the ability to communicate customer changes as an example, the broadcast framework uses components of the API Administration tool, XML, and UniData functionality. As receiving applications do not necessarily care about all customer changes, the API Administration utility is used to define data to be watched. When that specific data changes, a “return API” is defined to return only the data that applications want to receive. A facility also exists for synchronizing data upon initial setup.

For moving the data out of *InfoLease*, all data is XML-based and conforms to *Integration Manager XML* standards. There are two options for transport, depending on the UniData version in use. For UniData 5.2, there is native support within the database to send data using the HTTP protocol. For UniData 6.0, support for the Application Messaging Interface (AMI) was added. Note that WebConnect and MessageConnect are not involved with the data transfer, as communication is directly with third party applications, either through a MOM or through a standard XML transaction. When using the AMI functionality, the environment may look like the following.

Figure 4 – Broadcast AMI Overview

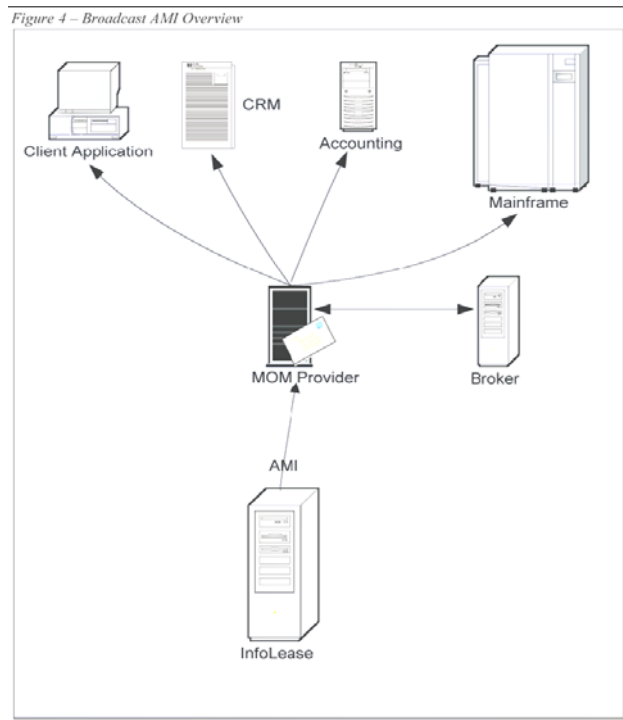
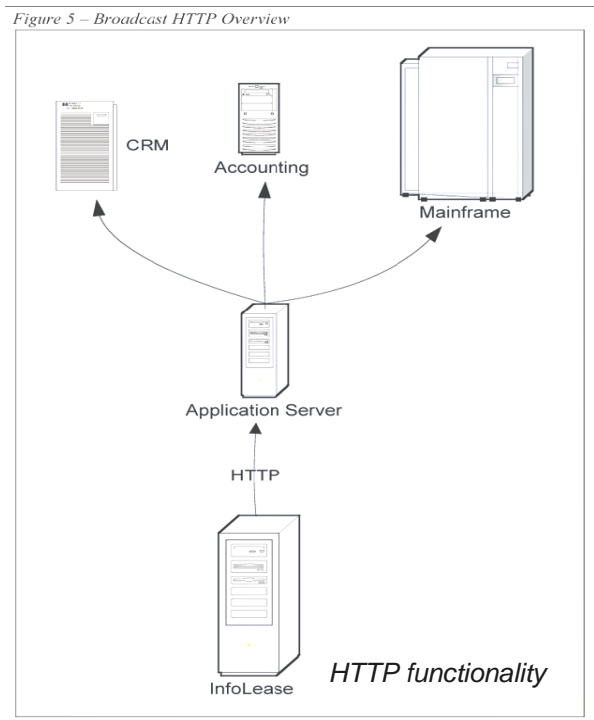


Figure 5 – Broadcast HTTP Overview



Actual installation locations and requirements, firewalls, etc. will vary with the specific environment.  
Stack Diagrams:

The following figures show Access Framework and Broadcast Framework stacks.

Figure 6 – Access Framework Stack

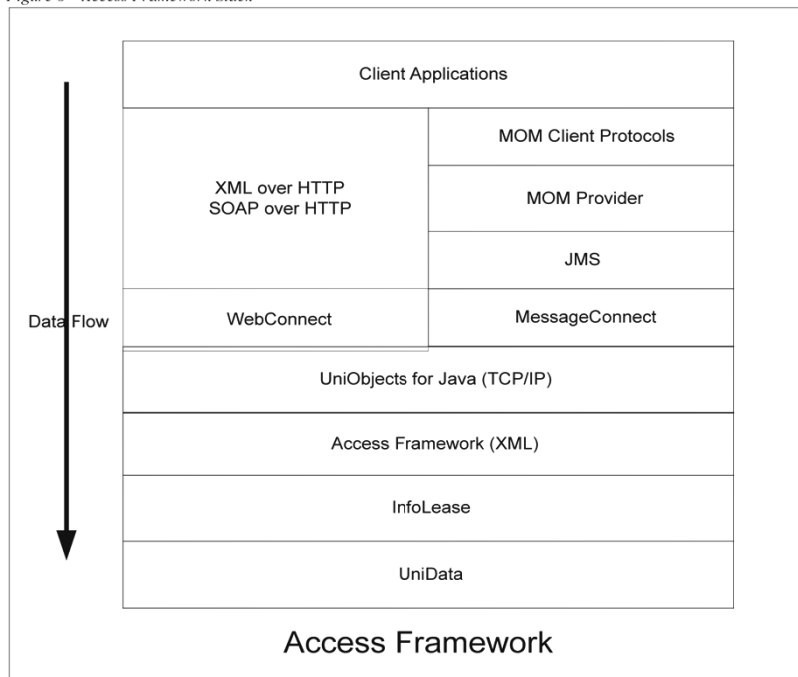
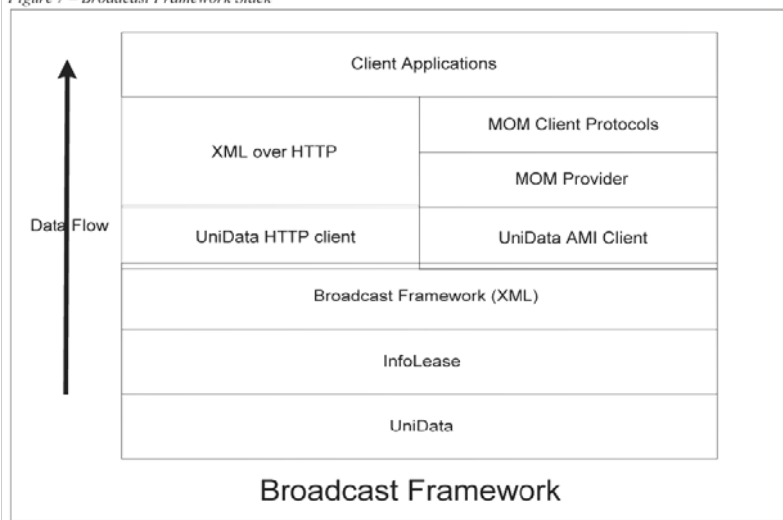


Figure 7 – Broadcast Framework Stack





INTERNATIONAL  
DECISION SYSTEMS®

#### **UNITED STATES**

International Decision Systems  
1500 IDS Center  
80 South 8th Street  
Minneapolis, MN 55402 USA  
Tel: +1.612.851.3200  
Fax: +1.612.851.3207  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

#### **EUROPE**

IDS Limited  
Norton House  
1 Stewart Road  
Basingstoke  
Hampshire RG24 8NF  
United Kingdom  
Tel: +44.1256.302000  
Fax: +44.1256.302005  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

#### **AUSTRALIA**

IDS Pty Ltd  
Level 5  
234 George Street  
Sydney, NSW 2000  
Australia  
Tel: +61.02.9247.9166  
Fax: +61.02.9247.9212  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

#### **ASIA/PACIFIC**

International Decision Systems  
9 Temasek Boulevard  
#15-02A Suntec Tower Two  
Singapore 038989  
Tel: +65.6333.9866  
Fax: +65.6333.9877  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

IDS Software Solutions (India) Private Ltd.  
6th Floor, Tower D, Corporate Block  
Diamond District, Airport Road  
Bangalore, 560 008 India  
Tel: +91.80.55150910  
Inquiries: [idsindia@idsgrp.com](mailto:idsindia@idsgrp.com)

04-03-1008IM