
MDL Information Systems Inc.
San Leandro, California

# The MDL® Discovery Framework: Data and Application Integration in the Life Sciences

© 2002 MDL Information Systems, Inc. All rights reserved. No part of this document may be reproduced by any means except as permitted in writing by MDL Information Systems, Inc.

The information in this document is subject to change and should not be construed as a commitment on the part of MDL to provide a system that conforms to this description. Any commands or menus shown in these pages are illustrative examples and may be modified in the final release of the product.

## *Executive Summary*

The MDL Discovery Framework is a new information system supporting discovery research in the pharmaceutical, agrochemical, and chemical industries. MDL's goal is to achieve a superior discovery informatics environment by providing an open and integrated set of applications and technologies that integrate the discovery process and interoperate cleanly with other technologies.

By focusing on the domain-specific needs of discovery chemistry and biology, MDL Discovery Framework is the ideal system for integrating discovery informatics. Organizations that implement this integrated informatics environment will help their scientists make better decisions more quickly, enabling the organization as a whole to bring good candidate compounds to market at a lower cost.

The MDL Discovery Framework is the foundation of all of MDL's new products. MDL Discovery Knowledge provides scientific content through in-house databases and the DiscoveryGate[SM] Internet site. MDL Discovery Experiment Management provides applications for managing chemistry, biology, and logistics workflows and data. MDL Discovery Predictive Science helps scientists assess drug properties of compounds for which complete experimental data are not available.

From a technology perspective, MDL Discovery Framework features a shift to a three-tier architecture, comprising

- A strong set of new applications and user interface tools.
- An integrating middle tier of business processing objects that power the applications and provide a standard interface for custom development.
- A powerful set of database services built on modern Oracle technology.

The MDL Discovery Framework benefits research organizations in a number of ways:

- New applications can be delivered with improved quality and timeliness.
- The developer interface follows a simple, object-oriented model that is more powerful, consistent, and suited to custom application development than current models.
- The flexible architecture provides a single interface that supports a broad range of thick- and thin-client applications.
- The technology is scalable to the high volumes of data and usage required in today's discovery workflow.
- The three-tier architecture is easier to deploy because most system configuration activities take place on the middle tier.
- By building everything on top of a common object layer, standard and custom applications interoperate at a much more fundamental level, achieving a single system that meets the organization's unique needs.

- Users benefit from the tighter integration and increased functionality of searching over multiple database systems, sharing objects between applications, and more consistent user interfaces.

MDL has an unparalleled set of products and services that support discovery research. MDL's core competency is the management of chemical structures and related data for biological activity, experimental methods, and physical properties. The MDL Discovery Framework provides a new platform for managing data in an integrated way.

The MDL Discovery Framework leverages the investment that companies have made in previous MDL products. As a result, companies can focus on building new applications on the MDL Discovery Framework while keeping existing, satisfactory applications running. Scientists do not need to wait for everything to be shifted to a new architecture before anything becomes available. Instead, new applications can be deployed as they are ready.

MDL's overall goal in developing the MDL Discovery Framework is *integration* of the discovery process by integrating its data and its applications.

## *The Importance of Data Integration*

The continuing expansion of high-throughput research techniques has led to an even larger expansion in the amount of data generated. This data represents a tremendously valuable resource, acquired at great cost, and its management merits considerable attention.

The importance of handling the unique data types encountered in life-science research, such as molecules (2D and 3D), reactions, DNA and protein sequences, and images, was recognized as early as the late 1970s. Commercial data management systems first appeared then, and efforts to combine these data types with text and numeric data began shortly thereafter. Early systems tended to capture and use data in monolithic, closed environments.

Today, data is generally captured, but organizations can greatly increase the value of that data by making it more accessible and by providing more flexible ways of combining data from genomic, proteomic, and small-molecule sources. By building on open standards that simplify the retrieval and use of data by any number of applications, an organization can enable its scientists to experiment with different ways of examining the data, looking for patterns, relationships, and trends. Through easy, consistent access to data, scientists can quickly work with new approaches as these appear. The overall goal is to support data- and knowledge-driven decisions to efficiently develop successful products.

## *The Importance of Application Integration*

In early monolithic systems, "integration" focused exclusively on sharing data. But an information management system is built up of more than data alone; it also incorporates the "business rules" that a user community agrees to operate by, in order to foster a rapid understanding of the context and meaning of the data itself. If each application that

performs a given task does so in exactly the same manner, the user of those applications can rely on that predictable behavior and focus on the scientific task, rather than having to mentally adjust to the differences in each application.

Historically, however, independent applications implemented key functionality multiple times, often introducing differences. The users then faced longer learning periods and increased chances of error. Further, the application development groups faced the burden of maintaining and enforcing consistent behavior in different implementations of key functionality, and risked introducing errors in both the applications and in the data.

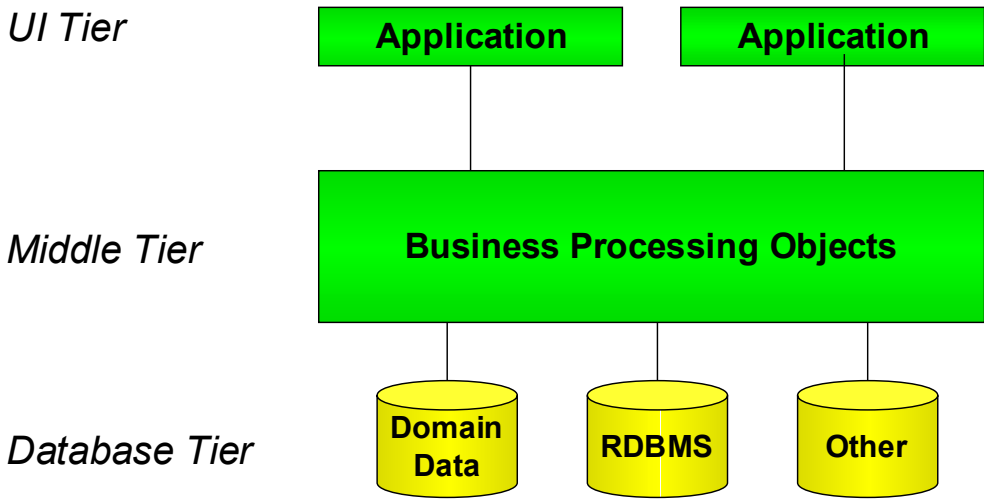
This need to establish standard ways to perform standard functionality has driven the movement to a three-tier application development architecture. In such an architecture, business rules are built as standard services, maintained in the middle tier, and made available to all applications that need those services. Examples of such needed services in the discovery environment include:

- User authentication and log in
- Data access management
- Structure normalization
- Structure uniqueness checking
- Compliance with required supporting data
- Structure registration
- Calculation of physical properties
- Unique identification number assignment
- Assay definition
- Protocol registration
- Assay results collection
- Literature cross linking
- List management

By implementing standard services once on a central server, it is possible to build a variety of applications with separate workflows and appearances, but with a consistent delivery of key functionality.

## *Multi-tier Architectures*

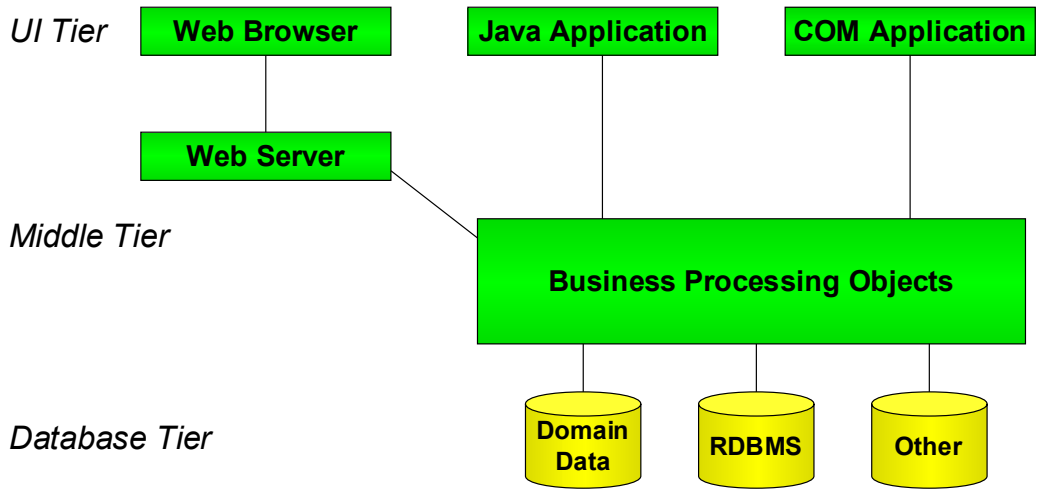
Modern multi-tier systems provide a basis for achieving these goals in both in-house systems and Internet-based services. As shown in the illustration below, a multi-tier architecture distributes functionality across a database tier, a middle tier and a UI (User Interface) tier:




The individual tiers are:

- *User interface tier*. This tier provides a front-end for the scientist using the system. Depending on the specific application's needs, this can range from a thin web client to a thick conventional application.
- *Middle tier*. This tier provides a set of standard services that can be shared by multiple applications. Applications then exhibit a common, predictable behavior, making them easier to learn. Applications can also easily share data and objects, and so are more tightly integrated. This tier is usually located on a separate, dedicated server
- *Database tier*. Typically located on one or more server computers, this tier provides the searching and registration engines for domain-specific and general-purpose database access.

Applications are written using standard technologies such as COM for Windows clients, Java for applets and cross-platform applications, and a web browser for truly thin clients. A web application is deployed on a web server, using Java servlets for user interface generation and other services specific to the thin web environment. The web server straddles the boundary between the UI and middle tiers, but relies on the common interface of the business processing objects for all of its standard services:




## *MDL Discovery Framework*

MDL Discovery Framework provides a set of tools and services that support an integrated discovery informatics environment based on this modern three-tier architecture. These are the design principles of the MDL Discovery Framework:

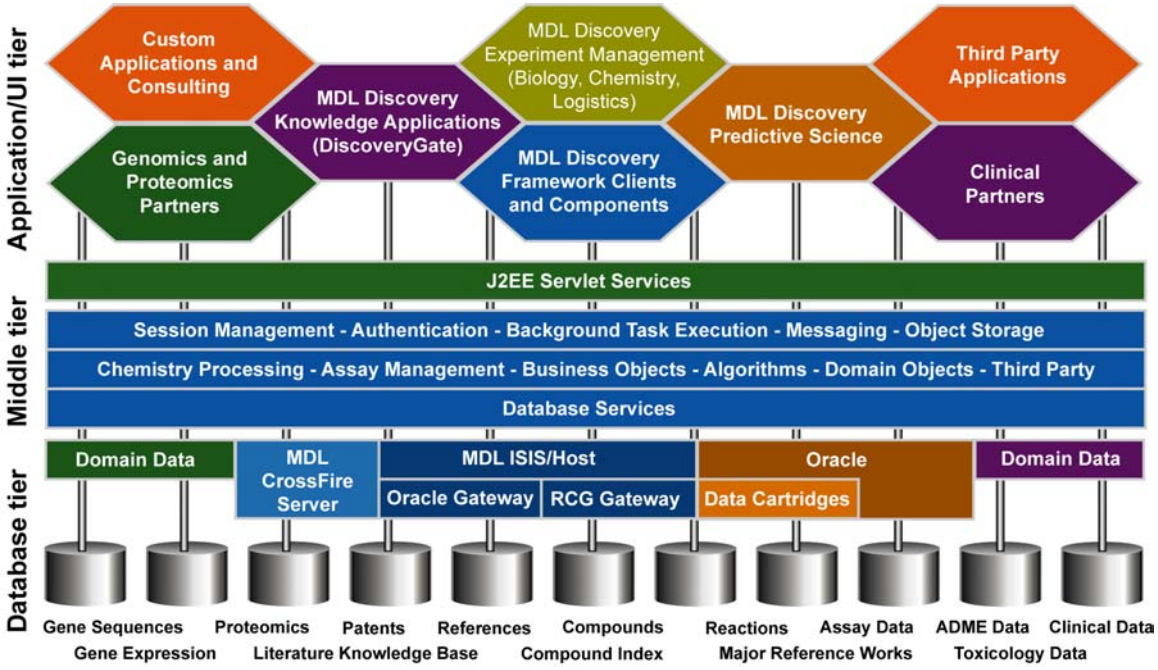
- Open – able to use components from other vendors, so that organizations can assemble research environments based on those components that best meet their needs.
- Standards-based – use established and de facto standards as much as possible.
- Flexible – usable from a wide variety of standard technologies.
- Scalable – able to handle the high volumes of data access that are required in a modern, distributed, high-throughput environment.
- Extensible – developers can extend the system with services they have written themselves or acquired from other parties.
- Transparent – end users see only the applications based on MDL Discovery Framework, not the technology behind it.

The goal is to make it easier – for MDL, for MDL's customers, and for other software providers – to develop robust applications faster. In addition, end users should find applications built with the help of MDL Discovery Framework to be easier to learn and use, thanks to consistent behavior as a result of the shared use of key services. The result is a tight integration of data and applications.

The MDL Discovery Framework integrates with the multi-tier architectures already in place in industry, rather than building its own proprietary system architecture. MDL Discovery Framework is not a replacement for such horizontal application servers such as Oracle9i Application Server, IBM's WebSphere, or BEA Systems' WebLogic. Instead, it works alongside such horizontal products, providing scientific services of high value to the discovery community. MDL Discovery Framework enriches and complements

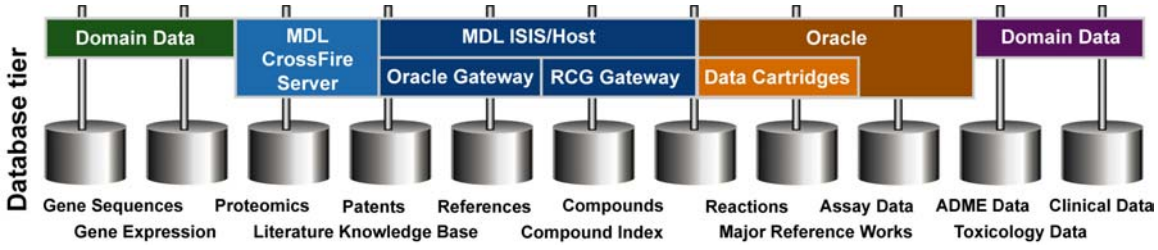
horizontal products by enabling domain specific information sources to participate fully within the overall corporate framework.

In the following picture, MDL's four main solution areas are shown in the top center, alongside applications from partners and third parties who are integrating with the MDL platform. The central area shows a middle-tier application server hosting a series of standard technologies as well as domain-specific scientific services. At the base of the picture, MDL's chemical database engines are shown together with other database systems of interest.




We will examine MDL's vision for delivering each of these tiers as part of the MDL Discovery Framework, starting from the database foundation.

## *The Database Tier*




At the base of the framework lie the database servers, which store and search the information required for the discovery workflow.

Most of the discovery process's data has been represented as traditional numeric and text data, which can be nominally stored in a normal relational database such as Oracle. MDL uses industry-standard relational databases for text and data storage. These MDL

products know how to work with this data and the relational data models used to organize it, and they must provide the context and interpretation required to make the data useful.

## MDL Relational Chemistry Server

Chemical data presents a complex challenge. Molecules and reactions are complex objects, and the operators for searching these objects require special logic not present in the traditional operators and indexing techniques provided by relational database vendors. MDL's expertise in cheminformatics, however, has enabled it to take the lead in developing solid functionality for handling chemical information in a modern, integrated database environment.

The best solution is to use object extensions that are starting to appear in modern database systems such as Oracle. MDL worked closely with Oracle on these extensibility features throughout the development of Oracle8*i*, and the result is a *data cartridge* technology that allows MDL to plug its chemical search engines directly into the Oracle server. The MDL cartridges provide definitions for new indexed types that can be called directly from SQL commands within any Oracle application. When a search or registration operation involves a chemical field or search operator, the Oracle server launches an external process with the MDL chemistry search engine and integrates the results of that search clause with the rest of the SQL operation.

The result is a fully transactional, integrated operation that works as part of the Oracle server, yet provides all of the power of MDL's chemistry database engine. The chemical types provide all of the richness and performance a scientist would expect of an MDL product, but the data storage and the developer's interface are within Oracle, not a proprietary layer.

The MDL Relational Chemistry Server delivers its two chemical data cartridges for storage, registration, and searching of chemical data in an Oracle database:

- A **molecule cartridge** provides the following search types: 2D and 3D substructure, similarity, molecular weight, formula, flexmatch and 3D conformationally flexible search. The flexmatch operator provides a powerful and highly tunable query tool that includes literal exact match, tautomer, salt, parent molecule, stereoisomer, and other common types of structural matching. The data storage for the molecule cartridge is the same Relational Chemical Gateway (RCG) format used by ISIS/Host 4.0. By keeping a common database format between the molecule cartridge and ISIS/Host, MDL has provided a smooth transition path for its customers, many of whom have large databases and applications based on the ISIS formats.
- A **reaction cartridge** provides a reaction type with the following search operators: reaction substructure search (RSS), reaction flexmatch search, and reaction similarity. If no other searching is desired, the reaction type can be stored as a very simple data type. This simplicity makes possible a variety of registration applications for combinatorial chemistry and other fields.

The two cartridges can be used together to provide a truly relational system for managing chemical data. Because the two tables are cross-referenced, a scientist can search for a

reaction using molecule information, for example "what reactions do I have that produce a product similar to this query?" By keeping the reaction and molecule types separate, the MDL approach creates simpler types that are more compliant with the relational model than other solutions based on a single large cartridge.

## Other Chemical Database Servers

MDL ISIS/Host remains broadly deployed in the industry, powering an entire generation of applications built on top of interfaces such as ISIS/Object Library, ISIS/Base PL, Chemscape, and the ISIS/Host API. MDL continues to enhance ISIS/Host and will continue to support the product during an extended transition period while its customers are moving to the new MDL Discovery Framework platform.

The Beilstein database is the world's largest reaction database, and it has earned an unsurpassed reputation for the quality and completeness of its information. This database is delivered through a proprietary search engine, CrossFire, which has been highly optimized for the size and special needs of this database. It is also widely used in academic environments where Oracle is not widely available. For this reason, MDL plans to continue to support the proprietary CrossFire database format for the foreseeable future.

In the longer term, MDL is coordinating its reaction cartridge research so that it can integrate the wealth of data in both the MDL and Beilstein databases. The long-term goal is a single relational-based searching system that builds upon the best aspects of both current products.

## Biology Data Sources

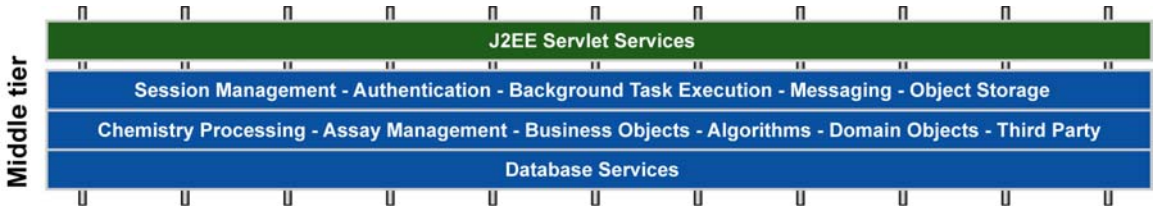
MDL Assay Explorer and other industry products are typically used to build databases of biological screening results. The data are stored in conventional Oracle databases in a variety of complex data models. Assay Explorer is unique in its ability to provide a system that can flexibly adapt to the data models present in an organization.

Descriptive information, sometimes called *metadata,* is the key to managing biology data. Although the basic data are searchable through standard Oracle tools without any need for a data cartridge, the data models are complex enough to require interpretation. Each protocol may have different data associated, and this must be organized to present the data in a meaningful context. MDL's middle tier technology provides an ideal way to expose the descriptive data that organize a biological data management system.

## Other Data Sources

The open MDL Core Interface middle-tier technology, described later in this document, provides a way to integrate multiple data sources from MDL and other vendors. Genomics, proteomics, analytical chemistry, and other disciplines are creating data that is critical to the discovery process. MDL continues to work with vendors of these databases to provide methods to integrate their data into the overall system.

## *The Middle Tier*




The middle tier is the heart of a modern informatics system. Business processes have become so complex that they can only be modeled through a sophisticated set of business rules. These rules control how structures and data are registered, how they are searched, and how the information is managed at higher levels. In a middle tier, rules can be implemented through transactional objects that provide a standard interface that models all the functionality of the underlying database systems.

MDL's middle tier technology provides the ideal place for exposing a variety of services used by applications. MDL has developed a variety of technologies for managing chemistry, biology, inventory, and other systems. By grouping these technologies together into a single layer, MDL's middle tier provides a complete, consistent set of tools for applications to be built upon. Integration with other middle-tier technologies can be provided in a uniform, focused way.

The table on the next page summarizes the primary middle-tier interfaces. *MDL Core Interface* provides the basic framework for security, database access, object storage, and other fundamental objects. MDL's *domain interfaces* provide additional scientific services that model and power the discovery domain. This list will grow in future years as MDL develops new applications and technologies in additional areas.

### MDL Core Interface

The MDL Core Interface provides the basic foundation technology for the MDL middle tier. With this, an application can connect to the system, search one or more databases, and manage basic tasks in the system. MDL Core Interface provides a common application programming interface (API) for the MDL Discovery Framework's central services.

MDL Core Interface provides an open API to developers wishing to access MDL's rich chemical and biological information technologies from a middle-tier application server. The core interfaces are based on Java, but can be called from many other environments. COM-based connectors are provided to enable development in Microsoft environments.

Cheminformatics systems are unique in their requirement to manage both relational and non-relational data. Most standard object-relational translation layers handle all database interactions through conventional SQL queries. For chemical data, this must be extended to the SQL 3 syntax required for data cartridge access, and in many cases the layer must provide access to non-relational data as well. For the near future, CrossFire and other important data sources will remain external to the relational systems, and any proper cheminformatics middle tier must provide access to these systems as well.

## MDL Core Interface

- Session management service – security, single sign-on, license management
- Data source service – internal/external database searches (integrated access to data), list management, registration, import/export data
- Object Storage Service – object repository
- Message Center Service – communicates results and status to and from other parts of the system
- Task Management Service – launching of asynchronous tasks
- Discovery Framework Console – web-based administration client

## MDL Domain Interfaces

### Chemical warehouse services

- Normalization (concordance)
- Tools for advanced searching of structures & data
  - Smart searching of properties with structures
  - Data pivoting
  - Drill-down
- Data mart services

### Calculation services

- Chemical structure manipulation (MDL Chemistry Rules Interface)
- Cheminformatics business rules manager (CBRM)
- Property calculators and predictors

### Registration services

- Compound & substance (identified vs. abstract) ID generation
- Novelty check and business rules
- Sample, lot & batch management
- Salt forms management

### Combinatorial chemistry services

- Reaction-based enumeration
- Product-based tools: enumeration, clipping, Rgroup decomposition

### Reaction management

- Searching
- Registration
- Keyword generation
- Advanced reaction tools: synthesis planning, retro-synthetic analysis

### Experiment management

- Assay management services
- Plate management services
- Robotics interfaces

### Procurement services

- Shopping cart
- Procurement engine links in-house inventory to external suppliers

### Literature services

- Literature linking
- Thesaurus tools
- Document parsing for scanning patents and other literature

***Table 1: Core and domain services available or under development***

MDL Core Interface focuses on standard query and database services:

- A *Session Management Service* provides single sign-on services for the system. Users connecting to the system provide a single password to a standard authentication service. If their authentication succeeds, they are given access to other services as appropriate and authorized.
- A *Data Source Service* provides unified access to searching of chemical structures and alphanumeric data. Chemical structures can be searched in any MDL data source, including MDL Relational Chemistry Server and CrossFire. List management provides a key extension to the relational model, allowing users to save hit lists and combine them using the logical operations union, intersection, and subtraction, without having to repeat the original searches. The data source interface is open: developers wishing to access an unsupported format can write a custom data source.
- An *Object Storage Service* allows storage and retrieval of public, group, and private objects such as hit lists and procurement shopping carts for reuse in later sessions or for sharing between users.
- A *Message Center Service* provides access to errors and asynchronous messages appearing from the system.
- A *Task Management Service* provides on-demand and scheduled access to calculations and predictions performed by external tools. This manager can be used in connection with the property calculation and cheminformatics business rules described below.
- A *Discovery Framework Console* provides a web-based tool for management and tuning of the middle-tier server.

As an extension to MDL Core Interface, MDL has built a *Servlet Application Framework* (SAF) that simplifies some of the repetitive aspects of building a servlet-based web application. These tools include standard objects for generating user interfaces in web pages.

## MDL Domain Interfaces

A primary value of the middle tier comes from its ability to host a variety of new domain objects and services. MDL is building its most important business processing objects as middle-tier services. These *domain interfaces* work together with MDL Core Interface to provide a common application programming interface to MDL's discovery technologies.

*Chemical warehouse services* provide integrated access to data for data management and data warehousing:

- *Normalization tools* for building and managing a unified concordance of chemical structure information coming from a variety of sources. The structures in the concordance are *deduplicated,* meaning that identical structures coming from different sources are stored only once, with links providing access to the supporting data in the original sources. The concordance is also *normalized* so

that identical structures with different representations are correlated and stored only once.

- *Smart searching* of a chemical warehouse with property information. Searching a chemical data warehouse involves more than just structure searching. Typically, a structure search needs to be combined with searches of one or more *properties,* data stored in other tables. *Data pivoting* allows access to common data models in a manner useful to scientists.
- *Data mart* tools allow an organization to organize its discovery data to facilitate better decisions.

*Calculation services* generate properties for registration, data mining, and other purposes:

- The *MDL Chemistry Rules Interface* engine provides the ability to calculate properties and change structures based on custom cheminformatics business rules. This service is based on MDL's industry-leading Cheshire™ technology, providing a simple scripting interface for developing powerful business rules and calculations. Scripts developed for ISIS are compatible with this new service.
- A *cheminformatics business rules manager* provides a consistent Oracle-based storage for business rules and structural calculations and a simple API for developer access. By using this manager throughout the cheminformatics system, rules can be developed once and made accessible to any client or server tool that needs them.
- *Property calculators* generate calculated values for immediate use or for storage in database tables. The calculated values can include chemical names, 3D models, and toxicity/metabolic activity predictors.

*Compound registration* tools provide sophisticated business logic for creating databases of discovery data:

- *Compound and substance* objects can be independently managed to reflect business processes for chemical entities. The system provides automated generation of compound and substance IDs.
- *Novelty check* tools apply business rules for determining structure uniqueness.
- *Sample, lot, and batch* data are linked and managed through business rules that reflect corporate processes.
- *Salt forms* management provides a variety of methods for managing distinct salts as a single compound.

*Combinatorial chemistry* services are based on MDL's leading set of technologies:

- A *reaction-based enumerator* and related tools provide intuitive storage and management of parallel synthesis experiments. These tools are a new generation of the technology pioneered by MDL Afferent, the industry leader in reaction-based experiment management. By integrating these tools into the middle tier, MDL gives a powerful connection between these reaction-based tools and discrete structure storage in ISIS databases.

- *Product-based generic structure* tools provide a Markush-style representation for creating, storing, and searching virtual libraries. This product-based representation provides the most efficient storage for large libraries. Using Central Library technology, it is possible to find libraries based on substructure search operations without physically enumerating the libraries. This makes it possible to represent libraries that are so large that it would be impractical to store and search all enumerated combinations.

*Reaction management* tools provide high-level access to the reaction objects that can be stored in the new reaction data cartridge:

- *Reaction searching* objects provide simple tools for generating the SQL required by all major reaction searches. These high-level objects provide a simple interface for the full range of sophisticated reaction searches across the entire reaction data model.
- *Reaction registration tools* provide transaction-safe updates to the reactions, molecules, and related tables in the new reaction data model. Using these tools, a developer can easily implement a registration application that conforms to MDL's recommended reaction data model.
- Tools for generating *keywords* and other properties for easy *reaction classification* and non-structural searching.
- *Advanced reaction tools* provide simplified search query generation and other tools for high-level reaction applications.

*Experiment management* services allow an organization to manage high-throughput screening processes:

- *Assay management services* manage high-throughput screening data and protocols.
- *Plate management services* provide a consistent interface for storing data related to multi-well containers used in today's automated synthesis and screening systems. The container object allows management of a broad range of data connected to locations on 96-well, 384-well, 1536-well, or any other array.
- *Robotics interfaces* provide standard connections to automated synthesis and analysis equipment.

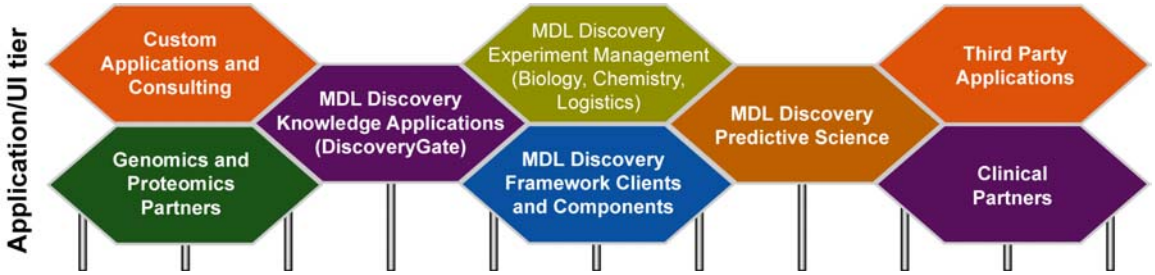
*Procurement services* model the central logistics process for organizations that select, buy, and physically manage chemicals:

- A *shopping cart* object provides a central mechanism for managing user selections from an inventory, catalog, or other collections.
- A *procurement engine* provides a standard interface for managing inventory and procurement data, including handling procurement business rules and interfacing to purchasing systems and e-commerce sites.

*Literature services* provide electronic access to the scientific literature:

- *Literature linking* uses intelligent parsing to transform a user's request into an optimized query into the native data sources.
- *Thesaurus* tools for organizing information based on user-defined metadata and standard classifications such as EMTREE and MESH.
- *Document parsing* tools allow scanning of patents and other critical public information in order to recognize, process, and automatically extract scientific content such as chemical names, chemical structures, keywords, and property data.

## *Applications and the UI Tier*




The User Interface tier enables applications to be integrated with a number of software and scientific application products. Some of these will be built by MDL, some by customers, and some from other vendors.

MDL is building a broad range of applications that form the front end for most users to the system. For cases where end users simply want to use standard applications, MDL licenses its Discovery Framework Standard Edition as a runtime platform with all required middle-tier objects and services required for delivering the application.

In other cases, customers may want to build their own applications based on the underlying developer interfaces. For these cases, MDL licenses its Discovery Framework Enterprise Edition, which includes the full MDL Core Interface and client tools for custom development and runtime use of the APIs.

### Discovery Framework Client Applications

MDL is building a new generation of general-purpose applications as part of the MDL Discovery Framework. These fill many of the same needs as current tools such as ISIS/Draw, ISIS/Base, and ISIS for Excel, but provide much greater flexibility and functionality.

Here are some of the general-purpose applications that MDL has delivered or is developing:

- *MDL Draw Enterprise Edition*, available now, is a next-generation tool for drawing and displaying chemical structures. The MDL Draw user interface is very familiar to users of ISIS/Draw, and yet it provides new tools that represent a major leap in usability and functionality. Occasional users can easily draw Rgroup queries and create new templates, features that were previously practical only for well-trained power users. MDL Draw also provides rendering

components that can be used in place of Chime Pro in a web browser or client application.

- A forms-based client for creating queries, displaying search results, and performing basic data input. Using a simple, standard interface, users can add their own boxes and create custom views of the data. A standard database browser provides a user interface customized to the needs of users of each MDL database.
- A reporting module for flexible output of data in printed or onscreen form. Reports can be created in a variety of formats that expand or adapt to the amount of data being presented.

The Discovery Framework client applications are designed to work out of the box. But for Enterprise Edition customers, they also provide a flexible set of components for application development:

- XML-based configuration files allow an application developer to deploy custom menus, tools, and user interface elements for specific applications. Unneeded tools or menus can be removed for specific applications. Dialogs can be changed to match in-house conventions and requirements. Templates can be modified for the entire organization or personalized by the end user.
- Component interfaces are provided to the industry standard technologies of Microsoft's ActiveX (for Visual Basic and other tools) and Java. Using these interfaces, a developer can integrate these components into applications or web applets.

## A Foundation for All MDL Applications

In addition to the Discovery Framework, MDL has three other primary solution areas:

- MDL Discovery Knowledge provides scientific content through in-house databases and the DiscoveryGate Internet site.
- MDL Discovery Experiment Management provides applications for managing chemistry, biology, and logistics workflows and data. MDL Chemistry Explorer and MDL Assay Explorer provide sophisticated user interfaces for managing chemical and biological data in a high-throughput discovery environment.
- MDL Discovery Predictive Science helps scientists assess drug properties of compounds for which complete experimental data are not available.

Applications in all of these solution areas use the same basic user interface components and architecture delivered by the MDL Discovery Framework client. MDL Chemistry Explorer, for example, adds advanced user interface elements to manage combinatorial chemistry and compound registration, yet it shares its basic user interface with the MDL Discovery Framework client. The two applications are so closely integrated that the user seems to be using a single system.

## *A Framework for Custom Development*

MDL Discovery Framework Enterprise Edition allows an application developer to build custom applications on top of the same tools that MDL uses to build its own applications:

- Client-side components and interfaces, available in both Microsoft's ActiveX and Sun's Java technologies.
- The Java-based interfaces exposed on the middle tier by MDL Core Interface and the domain interface products, as well as the web server extensions provided by the Servlet Application Framework.
- The Oracle SQL extensions provided by the MDL Relational Chemistry Server on the database tier.

MDL's customers build applications in a wide variety of technologies and programming languages. Some use thin clients where most or all of the application is hosted on a middle-tier web or application server. Some use thick clients that deploy more application components to the client workstation. MDL Discovery Framework supports application building in all of the technologies in general use:

- Microsoft technologies such as COM, ActiveX, and .NET are commonly used to build thick client applications in development environments such as Microsoft Visual Basic. All of MDL's middle-tier Java interfaces and client components include COM wrappers so that a Visual Basic developer can write client code in a familiar environment. The newer Microsoft .NET technologies are also supported.
- Java clients and applets can be built directly using the Java interfaces provided by the middle tier and client components such as MDL Draw Enterprise Edition. Calls across machine boundaries are provided using standard Java Remote Method Invocation (RMI) technology.
- Web applications can be built using a variety of technologies. MDL provides specific support for the Java servlet technologies that are generally considered the most scalable and maintainable web development environments. Web server based applications can also call the middle-tier Java interfaces directly.
- Java-based application servers based on technologies such as Enterprise Java Beans (EJBs) and web services are becoming widely used in industry. Applications written in these standard technologies can simply call the Java interfaces exposed by MDL's middle tier.

Because both thick and thin clients are built on the same underlying middle-tier objects, both can work as part of a unified system. Even in thick clients, most of the application's logic is maintained on the middle tier. All applications have access to the same set of services and all can share the same objects. This consistent underlying layer means that all applications have the same implementation, even if their user interfaces are delivered through different technologies.

MDL uses its own interfaces to deliver its applications. This approach ensures that an interface is proven for stability and performance before it is released for general custom application development.

## *Transition to the New Technology*

MDL and its customers must consider the impact of any new technologies on all of the applications already built and deployed on ISIS client-server technology. These applications are doing useful work, and their logic represents a tremendous investment on the part of MDL and its customers.

The most important aspect of any transition is the migration of the data. MDL's molecule data cartridge strategy provides a clean migration for its customers, because both ISIS/Host and cartridge-based applications can read and write to the same database. This means that an organization can introduce an application based on new tools without fearing that it will cut off its existing ISIS users from their data. This data migration strategy is the most important step MDL can take to ensure a clean transition.

Developer interfaces require more attention. Applications built on ISIS/Base PL, Object Library, and Chemscape are firmly rooted in the assumptions of the ISIS client-server system. It is not possible to transition these tools automatically to a new interface without inheriting the limiting aspects of the old design.

The solution is to introduce new interfaces but continue to support the old for a long transition period. MDL's three-tier architecture is designed to coexist with the existing ISIS/Host client-server tools. This means that the existing ISIS/Base PL, Object Library, and Chemscape applications can remain in operation for a considerable period while MDL and its customers develop new applications built on the three-tier architecture.

But what if you are building or enhancing an application before the new tools are complete? Do you build it with existing tools or wait for future products? The answer depends on your organization's needs. In some cases, it may be reasonable to use existing MDL technologies, architecting the application in a three-tier manner so that you can easily move to new tools once they become available. In other cases, it may be best to use a hybrid approach. MDL's Consulting Services organization can assist customers in designing a project so that it will be consistent with the three-tier model and so that it will have as long a lifetime as possible.

## *Delivering the New Technology*

MDL's Discovery Framework and its underlying technology are a bold initiative, involving a large shift in technology and approach. With an integrated set of applications, database content, and technology, these developments provide a broad solution for the complex informatics needs of discovery. Because this solution is focused on delivering domain-specific requirements, it is the ideal framework for integrating discovery informatics.

MDL's strategy is to validate its Core Interface by building applications before releasing it for general customer use. This ensures that the documented developer interfaces are

stable and maintainable into future releases. In this way, MDL can develop, test, and validate the design, even making major changes to early versions as a result of the experience gained, without the risk of disrupting customer applications. Once MDL releases a documented interface, it makes a fundamental commitment to support those interfaces going forward, so that application developers can build applications with confidence that they can keep using those interfaces in future releases without major changes to their applications.

As a result, the MDL Core Interface has gone through two generations before release to customers. This means that the commercial product is already a third-generation product that has already proven robust and flexible.

The following applications and technologies are already available as part of the first and second generations:

- **MDL Chemistry Rules Interface** (formerly Cheshire for ISIS and CBRM) was first released in 1998. This product is based on a technology that is portable to the new architecture so that existing business rules are immediately usable in the new environment.
- **MDL Relational Chemistry Server** (formerly ISIS/Direct) was released in 1999, and its second major release followed in 2001. It has been put in production by a number of major companies.
- **MDL CrossFire Web Client** (released in Q4 2001). This application specifically relied on the authorization capabilities of the first generation of MDL Core Interface.
- **MDL Draw Enterprise Edition** was released for general use in August 2002, and major parts of it were included earlier in MDL's CrossFire Web and DiscoveryGate products.
- The **DiscoveryGate** Internet site went online for customer use in September 2002. DiscoveryGate includes the MDL Compound Locator application. DiscoveryGate integrates, indexes, and links over 15,000 journal titles and patent archives with more than 11 million structures from commercial databases, including their associated synthesis, bioactivity, physical property, and sourcing data. All of this is also linked to authoritative majored reference volumes that critically review synthetic methods. Discovery Gate is built using MDL's Compound Index and Compound Locator, as well as MDL's Literature Linking technology.
- **MDL Decision Manager 1.0,** currently in beta testing, provides a single interface for querying a project database, and then viewing, assessing, and collaborating on all the relevant project information. Among other features, users can cluster hits, filter on structure features, and calculate properties using preferred methods and algorithms. The resulting analysis can be tested as a new query, to identify additional information of interest.

MDL Core Interface 1.0 is scheduled for release by the end of 2002. MDL's Consulting Services group is already developing customer projects using this new

technology. At the same time, MDL is using this new version as the foundation for additional applications and services under development:

- **MDL Plate Manager 2.0** is an integrated plate management system that makes it possible for researchers to manage the information flow generated when making, screening, and analyzing thousands of chemical entities a day. Tasks enabled include: create, import, and reformat plates, search plate and sample information, and manage plate inventory including plate history and expiration information.
- **MDL Assay Explorer 3.0** refactors the existing Assay Explorer product on top of the new middle-tier technology. The existing MDL Assay Explorer enables a fully flexible definition of experiments, without rigid results templates, and without customization. Results information can be automatically read in, calculations can be automated, vocabulary can be controlled, and data can be filtered at will. A variety of graphical data representations are also available, including IC50 curves. Structures can be retrieved from ISIS/Host and MDL Relational Chemistry Server. Data from Assay Explorer is already accessible through MDL Core Interface 1.0.
- **MDL Compound Logistics and Procurement** is aimed at acquiring and managing compounds and plates from commercial sources and in a company's inventory, including the selection and/or purchase of such materials. It will rely on the MDL Plate Manager for handling plates, as well as other MDL Discovery Framework capabilities.
- **MDL Compound Locator 2.0** will offer tools to create in-house warehouses of normalized, deduplicated structures, indices to original sources, and normalized searching queries. Using technology developed for DiscoveryGate, the Compound Locator will offer integration of commercial data such as provided via DiscoveryGate with internal, proprietary information. Users can select compounds on the basis of a summary of the information that is available, and then drill down to full data in native databases via custom indexes called "topics".

## *Summary of Benefits*

The MDL Discovery Framework and its underlying technology benefits research organizations in a number of ways:

- **Preserves current investment; no need to replace all existing applications immediately.** Since ISIS/Host Relational Chemical Gateway and MDL Relational Chemistry Server can access the same physical database, existing applications based on ISIS/Host can continue to be used while new applications are built using MDL Discovery Framework. MDL's Discovery Framework allows co-existence of both new and existing applications.
- **In-house or third-party tools can readily be integrated.** MDL Discovery Framework has been designed to be an open system. A published API defines how applications integrate within the Discovery Framework.

- **Updates to applications can be straightforward.** The MDL Discovery Framework allows for in-house or third-party tools to be added or removed as needed, without altering the infrastructure already in place or interrupting the discovery workflow. Tools can be chosen to fit the specific user needs, and can be integrated in a predictable, safe, and maintainable fashion.

End users of the MDL Discovery Framework benefit directly from:

- **Application integration.** Because of the increased consistency of behavior between applications, end users find it easier to learn and use each new, related application, resulting in improved usability and increased productivity. Single log-on to the application reduces the number of usernames and passwords the user must remember. Properly designed, applications can appear truly seamless.
- **Data integration.** The enhanced ability to share data between applications and query multiple data sources streamlines the overall workflow, increasing productivity, and reducing the opportunity to introduce error by reducing the number of redundant tasks that a user must perform in order to complete a synthesis, an experiment or the analysis of an experimental dataset. Today, these tasks require the transfer of a single dataset into multiple applications.

End users of the MDL Discovery Framework also benefit from a number of indirect effects:

- **Faster application development.** End users will see more of their desired applications implemented more quickly, as a result of the improved application development environment.
- **Reduced application maintenance and deployment costs.** Centralized deployment and maintenance of business logic allows Research IT groups to focus on faster delivery of urgently needed new capabilities.
- **Easier integration of custom and commercial applications.** With all applications based on a common integration layer, end users will benefit from an easier-to-learn and easier-to-use system.

Application developers directly benefit from the use of the MDL Discovery Framework:

- Applications developers can focus on custom functionality rather than system architecture.
- The developer interface follows a simple, object-oriented model that is more powerful, consistent, and suited to custom application development than current models.
- The flexible architecture provides a single interface that supports a broad range of thick- and thin-client applications.
- The technology is scalable to the high volumes of data and usage required in today's discovery workflow.
- The three-tiered architecture is easier to deploy because most system configuration activities take place on the middle tier.

- By building everything on top of a common object layer, both standard and custom applications interoperate and integrate at a much more fundamental level than was possible with previous technologies.
- New applications can be delivered with higher quality and timeliness, increasing constituency approval.

Please contact MDL if you would like more information about how MDL Discovery Framework could benefit your organization.