BroadQuest
THE DATAWEB COMPANY™

Dataweb Architecture
White Paper

CONTENTS

PART ONE—A HIGH-LEVEL VIEW

I. THE DIVIDED ENTERPRISE

The "Divided" Enterprise for Business

Enterprise-wide data access suffers from a lack of comprehensive, real-time integration—both from a customer sales, service, and support perspective, and from an interdepartmental perspective.

A customer calls a service agent and expects not only that the sales agent knows that the customer talked with another sales agent a day before and a particular technical support agent a week before that, but also the exact nature of both calls and the specific ways each agent attempted to satisfy the customer. In our highly technological society, the customer rightfully expects every company representative to have accurate, complete, and real-time customer information.

Likewise, a sales or support agent wants accurate, complete, and real-time customer histories for sales, payments, warranties, and problem calls—everything that will help that sales or support agent immediately satisfy any customer request or demand. Furthermore, when the support department implements a new warranty policy, or when the inventory on a particular item falls to zero, the sales agent wants real-time access to that information. Marketing also wants real-time information for product orders or technical support problems, and management wants to create reliable real-time reports that accurately reflect the current state of the company.

Any backlogs in information and policy updates divide the enterprise, inhibit cross-department teamwork, and result in lost sales, higher customer turnover, poor product planning, and higher support costs.

The "Divided" Enterprise for the Internet

This divided enterprise also manifests itself in a somewhat different form on the Internet—researchers and other information gatherers have trouble locating specific, concise sets of relevant information. The current search and retrieval tools return notoriously large amounts of useless documents and Web sites, and even fail to successfully retrieve many that *are* specifically relevant.

A search engine seeks documents that only *generally* match. What everyone wants is the ability to access and query every database and every document specifically, and to ask each one in effect, "Are you precisely what I am looking for?" Or even more powerfully, "I don't know exactly what I am looking for, but I will know it when I see it."

The solution to both divided enterprises is a new kind of communications model: dataweb architecture, which allows you to quickly see what is available and efficiently drill down to the files you want and even to files you didn't know you wanted.

II. THE BROADQUEST DATAWEB

The BroadQuest Solution: Dataweb Architecture
Unlike current approaches to solving the problem of the divided enterprise that offer centralized models—such as harvest and search, harvest and push, or publish and subscribe—BroadQuest has built a set of intranet- and Internet-based software technologies that exploit a many-to-many transaction model: dataweb architecture.

This architecture unifies the divided enterprise so that every user has direct, comprehensive, and real-time access to all databases needed by users to most effectively accomplish their jobs.

The dataweb provides:

- Revolutionary Search capabilities that filter out irrelevant data and get users what they need fast
- Powerful individualized or user-group configuration capabilities
- Customized monitoring of data sources anywhere on the enterprise
- A complete palette of data transaction types beyond Search, including Post, Publish, and Subscribe.

The dataweb also delivers a single system to unify the entire enterprise and to provide:

- The leveraging of existing intranet, application, and database investments
- Secure access for multiple user types via internet, extranet, and the Internet
- High configuration capabilities to meet changing business needs
- High performance data access regardless of network size
- Data access for channel partners and customers via the corporate Web site.

Abstract of a Dataweb
A dataweb represents the all-encompassing “potential” for all possible kinds of particular “datawebs." A dataweb is an administrator-customized, user-configurable roadmap of “channels” for sets of data. At the highest level, a dataweb comprises three sets of partially overlapping technologies: user technologies, channel technologies, and database technologies.

User Technologies

BroadQuest Consoles are browser-compatible application software -- completely data-driven by its Broker software on the server – that intelligently displays a real-time, unified view of disparate data sources. Users can Search, Post, Subscribe, and Publish from the Console GUI. Each BroadQuest Console can customize views of customer and product data according to the user's function, role, task, and personal preferences. Sales reps, product managers, support reps, and system engineers can now all be linked to the same data sources while each department has a different look on their Console. And the Console Broker can even drive data to be displayed through alternate applications, such as Vantive, SAP, and PeopleSoft.

BroadQuest Console Brokers act as “intelligent” multi-user, multi-channel translators that access, monitor, update, and deliver information within the dataweb framework. A set of Console Brokers and Data Brokers establish the potential for an enterprise-wide dataweb of information across all types of datasources. Every BroadQuest Console connects to a BroadQuest Broker, as does every database and application. Every Broker has specialized subsystems, called “Engines," “Routers," or "Wrappers" that authenticate, consolidate, filter, map, and channel incoming and outgoing data. Every Console Broker can also function potentially as a Data Broker whenever the user is allowed to Post or Publish data.

Channel Technologies

The BroadQuest Dataweb is an intranet- or Internet-based “many-to-many broadcasting” technology that uses IP multicasting when available and software routers that function like IP multicasting. Unlike unicasting, the Internet’s standard means of communication that sends packets of information to one site at a time, IP multicasting sends packets to multiple sites simultaneously. BroadQuest Brokers are built to take full advantage of this protocol as it becomes more widely available on the Internet. When IP multicasting is unavailable, the BroadQuest Dataweb employs self-contained software routers that accomplish the same multi-site distribution that IP multicasting provides. The BroadQuest Dataweb “broadcasts” in a kind of circuit that allows one Broker to communicate with every other Broker in the dataweb. The dataweb administrator establishes the boundaries of possible dataweb configurations and channels through BroadQuest Director.

BroadQuest Director is the primary control console for the Dataweb Administrator. Through Director, the Administrator can create and extend a variety of customized datawebs, dataweb components, and data sources. Director can create, modify, and delete user and group privileges. Current network administrators, database administrators, and webmasters all have the required skills to manage dataweb services via the director.

The dataweb administrator creates custom datawebs to meet the particular ongoing and changing needs of a company. BroadQuest works with each company to help determine their special data needs and to develop datawebs to meet those needs. For example, a sales department may need a difference set of databases and retrieval tools than the accounting department. A different dataweb would be customized for each department. For the Internet, merchandise sales datawebs and job datawebs are among the countless possible datawebs that can be created.

DB Technologies

BroadQuest Data Brokers act as “intelligent” multi-user, multi-channel translators that access, monitor, update, and deliver information within the dataweb framework. A Data Broker is attached to a database or set of databases, or to an application. Each Broker is designed to let the dataweb know the types and categories of data that are available on its data sources. Brokers are assigned certain "channels" that enhance the efficiency of the dataweb by limiting access only to those user actions that are relevant to that data source. Like Console Brokers, every Data Broker has specialized subsystems, called “Engines," “Routers," or "Wrappers" that authenticate, consolidate, filter, map, and channel incoming and outgoing data. In fact, all Brokers are identical apart from their Wrappers, which are specially designed to integrate each data source or Console into the dataweb.

The Broadcasting Analogy for the Internet

A dataweb functions like broadcasting. The atmosphere is constantly filled with radio, television, and satellite frequencies that exist simultaneously. The Federal Communications Commission (FCC) determines the different “kinds” of frequency sets (AM, FM, VHF, UHF, emergency bands, and satellite) and who is allowed to broadcast on what bands. Your radio tuner is configured to pick up a select range of frequencies, and you can tune to different stations than your neighbor.

The dataweb is full of information just like the airwaves. The various databases are like broadcasting stations; however, these stations have the unusual ability to broadcast on several frequencies or "channels" simultaneously. The dataweb administrator through BroadQuest Director is like the FCC, who determines which "stations" (databases) are allowed to broadcast on which channels. And your BroadQuest Console is the tuner. Even though the dataweb has the “potential” of all information, you and your neighbors can selectively “tune” into those channels that your administrator has created. And you can specify data sets that you need right on your console.

But the dataweb provides capabilities that your tuner lacks.

Your dataweb "tuner" (your Console Broker) knows what channels to "listen in" on once you perform a search. Suppose broadcasting of all types were architected like the dataweb. With the BroadQuest dataweb architecture, you could perform a search for details on all programming related to The Beatles’ live performances in all formats (audio, video, photos). Your tuner (Console Broker) would “broadcast” the search order to all stations (Data Brokers) that are part of the channels relevant to your request. In other words, your tuner knows specifically which stations it needs to broadcast to in order to retrieve the results you need. Those stations would search their musical catalog abstracts for everything related to live performances by The Beatles and return the results that partially or wholly match your original search order. Furthermore, your tuner provides you with the ability to drill down quickly through all kinds of categories in such a manner that you can quickly make discoveries of data you didn't know you wanted. For example, you might find that someone had created a file of articles related to road personnel who told stories about their experiences with individual Beatles after the group had broken up.

Furthermore, with dataweb you could establish a subscription that would automatically notify you of all future live Beatles programming. Or you could actually “post” a live recording you have for others who are interested in it. You could even provide a “publication” to others who have notified you of their particular interests, should you have something they can use. You could establish a "Live Beatles User Group" that filtered out all information except that in which you were specifically interested, and the filtering would be so precise that not only would you *not* receive irrelevant responses, but you would never miss one that *was* relevant.

And your neighbor could be using the same dataweb for similar searches, postings, publications, and subscriptions, all configured or applied to meet their needs without affecting your "broadcasting network" in any way.

III. MORE ABOUT DATAWEBS

The Auto Dataweb: A Simple Example of an Internet Dataweb

The following dataweb profile shows how a BroadQuest Dataweb can work to facilitate the purchase of an automobile. In this case, a regional newspaper sponsors the "Auto Dataweb," a forum where users go to buy and sell automobiles.

As the original sponsor, the newspaper uses its abundant experience in automobile classifieds to define the dataweb classification system, which describes for all vehicles the relevant categories, attributes, and values (category attribute "Automobile," category value "Ford Mustang," attribute "Top," value "Convertible"). It then recruits local dealers, car Brokers and financiers to license seats on the Auto Dataweb. The newspaper also allows individuals trying to sell used vehicles to list them using the Broker sponsored by the newspaper on the dataweb. With all the sellers and financiers participating, the newspaper provides critical mass for starting the dataweb, much like a number of auto dealers who choose to co-locate to attract comparison shoppers.

The dataweb goes on-line with thirty new-car dealers, eleven auto Brokers and 35 used car lots who have purchased BroadQuest Broker software and installed it on their servers. In essence, all the participants are then "listening" to the Auto Dataweb for information requests.

A typical transaction goes as follows: An Internet user contacts the Auto Dataweb home page provided by the newspaper and is presented with a simple template. He selects Automobile from a menu of categories, and is then prompted to select the characteristics of the vehicle he is seeking. He completes the template by selecting the color red, 1994 to 1996 for the years, sedan for the model, 4-door as a value and Germany for the country of manufacture. When he finishes, he clicks a Broadcast button that transmits his request for information onto the Auto Dataweb.

All the participating Brokers listening on the dataweb receive the request. Each server decides if it has any relevant information to offer the requester. Several of the dealers do not respond because they sell only new cars, others because they do not sell German vehicles. But two European vehicle dealers, a used car lot and two individual sellers have vehicles that match the buyer's criteria. All five of these servers respond to the buyer with a URL pointer and a citation explaining what they have.

The buyer can then visit each site directly, viewing exactly the information desired: a picture of the listed vehicle and detailed information about it. What could have been an exhausting search takes just minutes to complete. If the buyer fails to find exactly what he wants, using the data already entered in the template he can create a standing request that he be notified when new listings occur that match his criteria. After making an auto selection, the buyer could use a similar process to shop for a loan to finance his vehicle purchase.

The Customer Dataweb: A Simple Example of an Corporate Intranet Dataweb
Suppose Amazing Silicon, Inc. is a $5 billion international hi-tech company whose products include both hardware (mainframes, servers, personal computers, storage, and printers) and software (operating systems, networking, systems management, and applications). They also provide services (support, installation, integration, and outsourcing) and, as a consequence, face the classic issues of a large company with fragmented silos of activity.

With 5,000 sales agents and 800 support reps worldwide, the company has decided to overcome the barriers of its divided enterprise with a customized intranet dataweb. Those barriers have prevented support reps from accessing 1) any product information outside of the one they are assigned, 2) any billing, shipping or install status information, and 3) any information on the products or services that the sales force is currently trying to sell. Those same barriers have prevented sales agents from accessing 1) information on support problems their customers are having, 2) information on the kind of business the company's business partners are doing with their customers, and 3) "big-picture" information on their customers' financial relationship with the company, such as accounts payable or maintenance renewals.

With their new dataweb, the company now can get support reps and sales agents to work together. When a customer calls, a support rep can immediately get the customer's sales and support history, including calls to other support reps, and the support rep is able to help the customer combine support issues with new product sales and even current product shipping status. Customers are impressed with the single point of contact for all their needs.

Sales agents use the dataweb to be immediately notified whenever their customers are facing unusual support issues, or when a customer is doing business with a business partner. Like the support rep, the sales agent can bring up every customer's current billing, shipping, and support status so that customers need never be transferred to another department or find themselves having to repeat the same story more than once.

With the dataweb, the executive sales staff has chosen a configuration that allows worldwide access to the daily numbers regarding sales and support, and they have instituted "notification flags" that automatically let them know when certain high or low thresholds are reached by continent, country, region, or city; or by product or service class, individual product or service; or by sales or support groups or individuals.

The power of the dataweb continues to grow as the company finds new ways of configuring and correlating its worldwide databases on a real-time basis, and setting up anyone or any group in the company to access any set of real-time data that helps them do their job most efficiently and completely.

IV. BUILDING A DATAWEB

The Internet Dataweb

The three participants required to build a successful Internet dataweb include the creator or owner of the dataweb, a Broker owning a seat on the dataweb, and a user of the dataweb.

Internet Dataweb Creator/Owner

The first step in establishing a new Internet dataweb is for a sponsoring organization to contract with BroadQuest as the Internet dataweb owner. The prospective dataweb creator needs to:

- Obtain appropriate software and dataweb licenses from BroadQuest
- Use BroadQuest development tools to define and specify the dataweb rules for the dataweb
- Have ready access to buyers and sellers who are interested in participating in the topic of the dataweb

The prospective creator/owner may vary depending on the market segment being targeted. Following are the likely sponsors by market segment:

- On-line corporations bringing many distributed sellers together with a large body of buyers to facilitate their matching.
- Corporations building intranet datawebs to connect employees and distributed information sources, as well as extranet datawebs to link in customers and suppliers.
- Associations and affiliated groups creating a data market to facilitate communications and sharing of information among members.

Internet datawebs are inherently scalable and highly customizable. As a particular dataweb develops and matures, the owner can alter the dataweb by adding or deleting Brokers, create links to other dataweb or adapt the dataweb classification system to reflect current topics of interests or changing needs.

Attract Content Providers to the Dataweb

The next step in building a dataweb is to attract content providers who license seats on the dataweb. The incentives for doing so are similar to those for leasing space in a shopping mall. Brokers join a dataweb when they have content of interest to the community of buyers who will access that dataweb. A prospective Broker needs to:

- Obtain a seat on the dataweb from the dataweb owner
- License the necessary software from the dataweb owner
- Use the software to configure the dataweb to attract target users to the site

When content providers have information of interest to several different types of buyers, they may choose to belong to more than one dataweb. In a corporate sponsored dataweb for example, individual divisions might have their own dataweb, tailored to their own needs, and then use a Broker to become part of the Corporate dataweb.

Recruit Users of the Dataweb
The final step in building a dataweb is to attract its users. By participating in a dataweb where all members have common interests, a user gains important benefits that include the ability to:

- Quickly construct well-formed requests specific to the dataweb
- Receive relevant responses from multiple disparate databases and web sites
- Receive publications from relevant data sources represented on the dataweb
- Save time in searching for information or completing a transaction

Therefore, by participating in a dataweb, a user can not only find information faster but also be guaranteed that all responses received are relevant and current.

The Corporate Intranet Dataweb
A company wishing to develop its own dataweb may begin by

- Obtaining appropriate software and dataweb licenses from BroadQuest
- Using BroadQuest development tools to define and specify the dataweb rules for its dataweb that best match the needs of its departments

Or the company may wish to look over BroadQuest's pre-created datawebs, such as CustomerQuest, which provides the kind of unified enterprise that many companies have been searching for.

PART TWO—A TECHNICAL VIEW

V. BROADQUEST CONSOLES

With the BroadQuest Console, a user to perform one of four actions: Search, Post, Subscribe, and Publish. The dataweb Search action is at once powerful and revolutionary by offering a detailed hierarchical "drill down." This approach allows the user not only to find specific data files quickly, but also to accomplish the kind of search that helps a user who says, in effect, "I don't know exactly what I want, but I will know it when I see it."

Search

Once the dataweb administrator has categorized all data sources, the user can "drill down" through the Categories to quickly Search for exactly the right data. For example, a product manager may choose "New Products" and see that the 1,427 matches can be accessed geographically. Choosing "Australia (21)," she may note twelve matches for "Conceptual," six for "In Development," and three for "Ready to Ship." Furthermore, they can be chosen based on "Text Document (15)," "Presentations (4)," and "Pictures/Graphics (2)."

The Product Manager can then immediately switch to a Summary view of all 21 records and drill down further to refine the Search based on highly specific attributes. Or she can switch to a Results view that sorts in ascending or descending order the most relevant matches, or switch to a highly detailed description of all the actual matching attributes. Searches can be saved for other team members to use.

Post

The user can Post data that can be accessed by other team members who Search. They do not have to know that such data has been posted to find it. Which user or user groups can actually Post data is determined by the dataweb administrator, who can also attach access privileges by individual, department, or any other classification.

Subscribe

Users can arrange to Subscribe for data; i.e., establish a standing order for certain customer or product data that is automatically sent to the user when available on the dataweb. This function meets real-time needs so that any user or department can be immediately informed of relevant customer or product data as soon as it is added to a dataweb data source.

Publish

Users can Publish data that goes out to subscribers. The dataweb allows continual broadcast of all updated data, but only authorized subscribers will receive the published data. Every department can arrange for separate sets of published data.

The BroadQuest Console GUI consists of two frames: An LHS frame for selecting a category or categories and an RHS frame that displays one of four pages, depending on the user's actions and the number of matches found:

1. A table of Attributes and forms for constraining their Values

2. A summary of the records found in a search order, showing for each Attribute, an abstraction of the distribution of Values found for that Attribute. User's can refine searches by drilling down from this summary page.

3. A table of the most relevant matching orders. The results can be sorted in ascending or descending order based on any of the column headings (Attributes) in the table.

4. A detailed description of any of the actual matching orders returned

The BroadQuest Console provides many features for facilitating the creation of orders:

- An LHS frame allows users to drill down through the hierarchy of Categories to find the Category that best matches the order they want to create. To avoid getting lost in a forest of Categories, users dynamically control which Categories display their subcategories.

- Choosing a Category in the LHS frame determines the set of Attributes that appear in the RHS frame. These Attributes are used to specify more precisely the order desired.

- A user-controlled switch is provided to enable/disable the selection of multiple Categories in an order.

- The Console (and the underlying dataweb) is designed to handle very large matching answer sets and to display them in a way that facilitates intuitive exploration. Rather than simply showing "the first 20 of 30,000 responses," the Console displays a high-level summary of the answer set, which can be browsed further by adding constraints to the search order.

- The format used in the RHS frame to display the Attributes that can be constrained in forming an initial search order is the same as that used to display the summary results produced by submitting search orders. This facilitates a natural "drilling down" paradigm of successively refining searches until the desired matches are found.

At any time in the drilling-down process, the currently constrained Attributes can be used to submit a search order or a subscription order. This capability makes it easy for users to submit orders that are similar to existing orders and increase the likelihood of consistency in the resulting databases of submitted orders.

VI. THE BROADQUEST BROKER

Brokers act as "intelligent" multi-user, multi-channel, and multi-database translators that access, monitor, update, and deliver information within a consistent dataweb framework. A set of Brokers establish the potential for an enterprise-wide or Internet-wide dataweb of information across all types of users and datasources, and support individualized "views" of that dataweb for each Consoles or set of Consoles.

Brokers communicate with each other via the dataweb, and whenever possible, the architecture takes advantage of IP multicasting; otherwise, internal software routers within the Brokers take over and provide the same functions. Brokers have six customizable subsystems:

Wrapper: Has specific intelligence about a single data source or class of data sources. Uses a mapping script to allow orders to be optimized between dissimilar data sources. Different Wrappers are needed to connect a Broker to a Console, to a database, to an application, and to other datawebs. A single Broker can use multiple Wrappers to connect with several databases.

Authenticating Engine: Provides user, data, and transport security; defends against unauthorized clients; encrypts communications; allows custom configuration to restrict data results; and provides Security API for third-party security software.

Channel Router: Routes orders to relevant Brokers; and maintains high-level data abstractions and Broker subscriptions to identify Brokers having query answers or maintaining publications.

Indexing Engine: Maintains a dataset abstraction to protect from access overload and to resolve most queries directly.

Consolidating Engine: Combines and consolidates incoming data results to protect user from information overload.

Matching Engine: Verifies that incoming data conforms to the order.

A Broker interfaces with specific data sources through Wrappers.

Wrappers exist that can connect with:
Relational Databases: Oracle Sybase, MS SQL, Informix, DB2, and ODBC
Messaging Systems: IBM MQ Series, MS MQ, and Oracle AQ
Search Engines: Verity
Flat Files: Spreadsheets, word processors, and CSV files

Wrappers also can integrate with specific applications:

> **Customer Interaction Systems:** Vantive, Scopus, Clarify, Siebel, and Aurum
> **Groupware Systems:** Domino, MS Exchange, HTML and Web Servers
> **Enterprise Resource Planning Systems:** SAP, Oracle Financials, and PeopleSoft

Mainframe legacy systems such as CICS or IMS are accessible via drivers to IBM MQ, MS MQ, and Oracle AQ.

A Low-level Example
(For this example, all Brokers apart from the Console Brokers that connect to data sources are called "Data Brokers.) The user's search action is encapsulated into BQ Format (the language of the dataweb) by the Wrapper, which uses Datamap information to map the search into an "order." (See *An Example of How the Datamap Works* in section VII. BroadQuest Director.) The order is then broadcast to the Transactional Database of the Console Broker through the dataweb.

The Authenticating Engine
The Console Broker passes the order to the Authenticating Engine, which checks the user's credentials against its internal database or some external authenticator. When the user is authenticated, the Authenticating Engine sends the order to the Channel Router. On the receiving end, the Data Broker also uses its Authenticating Engine to verify that the order has been received from an authenticated Console Broker.

The Channel Router
The customizable Channel Router in the Console Broker determines to which Data Brokers the order is sent. The set of Data Brokers can be widened or narrowed to optimize the number of relevant responses to the order.

When relevant channels are discovered, both the User and Data Brokers create a transaction record that is stored in each Broker's Transaction Database. The transaction contains the originator address, an interactive timer, and a termination timer.

Each channel has zero or more member connections, and each connection corresponds to a Broker or database client. A channel can be in multiple membership lists, and information is maintained for each connection (at minimum, a receiving address). Channel memberships are determined either statically by an administrator, or dynamically by sending a message to each connection asking which channels are of interest.

The Channel Router's input consists of:

> **An Order**, which is compared against channel definition orders to determine relevant channels;
> **A verb describing the intended action**, which selects the appropriate channel definition orders;
> **A channel definition file**, which holds channel information: channel definition orders, memberships, and members' addresses.

(For a discussion defining "definition orders" see the next section BroadQuest Director under "Data Administration.")

The Indexing Engine
The Indexing Engine works in the Data Broker with the Matching Engine and the Consolidating Engine to protect a database from access overload by maintaining all the searchable fields of the database's records in compressed from in memory and sequentially scanning them to determine matches. The Indexing Engine uses compact identifiers to represent actual values.

Orders are passed to the Indexing Engine to see if possible matching responses can be made without consulting the original database. If the Indexing Engine does not have sufficient information to respond, the order is passed to the Wrapper, which initiates a process that consults the database directly. The Wrapper queries the database for "candidates" of possible matching responses. The Wrapper converts these candidates into BQ Format and sends them back to Data Broker. Once all responses are ready, or when the interactive or termination timers expire, the responses go to the Matching Engine.

The Matching Engine
The Matching Engine compares the original order to the responses and determines when the responses partially or exactly match. Both Console Brokers and Data Brokers use the Matching Engine to guarantee that all responses obey all the constraints of the original order. All final matches are then sent to the Consolidating Engine.

The Consolidating Engine
The Consolidating Engine consolidates all the matching responses and summarizes them. When the interactive timer expires, the Consolidating Engine sends to the Console Broker all responses accumulated to that point. When the termination timer expires, the Consolidating Engine sends to the Broker all responses accumulated to that point and closes the entire transaction.

The Consolidating Engine retains the most relevant records given to it. It keeps track of the number of occurrences of each field value in the records it processes. It records the total number of records that have at least a partial match and also the number of records having exact matches. It keeps the attribute count for all the values in that category and the most frequent values with their counts for that particular category. It also generates the result of all the processed records in BQ Format. The result contains different categories with the total attribute count for all the values in that category, the most relevant values with their counts, the number of records that have exact matches, and the number of those that have at least partial matches.

VII. BROADQUEST DIRECTOR

BroadQuest Director is the primary control console for the dataweb administrator. Through Director, the dataweb administrator can create a dataweb, define channels within the dataweb, and configure Brokers and Consoles. The dataweb administrator can choose among several administering options:

- Dataweb Administration
- Broker Administration
- User Administration
- Data Source Administration

Dataweb Administration

The Dataweb Administration GUI offers the dataweb administrator an HTML forms-based interface to administer dataweb-wide settings for the Datamap and Broker Channels. The dataweb administrator can:

- Load a Datamap
- View the currently loaded Datamap in its source form and save it to a file
- Modify the Datamap with the Datamap Editor
- Create or modify channel definition orders
- Install new channel definition orders derived from the available standing channel definition orders
- Update Broker Channel memberships
- Force all Brokers to synchronize with BroadQuest Director

"Channel definition orders" are orders that are defined by the dataweb administrator to partition the dataweb into channels. Until channels are created, the dataweb remains entirely general in scope—all Brokers are transmitting to and receiving from all other Brokers. The easiest way to create a viable channel within the dataweb is to associate a particular order with that channel. The order imposes constraints by defining the kind of orders allowed on that channel. Thus, administrator-defined channel definition orders effectively filter orders generated from other Brokers. The GUI used by the dataweb administrator to define a channel is identical to the general user's GUI, because that is the natural interface for defining orders.

The Datamap

Each piece of BroadQuest software depends on and is driven by the Datamap, which holds the information that distinguishes the behavior of each dataweb. It holds the dataweb Classification System, it tells the Wrapper how to convert between BQ Format and external value representations, and it provides Brokers with the necessary information to determine whether orders and responses match.

An Example of How the Datamap Works

Brokers communicate by transmitting orders to other Brokers in the dataweb. An order is a set of attribute-value pairs, some of which may be system properties of the dataweb:

Color => [red], Make => [Ford, Honda], BQArrivalTime => [19971109324567],

and

BQOrderType => [Offer].

The Matching Engine is a very general mechanism that uses the Datamap to determine when orders match. For a very simple example of two orders that match consider a search for red or yellow cars (represented as Order1) and a database entry offering a red Ford (represented as Order2):

```
Order1) {BQCategory => [car],
         BQOrderType => [Request],
         Color => [red, yellow] }

Order2) {BQCategory => [car],
         BQOrderType => [Offer],
         Color => [red],
         Make => [Ford]    }
```

In this greatly oversimplified example, the Datamap is used to determine the matching criteria for each of the attributes:

for BQOrderType, Requests match Offers,

for Color and BQCategory, the Request set must intersect the Offer set.

In reality, the matching criteria are much more powerful and much more symmetric; orders that represent Offers can impose conditions on matching Requests.

Classification System
The purpose of the Classification System is to provide a common language for expressing succinctly and with great precision what it is that users seek and that publishers have to offer. The system is based on three notions: Categories, Attributes, and Values. Intuitively, Categories are a kind of "super" attribute used to identify the general domain in which one is searching; for example, "Vehicles," "Merchandise," or "Customer Service Information." Associated with each Category are Attributes—fields that further distinguish properties within that Category. For example, Attributes associated with the "Vehicles" Category might be "Make," "Year," "Color," and "Price-range."

Finally, each Attribute field can be required to have particular Values. For example, someone might specify that she is looking for cars whose Make is "Ford," "Chevrolet," or "Plymouth," and whose Color is "Red."

The Datamap contains the formal specification of the Classification System together with certain "properties" of attributes and values. It consists of nine sections:

AllAttributes
a list of all the attributes that appear in the Classification System

Properties
a hash table whose keys are attributes and values; gives, e.g., display properties for attributes and print-names for values

InputAttributes
a hash table whose keys are categories; tells the Console which attributes to display when a category is first selected

SummaryAttributes
a hash table whose keys are categories; tells the Console the attributes for which summary/consolidation information is to be displayed

ResultAttributes
a hash table whose keys are categories; tells the Console which attributes to display in the results table

DetailAttributes
a hash table whose keys are categories; tells the Console which attributes to display in the details frame

ShowInitialCategories
a list of the categories that should start "open" in the lhs frame

SubCategories
a hash table whose keys are (non-leaf) categories; contains the hierarchy of categories ('BQTop' represents the root)

SubValuesCategories
a hash table whose keys are (non-leaf) values; contains the forest of value hierarchies

In defining a search query or describing the contents of a publisher's database, it is much more convenient to be able to browse a set of possible values than to have to remember or guess the appropriate terms. For this reason the values in the Categories and Values tables are hierarchical. Associated with each attribute is a particular "control," which describes how that attribute's values are to be presented and selected in the user interface.

The Datamap Editor
The Datamap Editor GUI is designed for creating and modifying the Datamap. Since many of the operations of the Datamap Editor require the user to assign Values to properties of Attributes or Categories, the Datamap Editor is based on the BroadQuest Console. Therefore, a thorough understanding of the Datamap Editor requires thorough knowledge of the BroadQuest Console as well.

The Datamap Editor interface consists of two frames:

- An LHS frame for selecting which Attribute or Category to edit
- An RHS frame for entering the specific information for the entity selected

The LHS offers the options of editing:

- The set of all Categories - Categories can be added hierarchically or deleted from the Datamap
- The set of all Attributes - Attributes can be added or deleted
- System properties - used for customizing the look and feel of the resulting BroadQuest Console
- The properties of individual Categories

Broker Administration
The Broker Administration GUI of BroadQuest Director (the Director Host) allows dataweb creation, deletion, and connection. A Broker Administration GUI is also available with each Broker (the Broker Host), but it has restricted capabilities. On any Broker host, the Broker Administration GUI allows administration of Broker-specific settings such as a Broker's connection (subscription) to datawebs, its user accounts, and its data source connections.

On the Director Host, the Broker Administration GUI offers extra functionality relevant to the Director Host's capabilities. Its use is restricted by a master installation password. The Broker Administration GUI allows the Dataweb Administrator to perform the following actions:

- Create a new dataweb
- Delete an existing dataweb
- Administer an existing dataweb

On all Broker Hosts, including the Director Host, the Broker Administration GUI allows the following actions:

- Subscribe to a dataweb created on a Director Host. To establish this connection, a set of credentials supplied by the Dataweb Administrator has to be supplied
- Unsubscribe from a dataweb created on the Director Host
- Synchronize with the Director Host by retrieving the latest dataweb-wide configuration information
- Administer user accounts using the User Administration GUI
- Administer data source connections using the Data Source Administration GUI

The Director Host has a superset of a Broker's capabilities. As such, it supports the Broker Administration functionality described above, and can subscribe to datawebs created by other dataweb authorities, towards which it acts as a simple Broker. However, it does not support the ability to Subscribe and Unsubscribe to datawebs created by itself. The Director Host is always implicitly subscribed to all datawebs created through it, even if its role is only administrative and isn't used for user and database client connections.

The main components of the Broker Administration GUI on the Director Host are:

- State maintenance
- Authorization control
- Dataweb installation verification
- User action selection
- Dataweb Creation, Deletion, and Administration
- Dataweb Subscriptions
- Broker to dataweb authority synchronization
- HTML form generation and user interface interaction

User Administration

The User Administration GUI allows the administrator to determine which users are added to and deleted from the dataweb, and how password security is designed. The administrator can also create User Groups; for example, a corporate dataweb may group users by department.

Outside of the GUI, the administrator has a powerful option to create a text file that limits each user's actions; i.e., Search, Post, Publish, and Subscribe. For example, some users can be limited to Search, some to Search and Subscribe, while others can Search, Post, Publish, and Subscribe. In a corporate environment, Level 1 support reps may be limited to Search, Level 3 support reps to Search and Post, while the executive staff may perform all actions.

Data Source Administration
The Data Source Administration GUI allows administration of a Broker's data source connections. It is implemented as a CGI program and offers an HTML forms-based interface. Data source connections are private to each Broker. Even though data retrieved from a Broker's data source connection is available to queries originating from other Brokers, the data sources themselves are not.

The Data Source Administration GUI can be used on any Broker Host. Its use is restricted to members of that Brokers Administrators group. It allows them to perform the following actions:

- Create a new data source connection
- Delete a data source connection. This action removes connection information and locally held (cached) copies of data, but does not actually affect the original source of data.
- Edit a data source's mapper configuration using the Mapper Editor

For uploaded data sources (Flat-file type) the following actions are also available:

- Load a data file containing data in comma-separated-value (CSV) text file format.
- Download (export) data into a CSV text file

- Delete records as specified by a CSV text file or a list of Ids
- Delete all records held by the data source

The main components of the Data Source Administration GUI on the Director Host are:

- State maintenance
- Authorization control
- User action selection
- Data source connection creation/deletion Uploaded data maintenance
- HTML form generation and user interface interaction

The Mapper Editor
The Mapper Editor is designed to be a very general tool; it is used both to create new mapper scripts and to edit existing mapper scripts, and it can handle a wide variety of data sources, ranging from ODBC clients to Domino full text searches to simple csv files. As such, it offers several features, whose operation may take some time to understand fully and to use efficiently.

When creating a new script, there are conceptually two phases:

Phase 1:	Entering formatting and connecting information
Phase 2:	Mapping between BQ concepts (Attributes) and database concepts (Fields)

Information entered in Phase 1 is used to enable Brokers to know how to connect to data sources and how to interpret data received from those sources. New script creation is logically two-phased since mapping Attributes to Fields in Phase 2 cannot begin until the editor has sufficient information from Phase 1 to know how to retrieve the Field names from the data source being wrapped.

In the process of creating a new script, as information is entered the Mapper Editor attempts to go on to Phase 2 whenever it has sufficient information to retrieve Field names and make an attempt at suggesting potentially useful mappings.

When editing an existing script, the Mapper Editor will display the information currently contained in the script and allow a variety of changes and operations. Experienced data source administrators will be able to exploit this feature to save time and labor by using mapper scripts from previously wrapped sources as starting points for wrapping new sources.

VIII. HOW DATA MOVES

Dataweb Overview

The BroadQuest dataweb consists of Broker applications that provide some user-interface options to form both "active" and "standing" requests and offers, to broadcast those requests and offers to a community of other Brokers listening for such requests and offers, and to receive intelligible responses to those requests and offers. A "datamap" (catalog) specifies the user-interface options. The system also allows users to "listen" through Broker applications for information being broadcast from other Brokers.

This symmetrical architecture enables users to query a community of databases with a single broadcast request, and it enables data sources to notify a community of users with a single broadcast publication. Whenever possible the BroadQuest dataweb takes advantage of IP multicasting, which facilitates broadcasting on a network. Otherwise, special software routers accomplish the same network-broadcasting functions.

How Orders are Processed

The user can Search, Publish, Subscribe, and Post through an interactive session with the Broker. Each of these user "actions" creates an internal “order." An order is a special list of pairs, the first part being an "Attribute" and the second being a specific "Value." For example, the Attribute may be "Country" and the Value "Spain."

A "request" indicates a user's desire for data. An "offer" indicates a user making data available for others. "Active" means the user proactively requests (Searches) or offers (Publishes) data. Active orders generally involve a Broker broadcasting out into the dataweb.

"Standing" means the user passively waits for data to be made available (Subscribes) or makes data available (Posts) that can only be accessed through an active request generated by another Broker. Standing orders generally reside inside a Broker and "listen for" active orders that match it.

Actions	**Request**	**Offer**
Active	*Search*	*Publish*
Standing	*Subscribe*	*Post*

Orders	**Request**	**Offer**
Active	*Search Order*	*Publication Order*
Standing	*Subscription Order*	*Posting Order*

Search Orders

A user's search action is sent to the Console Broker, which uses information from the datamap to map the search action into a search order. The Console Broker checks the user’s credentials against the datamap or against an externally available authenticator. The Console Broker then analyzes the search order and compares it against the databases on the dataweb channels that the user is qualified to use. If dataweb channels are discovered that respond to the criteria of the search order, the Console Broker sends the search order though the dataweb to all the relevant receiving Data Brokers.

Each Data Broker authenticates the search order, verifying that it was received from a Broker that is authenticated for that particular dataweb. The order is compared against an internal database to identify any relevant local channels. If relevant channels are found, the order is sent through the dataweb to all the locally connected Wrappers that are members of the relevant channels. (A single Broker can accommodate more than one wrapper to access more than one database.)

Each Wrapper consults a special “abstracted” database and attempts to produce a response to the order without consulting the main database. If the attempt is successful, a response is returned back to the Data Broker through the dataweb along with the original order. If there is not have sufficient information to answer the order, the Wrapper uses information from the datamap together with information supplied by the Administrator when the data source was wrapped, to translate the order into a valid database request. The request is then encapsulated into the appropriate form for that particular type of database and the Wrapper requests access to the database. If the DBMS responds, the Wrapper creates a list of candidate matching responses.

The Data Broker finds “true” matches by comparing the responses against the original order. This step is needed since the database that performed the original matching is out of the control of the BroadQuest Dataweb. The final matches are consolidated with any other matches that may have come from other DB Wrappers and sent back to the original Console Broker.

The Console Broker processes the responses in a manner similar to that of the Data Broker, finally routing the filtered, consolidated, and summarized responses to the User Mapper, which translates and encapsulates the responses into an internal form. The User Mapper submits the responses to the User Driver, which is responsible for the final presentation of the results to the user through the BroadQuest Console.

Posting Orders
By definition, whenever a record is added to a BroadQuest wrapped database (a database that has been connected to a Broker via the database's own customized Wrapper), a posting is created. Users can create postings as well.

When a user posts, the posting order initially follows the same path as that followed by a search order: A user's post action is sent to the Console Broker, which uses information from the datamap to map the post action into a posting order.

At this point the paths of searches and postings diverge. The Console Broker updates its internal storage structure with the newly added posting order and then checks to see if the posting order is relevant to any dataweb channel of which the Console Broker is not currently a member. If the posting order is relevant, the Console Broker will become a member and start listening on that channel.

Publication Orders
The publication order follows exactly the same path as the query order from the Console to the Wrapper. The only difference is that the publication order can match only subscription orders (in the same way that search orders match only posting orders). Consequently, the publication order may be sent to different Wrappers.

The Wrapper submits the publication order to the DBMS as if it were a query order and returns the candidate matches to the Data Broker. Note that these matches are subscription orders. The candidate matches are further filtered by the Data Broker, at which point the path of the publication order diverges from that of a query order. Instead of consolidating the matches and sending the summary back to the original Console Broker, the publication is saved in the Data Broker's publication database.

Later the user can request the Console Broker to download all matches to a specific subscription (from all Data Brokers), to clear all subscription matches, or to download a publication that has been matched by a subscription.

Subscription Orders
Subscription processing is very similar to posting processing. Subscriptions are usually created through an interactive session. Such a session results in a subscription record being added in a BroadQuest wrapped database. The subscription is then translated by the Wrapper into BQ Format and then submitted to the Data Broker. The Data Broker then starts listening as appropriate to new channels. Subsequently, publication orders may match subscription orders as described in the *Publication Orders* section above.

IX. HOW DATA IS CONSOLIDATED, FILTERED, AND MAPPED

Consolidation
BroadQuest Brokers have a customizable subsystem called the Consolidating Engine, which combines and consolidates incoming datasets to protect the user from information overload. The Consolidating Engine plays a dual role. First, it protects the dataweb from a flood of results by limiting the number of responses allowed for any order. Second, and more importantly, it acts as an intelligent agent, tabulating summary information at the right level of abstraction for any given answer set.

For example, if an order for company information resulted in 100 company records, the Consolidating Engine could return summary information for the Address attribute of those records. The summary information could be abstracted and displayed at the city level, allowing the user to easily browse those few cities with companies that interest the user, rather than displaying all 100 individual address records. If the query result were 100,000 records, the abstraction for the Address attribute could be as broad as “US," “Europe," and “The Pacific Rim." The summary information includes precisely those areas that appear in the results.

How does the Consolidating Engine of the Data Broker achieve this result and at what cost? Typically the order is submitted by the Broker to all relevant databases, and the Consolidating Engine deals with consolidating a single, and potentially long, result. It uses scores generated by the "Wrapper" of the Broker to keep a “best-hit” list, and, as it filters the results, it maintains the set of values for all attributes appearing in the result recorded. As the value set grows, the Consolidating Engine keeps moving into higher abstractions, keeping a cap on the total size needed for the result. In many cases the Consolidating Engine may be unable to find a good abstraction for a particular attribute; for example, the titles of the articles for companies may not be “abstractable." In that case the Consolidating Engine only maintains the information that “title” is an attribute of the results.

The Consolidating Engine of the Console Broker has a similar role. However, it typically receives many short answers, each having been through a prior, separate consolidation step. Because of these separate consolidations, the Console Broker may receive results that vary significantly in their abstraction level. In these cases the Consolidating Engine keeps the higher abstraction level, simply incrementing the counts of any lower abstraction level. The overhead of the Consolidating Engine’s processing is minimized by filtering.

Filtering
Filtering protects the database from irrelevant orders. A database has content that is more tightly defined than the channels that would access it. Furthermore, a Broker may need to listen to multiple channels, even with relatively small data sets to offer. Processing all orders would require a DBMS capable of handling millions of orders per day. Filters enable a Broker to specify a much more precise definition of the data, allowing the Broker to submit only those orders that pass through the filter.

By caching the results of the Consolidating Engine, the Data Broker can maintain a high level of abstraction of the actual data residing in the database. According to the above example, these would include Date Ranges, Address Abstractions, and Company Classifications. For example, if the Address abstraction is "US," any request that contains the Address attribute to a foreign address would be dropped. Most orders that result in many records (and thus appear to be general purpose) are cached. Then the Data Broker can maintain a fairly accurate view of the scope of the data and in some cases it may be able to produce results without bothering the database.

Attribute Mapping
Information owners and publishers store data in a great variety of ways. These systems vary in the technology being used to store and access the data, ranging from plain file systems with indexes and text retrieval databases, to full fledged DBMSs. They differ in the way they break and archive the data (i.e., using relational tables, inverted indexes, or hierarchical structures), and finally they differ in the way they represent the actual data properties and values—the attribute Company Name may be described as “name," or “company," or “Company Name," or even “comp_name." Furthermore, different databases use different attribute values to mean the same thing—for example, the company name field for IBM may be found as “I.B.M.," as “International Business Machines," or simply “IBM."

Exposing the user to this heterogeneity would prevent BroadQuest from reaching its goal of connecting information seekers directly to information providers. Not only must the dataweb send a user's request to every relevant database in the community, it must bridge the language barrier between the user request and the database. BroadQuest’s solution to this problem is threefold.

First is the Datamap, which makes it easy to describe orders in BroadQuest terms. Second is the Wrapper, which deals with the issues of data heterogeneity and software heterogeneity, abstracting the differences of the various access methods available for different databases. Third is the Mapper Script, which maps the order onto the database and, when results are returned, maps the database results back into BQ Format.

To understand the function of the Wrapper, recall the earlier example. Suppose a publisher has both an archive of Usenet newsgroups containing company information and an archive of press releases for various companies. Suppose that each archive is indexed through some tables in an ODBC database. First the publisher needs an ODBC Wrapper that enables access to ODBC databases. Wrappers for several popular information systems and databases are available from BroadQuest, and an API is also available that enables custom-built Wrappers for currently unsupported databases. Next the publisher needs to specify an interface between the contents of the database and the BroadQuest system. This interface provides:

- A specification of the conditions that must be met before an order will be submitted

- A description of which attributes in the table are to be included in result records and how the attribute names are to be mapped into the BroadQuest terminology
- A URL to associate with each result record.

Using the Mapper Editor, the publisher specifies all of this information by writing Mapper Script that creates a Mapper Configuration File.

The Wrapper translates the request into an SQL query on the database tables, then applies the SQL queries to the ODBC database and streams back the results. The results are then mapped into BQ Format and streamed to the Broker.

BroadQuest: Glossary of Terms

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
Action	User Process	A User-initiated or recurring Request for or Offer of data. Actions are categorized as "active" (Search, Publish) and "standing"(Subscribe, Post). Requests for data include the Search and Subscribe functions, and Offers of data include the Publish and Post actions. Each Action results in an Order that is sent out to relevant Brokers.	Search, Post, Publish, Subscribe, Order, Request, Offer	Permission
Attributes	Data Characteristic	A list of characteristics that are generated once a User chooses a Category. They are always associated with Values, and a list of such pairs form an Order. A Category is chosen as a kind of "super" attribute that begins the process of creating an Order. (Category/Cars, Make/Ford, Color/Red) "Category," "Make," and "Color" are Attributes. "Cars," "Ford," and "Red" are Values. The four classifications of Attributes (as controlled by the Dataweb Administrator through BroadQuest Director) are Input, Result, Summary, and Detail.	Order, Category, Values, Datamap, Input, Result, Summary, Detail	
Authenticating Engine	Broker Subsystem	A BroadQuest Broker subsystem that implements multi-level security for users, data sources, and network access. Allows the Dataweb Administrator to implement data and user security levels to Users and User Groups.	BroadQuest Broker	Authenti-cator
BroadQuest	Company Name	The Dataweb Company		
BroadQuest Broker	Software Component	The Dataweb software component that processes Orders between users, applications, and databases. BroadQuest Brokers act as “intelligent” multi-user, multi-channel, and multi-database communicators that access, monitor, update, and	Order, Console Broker, Channel, Authenticating Engine, Channel Router, Consolidating Engine,	

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
		deliver information within the Dataweb. Brokers provide access to Data Sources. They communicate with each other via Channels. Each Broker contains six subsystems: Authenticating Engine, Channel Router, Consolidating Engine, Indexing Engine, Matching Engine, and Wrapper. Every Broker in a Dataweb also contains identical Datamaps.	Indexing Engine, Matching Engine, Wrapper, Datamap	
BroadQuest Console	User Interface	Browser-compatible application software -- completely data-driven by its Broker software on the server – that intelligently displays a real-time, unified view of disparate data sources. Users can Search, Post, Subscribe, and Publish from the Console GUI. Each BroadQuest Console can customize views of customer and product data according to the user's function, role, task, and personal preferences. The main interface allows the User to choose from among several Categories (set by the Dataweb Administrator in BroadQuest Director). Within the chosen Category are many sub-categories from which to choose along with the number of Matches for each. The User chooses what Action to take (Search, Post, Subscribe, or Publish—the Dataweb Administrator determines if one or all of these Actions are available).	Action, User Group, User, Search, Post, Subscribe, Publish, Dataweb Administrator, BroadQuest Director	
BroadQuest Director	User Interface	**T**he primary control console for the Dataweb Administrator. Through Director, the Administrator can create and extend a variety of customized datawebs, dataweb components, and data sources. Director can create, modify, and delete user and group privileges.	Dataweb, Channels, BroadQuest Broker, BroadQuest Console, Dataweb Administrator	Admini-strator

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
Category	Data Characteristic	A section of the Datamap that defines a kind of "super" Attribute (i.e., “vehicles," “furniture," musical instruments”). Once the User picks a certain Category, the Dataweb knows what Attributes to assign to that Category.	Datamap, Attributes, Values	
Channels	Channel	A group of BroadQuest Brokers that are defined by their Interests (that have some data Attributes in common). The Dataweb Administrator uses BroadQuest Director to define Channels so that Users can access data sources relevant only to their Actions (like using a tuner on a radio). Every time an Order is created, its content is compared to the existing, or defined Channels, and then the Dataweb determines which Brokers should receive the Order. This filtering process dramatically reduces the amount of network traffic on the Dataweb and improves performance and scalability.	BroadQuest Broker	
Channel Definitions		Orders that are defined by the Dataweb Administrator to partition the Dataweb into Channels. Until channels are created, the dataweb remains entirely general in scope—all BroadQuest Brokers are transmitting to and receiving from all other Brokers. The easiest way to create a viable Channel within the Dataweb is to associate a particular Order with that Channel. The Order imposes constraints by defining the kind of Orders allowed on that Channel. Thus, administrator-defined channel definition orders effectively filter Orders generated from other Brokers. The GUI used by the Dataweb Administrator to define a channel is identical to the general user's GUI, because that is the natural interface for defining Orders.	Dataweb Administrator, Channels, Orders	

BroadQuest: Glossary of Terms

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
Channel Router	Broker Subsystem	A Broker subsystem that determines to which Brokers an Order is sent.	BroadQuest Broker, Order	Channelizer
Console Broker	Type of Broker	A Broker that is connected to a BroadQuest Console for processing Actions and gathering data upon request.	BroadQuest Broker, BroadQuest Console	
Consolidating Engine	Broker Subsystem	A BroadQuest Broker subsystem that filters and organizes incoming data sets by merging and prioritizing Data Results to minimize data overload for both Users and Data Sources.	BroadQuest Broker	Consoli-dator
Data Broker	Broker	A generic designation for all Brokers (database, application, gateway) apart from the Console Broker.	BroadQuest Broker, Console Broker	
Datamap	Dataweb Database	The "catalog" of Attributes that hold Values. Attributes are the fields within a Datamap and Values are the specific instances. A logical archive of rules that determine how the Dataweb functions, and as such is the central feature of the Dataweb. The Datamap options are controlled and changed by the Dataweb Administrator (using BroadQuest Director), who determines the way in which information is published and retrieved, how queries are structured, and what a user does or does not see. Identical Datamaps are stored in every Broker.	Categories, Attributes, Values	Catalog
Datamap Editor	User Interface	A tool of BroadQuest Director that allows the Administrator to edit the Datamap.	BroadQuest Director , Datamap	
Data Mart	Data Archive System	A type of data warehouse designed to meet the needs of a specific group of users such as a single department or part of an organization. Typically a data mart focuses on a single subject	Data Warehouse	Data Mart

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
		area such as sales data. Data marts may or may not be designed to fit into a broader enterprise data warehouse design.		
Data Results	Type of Data	The matching responses (Matches) received by the User from an Order being sent out on the Dataweb.	User, Order, Dataweb	
Data Sources		A database, application, or User connected to the Dataweb via a BroadQuest Broker. Since Users can Post and Publish data, every User is a potential Data Source.	User, BroadQuest Broker	Data Source
Data Warehouse	Data Archive System	A subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process. A repository of consistent historical data can that can be easily accessed and manipulated for decision support.	Data Mart	
Dataweb	Virtual Data Architecture	An architecture created by BroadQuest that allows corporations to connect Users, applications and datasources—regardless of disparate applications or data types—in order to overcome data barriers, and to access, subscribe and share data across the enterprise. An intranet- or Internet-based “many-to-many broadcasting” technology that uses IP multicasting, when available; otherwise, software routers are used that function like IP multicasting. The Dataweb “broadcasts” in a kind of circuit that allows one Broker send an Order to every other Broker in the Dataweb. The Dataweb Administrator uses BroadQuest Director to develop and administer the Dataweb. The term "Dataweb" also applies to each instance of a Dataweb.	Channels, BroadQuest Broker	Exchange, Internet Exchange, Trading Well
Dataweb Administrator		The person who develops and administers the Dataweb through BroadQuest Director.	Dataweb, BroadQuest Director	

BroadQuest: Glossary of Terms

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
Detail	Attribute Classification	Indicates what Attributes are to be displayed in the detail table to the User.	Category, Attribute, Input, Result, Summary.	
Indexing Engine	Broker Subsystem	A BroadQuest Broker subsystem that maintains a dataset abstraction to protect from access overload and to resolve most Orders directly.	BroadQuest Broker	Indexer
Input	Attribute Classification	Indicates what Attributes are to be displayed to the User.	Category, Attribute, Result, Summary, Detail.	Input Attribute
IP Multicasting		Unlike unicasting, the Internet's standard means of communication that sends packets of information to one site at a time, IP multicasting sends packets to multiple sites simultaneously. Through the Dataweb, BroadQuest Brokers take full advantage of this protocol as it becomes more widely available on the Internet.	Dataweb, BroadQuest Brokers	
Interests		That part of a BroadQuest Broker that defines the nature of the data it can access. Interests are specific lists of Attributes (with perhaps some related Values). Interests are used by the Dataweb Administrator to create Channels. For example, the Dataweb Administrator can establish which Brokers can access data regarding "Chevy" vehicles as one Channel.		
Mapper Editor	Use Interface	A tool of BroadQuest Director that allows the Dataweb Administrator to create and edit Mapper Scripts.	BroadQuest Director , Datamap, Mapper Script	
Mapper Script		A configuration file for the Wrapper that tells the Wrapper how to match the characteristics of the Dataweb to the individual	Wrapper, Data Source	

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
		fields in each Data Source. The Mapper Script supplies control information and data Mapping that determine how a Wrapper attaches a Data Source to a Dataweb.		
Mapping	Software Process	The process in the Wrapper of matching the characteristics of the individual fields in each Data Source to the Dataweb.	Data Source, Wrapper	
Matching Engine	Broker Subsystem	A Broker subsystem that verifies incoming data conforms to the Order.	BroadQuest Broker	Matcher
Matches	Relevant Datasets	The Data Results of a Dataweb search of all Data Sources. Full and partial matches are displayed on a summary or consolidation screen. The total number of matches found is displayed in brackets [##] after each Attribute.		
Offer		A sub-category of User Actions when the User wants to offer data to the Dataweb. Both "Publish" and "Post," are Offers, as opposed to "Search" and "Subscribe," which are categorized as "Requests."	Actions	
Order	Type of Data	A special list of pairs, the first part being an "Attribute" and the second being a specific "Value." (Category/Cars, Make/Ford, Color:/Red) "Category," "Make," and "Color" are Attributes. "Cars," "Ford," and "Red" are Values. Orders are internally generated from each of the four User Actions.	Category, Attributes, Values, Action	
Post	Type of Action	A stored user Action that places specific data within the Dataweb allowing it to be located by other users and applications. A Post generates a Posting Order.	Action, Posting Order	Post

BroadQuest: Glossary of Terms

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
Posting Order	Software Term	The Order that results when a User Posts.	Post, Order	
Publication Order	Software Term	The Order that results when a User Publishes.	Publish, Order	
Publish	Type Action	A User Action that distributes specific data to a named group of interested users. Publish generates a Publication Order.	Action, Publication Order	
Request		A sub-category of User Actions when the User wants data from the Dataweb. Both "Search" and "Subscribe" are Requests, as opposed to "Publish" and "Post," which are categorized as "Offers."	Actions	
Result	Attribute Classification	Tells the Dataweb what Attributes are to be displayed in the results table to the User.	Category, Attribute, Input, Summary, Detail.	Result Attribute
Search	Type of Action	A real-time or stored user Action to locate data with specific characteristics. Search generates a Search Order.	Action, Search Order	
Search Order	Software Term	The Order that results when a User Searches.	Search, Order	
Subscribe	Type of Action	A stored action that notifies interested users or applications when data of interest arrives in the Dataweb. Subscribe generates a Subscription Order.	Action, Subscription	
Subscription Order	Software Term	The Order that results when a User Subscribes.	Subscribe, Order	
Summary		Tells the Dataweb what Attributes are to be displayed in the summary/consolidation table to the User.	Category, Attribute, Input, Result, Detail.	

CURRENT NAME	ITEM TYPE	DEFINITION	RELATED TERMS	OLD NAME
User		Users are people who use the Dataweb via the BroadQuest Console to interact with meaningful data.		
User Group		A method of defining a group, or class, of Users that makes administrating the use of the Dataweb easier.		
Values	Data Characteristic	The specific description of an Attribute. The User chooses from a list of Values to specify the Attributes for an Order. (Category: "Cars"/ Make: "Ford"/ Color: "Red") "Category," "Make," and "Color" are Attributes. "Cars," "Ford," and "Red" are Values.	Category, Attribute, Order, Datamap	
Wrapper	Broker Subsystem	A subsystem of the BroadQuest Broker that has specific intelligence about a single Data Source or class of data sources. The Wrapper uses a Mapping Script to allow Orders to be optimized between dissimilar data sources. Different Wrappers are needed to connect a BroadQuest Broker to a BroadQuest Console, to a database, to an application, and to other Datawebs. A single BroadQuest Broker can use multiple Wrappers.	BroadQuest Broker, Mapping Script, Dataweb	Wrapper