Embedding Data Integration in Software Applications
Abstract

This article outlines the requirements for a robust data integration platform and the capabilities of the Informatica data integration platform. It also provides a high level introduction to the Informatica data integration platform interfaces and how you can leverage them in your applications.

Table of Contents

Overview ........................................................................................................................................................................... 2
Anatomy of a Software Application ................................................................................................................................... 2
User Interface Logic .......................................................................................................................................................... 3
Application Logic ............................................................................................................................................................ 3
Data Integration Logic ..................................................................................................................................................... 4
Application Metadata Repository and Data Stores ....................................................................................................... 4
Administration Services .................................................................................................................................................. 4
Execution Control ........................................................................................................................................................ 4
Embedded Data Integration .............................................................................................................................................. 4
  Application-Specific User Interface ............................................................................................................................... 5
  Requirements for the Data Integration Platform ........................................................................................................... 5
The PowerCenter Data Integration Platform ..................................................................................................................... 7
  PowerCenter Data Integration Platform and Interfaces ................................................................................................ 7
  Extending PowerCenter with the PowerCenter APIs: Partner Case Studies .................................................................. 8
  Informatica Developer Platform APIs ........................................................................................................................... 9
Conclusion ...................................................................................................................................................................... 10

Overview

Whether you are developing a software-as-a-service (SaaS) application or a classic on-premise deployed application, data integration plays a major role in successful adoption and end customer delivery. The success of the application is closely tied to how well the application manages and resolves data issues.

In most cases, the initial attempt at adding data integration to applications involves hand coding, which is complex and expensive to maintain. A better way to include data integration in applications is to adopt the industry’s leading data integration platform and embedding and customizing it to your specific requirements.

This article outlines the requirements for a robust data integration platform and the capabilities of the Informatica data integration platform. It also provides a high level introduction to the Informatica data integration platform interfaces and how you can leverage them in your applications.

Anatomy of a Software Application

Let us take a look at the anatomy of a typical application and how and where data integration comes into play.
The following diagram shows the logical components of a typical software application:

The components provide distinct services. Together, they drive the application behavior. Note that these components are logical. In the actual physical implementation, these components can be combined or further broken down into smaller components based on the architecture of the application.

The components and logic of the application can be broken down into the following components:

- User interface logic
- Application logic
- Data integration logic
- Application metadata repository and data stores
- Administration services
- Execution control

**User Interface Logic**

The user interface drives the user interaction for the application and interfaces with the application layer to carry out specific end user functions and utilities. For example, in a Customer Hub application, the user interface provides the screens for editing and managing the customer information stored in the hub.

**Application Logic**

Application logic drives application behavior. This layer contains the core business rules and processes for the application. It interfaces with the user interface layer to control the user interface behavior of the application and drive user interaction. It also interfaces with the data integration layer to carry out the data integration functions of the application. In the example of a Customer Hub application, the application logic controls the processes that govern how customer information is synchronized with the hub, how duplicate customers are handled, and how data is written back to CRM systems. In many cases, this component includes process engines to orchestrate specific business processes.
**Data Integration Logic**

The data integration layer carries out all the data interactions of the application. The data integration layer can be tightly embedded in the application layer or can be spread across different layers. The data integration layer provides the following services:

- Accessing and updating the application data stores and other external data sources that the application depends on.
- Transforming data in various formats, from external data formats to application-specific data formats or from application formats to external formats.
- Fixing errors and checking the quality of data entered into the system through user interfaces and other data adapters.

In the example of the Customer Hub application, the data integration layer manages the loads and incremental updates for the hub. It handles all the data transformation and quality issues during the load process. It also transforms customer representations of the data from external applications to the model used by the hub. The data integration logic cleans up the customer data attributes, augments missing information, and detects and handles duplicate data. The data integration layer also defines the logic for writing data back into the CRM and other systems so that customer data is in sync across the hub and all systems. In addition, the data integration layer provides data services to access customer data for display in customer portals.

**Application Metadata Repository and Data Stores**

The repository contains the metadata for the application and drives the application behavior. The metadata may be in a database repository or stored in configuration files. The application data catalog, which includes the description of the application and external data structures, is typically stored in the metadata repository. The data integration layer uses the repository when it accesses data from external systems.

The application data stores contain the data required by the application. Typically, the data is stored in relational databases, XML files, or other proprietary data storage. The application accesses data from the data stores through SQL queries or web services, or through application-specific APIs. In the example of the Customer Hub application, the metadata repository contains metadata that define the hub structure, the data structures of the customer tables in external applications, and the data and business processes. The data store for the Customer Hub application is the data hub, which is also accessible through SQL, web services, or an API layer.

**Administration Services**

The administration component of the application includes application setup and configuration, user security and management, data security, deployment and migration, and backup.

**Execution Control**

Processing control drives the business processes and operations of the application. Typically, schedulers invoke the data processes. Execution control also includes monitoring of the system.

**Embedded Data Integration**

A data integration platform can provide all the data integration services required by a software application. However, the software application needs to control not only the application behavior but also the underlying data integration functions. This implies that the application interface must be able to invoke and control the data integration functions provided by the data integration platform. Essentially, the data integration capability must be embedded within the application.

Embedding a data integration engine in an application is an excellent alternative to implementing data integration functions from the ground up. This approach not only ensures appropriate processing of data, it also enforces a loosely
coupled architecture between the application components and the data integration component. This approach also allows applications and data integration functions to evolve without impacting one another.

**Application-Specific User Interface**

When data integration capabilities are embedded in an application, the application can provide custom user interfaces for configuring and controlling the underlying data integration activities. Custom interfaces greatly improve the user experience with the application. Typically, the generic user interface for the data integration platform exposes all the capabilities of the platform. A custom or purpose-built user interface can expose and provide access only to the data integration capabilities used by the application. For instance, the custom user interfaces can be built to expose only the data sources and the business rule parameters that are specific to the application.

In the example of the Customer Hub application, the user interface can be designed specifically to handle the data hub schema and attributes. The custom interface greatly simplifies how users specify the data mapping rules.

**Requirements for the Data Integration Platform**

The following diagram shows the services that a data integration platform should provide to allow the application control over data integration processes:

A robust data integration platform must provide the following capabilities, customization, and levels of control to the software application:

- Access to custom data sources
- Extensible transformation processing
- Application-driven data integration logic
- Execution controls and monitoring
- Security and access control
• Configuration
• Auditing and reporting services
• Multi-tenancy management for SaaS

**Access to Custom Data Sources**

The data integration platform typically provides out of the box access to common data sources such as relational databases, files, XML, mainframe data sources, and ERP and CRM applications. The software application can use the adapters provided by the integration platform to access the data sources. The application might also need to process external data stored in custom data sources. Or the application might need to expose an API layer to access its own data store. In either case, the data integration platform must provide a way to extend data connectivity to access and update data in these custom data sources.

**Extensible Transformation Processing**

The data integration platform can perform the majority of data transformation processing requirements for the software application. In some cases, however, an application can require custom transformation processing beyond what is provided by the data integration platform. For example, certain business transformation rules can require access to an application-specific API during the data load process. In the example of the Customer Hub application, the application must control surrogate key generation for the customer data hub load through an API. The data integration process invokes the API functions to request a key value for a new customer. The data integration platform must provide a way to extend the transformation capability.

**Application-Driven Data Integration Logic**

The user interface provided by a software application affects the behavior of the application. The data integration platform must be able to provide a user interface customized for the application data requirements. It should also be able to handle data according to the application data requirements. For example, an application can allow an end user to pick and choose the data sources and attributes that need transformation. The end user also indicates the types of data transformations and augmentation that is needed.

In the example of the Customer Hub application, the user picks the relevant customer tables and attributes from external CRM systems and provides a mapping to the corresponding entities and attributes in the customer hub model. The application must translate the mapping into the appropriate data integration logic to process the data as required by the user. The data integration platform must provide the automation interfaces and APIs that allow the application to control the data integration. The data integration platform must also provide a way for the application to manage the data integration process through its own user interface, without exposing the data integration platform user interface.

In addition, the data integration platform must provide an interface for the application to correlate application-specific metadata with the data integration metadata. This permits a single point of maintenance for all the metadata.

**Execution Controls and Monitoring**

The application needs to run data integration processes, such as data cleansing, transformation, and loading, on demand or on a schedule. The data integration platform must provide appropriate interfaces to allow the software application to invoke the operations.

The data integration platform must also provide a web service interface that allows the application to invoke external web services and extend the data integration capabilities with standard web services technology. This would allow external business process engines, such as BPEL engines, to invoke and orchestrate the data integration processes. Loosely coupled integration between processes provides more flexibility and resiliency to application changes.

In addition, the data integration platform must be able to monitor the data integration processes, capture run-time statistics, and provide reports on the status of the processes. In the example of the Customer Hub application, the application must be able to trigger data hub synchronization processes and monitor the number of customers that were added or rejected, and the reasons for rejection. This will allow the application to trap bad data rows and reprocess them as needed.
Security and Access Control

The application needs to ensure that it has secure access to data in external data sources. It also needs to protect all sensitive data that it stores in its own data stores. The data integration platform must support software application security protocols for network connections and data transmission to ensure security in data access. In the example of the Customer Hub application, access to enterprise CRM systems must be governed by security and access controls provided by the CRM application.

The application level permissions and privileges must drive the permissions and privileges available in the data integration platform. The data integration platform can also provide an additional level of security by leveraging a common enterprise directory, such as LDAP, as the single point of control for access to all applications.

Configuration

The software application must be able to administer data integration functions and processes, including installation and configuration, user administration, backup, and migration. The data integration platform must provide an interface to allow administration of the data integration process through its own application interface or through an external application.

Auditing and Reporting Services

The software application requires a reliable system to manage and audit changes to the data integration rules and to operational metadata. The data integration platform must provide interfaces to query and retrieve change management information. It must provide a reporting interface to operational information and statistics.

Multi-Tenancy Management for SaaS

For applications that use the SaaS model, separation of client application metadata and data is critical. Data integration metadata and processing must be isolated for each client. The data integration platform must provide a way to manage and store data and data integration rules that ensures privacy and security. The platform must also provide assurance of scalability for large clients and a large number of clients.

The PowerCenter Data Integration Platform

The PowerCenter data integration platform provides an industry-leading stand-alone platform for data integration. In addition, the platform can provide embedded data processing services for software applications with complex data integration requirements.

Application vendors can rely on the robust data integration services provided by the PowerCenter platform to meet their data integration requirements. By embedding PowerCenter data integration services within their software application, they can leverage the data integration capabilities of PowerCenter and focus their development efforts on application-specific components.

In addition to robust data integration services, the PowerCenter platform provides the following advantages:

- A highly scalable environment to process very large data volumes and an advanced load balancer that distributes processing optimally in both SMP and grid environments.
- A highly available data integration environment to ensure that mission critical applications can meet SLAs.
- A highly secure environment that ensures user access to enterprise data sources adheres to security guidelines.
- A metadata repository that enables a high degree of metadata reuse and which can be audited to meet governance and compliance requirements.

PowerCenter Data Integration Platform and Interfaces

PowerCenter provides programming interfaces that enable you to access its data integration functionalities and manage and control all aspects of application behavior. You can use the interfaces to embed data integration capabilities in your application.
The following table summarizes the capabilities provided by PowerCenter through various programming interfaces:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensible data sources</td>
<td>PowerExchange API</td>
<td>Provide connectivity to custom data sources and data formats.</td>
</tr>
<tr>
<td>Extensible transformation processing</td>
<td>Transformation API</td>
<td>Create custom transformations that invoke application-specific APIs.</td>
</tr>
<tr>
<td>Custom data integration user interfaces and application-driven data integration logic</td>
<td>Design API</td>
<td>Provide custom user interfaces and define data integration logic.</td>
</tr>
<tr>
<td>Execution control and monitoring</td>
<td>Operations API, web services, command line interface</td>
<td>Execute and monitor PowerCenter integration processes through the command line interface, Java and web services, and SQL Views.</td>
</tr>
<tr>
<td>Security and access control</td>
<td>Repository administration command line interface</td>
<td>Administer PowerCenter users and manage application connections.</td>
</tr>
<tr>
<td>Administration control</td>
<td>Repository administration command line interface</td>
<td>Administer the PowerCenter. Migrate applications across environments. Perform backups.</td>
</tr>
<tr>
<td>Services control</td>
<td>Domain administration command line interface</td>
<td>Start and stop integration services. Add grid nodes and manage grid resources for load distribution.</td>
</tr>
<tr>
<td>Metadata reporting</td>
<td>Metadata Views</td>
<td>Report on design metadata and run-time statistics.</td>
</tr>
</tbody>
</table>

**Extending PowerCenter with the PowerCenter APIs: Partner Case Studies**

The following are examples of how software vendors have leveraged the APIs provided by the PowerCenter integration platform.

**Persistent**

“As an Informatica development partner, Persistent relies on the Informatica SDK and corresponding API’s to deliver rich and flexible functionality. We’ve used this environment to deliver a number of PowerExchange connectivity solutions now used by hundreds of Informatica customers.”

Dhwani Katagade – Lead Developer

**Data Synapse**

“As a leader in dynamic application service management (DASM) software, DataSynapse relies on ISV partner APIs to build and manage enterprise-class application services. To provide a joint solution with Informatica, we used PowerCenter’s Load Manager API and were able to develop a flexible, joint integration quickly and with minimal effort.”

**Leading Performance Management Solution**

PowerCenter is embedded in an industry leading performance management solution that integrates financial management applications with the business integration (BI) platform. PowerCenter provides data management services for this solution, which allows the integration of financial data, master data, and metadata with other enterprise CRM and ERP systems. Leveraging PowerCenter for the data management services provided a robust and easy deployment solution to integrating information across systems. The solution eliminated data fragmentation across the enterprise and optimized BPM and BI deployments.
**Informatica On Demand**

Informatica has the broadest portfolio of data integration solutions in the industry specifically designed to integrate Salesforce CRM and Force.com with on-premise databases and applications. These solutions enable a company to ensure data accuracy, improve business decisions and operations, and derive maximum value from your Salesforce investment.

Informatica on Demand leverages the PowerCenter data integration platform and interfaces to provide a purpose-specific multi-tenant hosted application to integrate Salesforce CRM with customer data in software applications.

**Informatica Developer Platform APIs**

**PowerExchange API**

*Purpose: Extend connectivity to custom data sources*

You can use the PowerExchange API to develop custom adapters for new data sources. The custom PowerExchange adapters allow PowerCenter sessions to read and write data to these data sources.

The PowerExchange API is available in Java and C++.

**Transformation API**

*Purpose: Develop custom transformations for application-specific processing*

PowerCenter provides built-in transformations that address the majority of the transformation requirements. In some cases, however, custom transformations are needed to invoke external APIs and programs that perform application-specific processing of the data. You can use the Transformation API to develop custom transformations that can be used in PowerCenter mappings to process data based on your requirements.

The Transformation API is available in C and Java.

**Design API**

*Purpose: Programmatically generate data integration logic or metadata and build custom data integration user interfaces*

The Design API allows programmatic creation of PowerCenter design metadata. You can leverage this API to build custom user interfaces and purpose-built interfaces for creating PowerCenter mappings and other design metadata such as sessions and workflows. You can also leverage the Design API to build automation interfaces to generate design metadata from other application metadata. You can use this API to substitute the user interface provided by PowerCenter with custom user interfaces specific to the application.

The Design API is available in Java. It was formerly known as the Java Mapping Framework (JMF).

**Operations API and Web Services**

*Purpose: Control the execution and monitor the progress of PowerCenter jobs*

The Operations API allows programmatic execution of PowerCenter workflows. This API allows external applications to start workflows and monitor the progress. You can use this API to retrieve messages and statistics while the workflow is executing.

The Operations API is available in Java. It was formerly known as the Load Mapping API (LMAPI). You can also access the Operations API through the PowerCenter web service operations.
Repository Administration CLI (pmrep)

*Purpose: Administer PowerCenter users and repository*

`pmrep` is a command line utility that allows you to automate PowerCenter repository management functions. `pmrep` also allows you to manage PowerCenter users and roles on the command line. External applications can leverage `pmrep` to manage the PowerCenter repository.

Domain Administration CLI (nmcmd)

*Purpose: Administer PowerCenter Services*

`nmcmd` is a command line utility to administer PowerCenter services such as the Integration Service and the Repository Service. Using this command line utility, you can create new PowerCenter services and start or stop these services. External applications can use this utility to manage the services in the PowerCenter domain.

You can also use this utility to administer a grid and manage resources available for the nodes in the grid. External grid management tools can use this utility to configure and manage the PowerCenter grid environment.

Metadata Views (MXViews)

*Purpose: Administer PowerCenter Services*

MXViews are a set of database SQL views provided for the PowerCenter metadata repository tables. You can leverage these views to extract design time metadata and run-time statistics and logs from the PowerCenter metadata repository.

Conclusion

The viability and adoption of software applications is largely dependent on its ability to address the challenges associated with enterprise data. By leveraging the Informatica data integration platform, software applications can address all data integration requirements in a robust and optimal manner. Application vendors can increase their agility and focus their efforts on adding more application-specific functionality for their end users. Robust data integration capability coupled with application functionality will increase the competitive positioning of the application.

Author

Naresh Govindaraj  
Senior Director, Product Management

Don Tirsell  
Senior Director, Alliances Technology

Marissa Johnston  
Staff Technical Writer