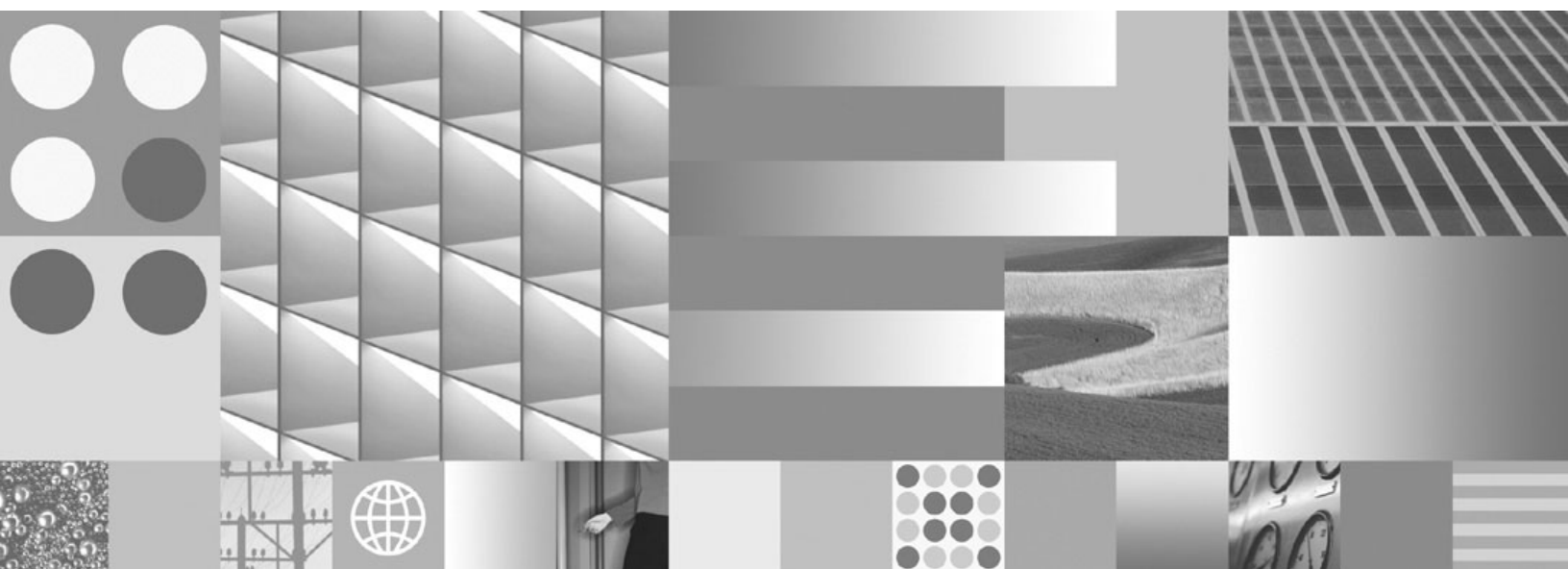
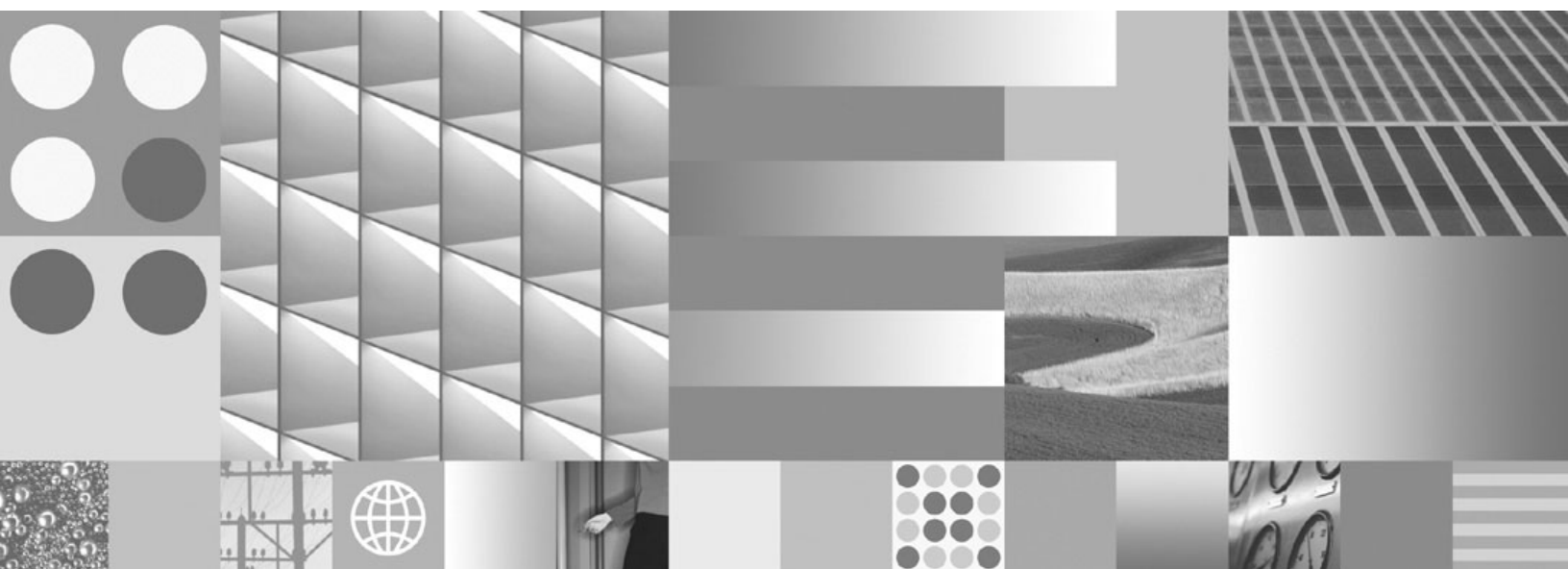




IBM Information Server Introduction



IBM Information Server Introduction

Note

Before using this information and the product that it supports, read the information in “Notices” on page 137.

Contents

Chapter 1. Introduction 1

Chapter 2. Architecture and concepts . . . 5

Parallel processing in IBM Information Server	7
Parallelism basics in IBM Information Server	8
Scalability in IBM Information Server	11
Support for grid computing in IBM Information Server	12
Shared services in IBM Information Server	13
Administrative services in IBM Information Server	13
Reporting services in IBM Information Server	15

Chapter 3. Metadata services 17

Metadata services introduction	17
A closer look at metadata services in IBM Information Server	20
WebSphere Business Glossary	20
WebSphere Business Glossary tasks	21
WebSphere Metadata Server	23
Information resources for metadata services	28

Chapter 4. Service-oriented integration 29

Introduction to service-oriented integration in IBM Information Server	29
A closer look at service-oriented integration in IBM Information Server	32
SOA components in IBM Information Server	35
WebSphere Information Services Director tasks	36
SOA and data integration.	40
Information resources for WebSphere Information Services Director.	42

Chapter 5. WebSphere Information Analyzer 45

WebSphere Information Analyzer capabilities	45
A closer look at WebSphere Information Analyzer	48
WebSphere Information Analyzer tasks	52
Data profiling and analysis	53
Data monitoring and trending	57
Results of the analysis.	60
Information resources for WebSphere Information Analyzer	61

Chapter 6. WebSphere QualityStage . . . 63

Introduction to WebSphere QualityStage.	63
A closer look at WebSphere QualityStage	67
WebSphere QualityStage tasks	70
Investigate stage.	71
Standardize stage	73
Match stages overview	74

Survive stage.	79
Accessing metadata services.	79
Information resources for WebSphere QualityStage	80

Chapter 7. WebSphere DataStage 81

Introduction to WebSphere DataStage.	81
A closer look at WebSphere DataStage	83
WebSphere DataStage tasks	87
WebSphere DataStage elements.	87
Overview of the Designer, Director, and Administrator clients	89
Data transformation for zSeries	107
WebSphere DataStage MVS Edition	107
WebSphere DataStage Enterprise for z/OS.	109
Information resources for WebSphere DataStage	110

Chapter 8. WebSphere Federation Server 111

Introduction to WebSphere Federation Server.	112
A closer look at WebSphere Federation Server	115
The federated server and database	115
Wrappers and other federated objects	116
Query optimization	117
Two-phase commit for federated transactions	118
Rational Data Architect	119
WebSphere Federation Server tasks	120
Federated objects	120
Cache tables for faster query performance.	121
Monitoring federated queries	122
Federated stored procedures	123
Information resources for WebSphere Federation Server	124

Chapter 9. Companion products 125

WebSphere DataStage Packs	125
A closer look at WebSphere DataStage Packs	127
WebSphere DataStage Change Data Capture	130
WebSphere Replication Server.	131
WebSphere Data Event Publisher	133
Information resources for IBM Information Server companion products	133

Accessing information about the product 135

Providing comments on the documentation	135
---	-----

Notices 137

Trademarks	139
----------------------	-----

Index 141

Chapter 1. Introduction

Most of today's critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

IBM® Information Server is the industry's first comprehensive, unified foundation for enterprise information architectures, capable of scaling to meet any information volume requirement so that companies can deliver business results within these initiatives faster and with higher quality results.

IBM Information Server combines the technologies within the IBM Information Integration Solutions portfolio (WebSphere® DataStage®, WebSphere QualityStage, WebSphere Information Analyzer, and WebSphere Information Integrator) into a single unified platform that enables companies to understand, cleanse, transform, and deliver trustworthy and context-rich information.

Over the last two decades, companies have made significant investments in enterprise resource planning, customer relationship management, and supply chain management packages. Companies also are leveraging innovations such as service-oriented architectures (SOA), Web services, XML, grid computing, and Radio Frequency Identification (RFID).

These investments have increased the amount of data that companies are capturing about their businesses. But companies encounter significant integration hurdles when they try to turn that data into consistent, timely, and accurate information for decision-making.

IBM Information Server helps you derive more value from complex, heterogeneous information. It helps business and IT personnel collaborate to understand the meaning, structure, and content of information across a wide variety of sources. IBM Information Server helps you access and use information in new ways to drive innovation, increase operational efficiency, and lower risk.

IBM Information Server supports all of these initiatives:

Business intelligence

IBM Information Server makes it easier develop a unified view of the business for better decisions. It helps you understand existing data sources, cleanse, correct, and standardize information, and load analytical views that can be reused throughout the enterprise.

Master data management

IBM Information Server simplifies the development of authoritative master data by showing where and how information is stored across source systems. It also consolidates disparate data into a single, reliable record, cleanses and standardizes information, removes duplicates, and links records together across systems. This master record can be loaded into operational data stores, data warehouses, or master data applications such as WebSphere Customer Center. The record can also be assembled, completely or partially, on demand.

Infrastructure rationalization

IBM Information Server aids in reducing operating costs by showing relationships between systems and by defining migration rules to consolidate instances or move data from obsolete systems. Data cleansing and matching ensure high-quality data in the new system.

Business transformation

IBM Information Server can speed development and increase business agility by providing reusable information services that can be plugged into applications, business processes, and portals. These standards-based information services are maintained centrally by information specialists but are widely accessible throughout the enterprise.

Risk and compliance

IBM Information Server helps improve visibility and data governance by enabling complete, authoritative views of information with proof of lineage and quality. These views can be made widely available and reusable as shared services, while the rules inherent in them are maintained centrally.

Capabilities

IBM Information Server features a unified set of separately orderable product modules, or suite components, that solve multiple types of business problems. Information validation, access and processing rules can be reused across projects, leading to a higher degree of consistency, stronger control over data, and improved efficiency in IT projects.

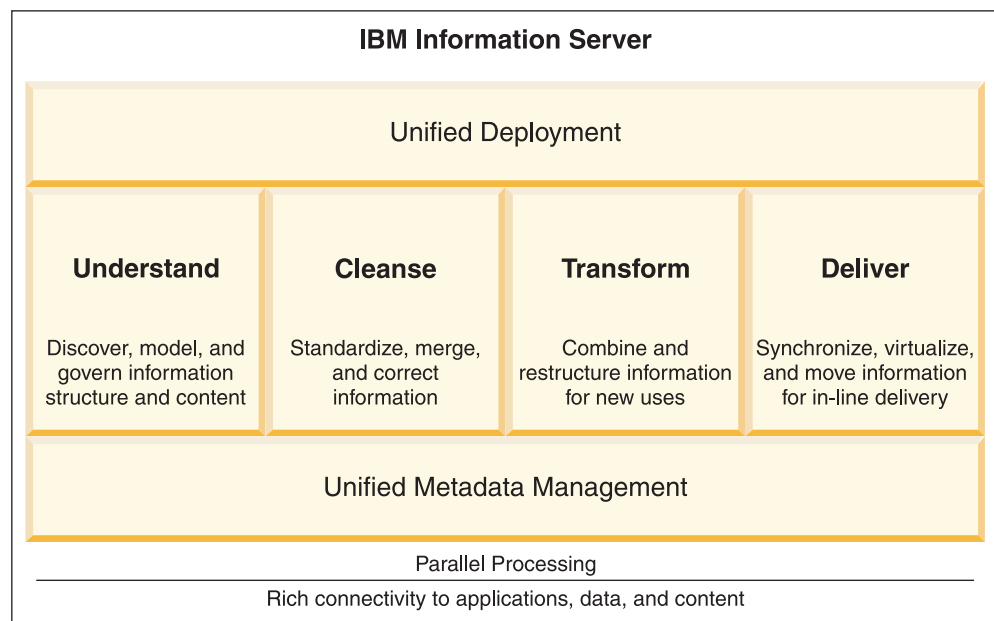


Figure 1. IBM Information Server

As Figure 1 shows, IBM Information Server enables businesses to perform four key integration functions:

Understand your data

IBM Information Server can help you automatically discover, define, and model information content and structure and understand and analyze the

meaning, relationships, and lineage of information. By automating data profiling and data-quality auditing within systems, organizations can achieve these goals:

- Understand data sources and relationships
- Eliminate the risk of using or proliferating bad data
- Improve productivity through automation
- Leverage existing IT investments

IBM Information Server makes it easier for businesses to collaborate across roles. Data analysts can use analysis and reporting functionality, generating integration specifications and business rules that they can monitor over time. Subject matter experts can use Web-based tools to define, annotate, and report on fields of business data. A common metadata foundation makes it easier for different types of users to create and manage metadata by using tools that are optimized for their roles.

Cleanse your information

IBM Information Server supports information quality and consistency by standardizing, validating, matching, and merging data. It can certify and enrich common data elements, use trusted data such as postal records for name and address information, and match records across or within data sources. IBM Information Server allows a single record to survive from the best information across sources for each unique entity, helping you to create a single, comprehensive, and accurate view of information across source systems.

Transform your data into information

IBM Information Server transforms and enriches information to ensure that it is in the proper context for new uses. Hundreds of prebuilt transformation functions combine, restructure, and aggregate information.

Transformation functionality is broad and flexible, to meet the requirements of varied integration scenarios. For example, IBM Information Server provides inline validation and transformation of complex data types such as U.S. Health Insurance Portability and Accountability Act (HIPAA), and high-speed joins and sorts of heterogeneous data. IBM Information Server also provides high-volume, complex data transformation and movement functionality that can be used for standalone extract-transform-load (ETL) scenarios, or as a real-time data processing engine for applications or processes.

Deliver your information

IBM Information Server provides the ability to virtualize, synchronize, or move information to the people, processes, or applications that need it. Information can be delivered by using federation or time-based or event-based processing, moved in large bulk volumes from location to location, or accessed in place when it cannot be consolidated.

IBM Information Server provides direct, native access to a wide variety of information sources, both mainframe and distributed. It provides access to databases, files, services and packaged applications, and to content repositories and collaboration systems. Companion products allow high-speed replication, synchronization and distribution across databases, change data capture, and event-based publishing of information.

Chapter 2. Architecture and concepts

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

The architecture is service oriented, enabling IBM Information Server to work within an organization's evolving enterprise service-oriented architectures. A service-oriented architecture also connects the individual suite components of IBM Information Server.

By eliminating duplication of functions, the architecture efficiently uses hardware resources and reduces the amount of development and administrative effort that are required to deploy an integration solution.

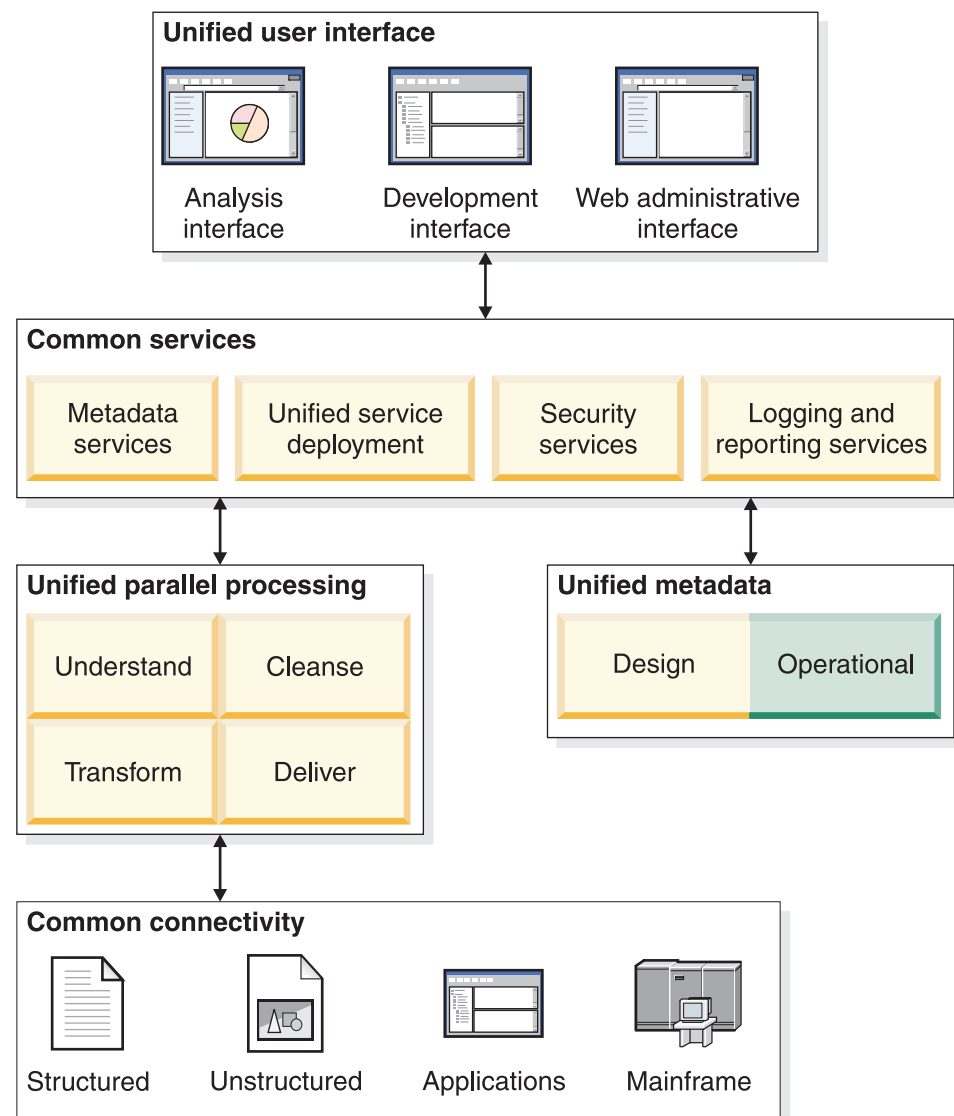


Figure 2. IBM Information Server high-level architecture

Figure 2 on page 5 shows the top levels of the IBM Information Server architecture.

Unified parallel processing engine

Much of the work that IBM Information Server does takes place within the parallel processing engine. The engine handles data processing needs as diverse as performing analysis of large databases for WebSphere Information Analyzer, data cleansing for WebSphere QualityStage, and complex transformations for WebSphere DataStage. This parallel processing engine is designed to deliver:

- Parallelism and pipelining to complete increasing volumes of work in decreasing time windows
- Scalability by adding hardware (for example, processors or nodes in a grid) with no changes to the data integration design
- Optimized database, file, and queue processing to handle large files that cannot fit in memory all at once or with large numbers of small files

Common connectivity

IBM Information Server connects to information sources whether they are structured, unstructured, on the mainframe, or applications.

Metadata-driven connectivity is shared across the suite components, and connection objects are reusable across functions.

Connectors provide design-time importing of metadata, data browsing and sampling, run-time dynamic metadata access, error handling, and high functionality and high performance run-time data access. Prebuilt interfaces for packaged applications called *Packs* provide adapters to SAP, Siebel, Oracle, and others, enabling integration with enterprise applications and associated reporting and analytical systems.

Unified metadata

IBM Information Server is built on a unified metadata infrastructure that enables shared understanding between business and technical domains. This infrastructure reduces development time and provides a persistent record that can improve confidence in information. All functions of IBM Information Server share the same metamodel, making it easier for different roles and functions to collaborate.

A common metadata repository provides persistent storage for all IBM Information Server suite components. All of the products depend on the repository to navigate, query, and update metadata. The repository contains two kinds of metadata:

Dynamic

Dynamic metadata includes design-time information.

Operational

Operational metadata includes performance monitoring, audit and log data, and data profiling sample data.

Because the repository is shared by all suite components, profiling information that is created by WebSphere Information Analyzer is instantly available to users of WebSphere DataStage and WebSphere QualityStage, for example.

The repository is a J2EE application that uses a standard relational database such as IBM DB2®, Oracle, or SQL Server for persistence (DB2 is provided with IBM Information Server). These databases provide backup, administration, scalability, parallel access, transactions, and concurrent access.

Common services

IBM Information Server is built entirely on a set of shared services that centralize core tasks across the platform. These include administrative tasks such as security, user administration, logging, and reporting. Shared services allow these tasks to be managed and controlled in one place, regardless of which suite component is being used. The common services also include the metadata services, which provide standard service-oriented access and analysis of metadata across the platform. In addition, the common services layer manages how services are deployed from any of the product functions, allowing cleansing and transformation rules or federated queries to be published as shared services within an SOA, using a consistent and easy-to-use mechanism.

IBM Information Server products can access three general categories of service:

Design

Design services help developers create function-specific services that can also be shared. For example, WebSphere Information Analyzer calls a column analyzer service that was created for enterprise data analysis but can be integrated with other parts of IBM Information Server because it exhibits common SOA characteristics.

Execution

Execution services include logging, scheduling, monitoring, reporting, security, and Web framework.

Metadata

Using metadata services, metadata is shared “live” across tools so that changes made in one IBM Information Server product are instantly visible across all of the suite components. Metadata services are tightly integrated with the common repository and are packaged in WebSphere Metadata Server. You can also exchange metadata with external tools by using metadata services.

The common services layer is deployed on J2EE-compliant application servers such as IBM WebSphere Application Server, which is included with IBM Information Server.

Unified user interface

The face of IBM Information Server is a common graphical interface and tool framework. Shared interfaces such as the IBM Information Server console and Web console provide a common look and feel, visual controls, and user experience across products. Common functions such as catalog browsing, metadata import, query, and data browsing all expose underlying common services in a uniform way. IBM Information Server provides rich client interfaces for highly detailed development work and thin clients that run in Web browsers for administration.

Application programming interfaces (APIs) support a variety of interface styles that include standard request-reply, service-oriented, event-driven, and scheduled task invocation.

Parallel processing in IBM Information Server

Companies today must manage, store, and sort through rapidly expanding volumes of data and deliver it to end users as quickly as possible.

To address these challenges, organizations need a scalable data integration architecture that contains the following components:

- A method for processing data without writing to disk, in batch and real time.
- Dynamic data partitioning and in-flight repartitioning.
- Scalable hardware that supports symmetric multiprocessing (SMP), clustering, grid, and massively parallel processing (MPP) platforms without requiring changes to the underlying integration process.
- Support for parallel databases including DB2, Oracle, and Teradata, in parallel and partitioned configurations.
- An extensible framework to incorporate in-house and vendor software.

IBM Information Server addresses all of these requirements by exploiting both pipeline parallelism and partition parallelism to achieve high throughput, performance, and scalability.

Parallelism basics in IBM Information Server

The pipeline parallelism and partition parallelism that are used in IBM Information Server underly its high-performance, scalable architecture.

Data pipelining

Data pipelining is the process of pulling records from the source system and moving them through the sequence of processing functions that are defined in the data-flow (the job). Because records are flowing through the pipeline, they can be processed without writing the records to disk, as Figure 3 shows.

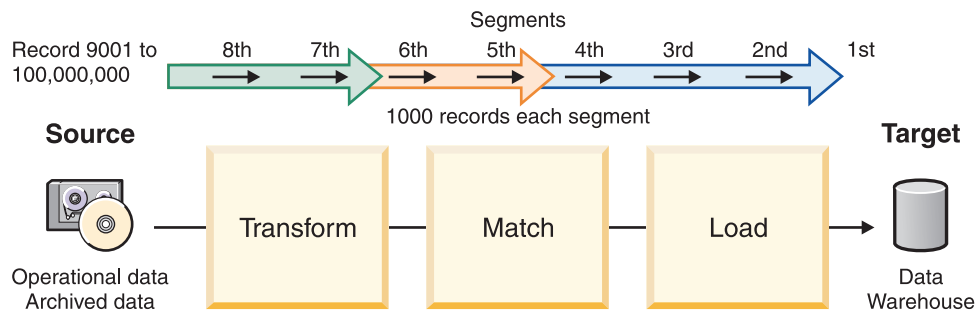


Figure 3. Data pipelining

Data can be buffered in blocks so that each process is not slowed when other components are running. This approach avoids deadlocks and speeds performance by allowing both upstream and downstream processes to run concurrently.

Without data pipelining, the following issues arise:

- Data must be written to disk between processes, degrading performance and increasing storage requirements and the need for disk management.
- The developer must manage the I/O processing between components.
- The process becomes impractical for large data volumes.
- The application will be slower, as disk use, management, and design complexities increase.
- Each process must complete before downstream processes can begin, which limits performance and full use of hardware resources.

Data partitioning

Data partitioning is an approach to parallelism that involves breaking the record set into partitions, or subsets of records. Data partitioning generally provides linear increases in application performance. Figure 4 shows data that is partitioned by customer surname before it flows into the Transformer stage.

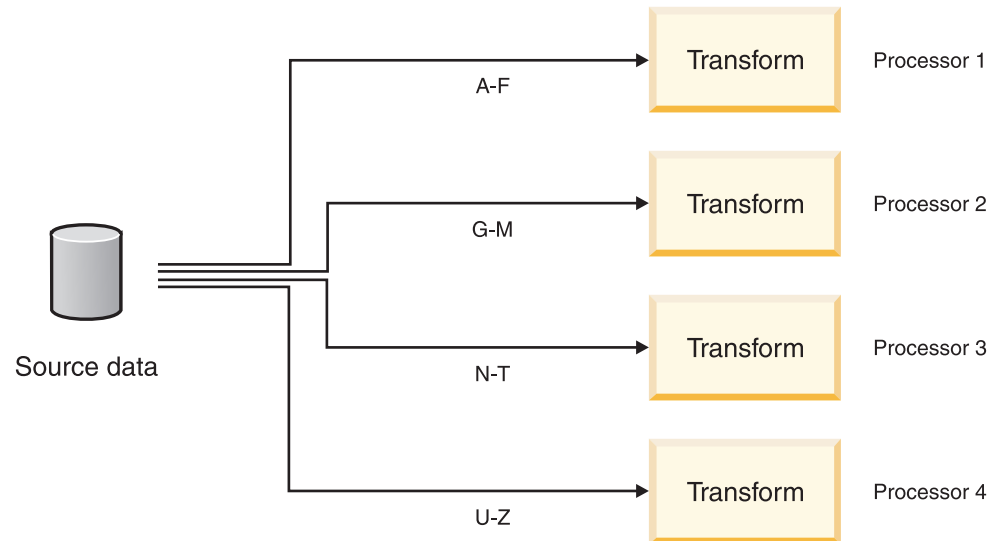


Figure 4. Data partitioning

A scalable architecture should support many types of data partitioning, including the following types:

- Hash key (data) values
- Range
- Round-robin
- Random
- Entire
- Modulus
- Database partitioning

IBM Information Server automatically partitions data based on the type of partition that the stage requires. Typical packaged tools lack this capability and require developers to manually create data partitions, which results in costly and time-consuming rewriting of applications or the data partitions whenever the administrator wants to use more hardware capacity.

In a well-designed, scalable architecture, the developer does not need to be concerned about the number of partitions that will run, the ability to increase the number of partitions, or repartitioning data.

Dynamic repartitioning

In the examples shown in Figure 4 and Figure 5 on page 10, data is partitioned based on customer surname, and then the data partitioning is maintained throughout the flow.

This type of partitioning is impractical for many uses, such as a transformation that requires data partitioned on surname but must then be loaded into the data warehouse by using the customer account number.

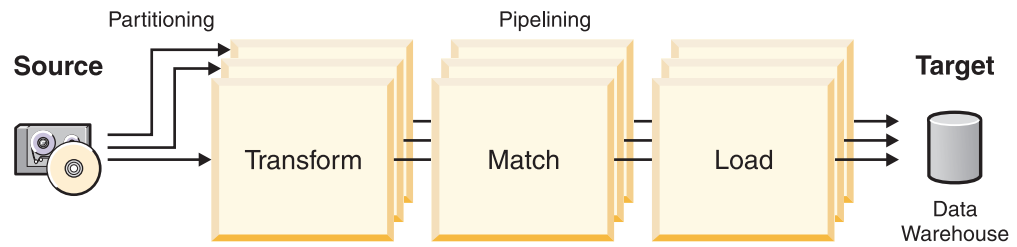


Figure 5. Data partitioning and parallel execution - a less practical approach

Dynamic data repartitioning is a more efficient and accurate approach. With dynamic data repartitioning, data is repartitioned while it moves between processes without writing the data to disk, based on the downstream process that data partitioning feeds. The IBM Information Server parallel engine manages the communication between processes for dynamic repartitioning.

Data is also pipelined to downstream processes when it is available, as Figure 6 shows.

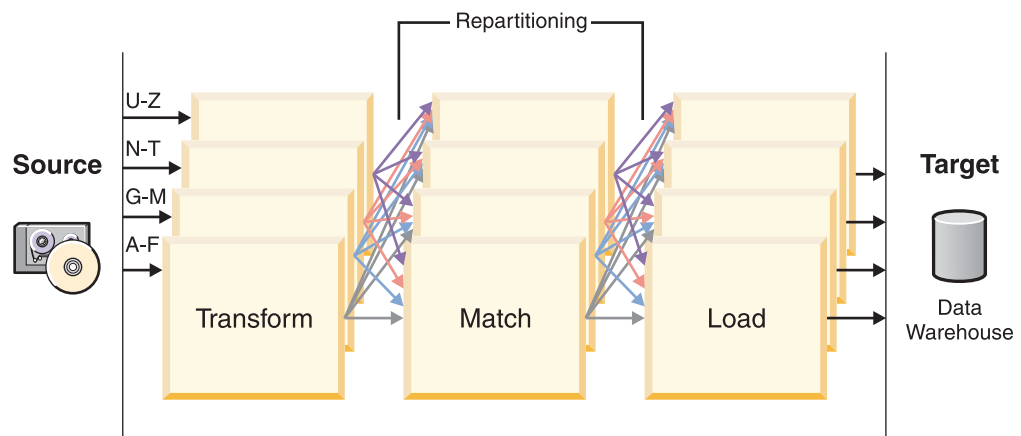


Figure 6. Dynamic data repartitioning - a more practical approach

Without partitioning and dynamic repartitioning, the developer must take these steps:

- Create separate flows for each data partition, based on the current hardware configuration.
- Write data to disk between processes.
- Manually repartition the data.
- Start the next process.

The application will be slower, disk use and management will increase, and the design will be much more complex. The dynamic repartitioning feature of IBM Information Server helps you overcome these issues.

Related concepts

“SOA components in IBM Information Server” on page 35

The run-time components that enable service-oriented architectures are contained in the run-time environment of the common services of IBM Information Server.

“WebSphere DataStage elements” on page 87

The central WebSphere DataStage elements are projects, jobs, stages, links, containers, and table definitions.

Scalability in IBM Information Server

IBM Information Server is built on a highly scalable software architecture that delivers high levels of throughput and performance.

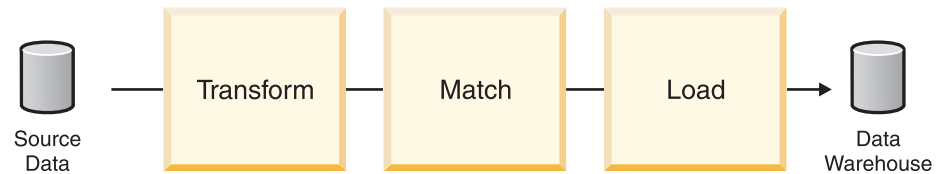
For maximum scalability, integration software must do more than run on Symmetric Multiprocessing (SMP) and Massively Parallel Processing (MPP) computer systems. If the data integration platform does not saturate all of the nodes of the MPP box or system in the cluster or grid, scalability cannot be maximized.

The IBM Information Server components fully exploit SMP, clustered, grid, and MPP environments to optimize the use of all available hardware resources.

For example, when you create a simple sequential data-flow by graph using the WebSphere DataStage and QualityStage Designer, you do not need to worry about the underlying hardware architecture or number of processors. A separate configuration file defines the resources (physical and logical partitions or nodes, memory, and disk) of the underlying multiprocessor computing system.

As Figure 7 on page 12 shows, the configuration provides a clean separation between creating the sequential data-flow graph and the parallel execution of the application. This separation simplifies the development of scalable data integration systems that run in parallel.

Application Assembly: One dataflow graph



Application Execution: Sequential or parallel

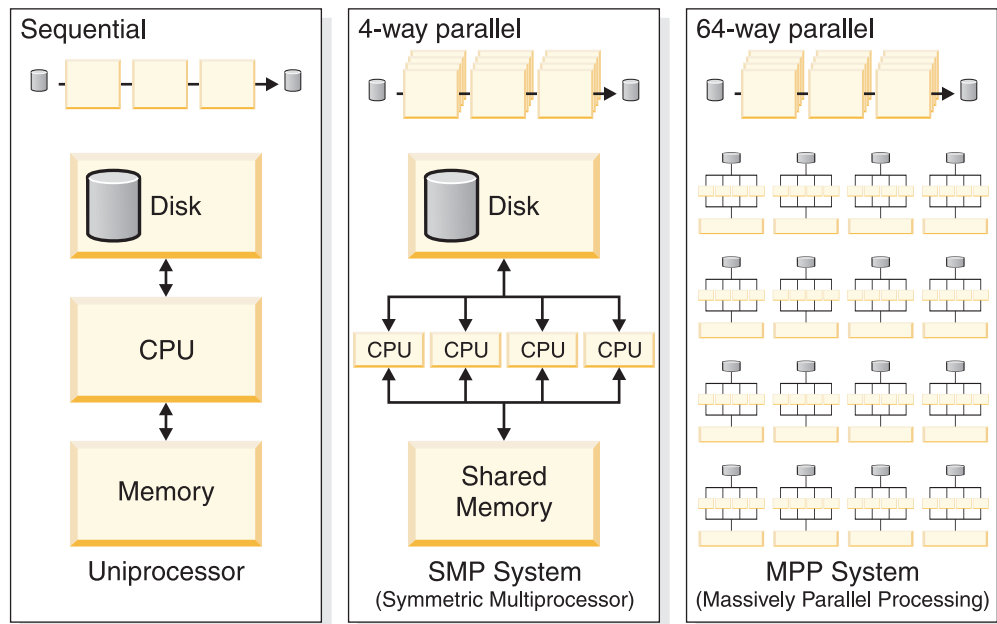


Figure 7. Hardware complexity made simple

Without support for scalable hardware environments the following problems can occur:

- Processing is slower, because hardware resources are not maximized.
- Application design and hardware configuration cannot be decoupled, and manual intervention and possibly redesign is required for every hardware change.
- Scaling on demand is not possible.

IBM Information Server leverages powerful parallel processing technology to ensure that large volumes of information can be processed quickly. This technology ensures that processing capacity does not inhibit project results and allows solutions to easily expand to new hardware and to fully leverage the processing power of all available hardware.

Support for grid computing in IBM Information Server

With hardware computing power a commodity, grid computing is a highly compelling option for large enterprises. Grid computing allows you to apply more processing power to a task than was previously possible.

Grid computing uses all of the low-cost computing resources, processors, and memory that are available on the network to create a single system image. Grid

computing software provides a list of available computing resources and a list of tasks. When a computer becomes available, the grid software assigns new tasks according to appropriate rules.

A grid can be made up of thousands of computers. Grid-computing software balances IT supply and demand by letting users specify processor and memory requirements for their jobs, and then find available machines on a network to meet those specifications.

The parallel processing architecture of IBM Information Server leverages the computing power of grid environments and greatly simplifies the development of scalable integration systems that run in parallel for grid environments.

IBM Information Server's pre-bundled grid edition provides rapid out-of-the-box implementation of grid scalability. It includes an integrated grid scheduler and integrated grid optimization. These capabilities help you easily and flexibly deploy integration logic across a grid without impacting job design, which provides unlimited scalability.

Shared services in IBM Information Server

IBM Information Server provides extensive administrative and reporting facilities that use shared services and a Web application that offers a common look and feel for all administrative and reporting tasks.

Administrative services in IBM Information Server

IBM Information Server provides administrative services to help you manage users, roles, sessions, security, logs, and schedules. The Web console provides global administration capabilities that are based on a common framework.

The IBM Information Server console provides these services:

- "Security services"
- "Log services" on page 14
- "Scheduling services" on page 15

Security services

Security services support role-based authentication of users, access-control services, and encryption that complies with many privacy and security regulations. As Figure 8 on page 14 shows, the console helps administrators add users, groups, and roles and lets administrators browse, create, delete, and update operations within IBM Information Server.

Directory services act as a central authority that can authenticate resources and manage identities and relationships among identities. You can base directories on IBM Information Server's own internal directory or on external directories that are based on LDAP, Microsoft's Active Directory, or UNIX.

Users only use one credential to access all the components of IBM Information Server. A set of credentials is stored for each user to provide single sign-on to the products registered with the domain.

The screenshot shows the 'Users' tab in the IBM Information Server Web console. The 'Create New User' form is displayed with the following fields and values:

Field	Value
User ID: *	Test
Password: *	••••
Confirm Password: *	••••
Title:	
First Name: *	Test
Last Name: *	User
IM Name:	
Job Title:	
Home Phone Number:	
Office Phone Number:	
Cell Phone Number:	
Pager Number:	
Fax Number:	
E-mail Address:	

The 'Roles' section on the right is divided into two categories:

- Suite Roles:**
 - ☐ Role
 - ☐ Suite Administrator
 - ☒ Suite User
- Product Roles:**
 - ☐ Role
 - ☐ Business Glossary Administrator
 - ☐ Business Glossary Author
 - ☐ Business Glossary User
 - ☐ DataStage Administrator
 - ☒ DataStage User

Figure 8. Adding a new user to a group

Log services

Log services help you manage logs across all of the IBM Information Server suite components. The console provides a central place to view logs and resolve problems. Logs are stored in the common repository, and each IBM Information Server suite component defines relevant logging categories.

You can configure which categories of logging messages are saved in the repository. Log views are saved queries that an administrator can create to help with common tasks. For example, you might want to display all of the errors in WebSphere DataStage jobs that ran in the past 24 hours.

Figure 9 on page 15 shows the IBM Information Server Web console being used to configure logging reports. Logging is organized by server components. The Web console displays default and active configurations for each component.

Select a Log View to Manage

▼ Create a new logging view

Name: *
Test Log

Log View Access:
Private ▼

Description: *
Demo

Log View Settings

▼ Format Message

Message:

▼ Severity Level

☒ All ☒ Error ☒ Info ☒ Trace
☒ Fatal ☒ Warn ☒ Debug

Categories

▼ Timestamp

☒ Range ☐ Real-Time Logging ☐ Interval

From: 2006-09-15 00:00:00 To: 2006-09-20 00:00:00

Event Count:

Refresh Rate (in sec.):

Last: Seconds ▼

Figure 9. Administrative console for setting up logs

Scheduling services

Scheduling services help plan and track activities such logging and reporting and suite component tasks such data monitoring and trending. Schedules are maintained using the IBM Information Server console, which helps you define schedules; view their status, history, and forecast; and purge them from the system.

Related concepts

Chapter 4, “Service-oriented integration,” on page 29

IBM Information Server simplifies the creation of shared data integration services by enabling integration logic to be used by any business process.

Reporting services in IBM Information Server

Reporting services manage run time and administrative aspects of reporting for IBM Information Server.

You can create product-specific reports for WebSphere DataStage, WebSphere QualityStage, and WebSphere Information Analyzer, and cross-product reports for logging, monitoring, scheduling, and security services.

All reporting tasks are set up and run from a single interface, the IBM Information Server Web console. You can retrieve and view reports and schedule reports to run at a specific time and frequency.

Figure 10 on page 16 shows the Web console.

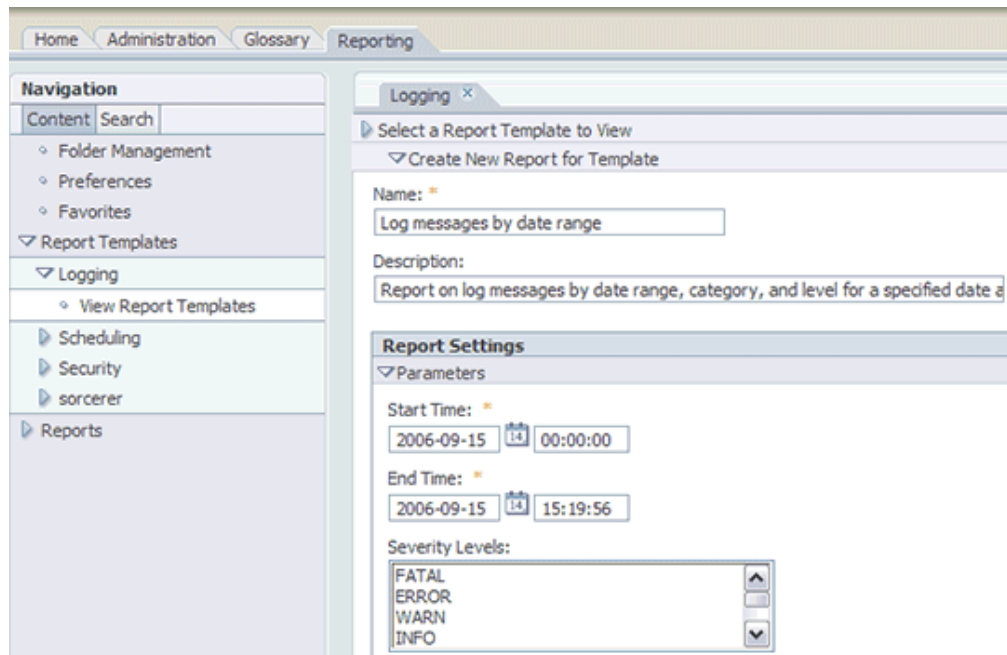


Figure 10. Creating a logging report by using the Web console

You define reports by choosing from a set of predefined parameters and templates. You can specify a history policy that determines how the report will be archived and when it expires. Reports can be formatted as HTML, PDF or Microsoft® Word documents.

Related concepts

Chapter 4, “Service-oriented integration,” on page 29

IBM Information Server simplifies the creation of shared data integration services by enabling integration logic to be used by any business process.

Chapter 3. Metadata services

When moving to an enterprise integration strategy, large organizations often face a proliferation of software tools that are built to solve identical problems. Few of these tools work together, much less work across problem domains to provide an integrated solution.

Data profiling, data modeling, data transformation, data quality, and business intelligence tools play a key role in data integration. Integration can become a mature, manageable process if these tools are enabled to work across problem domains.

The consequences of the inability to manage metadata are many and severe:

- Changes that are made to source systems are difficult to manage and cannot match the pace of business change.
- Data cannot be analyzed across departments and processes.
- Metadata cannot be shared among products without manually retyping the metadata.
- Without business-level definitions, metadata cannot provide context for information.
- Documentation is out-of-date or incomplete, hampering change management and making it harder to train new users.
- Efforts to establish an effective data stewardship program fail because of a lack of standardization and familiarity with the data.
- Establishing an audit trail for integration initiatives is virtually impossible.

The metadata services components of IBM Information Server create a fully integrated suite, eliminating the need to manually transport metadata between applications and provide a standalone metadata management application.

Metadata services introduction

Metadata services are part of the platform on which IBM Information Server is built. By using metadata services, you can access data and achieve data integration tasks such as analysis, modeling, cleansing, and transformation.

The major metadata services components of IBM Information Server are WebSphere Business Glossary, WebSphere Metadata Server, and WebSphere MetaBrokers and bridges

WebSphere Business Glossary

WebSphere Business Glossary is a Web-based application that provides a business-oriented view into the data integration environment. By using WebSphere Business Glossary, you can view and update business descriptions and access technical metadata.

Metadata is best managed by those who understand the meaning and importance of the information assets to the business. Designed for collaborative authoring,

WebSphere Business Glossary gives users the ability to share insights and experiences about data. It provides users with the following information about data resources:

- Business meaning and descriptions of data
- Stewardship of data and processes
- Standard business hierarchies
- Approved terms

WebSphere Business Glossary is organized and searchable according to the semantics that are defined by a controlled vocabulary, which you can create by using the Web console.

WebSphere Metadata Server

WebSphere Metadata Server provides a variety of services to other components of IBM Information Server:

- Metadata access
- Metadata integration
- Metadata import and export
- Impact analysis
- Search and query

WebSphere Metadata Server provides a common repository with facilities that are capable of sourcing, sharing, storing, and reconciling a comprehensive spectrum of metadata including business metadata and technical metadata.

Business metadata

Business metadata provides business context for information technology assets and adds business meaning to the artifacts that are created and managed by other IT applications. Business metadata includes controlled vocabularies, taxonomies, stewardship, examples, and business definitions.

Technical metadata

Technical metadata provides details about source and target systems, their table and field structures, attributes, derivations, and dependencies. Technical metadata also includes details about profiling, quality, and ETL processes, projects, and users.

WebSphere MetaBrokers and bridges

WebSphere MetaBrokers and bridges provide semantic model mapping technology that allows metadata to be shared among applications for all products that are used in the data integration lifecycle:

- Data modeling or case tools
- Business intelligence applications
- Data marts and data warehouses
- Enterprise applications
- Data integration tools

By using these components, you can establish common data definitions across business and IT functions.

- Drive consistency throughout the data integration lifecycle
- Deliver business-oriented and IT-oriented reporting

- Provide enterprise visibility for change management
- Easily extend to new and existing metadata sources

Scenarios for metadata management

A comprehensive metadata management capability provides users of IBM Information Server with a common way to deal with descriptive information surrounding the use of data. The following scenarios describe uses of this capability.

Web-based education: Profiling your customer

A Web-based, for-profit education provider needed to retain more students. Business managers needed to analyze the student lifecycle from application to graduation and direct recruiting efforts at individuals with the best chance of success.

To meet this business imperative, the company designed and delivered a business intelligence solution that uses a data warehouse that contains a single view of student information that is populated from operational systems. The IT organization uses WebSphere Metadata Server to coordinate metadata throughout the project. Other tools that were used included Embarcadero ER Studio for data modeling and Brio for Business Intelligence.

The overall project time was reduced by providing metadata consistency and accuracy across every tool. The business users now have trustworthy metadata about the information in their Brio reports. WebSphere Business Glossary provided business definitions to WebSphere Metadata Server. The net result is more confident decision-making about students and better student-retention initiatives.

Financial Services: Measuring levels of service

The data warehousing division of a major financial services provider needed to provide internal customers with critical enterprise-wide data about levels of service that are specified by signed service level agreements (SLAs). The data warehousing group also needed to provide business definitions of each field, including metrics that detailed actual versus promised levels of service.

The organization uses IBM Information Server to create an enterprise data warehouse and data marts to satisfy each SLA. The division used metadata services within WebSphere Information Analyzer, WebSphere QualityStage, and WebSphere DataStage to collaborate in a multiuser environment. The data warehousing group was also able to provide HTML reports that outlined the statistics that are associated with the loading of the data mart to satisfy the SLA.

The division met its service-level agreements and was able to demonstrate its compliance to internal data consumers. Additionally, end users received important business definitions from business intelligence reports.

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

Chapter 4, “Service-oriented integration,” on page 29

IBM Information Server simplifies the creation of shared data integration services by enabling integration logic to be used by any business process.

“WebSphere Business Glossary”

Managing business metadata effectively can ensure that the same data “language” applies throughout the organization. WebSphere Business Glossary gives business users the tools they need to author and own business metadata.

“WebSphere Metadata Server” on page 23

IBM Information Server can operate as a unified data integration platform because of the shared capabilities of WebSphere Metadata Server.

A closer look at metadata services in IBM Information Server

Metadata services encompass a wide range of functionality that forms the core infrastructure of IBM Information Server and also includes some separately packaged capabilities.

WebSphere Business Glossary

Managing business metadata effectively can ensure that the same data “language” applies throughout the organization. WebSphere Business Glossary gives business users the tools they need to author and own business metadata.

For example, one department refers to “revenues,” another to “sales.” Are they talking about the same activity? One subsidiary unit talks about “customers,” another about “users” or “clients.” Are these different classifications or different terms for the same classification?

WebSphere Business Glossary provides business users with a Web-based tool for creating and managing standard definitions of business concepts, called a *controlled vocabulary*. It also simplifies the building of a business-oriented classification system and the collaborative authoring of business metadata.

The tool simplifies the task of managing, browsing, and customizing the broad variety of metadata that is stored in the repository of WebSphere Metadata Server, metadata that includes details about tables, columns, models, schemas, operations, and other components of the data integration process.

The tool divides metadata into categories, each of which contains terms. You can use terms to classify other objects in the metadata repository based on the needs of your business. You can also designate users or groups as stewards for metadata objects.

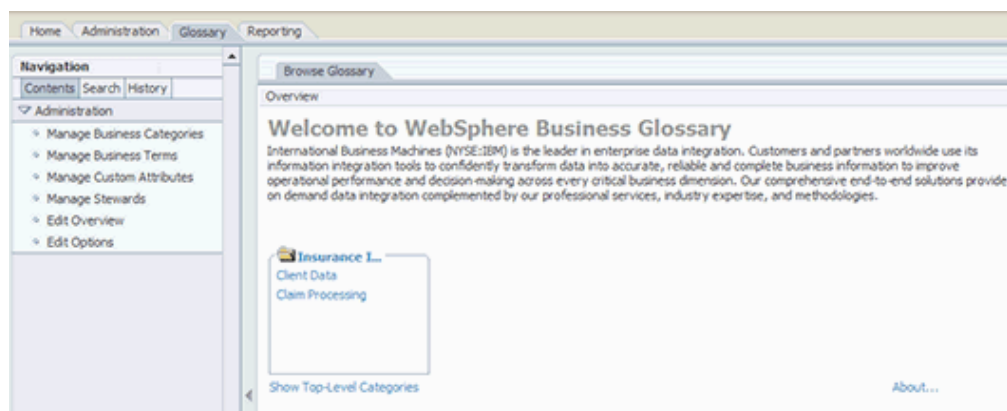


Figure 11. WebSphere Business Glossary user interface

WebSphere Business Glossary helps business users with the following tasks:

Developing a common vocabulary between business and technology

A common vocabulary allows multiple users of data to share a common view of the meaning of data. Users can assign categories and terms to data that are meaningful in a business context, and create a hierarchy of categories for ease of browsing.

Providing data governance and stewardship

Data assurance programs assign responsibility to business users (data stewards) for the management of data through its lifecycle.

Finding business information that is derived from metadata

Metadata helps business users to understand the meaning of the data, its currency, its lineage, and who is responsible for defining and producing the data. If a business user wants to know the definition of a term such as “corporate price,” the glossary will provide this insight.

Accessing metadata without complicated tooling and querying

Metadata objects can be arranged in a hierarchical fashion to simplify browsing of the data objects.

Providing collaborative enrichment of business metadata

Maintenance of business metadata is an ongoing process in which automated and manual data inputs evolve. Multiple business users can collaborate to add notes, annotations, categories, and synonyms to enrich business metadata.

For example, multiple systems may maintain tables of customer information, however the business may uncover a requirement for the concept of “high-value” customers. The business needs a way to define what a high value customer is, and how to recognize them (for example, a high-value customer is a customer with combined account balances over \$10,000). WebSphere Business Glossary provides a tool for recording these definitions, and relating business concepts together into taxonomies. This records the business requirements in the same metadata foundation that the profiling and analysis process uses.

WebSphere Business Glossary tasks

Major tasks in WebSphere Business Glossary include creating categories and terms, browsing and searching, enabling data stewardship, and annotating data for collaboration.

WebSphere Business Glossary is a browser-based application that you access by using Microsoft Internet Explorer.

Enabling data stewardship

Data stewardship is the management of data throughout its lifecycle. Stewardship includes making the data available to all those who are authorized to access it. It also includes the efficient management and integration with related data. Perhaps most importantly, stewardship includes the responsibility to ensure that data is properly defined, and that all users of the data clearly understand its meaning.

WebSphere Business Glossary supports the concept of data stewardship and helps you set and retrieve stewardship information for all data assets.

Administrators can designate a user or group as a steward. Administrators and authors can then specify that the steward is responsible for one or more metadata

objects. When you view the browse page for an object that has a steward, you can link to contact information for the steward.

Creating categories and terms

Although you can use several methods to find metadata in WebSphere Metadata Server, business users often find searching data by category is the best strategy. Data must be organized into meaningful taxonomies to aid the navigation of a business glossary by category.

Figure 12 shows the Create Category function in WebSphere Business Glossary. You create a business classification system or taxonomy that acts as the hierarchical browsing structure of the glossary Web site. You can also import structure from other tools or spreadsheets.

The screenshot shows the 'New Business Category (Properties)' form in the WebSphere Business Glossary. The interface includes a top navigation bar with 'Home', 'Administration', 'Glossary', and 'Reporting'. A left sidebar contains a 'Navigation' menu with options like 'Contents', 'Search', 'History', and 'Administration'. The main content area is titled 'Browse Glossary' and 'Manage Business Categories'. It features a 'Select a Business Category to Manage' dropdown and a 'New Business Category (Properties)' form. The form has several fields: 'Name' (set to 'Customer Information'), 'Parent Category' (set to 'USA Client Data' with 'Select' and 'Clear' buttons), 'Steward' (set to 'None' with a 'Clear' button), 'Add to Overview Page' (checkbox), 'Short Description' (set to 'Customer information'), and 'Long Description' (set to 'Category containing business terms that describe different customer objects'). At the bottom right, there are 'Cancel', 'Save', and 'Save and Close' buttons.

Figure 12. Creating a new category

A term is a word or phrase that can be used to classify and group objects in the metadata repository. For example, you might use the term “South America Sales” to classify some of the tables and columns in the metadata repository, and the term “Asian Sales” to classify other tables and columns.

When you create or edit a term, you can specify properties and relationships among terms, including synonyms and related terms. You can also specify parent categories to group similar terms and can designate stewards who have the responsibility for maintaining terms. Custom attributes enable administrators to define any number of new attributes to be applied to terms, categories, or both.

Annotating data for collaboration

While data stewards are responsible for specific types of data, creating a business glossary is a collaborative effort that requires subject matter experts from different parts of the enterprise. WebSphere Business Glossary provides tools for subject matter experts and others to annotate existing data definitions, edit descriptions, and assign data object to categories.

These annotations, or notes, help business users share insights about the information assets of the enterprise. For example, an analyst might discover that a database column for customer information also contains shipping information that does not belong in the column. The analyst could share that information by using the Notes® feature. Notes help you capture ideas in the form of unstructured metadata. This information might otherwise be unknown to a large portion of the enterprise.

Browsing the Business Glossary

You can start browsing the glossary structure from the Overview page, which displays the top-level categories that the glossary administrator has designated as most important for navigation in the metadata repository.

The browse by category function enables data stewards to find descriptions related to type of data even though they may not know the exact name of the data items in question.

When you select an object, its browse page is displayed on the Browse Glossary tab, which lists the object's name, class, steward and other important properties. You can inspect its attributes, browse its relationships to other objects, and send feedback to the administrator. Administrators and authors can add and edit notes about the object.

WebSphere Metadata Server

IBM Information Server can operate as a unified data integration platform because of the shared capabilities of WebSphere Metadata Server.

Common repository

By storing all metadata in a shared repository, IBM Information Server enables metadata to be shared actively across all tools. The repository provides services for two types of data:

- Design metadata, which is created as a part of the development process and can be configured to be either private or shared by a team of users.
- Operational metadata, which is created from ongoing integration activity. This metadata is message-oriented and time-stamped to help track the sequence of events.

With a shared repository, changes that are made in one part of IBM Information Server will be automatically and instantly visible throughout the suite. The repository offers the following key features:

Active integration

Application artifacts are dynamically integrated across tools.

Multiuser development

Teams can collaborate in a shared workspace.

The common repository is an IBM WebSphere J2EE application. The repository uses standard relational database technology (such as DB2 or Oracle) for persistence. These databases provide backup, administration, scalability, transactions, and concurrent access.

Common model

Metadata for data integration projects comes from both IBM Information Server products and vendor products. The repository uses metadata models (metamodels) to describe the metadata from these sources. Metadata models provide a means for others to understand and share metadata between applications.

The common model is the foundation of IBM Information Server. Metadata elements that are common to all metadata sources are discovered and represented once, in a form and format that is accessible to all of the tools. The common model enables sharing and reuse of artifacts across IBM Information Server.

Shared metadata services

WebSphere Metadata Server exposes a set of metadata manipulation and analysis services for use across IBM Information Server components. These services enable metadata interchange, integration, management, and analysis. They eliminate the need for a standalone metadata management product or repository product by actively managing metadata in the background, and by providing metadata functionality in the context of your normal daily activities.

For example:

- A WebSphere DataStage user wants to understand the dependencies between stages in an ETL job. By using metadata services, she can perform an impact analysis from the Designer client canvas, never needing to leave the application for another interface.
- A data analyst who is working with WebSphere Information Analyzer can add business terms, definitions, and notes to data under analysis for use by a data modeler or architect.
- A WebSphere QualityStage user needs to better understand the business semantics that are associated with a data domain. By using metadata services, he can access the business description of the domain and any annotations that were added by business users.
- A WebSphere DataStage component developer wants to find a function that performs a particular data conversion. By using metadata services, she can perform an advanced search for the function.

WebSphere Metadata Server offers the following key metadata services:

- Metadata interchange
- Impact analysis
- Integrated find

Metadata interchange

WebSphere MetaBroker[®] and bridges enable you to access and share metadata with the best-of-class tools for modeling, data profiling, data quality, ETL, OLAP, and business intelligence.

Figure 13 on page 25 shows how MetaBrokers work. MetaBrokers convert metadata from one format to another by mapping the elements to a standard model called the *hub model*. The selected metadata is then imported and stored in the repository. The metadata exchange enables decomposition and recomposition of metadata into simple units of meaning.

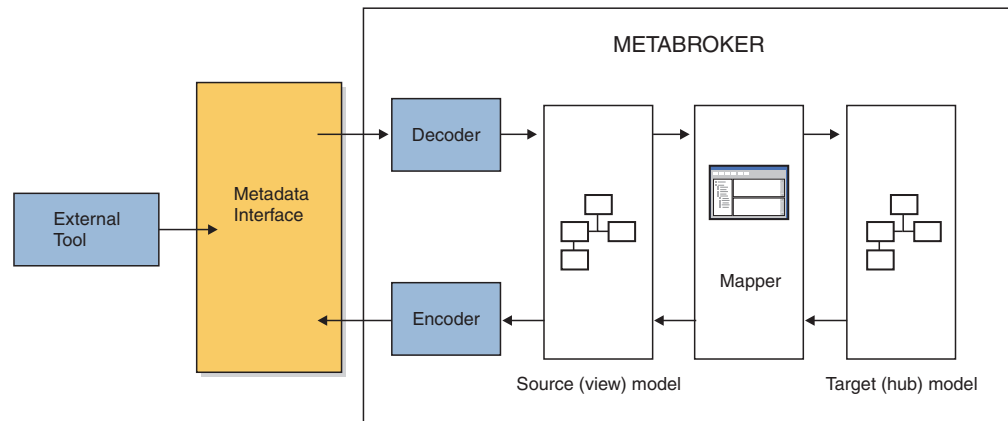


Figure 13. MetaBrokers convert metadata to hub model

IBM Information Server now supports more than 20 MetaBrokers and bridges to various technologies and partner products. You can use most MetaBrokers to import metadata from a particular tool, file, or database into the metadata repository of WebSphere Metadata Server.

Table 1 describes MetaBroker types and the different types of metadata that you can access.

Table 1. MetaBroker types

Type of MetaBroker	Type of metadata
Design tool	CA ERwin, Oracle Designer, Rational® Data Architect and the Unified Modeling Language (UML)
OLAP and business intelligence	Cognos PowerPlay, IBM Cube Views™, ReportNet, Business Objects, and Hyperion
Operational metadata	Metadata that describes operational events such as the time and date of integration process runs.

Impact analysis

Impact analysis helps you manage the effects of changes to data by showing dependencies among objects. This type of analysis extends across multiple tools, helping you assess the cost of change. For example, a developer can predict the effects of a change to a table definition or business logic.

Figure 14 on page 26 shows the WebSphere DataStage and QualityStage Designer being used to select a table definition called ProdDim from the metadata repository to show where used dependencies.

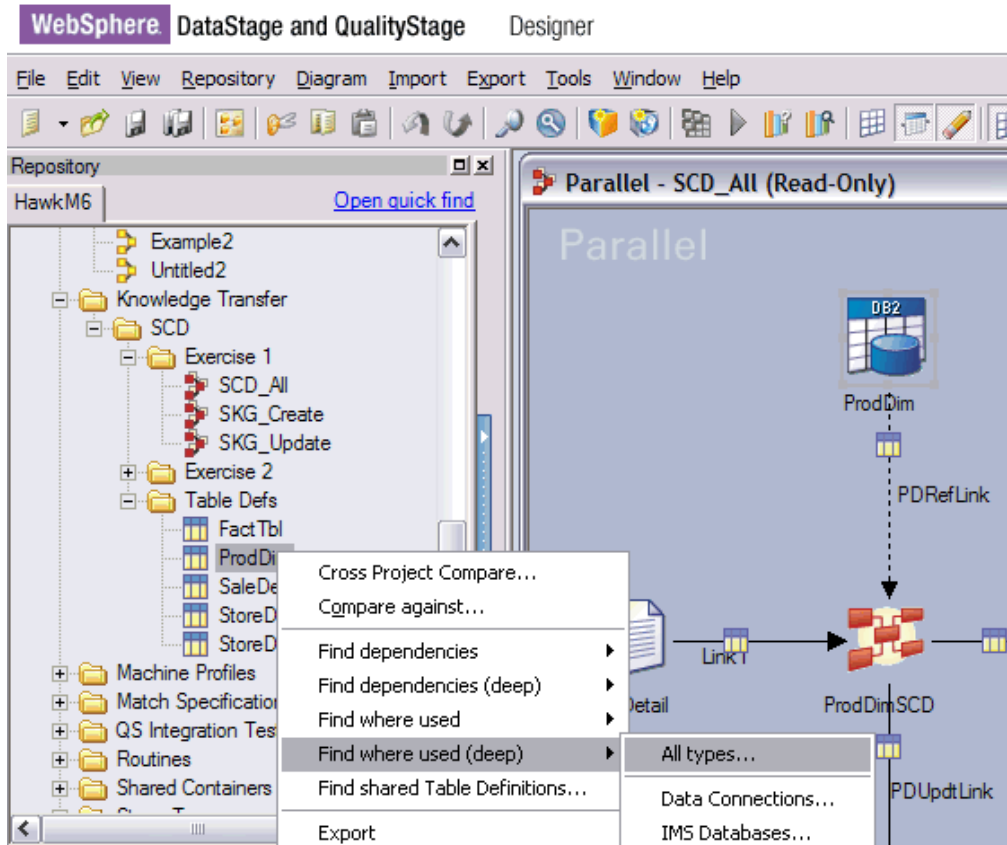


Figure 14. Using Find to show dependencies for a table definition in the repository

The Impact Analysis Path Viewer presents a graphical view of these relationships, as Figure 15 on page 27 shows.

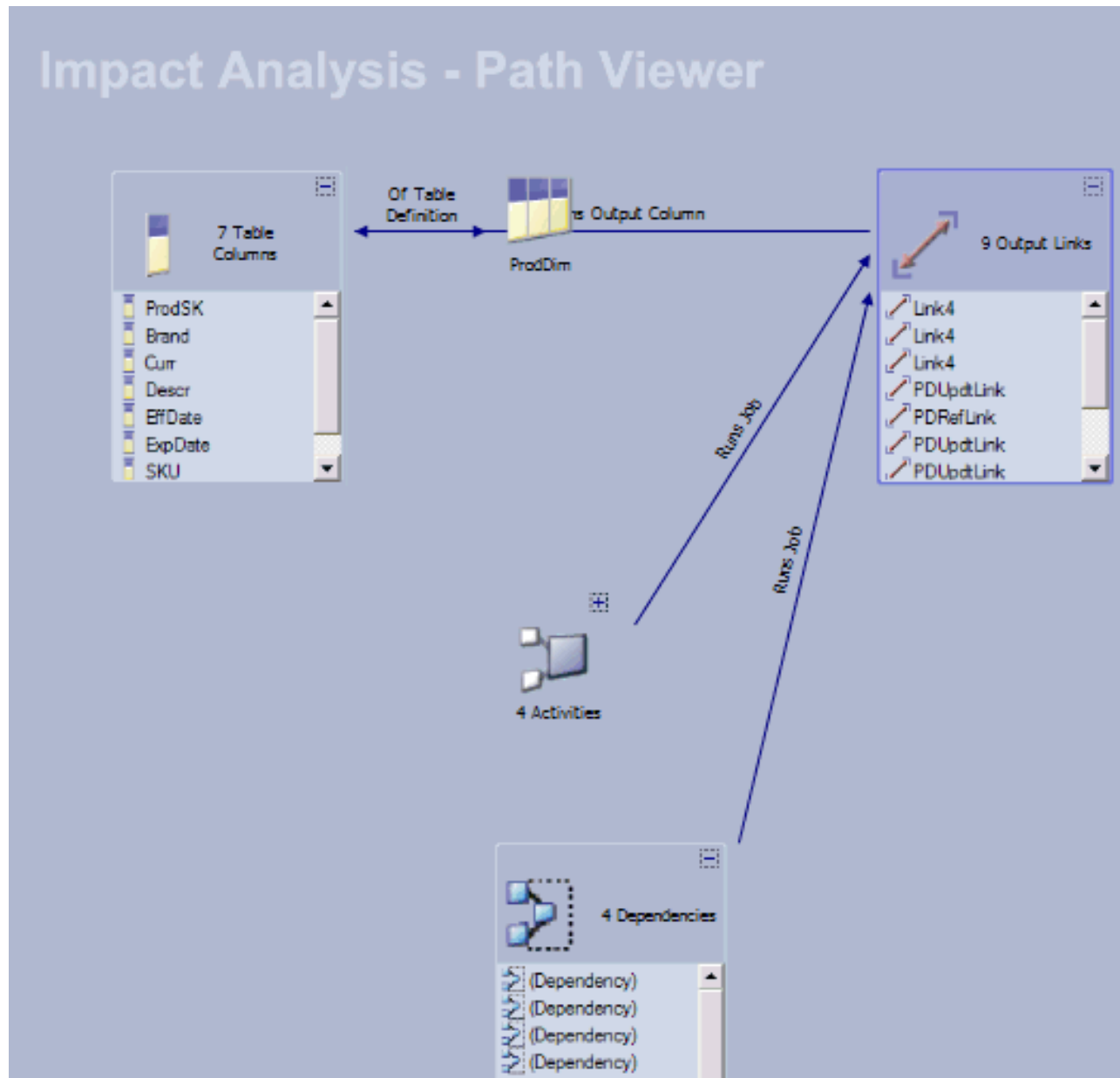


Figure 15. Impact Analysis Path Viewer

The dependencies can also be shown in a textual view. You can also run an impact analysis report that can be viewed from the Web console.

Integrated find

Metadata services help you locate and retrieve objects from the repository by using either the quick find feature or the advanced find feature. The quick find feature locates an object based on a full or partial name or description. The advanced find feature locates objects based on the following attributes:

- Type
- Creation data
- Last modified
- Where it is used
- Depends upon

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

Information resources for metadata services

A variety of information resources can help you get started with IBM Information Server’s metadata services.

WebSphere Business Glossary

The Getting Started pane that appears when you click the **Glossary** tab of the IBM Information Server console describes the purpose of the tab and how to get started. Each pane and tab on the console displays a line of context-sensitive instructional text.

The **Help** button links to online documentation for WebSphere Business Glossary in the IBM Information Server information center at <http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp>. The Information Center also provides all planning, installation, and configuration details for IBM Information Server and its suite components.

The *WebSphere Business Glossary Guide* PDF is also available on the Quick Start CD.

WebSphere MetaBrokers

Online help is available for all WebSphere MetaBrokers and bridges. The *Information Server Guide to WebSphere MetaBrokers and Bridges* PDF is also available on the Quick Start CD.

IBM Information Server and suite components

Planning, installation, and configuration details are also available in the following PDFs on the Quick Start CD:

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Chapter 4. Service-oriented integration

IBM Information Server simplifies the creation of shared data integration services by enabling integration logic to be used by any business process.

Invoking service-ready data integration tasks ensures that business processes such as quote generation, order entries, and procurement requests receive data that is correctly transformed, standardized, and matched across applications, partners, suppliers, and customers.

Many organizations are designing their next generation of infrastructure and applications as services. Implementing a service-oriented architecture (SOA) offers these benefits:

Adaptability

Functional components can be reassembled quickly and in new ways.

Consistency

Core rules for handling data and processes are reused across projects.

Reduced cost

Increased reuse and a single point of maintenance speed time to value and reduce development expense.

Federated ownership

Each service is owned and maintained independently by its own group.

IBM Information Server provides an SOA infrastructure that provides these capabilities by helping you create shared data integration services. A common services layer manages how services are deployed from any of the suite components. Cleansing and transformation rules or federated queries can be published as shared services by using a consistent and intuitive graphical interface, and managed after publication using the same interface.

Introduction to service-oriented integration in IBM Information Server

IBM Information Server provides standard service-oriented interfaces for enterprise data integration. The built-in integration logic of IBM Information Server can easily be encapsulated as service objects that are embedded in user applications.

These service objects have the following characteristics:

Always on

The services are always running, waiting for requests. This ability removes the overhead of batch startup and shutdown and enables services to respond instantaneously to requests.

Scalable

The services distribute request processing and stop and start jobs across multiple WebSphere DataStage servers, enabling high performance with large, unpredictable volumes of requests.

Standards-based

The services are based on open standards and can easily be invoked by standards-based technologies including enterprise application integration (EAI) and enterprise service bus (ESB) platforms, applications, and portals.

Flexible

You can invoke the services by using multiple mechanisms (bindings) and choose from many options for using the services.

Manageable

Monitoring services coordinate timely reporting of system performance data.

Reliable and highly available

If any WebSphere DataStage server becomes unavailable, it routes service requests to a different server in the pool.

Reusable

The services publish their own metadata, enabling them to be found and called across any network.

High performance

Load balancing and the underlying parallel processing capabilities of IBM Information Server create high performance for any type of data payload.

A data integration service is created by designing the data integration process logic in IBM Information Server and publishing it as a service. These services can then be accessed by external projects and technologies.

WebSphere Information Services Director provides a foundation for information services by allowing you to leverage the other components of IBM Information Server for understanding, cleansing, and transforming information and deploying those integration tasks as consistent and reusable information services.

As Figure 16 shows, service-ready data integration jobs can be used with process-centric technologies such as EAI, business Process Management (BPM), ESB, and application servers.

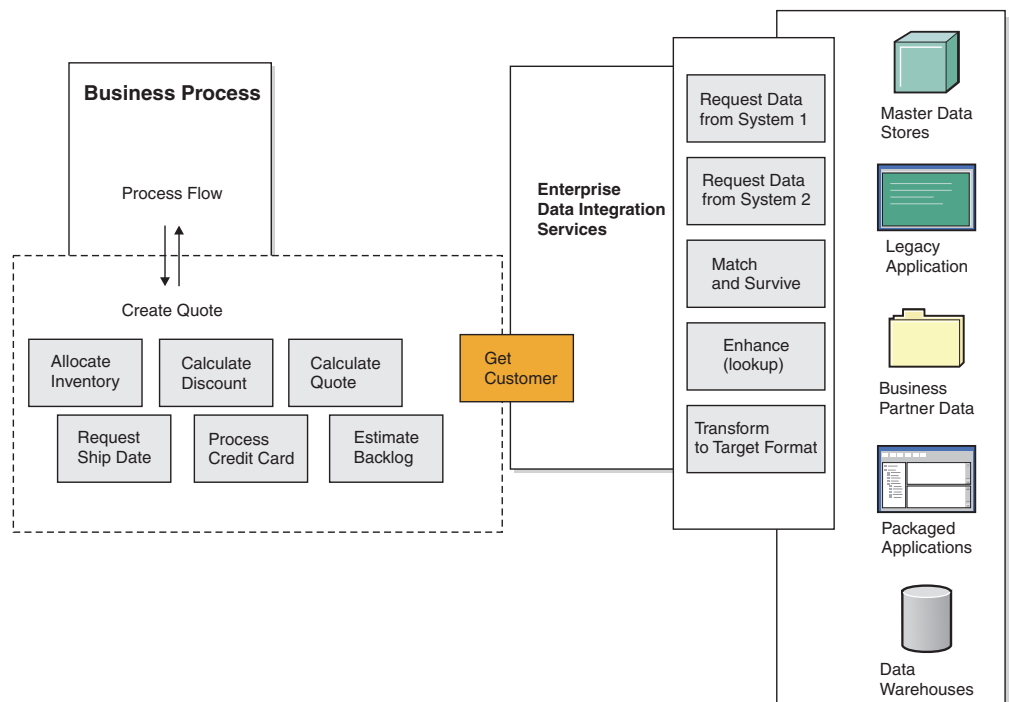


Figure 16. Service-ready integration tasks work with business processes

Scenarios for service-oriented integration

The following examples show how organizations have used service-oriented architectures in IBM Information Server.

Pharmaceutical industry: Improving efficiency

A leading pharmaceutical company needed to include real-time data from clinical labs in its research and development reports. The company used WebSphere DataStage to define a transformation process for XML documents from labs. This process used SOA to expose the transformation as a Web service, allowing labs to send data and receive an immediate response.

Pre-clinical data is now available to scientific personnel earlier, allowing lab scientists to select which data to analyze. Now, only the best data is chosen, greatly improving scientists' efficiency.

Insurance: Validating addresses in real time

An international insurance data services company employs IBM Information Server to validate and enrich property addresses by using Web services. As insurance companies submit lists of addresses for underwriting, services standardize the addresses based on their rules, validate each address, match the addresses to a list of known addresses, and enrich the addresses with additional information that helps with underwriting decisions. The company now automates 80 percent of the process and eliminated most of the errors. The project was simplified by using the SOA capabilities of IBM Information Server and the standardization and matching capabilities of WebSphere QualityStage.

Where SOA fits in a business context

By enabling integration tasks as services, IBM Information Server becomes a critical component of the application development and integration environment. The SOA infrastructure ensures that data integration logic that is developed in IBM Information Server can be used by any business process.

SOA allows you to use both analytical and operational data. The best data is available at all times, to all people and to all processes. The following categories represent common uses of SOA in a business context:

Real-time data warehousing

Enables companies to publish their existing data integration logic as services that can be called in real time from any process. This type of warehousing enables users to perform analytical processing and loading of data based on transaction triggers, ensuring that time-sensitive data in the warehouse is completely current.

Matching services

Enables data integration logic to be packaged as a shared service that can be called by enterprise application integration platforms. This method allows reference data (such as customer, inventory, and product data) to be matched to and kept current with a master store with each transaction.

In-flight transformation

Enables enrichment logic to be packaged as shared services so that capabilities such as product name standardization, address validation, or data format transformations can be shared and reused across projects.

Enterprise data services

Enables the data access functions of many applications to be aggregated and shared in a common service layer. Instead of each application creating its own access code, these services can be reused across projects, simplifying development and ensuring a higher level of consistency.

As Figure 17 shows, one of the major advantages of using an SOA approach is that you can combine data integration tasks with the leading enterprise messaging, Enterprise Application Integration (EAI), and Business Process Management (BPM) products by using binding choices.

Since most middleware products support Web services, there are often multiple options for how this is done. For example, WebSphere integration products such as WebSphere Federation Server or WebSphere Business Integration Message Broker can invoke IBM Information Server services to access service-ready jobs.

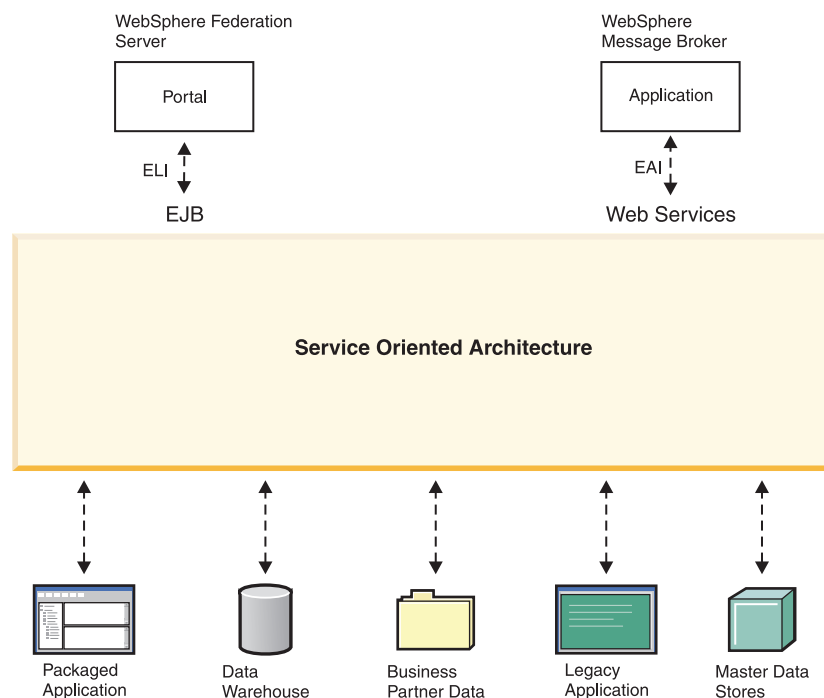


Figure 17. IBM Information Server used with WebSphere products

Related concepts

Chapter 1, "Introduction," on page 1

Most of today's critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

A closer look at service-oriented integration in IBM Information Server

IBM Information Server provides a SOA infrastructure that uses data transformation processes that are created from new or existing WebSphere DataStage or WebSphere QualityStage jobs or federated queries that are created by WebSphere Federation Server and exposes them as a set of services and operations.

After an integration services is enabled, any enterprise application, .Net or Java™ developer, Microsoft Office or integration software can invoke the service by using a binding protocol such as Web services.

The following features are central to the IBM Information Server SOA infrastructure:

Common administrative services

Host and publish service metadata, expose a choice of bindings for each service, and provide infrastructure services such as security management, session management, logging, and monitoring.

Foundation components for development

Provide a single set of data transformation rules for analytical and enterprise applications, business activity monitoring, federated data access, and business process integration.

Any-to-any connectivity

Provides technology independence for data transformation, standardization, matching, and legacy data access by using Web services (.NET and Java) or Enterprise JavaBeans™ (EJB) interface bindings.

Service-ready integration

A service-ready data integration job accepts requests from client applications, mapping request data to input rows and passing them to the underlying jobs. A job instance can include database lookups, transformations, data standardization and matching, and other data integration tasks that are supplied by IBM Information Server.

Figure 18 shows a WebSphere DataStage job with a service input and service output.



Figure 18. Service-ready job

The design of a real-time job determines whether it is always running or runs once to completion. All jobs that are exposed as services process requests on a 24-hour basis. The SOA infrastructure supports three job topologies for different load and work style requirements:

Batch jobs

Topology I uses new or existing batch jobs that are exposed as services. A batch job starts on demand. Each service request starts one instance of the job that runs to completion. This job typically initiates a batch process from a real-time process that does not need direct feedback on the results. This topology is tailored for processing bulk data sets and is capable of accepting job parameters as input arguments.

Batch jobs with a Service Output stage

Topology II uses an existing batch job and adds an output stage. The Service Output stage is the exit point from the job, returning one or more rows to the client application as a service response. As Figure 19 on page 34

34 shows, these jobs typically initiate a batch process from a real-time process that requires feedback or data from the results. This topology is designed to process large data sets and can accept job parameters as input arguments.

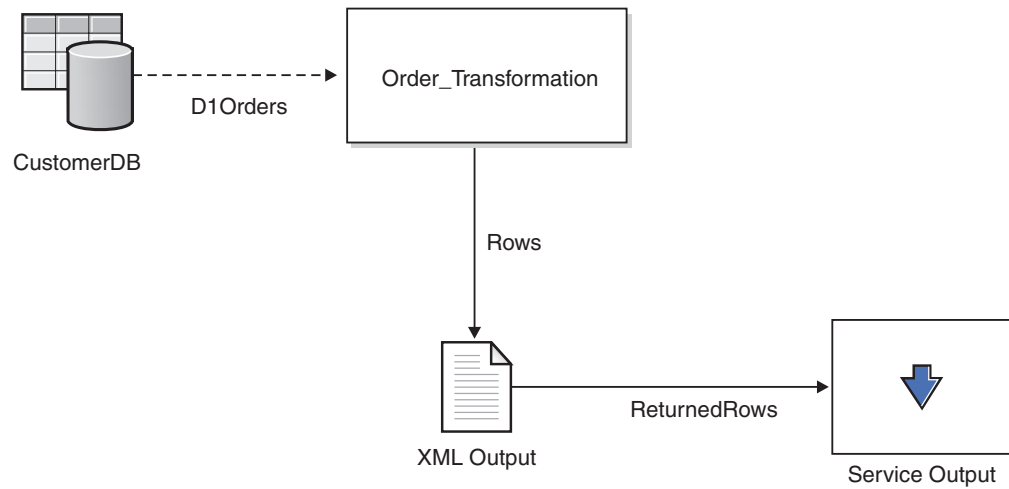


Figure 19. Batch jobs with a Service Output stage

Jobs with a Service Input stage and Service Output stage

In Topology III, jobs use both a Service Input stage and a Service Output stage. The Service Input stage is the entry point to a job, accepting one or more rows during a service request. These jobs are always running. This topology is typically used to process high volumes of smaller transactions where response time is important. It is tailored to process many small requests rather than a few large requests. Figure 20 on page 35 shows an example of this topology.

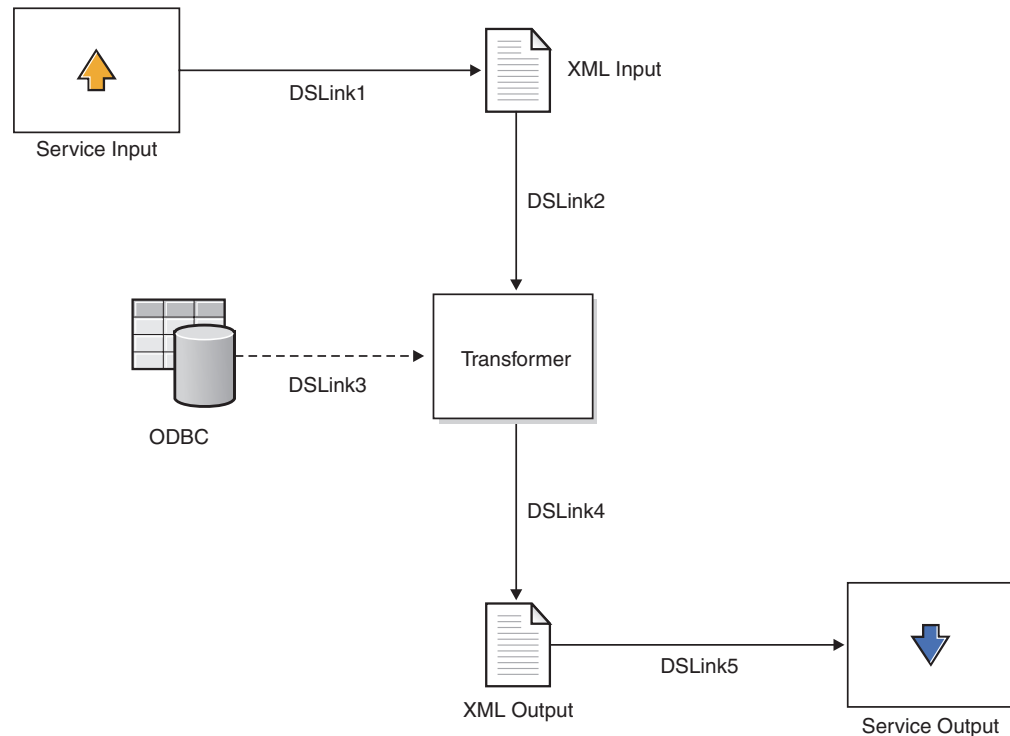


Figure 20. A more complex job with Service Input stage and Service Output stage

Related concepts

“A closer look at WebSphere Federation Server” on page 115

The components of WebSphere Federation Server include the federated server and database, wrappers, nicknames, and other federated objects. Capabilities of WebSphere Federation Server that provide performance and flexibility for integration projects include compensation, the query optimizer, and two-phase commit.

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

SOA components in IBM Information Server

The run-time components that enable service-oriented architectures are contained in the run-time environment of the common services of IBM Information Server.

These components are J2EE applications that distribute requests to WebSphere DataStage, WebSphere QualityStage, or WebSphere Federation Server based on load-balancing algorithms. Common core services include security and logging.

Threshold-balanced parallelism

The run-time environment combines parallel processing with load balancing and distribution to provide high performance data processing. It balances service requests by routing them to WebSphere Federation Server or WebSphere DataStage servers, each of which takes advantage of pipeline technology for parallel execution.

Threshold-balanced parallelism enables SOA platforms to automatically adjust resources based on thresholds that you set when you define services. The common services start and stop jobs in response to load conditions. The combination of these capabilities with parallel pipelining is unique to IBM Information Server and enables IBM Information Server to process data integration tasks faster than any other technology.

Multiple binding support

Virtually any protocol can be made to adhere to SOA principles. IBM Information Server supports this approach, enabling the same service to support multiple protocol bindings, all defined within the WSDL file.

An SOA interface should be able to handle multiple mechanisms (bindings) for calling services. This improves the utility of services and therefore increases the likelihood of reuse and adoption across the enterprise.

Projects for which Web services are not a viable option because of performance or architectural requirements can still leverage the services by using an interface better suited to their requirements. WebSphere Information Services Director can publish the same service using different bindings:

Simple Object Access Protocol (SOAP) over HTTP (Web services)

Any application that complies with XML Web services can invoke a WebSphere Federation Server or WebSphere DataStage integration process as a Web service. These Web services support the generation of literal document-style and SOAP encoded RPC-style Web services.

Enterprise JavaBeans (EJB)

For Java-centric development, WebSphere Information Services Director can generate a J2EE-compliant EJB (stateless session bean) where each data transformation service is instantiated as a separate synchronous EJB method call.

The design does not depend on the binding choice. As logic is built in WebSphere DataStage and WebSphere QualityStage, the designer does not need to be aware of how it will be used. After the service is deployed, additional bindings can easily be implemented without changing the logic.

WebSphere Information Services Director tasks

WebSphere Information Services Director provides an integrated environment for designing services that enables you to rapidly deploy integration logic as services without assuming extensive development skills.

With a simple, wizard-driven interface, in a few minutes you can attach a specific binding and deploy a reusable integration service. WebSphere Information Services Director also provides these features:

- Load-balancing and administrator services for cataloging and registering services
- Shared reporting and security services
- A metadata services layer that promotes reuse of the information services by actually defining what the service does and what information it delivers.

Information providers

An *information provider* is both the server that contains units that you can expose as services and the units themselves, such as WebSphere DataStage and WebSphere QualityStage jobs or federated SQL queries.

Each information provider must be enabled. To enable these providers, you use WebSphere Information Services Director.

You use the Add Information Provider window to enable information providers that you installed outside of IBM Information Server, such as WebSphere DataStage servers or federated servers.

Creating a project

A project is a collaborative environment that you use to design applications, services, and operations.

All project information that you create by using WebSphere Information Services Director is saved in the common metadata repository so that it can easily be shared among other IBM Information Server components.

You can export a project to back up your work or share work with other IBM Information Server users. The export file includes applications, services, operations, and binding information.

Creating an application

An application is a container for a set of services and operations. An application contains one or more services that you want to deploy together as an Enterprise Archive (EAR) file on an application server.

All design-time activity occurs in the context of applications:

- Creating services and operations
- Describing how message payloads and transport protocols are used to expose a service
- Attaching a reference provider, such as a WebSphere DataStage job or an SQL query, to an operation

Creating an application is a simple task from the Develop navigator menu of the IBM Information Server console. You can also export services from an application before it is deployed and import the services into another application.

You can change the default settings for operational properties when you create an application or later, as Figure 21 on page 38 shows.

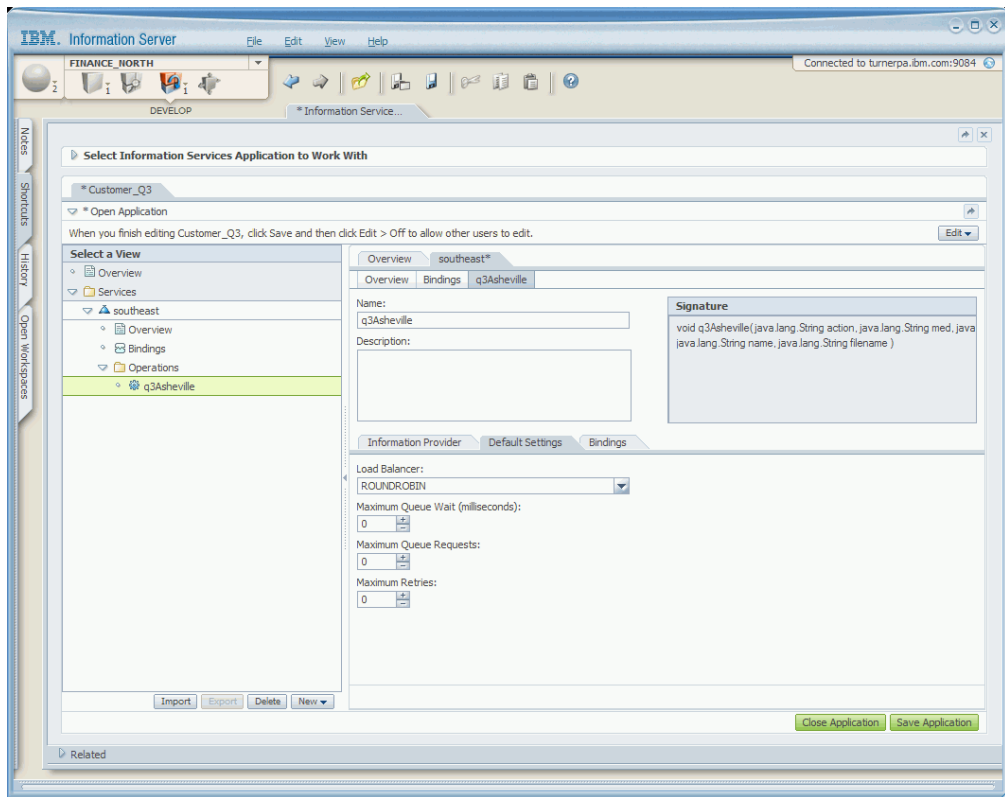


Figure 21. Setting operational properties for an application

Creating a service

An information service exposes results from processing by information providers such as WebSphere DataStage servers and federated servers. A deployed service runs on an application server and processes requests from service client applications.

An information service is a collection of operations that are selected from jobs, maps, federated queries, or other information providers. You can group operations in the same information service or design them in separate services.

You create an information service for a set of operations that you want to deploy together. You select a project and an application within the project in the Select a View area, as Figure 22 on page 39 shows.

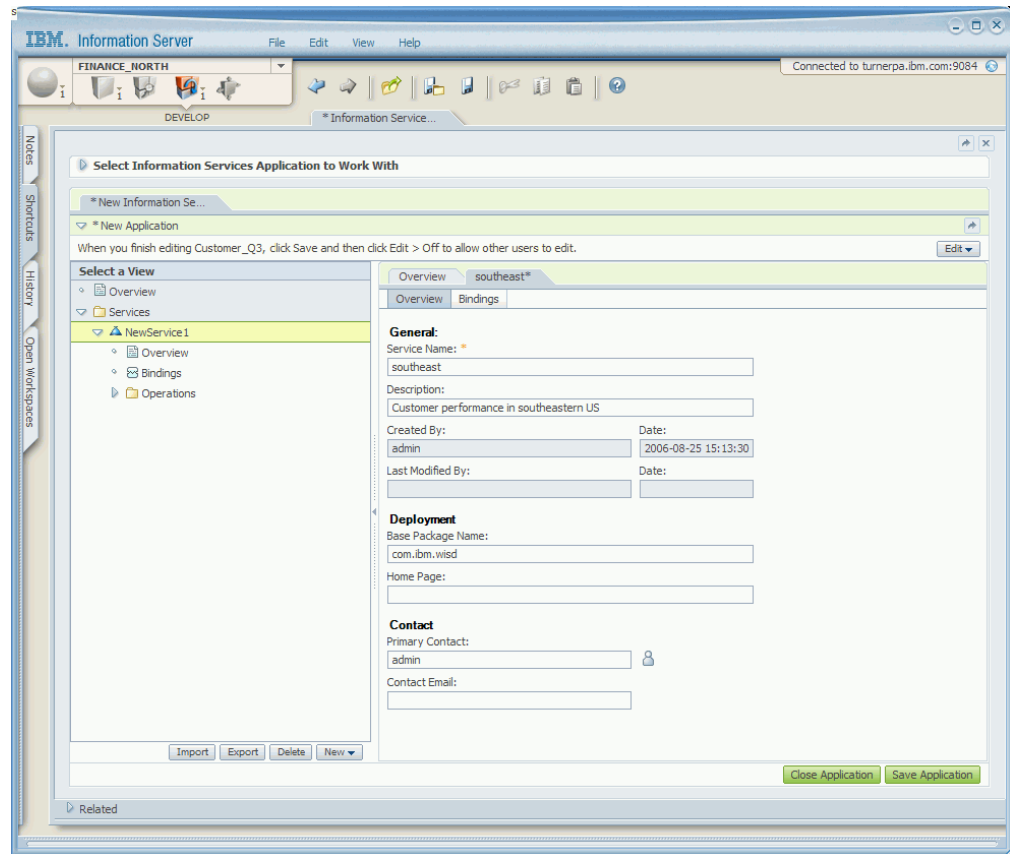


Figure 22. Identifying a service for a new application

When you create a service, you specify such options as name, base package name for the classes that are generated during the deployment of the application, and optionally the home Web page and contact information for the service.

After you create the service, you attach a binding for the service:

Simple Object Access Protocol (SOAP) over HTTP

To expose an information service as a Web service, attach the SOAP over HTTP binding to the information service.

Enterprise JavaBeans (EJB) interface

If your service consumers want to access an information service through an EJB interface, attach the EJB binding to the information service.

Deploying applications and their services

You deploy an application on WebSphere Application Server to enable the information services that are contained in the application to receive service requests.

The Deploy Application window in WebSphere Information Services Director guides you through the process, as Figure 23 on page 40 shows.

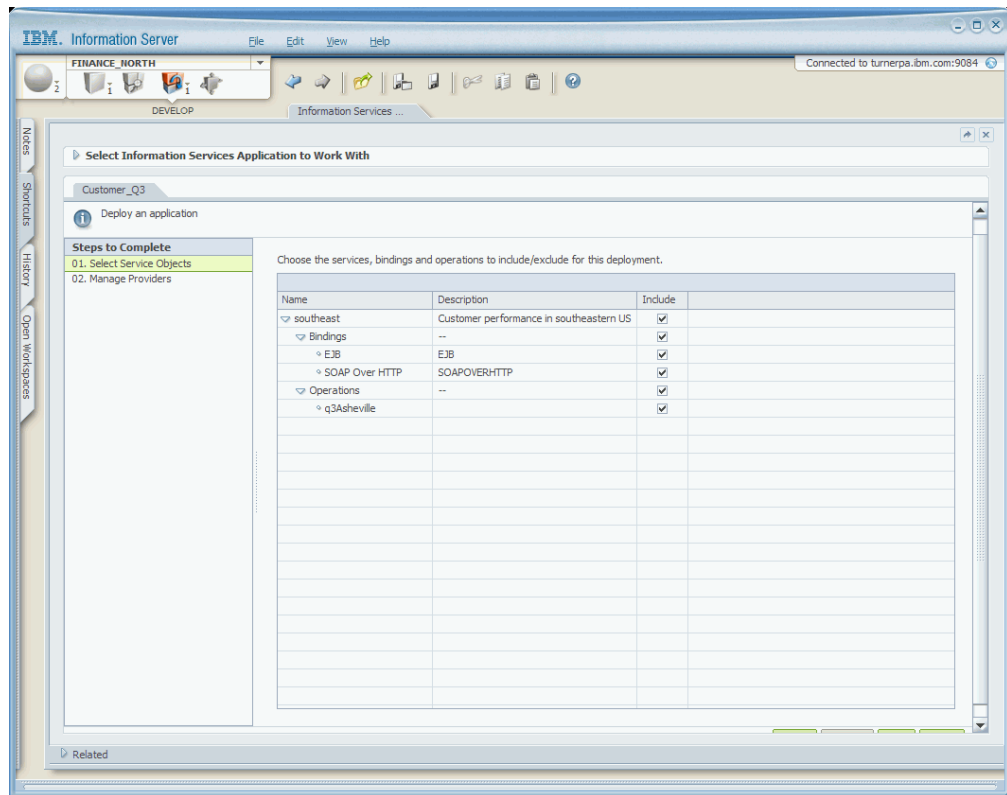


Figure 23. Deploying an application

You can exclude one or more services, bindings, and operations from the deployment, change runtime properties such as minimum number of job instances, or, for WebSphere DataStage jobs, set constant values for job parameters.

WebSphere Information Services Director deploys the Enterprise Archive (EAR) file on the application server.

Related concepts

“Introduction to WebSphere Federation Server” on page 112

WebSphere Federation Server allows organizations to virtualize their data and provide information in a form that applications and users need while hiding the complexity of the underlying sources. Data virtualization allows information to be accessed through a common interface that centralizes the control of data access.

SOA and data integration

Enabling an IBM Information Server job as a Web service enables the job to participate in various data integration scenarios.

Data integration enables users to federate heterogeneous data across several data sources. SOA allows WebSphere DataStage jobs to participate in federated queries by using WebSphere Federation Server.

Figure 24 on page 41 shows a business scenario in which a customer service manager needs to integrate information across multiple data stores to address new customer complaints. The manager needs to look at the actual invoice to compare

recent shipment data in XML format plus the historical data in the warehouse to ensure that the data is accurate.

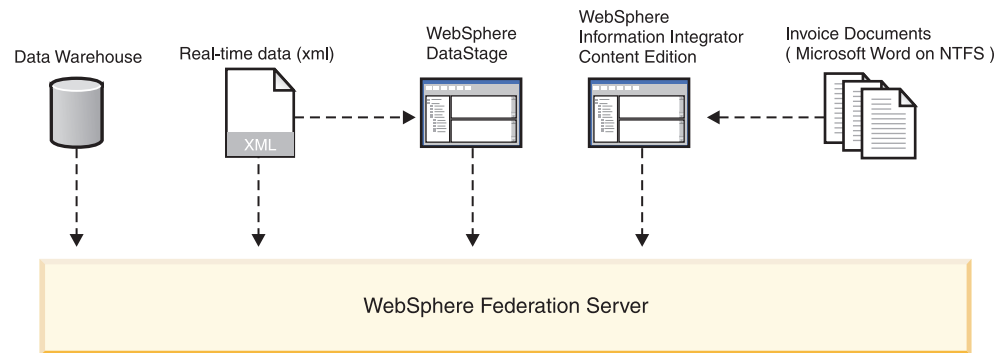


Figure 24. Combining WebSphere Information Integration products

In the example, real-time XML data is pulled out of a message queue by using WebSphere DataStage. The WebSphere DataStage job reuses the same transformation logic that it used to populate the warehouse. WebSphere Information Integrator Content Edition is invoked to display actual customer documents that reside on a document management system.

Figure 25 on page 42 shows how the data from each source is combined to present a virtual view of the most recent sales information. The following sequence is labeled in the diagram.

1. XML data is pulled from a queue by using a Shipto_Number to identify the XML files with the correct Sales_order_number, quantity of Cases_shipped and Gross_sales.
2. A WebSphere DataStage job that is deployed as a Web service provides real-time transformation of fact table data. Lookups are performed to locate values for Billto_key and Shipto_key surrogate keys.
3. Keys that are acquired from the WebSphere DataStage lookup are used to query the data warehouse to obtain company names that correspond to the keys.
4. The Sales_order_number is used to retrieve the URLs of the appropriate customer invoices from the document repository.

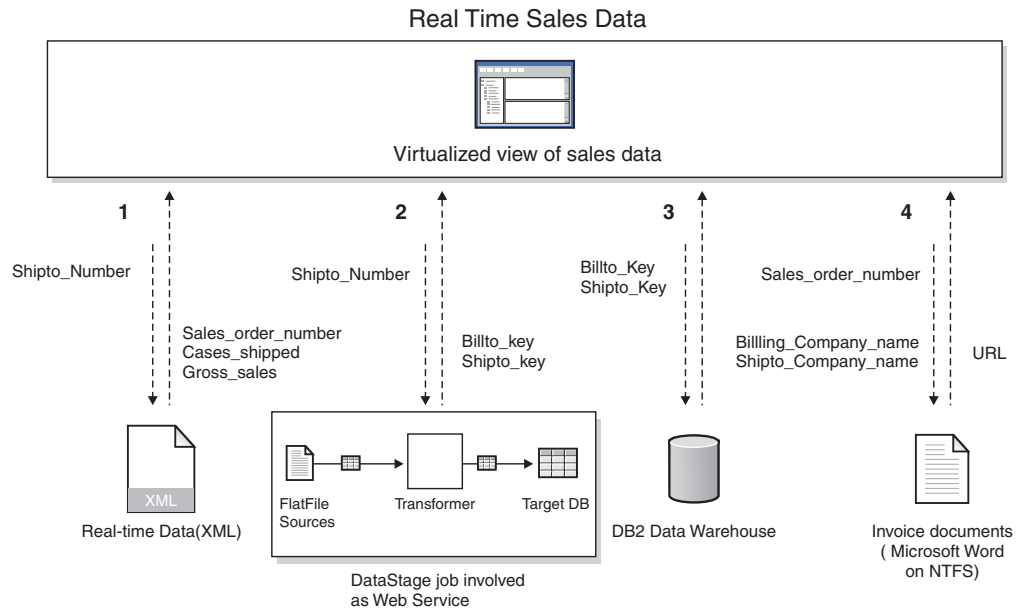


Figure 25. Combining data and content integration to create a federated query

Related concepts

“Federated stored procedures” on page 123

A *federated procedure* is a federated database object that references a procedure on a data source. Federated procedures are sometimes called federated stored procedures.

“Introduction to WebSphere Federation Server” on page 112

WebSphere Federation Server allows organizations to virtualize their data and provide information in a form that applications and users need while hiding the complexity of the underlying sources. Data virtualization allows information to be accessed through a common interface that centralizes the control of data access.

Information resources for WebSphere Information Services Director

A variety of information resources can help you get started with WebSphere Information Services Director.

When you first open the IBM Information Server console, the Getting Started pane describes all first steps that are required to begin your project. Each step includes two links:

- Open the related workspace and complete the task
- Open the Information Center to learn more about the task

Each pane and tab on the console also displays a line of context-sensitive instructional text, and online help is available from the interface. You can find more extensive online documentation for WebSphere Information Services Director in the IBM Information Server information center at <http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp>. The Information Center also provides planning, installation, and configuration details for IBM Information Server.

The *IBM Information Server Administration Guide* is available on the product documentation CD. You can also access the following PDFs from the Windows® **Start** menu and the product documentation CD:

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Chapter 5. WebSphere Information Analyzer

You use data profiling and analysis to understand your data and ensure that it suits the integration task. WebSphere Information Analyzer is a critical component of IBM Information Server that profiles and analyzes data so that you can deliver trusted information to your users.

WebSphere Information Analyzer can automatically scan samples of your data to determine their quality and structure. This analysis aids you in understanding the inputs to your integration process, ranging from individual fields to high-level data entities. Information analysis also enables you to correct problems with structure or validity before they affect your project.

In many situations, analysis must address data, values, and rules that are best understood by business users. Particularly for comprehensive enterprise resource planning, customer relationship management, or supply chain management packages, validating data against this business knowledge is a critical step. The business knowledge, in turn, forms the basis for ongoing monitoring and auditing of data to ensure validity, accuracy, and compliance with internal standards and industry regulations.

While analysis of source data is a critical first step in any integration project, you must continually monitor the quality of the data. WebSphere Information Analyzer enables you to treat profiling and analysis as an ongoing process and create business metrics that you can run and track over time.

WebSphere Information Analyzer capabilities

IBM WebSphere Information Analyzer automates the task of source data analysis by expediting comprehensive data profiling and minimizing overall costs and resources for critical data integration projects.

WebSphere Information Analyzer represents the next generation in data analysis tools, which are characterized by these attributes:

End-to-end data profiling and content analysis

Provides standard data profiling features and quality controls. The repository holds the data analysis results and project metadata such as project-level and role-level security and function administration.

Business-oriented approach

With its task-based user interface, aids business users in easily reviewing data for anomalies and changes over time, and provides key functional and design information to developers.

Adaptable, flexible, and scalable architecture

Handles high data volumes with common parallel processing technology, and leverages common services such as connectivity to access a wide range of data sources and targets.

Scenarios for information analysis

The following scenarios show how WebSphere Information Analyzer helps organizations understand their data to facilitate integration projects.

Food distribution: Infrastructure rationalization

A leading U.S. food distributor had more than 80 separate mainframe, SAP, and JD Edwards applications supporting global production, distribution, and CRM operations. This infrastructure rationalization project included customer relationship management, order-to-cash, purchase-to-pay, human resources, finance, manufacturing, and supply chain planning. The company needed to move data from these source systems to a single target system.

The company uses WebSphere Information Analyzer to profile its source systems and create master data around key business dimensions, including customer, vendor, item (finished goods), and material (raw materials). They plan to migrate data into a single master SAP environment and a companion SAP BW reporting platform.

Financial services: Data quality assessment

A major brokerage firm had become inefficient by supporting dozens of business groups with their own applications and IT groups. Costs were excessive, regulatory compliance difficult, and it was impractical to target low-margin, middle-income investors. When the federal government mandated T+1, a regulation that changed industry standard practices, the firm had to find a way to reduce the time to process a trade from 3.5 days to 1 day, a reduction of 71.4 percent.

To meet the federal mandate, the brokerage house uses WebSphere Information Analyzer to inventory their data, identify integration points, remove data redundancies, and document disparities between applications. The firm now has a repeatable and auditable methodology that leverages automated data analysis. By ensuring that all transactions are processed quickly and uniformly, the company is better able to track and respond to risk resulting from its clients' and its own investments.

Transportation services: Data quality monitoring

A transportation service provider develops systems that enable its extensive network of independent owner-operators to compete in today's tough market. The owner-operators were exposed to competition because they could not receive data quickly. Executives had little confidence in the data that they received. Productivity was slowed by excessive time reviewing manual intervention and reconciling data from multiple sources.

WebSphere Information Analyzer allows the owner-operators to better understand and analyze their legacy data. It allows them to quickly increase the accuracy of their business intelligence reports and restore executive confidence in their company data. Moving forward, they implemented a data quality solution to cleanse their customer data and spot trends over time, further increasing their confidence in the data.

WebSphere Information Analyzer in a business context

After obtaining project requirements, a project manager initiates the analysis phase of data integration to understand source systems and design target systems. Too often, analysis can be a laborious, manual process that relies on out-of-date (or nonexistent) source documentation or the knowledge of the people who maintain the source systems. But source system analysis is crucial to understanding what data is available and its current state.

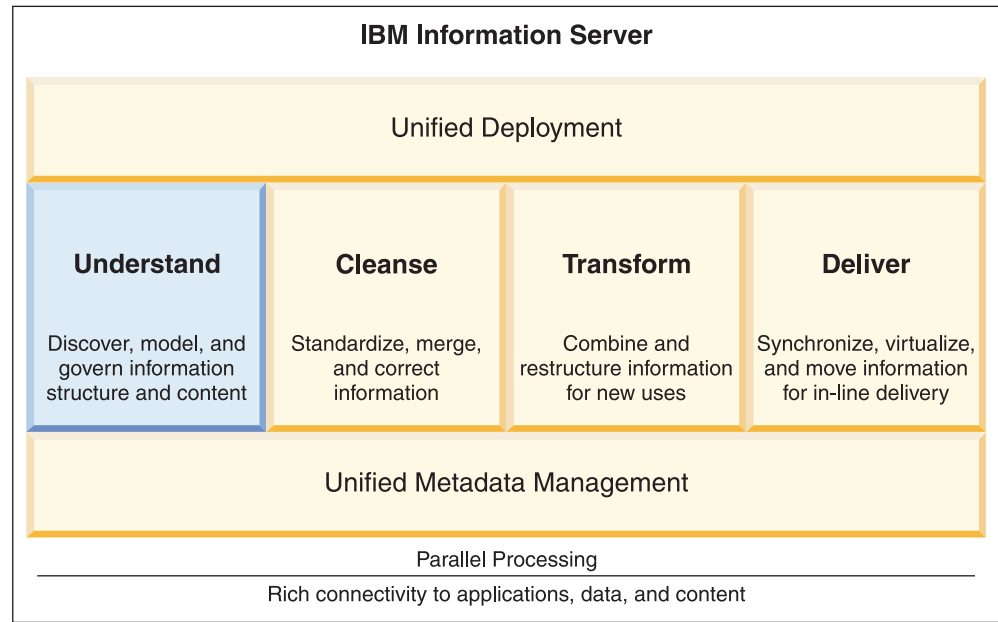


Figure 26. WebSphere Information Analyzer: Helping you understand your data

Figure 26 shows the role of analysis in IBM Information Server. WebSphere Information Analyzer plays a key role in preparing data for integration by analyzing business information to assure that it is accurate, consistent, timely, and coherent.

Profiling and analysis

Examines data to understand its frequency, dependency, and redundancy and validate defined schema and definitions.

Data monitoring and trending

Uncovers data quality issues in the source system as data is extracted and loaded into target systems. Validation rules help you create business metrics that you can run and track over time.

Facilitating integration

Uses tables, columns, probable keys, and interrelationships to help with integration design decisions.

Data analysis helps you see the content and structure of data before you start a project and continues to provide useful insight as part of the integration process. The following data management tasks use data analysis:

Data integration or migration

Data integration or migration projects (including data cleansing and matching) move data from one or more source systems to one or more target systems. Data profiling supports these projects in three critical stages:

1. Assessing sources to support or define business requirements
2. Designing reference tables and mappings from source to target systems
3. Developing and running tests to validate successful integration or migration of data into target systems

Data quality assessment and monitoring

Evaluates quality in targeted static data sources along multiple dimensions including completeness, validity (of values), accuracy, consistency,

timeliness, and relevance. Data quality monitoring requires ongoing assessment of data sources. WebSphere Information Analyzer supports these projects by automating many of these dimensions for in-depth snapshots over time.

Asset rationalization

Looks for ways to cut costs that are associated with existing data transformation processes (for example, processor cycles) or data storage. Asset rationalization does not involve moving data, but reviews changes in data over time. WebSphere Information Analyzer supports asset rationalization during the initial assessment of source content and structure and during development and execution of data monitors to understand trends and utilization over time.

Verifying external sources for integration

Validates the arrival of new or periodic external sources to ensure that those sources still support the data integration processes that use them. This process looks at static data sources along multiple dimensions including structural conformity to prior instances, completeness, validity of values, validity of formats, and level of duplication. WebSphere Information Analyzer automates many of these dimensions over time.

Related concepts

Chapter 1, “Introduction,” on page 1

Most of today’s critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

A closer look at WebSphere Information Analyzer

WebSphere Information Analyzer is an integrated tool for providing comprehensive enterprise-level data analysis. It features data profiling, analysis, and design and supports ongoing data quality monitoring.

The WebSphere Information Analyzer user interface performs a variety of data analysis tasks, as Figure 27 on page 49 shows.

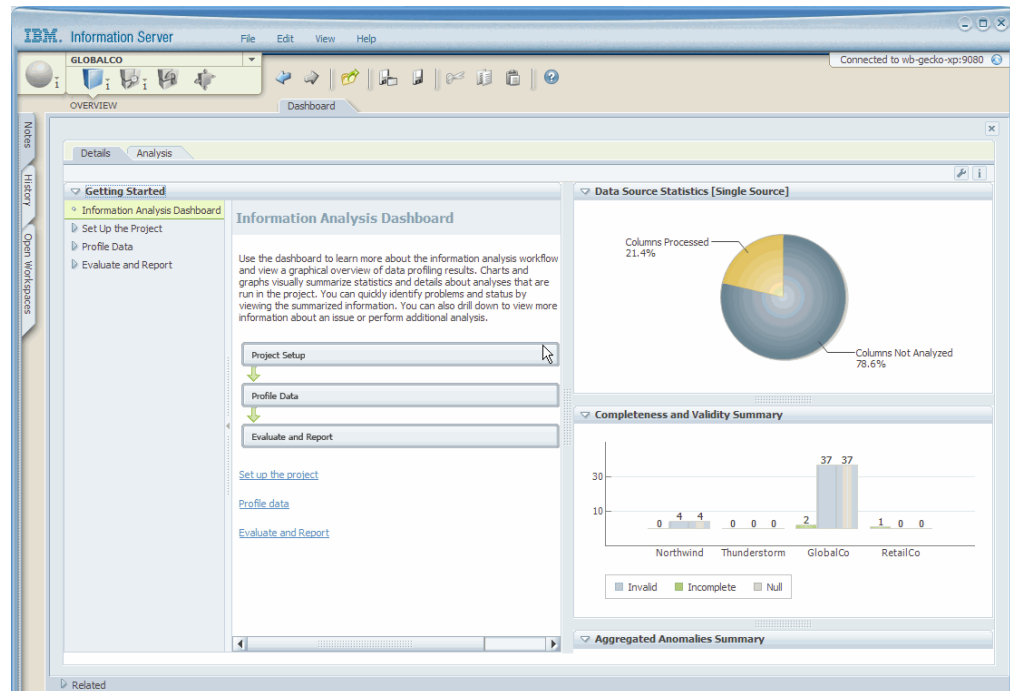


Figure 27. Dashboard view of a project provides high-level trends and metrics

WebSphere Information Analyzer can be used by data analysts, subject matter experts, business analysts, integration analysts, and business end users. It has the following characteristics:

Business-driven

Provides end-to-end data lifecycle management (from data access and analysis through data monitoring) to reduce the time and cost to discover, evaluate, correct, and validate data across the enterprise.

Dynamic

Draws on a single active repository for metadata to give you a common platform view.

Scalable

Leverages a high-volume, scalable, parallel processing design to provide high performance analysis of large data sources.

Extensible

Enables you to review and accept data formats and data values as business needs change.

Service oriented

Leverages IBM Information Server's service-oriented architecture to access connectivity, logging, and security services, allowing access to a wide range of data sources (relational, mainframe, and sequential files) and the sharing of analytical results with other IBM Information Server components.

Robust analytics

Helps you understand embedded or hidden information about content, quality, and structure.

Design integration

Improves the exchange of information from business and data analysts to developers by generating validation reference data and mapping data, which reduces errors.

Robust reporting

Provides a customizable interface for common reporting services, which enables better decision making by visually representing analysis, trends, and metrics.

IBM WebSphere AuditStage is a suite component that augments WebSphere Information Analyzer by helping you manage the definition and analysis of business rules. WebSphere AuditStage examines source and target data, analyzing across columns for valid value combinations, appropriate data ranges, accurate computations, and correct if-then-else evaluations. WebSphere AuditStage establishes metrics to weight these business rules and stores a history of these analyses and metrics that show trends in data quality.

Where WebSphere Information Analyzer fits in the IBM Information Server architecture

WebSphere Information Analyzer uses a service-oriented architecture to structure data analysis tasks that are used by many new enterprise system architectures. WebSphere Information Analyzer is supported by a range of shared services and reuses several IBM Information Server components.

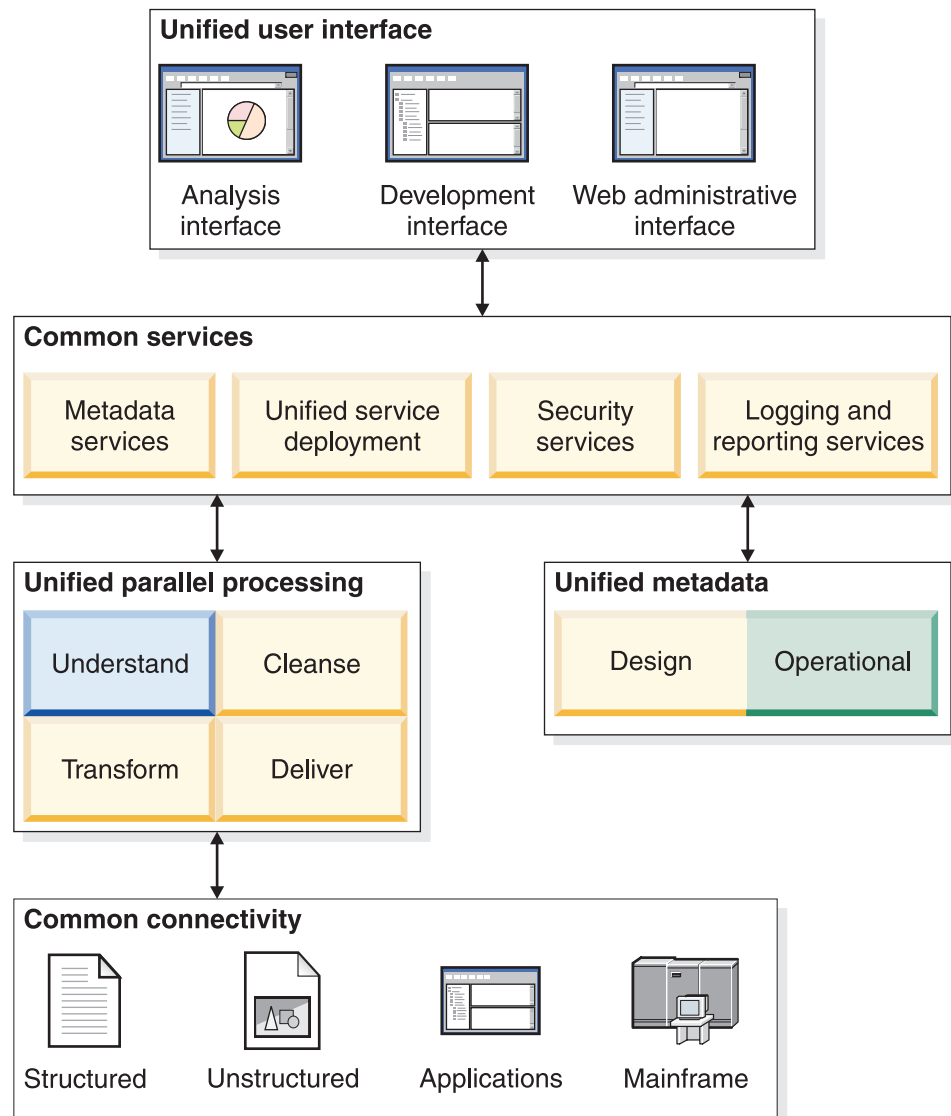


Figure 28. IBM Information Server architecture

Because WebSphere Information Analyzer has multiple discrete services, it has the flexibility to configure systems to match varied customer environments and tiered architectures. Figure 28 shows how WebSphere Information Analyzer interacts with the following elements of IBM Information Server:

IBM Information Server console

Provides a graphical user interface to access WebSphere Information Analyzer functions and organize data analysis results.

Common services

Provide general services that WebSphere Information Analyzer uses such as logging and security. Metadata services provide access, query, and analysis functions for users. Many services that are offered by WebSphere Information Analyzer are specific to its domain of enterprise data analysis such as column analysis, primary key analysis and review, and cross-table analysis.

Common repository

Holds metadata that is shared by multiple projects. WebSphere Information

Analyzer organizes data from databases, files, and other sources into a hierarchy of objects. Results that are generated by WebSphere Information Analyzer can be shared with other client programs such as the WebSphere DataStage and WebSphere QualityStage Designer by using their respective service layers.

Common parallel processing engine

Addresses high throughput requirements that are inherent in analyzing large quantities of source data by taking advantage of parallelism and pipelining.

Common connectors

Provide connectivity to all the important external resources and access to the common repository from the processing engine. WebSphere Information Analyzer uses these connection services in three fundamental ways:

- Importing metadata
- Performing base analysis on source data
- Providing drill-down and query capabilities

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

WebSphere Information Analyzer tasks

The WebSphere Information Analyzer user interface presents an intuitive set of controls that are designed for integration development workflow.

The WebSphere Information Analyzer user interface aids you in organizing data analysis work into projects. The top-level view is called a Dashboard because it reports a summary of your key project and data metrics, both in a graphical format and in a status grid format.

The high-level status view in Figure 29 on page 53 summarizes the data sources, including their tables and columns, that were analyzed and reviewed so that managers and analysts can quickly determine the status of work. The project view of the GlobalCo project shows a high-level summary of column analysis, an aggregated summary of anomalies found, and the Getting Started pane.

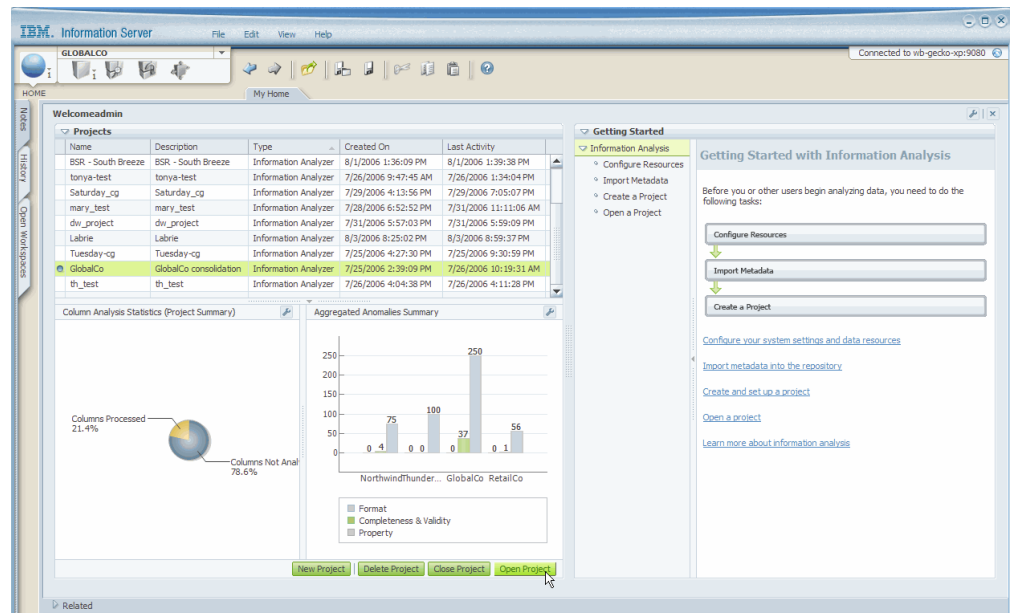


Figure 29. WebSphere Information Analyzer project view

While many data analysis tools are designed to run in a strict sequence and generate one-time static views of the data, WebSphere Information Analyzer enables you to perform select integration tasks as required or combine them into a larger integration flow. These tasks fall into three categories:

Profiling and analysis

Provides complete analysis of source systems and target systems, and assesses the structure, content, and quality of data, whether at the column level, the cross-column level, the table or file level, the cross-table level, or the cross-source level. This task reports on various aspects of data including classification, attributes, formatting, frequency values, distributions, completeness, and validity.

Data monitoring and trending

Helps you assess data completeness and validity, data formats, and valid-value combinations. This task also evaluates new results against established benchmarks. By using the WebSphere AuditStage component, business users develop additional data rules to assess and measure content and quality over time. Rules can be simple column measures that incorporate knowledge from data profiling or complex conditions that test multiple fields. Validation rules assist in creating business metrics that you can run and track over time.

Facilitating integration

Provides shared analytical information, validation and mapping table generation, and testing of data transformations by using cross-comparison of domains before and after processing.

Data profiling and analysis

WebSphere Information Analyzer provides extensive capabilities for profiling source data. The four main data profiling functions are column analysis, primary key analysis, foreign key analysis, and cross-domain analysis.

Column analysis

Column analysis generates a full frequency distribution and examines all values for a column to infer its definition and properties such as domain values, statistical measures, and minimum and maximum values. Each column of every source table is examined in detail. The following properties are observed and recorded:

- Count of distinct values or cardinality
- Count of empty values, null values, and non-null or empty values
- Minimum, maximum, and average numeric values
- Basic data types, including different date-time formats
- Minimum, maximum, and average length
- Precision and scale for numeric values

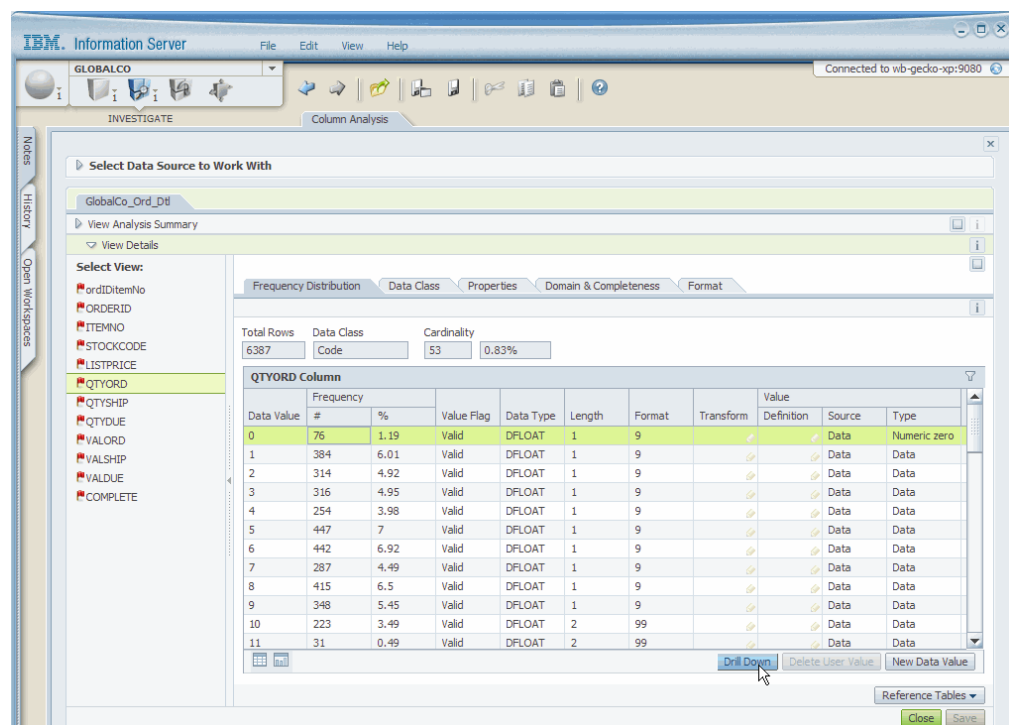


Figure 30. Column analysis example data view

WebSphere Information Analyzer also enables you to drill down on specific columns to define unique quality control measures for each column. Figure 30 shows a closer look at results for a table named GlobalCo_Ord_Dtl. At the top is a summary analysis of the entire table. Beneath the summary is detail for each column that shows standard data profiling results, including data classification, cardinality, and properties. When you select a column, additional tasks that are relevant to that level of analysis become available.

Another function of column analysis is domain analysis. A domain is a valid set of values for an attribute. Domain analysis determines the data domain values for any data element. By using a frequency distribution, you can facilitate testing by providing a list of all the values in a column and the number of occurrences of each. Domain analysis checks whether a data element corresponds to a value in a

database table or file. Figure 31 shows a frequency distribution chart that helps find anomalies in the Qtyord column.

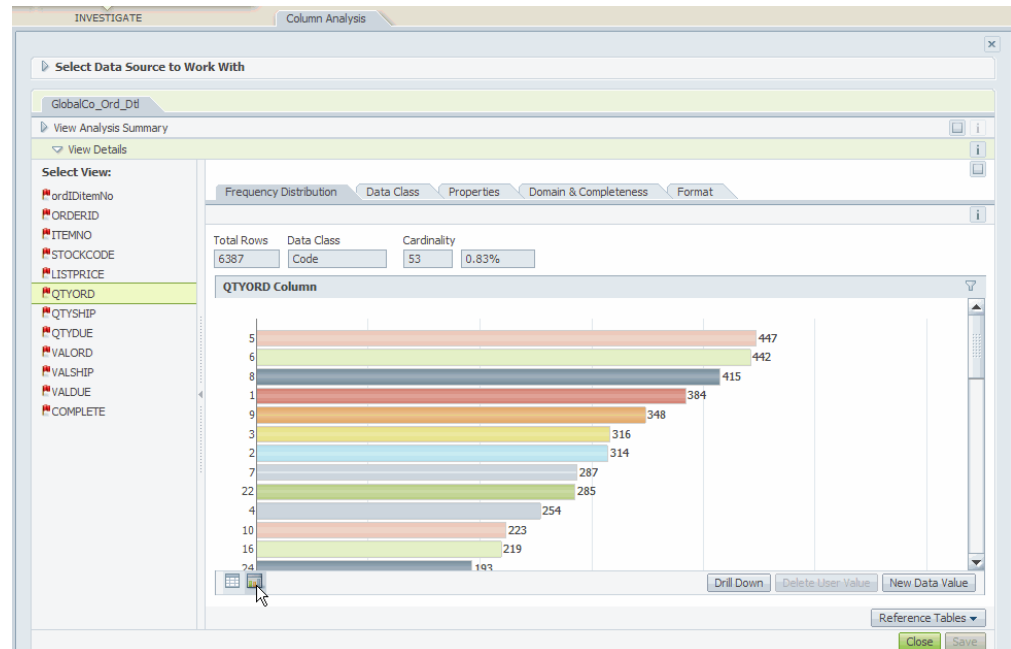


Figure 31. Column analysis example graphical view

The bar chart shows data values on the y-axis and the frequency of those values on the x-axis. This detail points out default and invalid values based on specific selection, ranges, or reference sources, and aids you in iteratively building quality metrics.

When you are validating free-form text, analyzing and understanding the extent of the quality issues is often very difficult. WebSphere Information Analyzer can show each data pattern of the text for a much more detailed quality investigation. It helps with the following tasks:

- Uncovering trends, potential anomalies, metadata discrepancies, and undocumented business practices
- Identifying invalid or default formats and their underlying values
- Verifying the reliability of fields that are proposed as matching criteria for input to WebSphere QualityStage and WebSphere DataStage

Primary key analysis

The primary key of a relational table is a unique identifier that a database uses to access a specific row. Primary key analysis identifies all candidate keys for one or more tables and helps you test a column or combination of columns to determine if it is a candidate for becoming the primary key. Figure 32 on page 56 shows a single-column analysis.

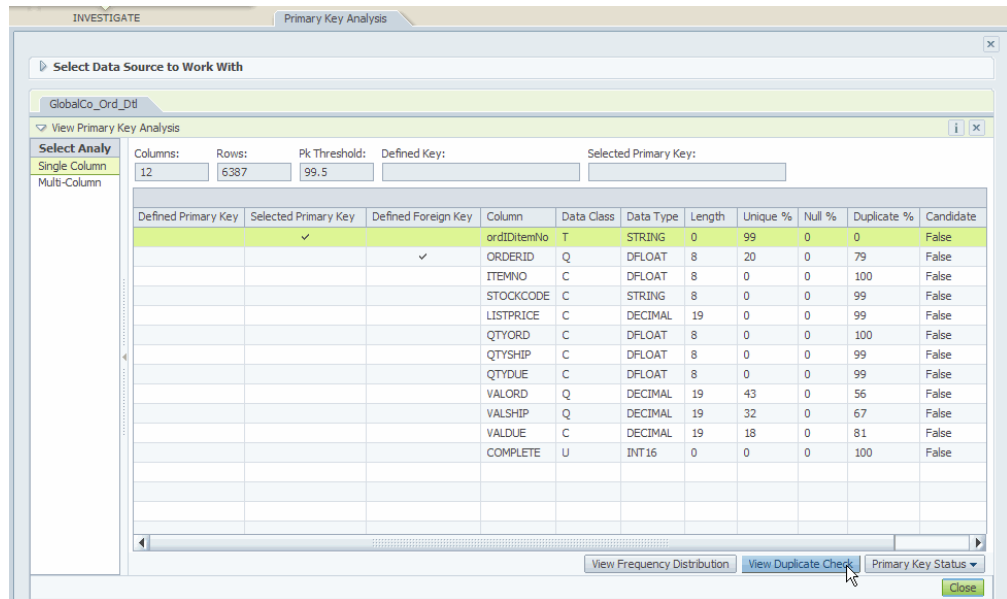


Figure 32. Primary key analysis

The analysis presents all of the columns and the potential primary key candidates. A duplicate check validates the use of such keys. You select the primary key candidate based on its probability for uniqueness and your business knowledge of the data involved. If you select a multi-data column as the primary key, the system will develop a frequency distribution for the concatenated values.

Foreign key analysis

Foreign key analysis examines content and relationships across tables. This analysis helps identify foreign keys, check their integrity, and check the referential integrity between the primary key and foreign keys. For example, in a Bill of Materials structure, the parent-child relationships among assemblies and subassemblies would require you to identify relationships between foreign keys and primary keys and validate their referential integrity.

A column qualifies to be a foreign key candidate if the majority (for example, 98 percent or higher) of its frequency distribution values match the frequency distribution values of a primary key column. As Figure 33 on page 57 shows, after you select a foreign key, the system performs a bidirectional test (foreign key to primary key, primary key to foreign key) of each foreign key's referential integrity and identifies the number of referential integrity violations and "orphan" values (keys that do not match).

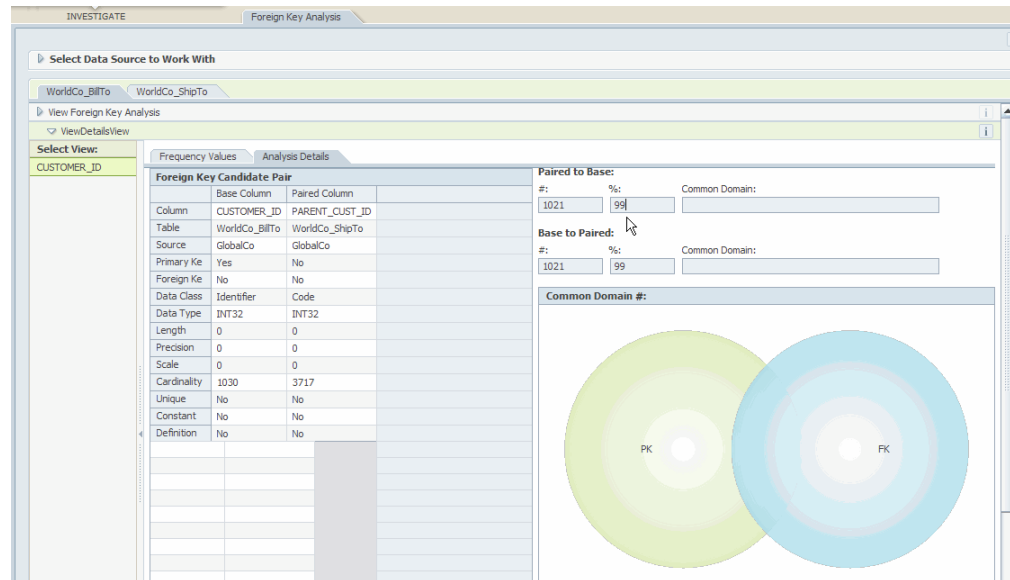


Figure 33. Foreign key analysis

Cross-domain analysis

Cross-domain analysis examines content and relationships across tables. This analysis identifies overlaps in values between columns, and any redundancy of data within or between tables. For example, country codes might exist in two different customer tables and you want to maintain a consistent standard for these codes. Cross-domain analysis enables you to directly compare these code values.

WebSphere Information Analyzer uses the results of column analysis for each set of columns that you want to compare. The existence of a common domain might indicate a relationship between tables or the presence of redundant fields. Cross-domain analysis can compare any number of domains within or across sources.

Data monitoring and trending

With baseline analysis, WebSphere Information Analyzer compares changes to data from one previous column analysis (a baseline) to a new, current column analysis.

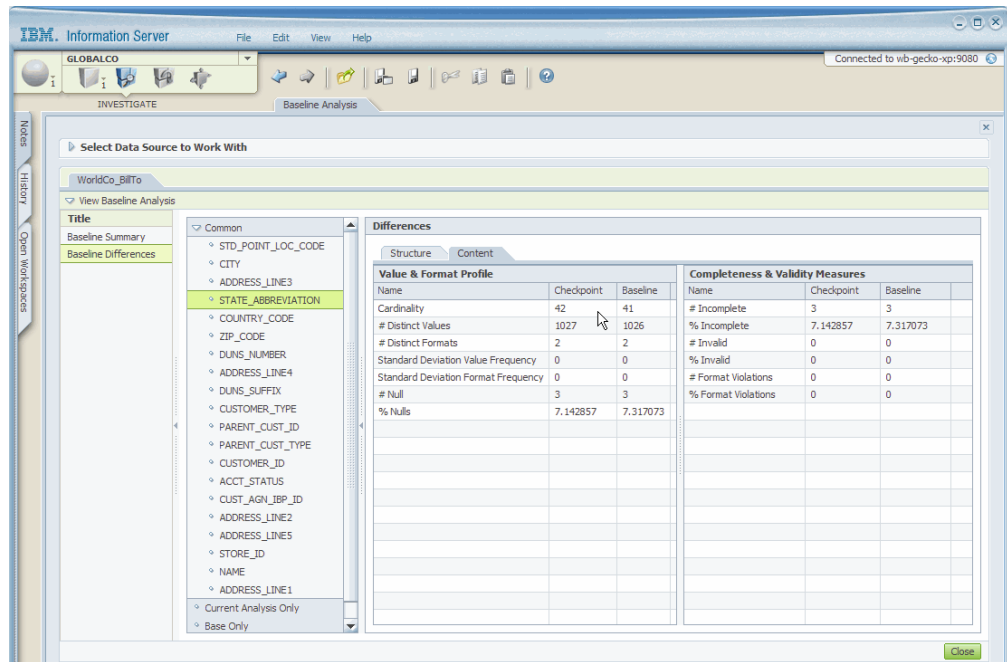


Figure 34. Baseline comparison results

Figure 34 shows the results of comparing two distinct analyses on the WorldCo_Bill_to table. The comparison provides a description of the structural and content differences, including the quality measures over time. The State_Abbreviation column shows a new data value, which should prompt a review of the column analysis for distinct changes that might affect overall data completeness and validity.

Data rules and metrics

With WebSphere AuditStage, you can create validation rules for data and evaluate data sets for compliance. These rules assist you in creating metrics that you can run and track over time.

Although validation rules of different organizations, particularly within the same industry, might be similar, each organization's rules will be specific to its processing operations and policies. Validation rule analysis can extend the evaluation of a data source or across data sources for defined relationships between and among data.

WebSphere AuditStage allows validation rules to be expressed in many ways. It can also check to see if data conforms to certain constraints:

Containment

Whether a field contains a string or evaluates to a certain expression that contains a certain string.

Equality

Whether a field equals a certain value.

Existence

Whether a source has any data.

Format

Whether values in the source data match a pattern string.

Occurrence

The number of times that values occur within a source table.

Range The range of the source data. A range can include a minimum value, maximum value, or both.

Reference column

Referential integrity of the source data against a reference column.

Reference list

Whether data fits a reference list of allowed values.

Type Whether the source data can be converted from a character to a number or date.

Uniqueness

Whether the source data has duplicate values. Certain fields such as account number must always be unique.

These rules can be combined with logical operators to find rows from one or more tables in which multiple columns have multiple characteristics. You can also combine the rules with logical operators to evaluate complex conditions and pinpoint data that is not invalid in itself but tests a broader constraint or business condition. For example, you might use a rule to measure a trend such as the number of orders from a given customer for a specific class of products.

WebSphere Information Analyzer supports the creation of benchmarks and metrics that are used to measure ongoing data quality. Viewing the benchmark over time provides valuable detail about data quality trends.

Facilitating integration

WebSphere Information Analyzer facilitates information integration by using the available source and target metadata and defined data rules and validation tables to initiate the design of new data integration tasks. These tasks include transformation and monitoring processes and generating new job designs. By generating a set of values against which data rules will compare the source data, WebSphere Information Analyzer can generate reference tables that are used for the following tasks:

Mapping

A mapping table is used to replace an obsolete value in a data table with an updated value as part of a transformation process.

Range checking

A range table helps you determine if a value in the data table falls within minimum and maximum values.

Validity checking

A validity table aids in determining whether a value in the data table is one of the valid domain values for the data element.

WebSphere Information Analyzer facilitates integration by sharing metadata with other components of IBM Information Server. You can bypass the data quality investigation stage by using published metadata from WebSphere Information Analyzer. Figure 35 on page 60 shows metadata that is being published to the WebSphere Metadata Server.

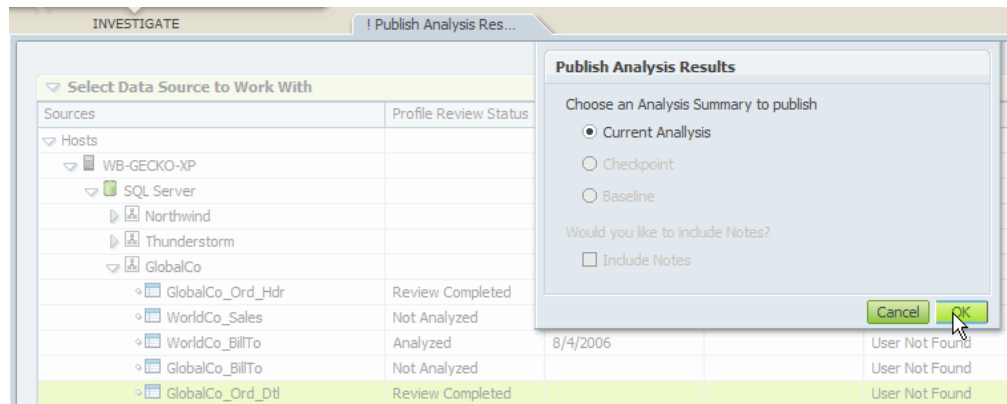


Figure 35. Publishing metadata

WebSphere Information Analyzer also supports the direct entry of associated business terms to data sources, tables, or columns. Such terms and associations can be used by WebSphere Business Glossary to expand the overall semantic knowledge of an organization or to confirm that business information is reflected in the actual data.

Results of the analysis

The results of a source system analysis support several key integration activities.

- Understanding the source data by using graphical displays and printed reports
- Generating validation reference tables
- Identifying source data for additional profiling and validation
- Generating mappings between the source database and a target database by using shared metadata (a WebSphere DataStage function)

Creating reference tables

You can create reference tables from the results of frequency distributions and use the reference tables with other IBM Information Server suite components or other systems to enforce domain and completeness requirements or to control data conversion. You can create Validity, Range, Completeness, and Mapping reference tables.

Reference tables improve the exchange of information from business and data analysts, speed the task of mapping data between source and target systems, and help you flag issues for review.

Securing analytical results

In meeting current regulatory and compliance needs, organizations must track and secure their data, particularly the privacy needs of their customers. From customer-sensitive tax IDs to employee salaries to potential fraud indicators, organizations must ensure that access to these and other highly sensitive data fields is appropriately restricted.

WebSphere Information Analyzer helps meet those critical requirements by using a multilevel access and security environment. Access to the functions of WebSphere Information Analyzer is controlled with both server-level and project-based user access, which is associated with appropriate roles from the organization's underlying security framework.

At the project level, users are granted rights to both functions and data sources, to the level of a specific column or field, giving you the flexibility to meet specific compliance needs.

Information resources for WebSphere Information Analyzer

A variety of information resources can help you get started with WebSphere Information Analyzer.

When you first open the IBM Information Server console, the Getting Started pane describes all first steps that are required to begin your project. Each step includes two links:

- Open the related workspace and complete the task
- Open the Information Center to learn more about the task

Each pane and tab on the console also displays a line of context-sensitive instructional text, and online help is available from the interface. You can find more extensive online documentation for WebSphere Information Analyzer in the IBM Information Server information center at <http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp>. The information center also provides planning, installation, and configuration details for IBM Information Server and WebSphere Information Analyzer.

The *WebSphere Information Analyzer User Guide* is available on the Quick Start CD. You can also access the following PDFs from the Windows **Start** menu and the Quick Start CD:

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Chapter 6. WebSphere QualityStage

The data that drives today's business systems often comes from a variety of sources and disparate data structures. As organizations grow, they retain old data systems and augment them with new and improved systems. Data becomes difficult to manage and use, and a clear picture of a customer, product, or buying trend can be practically impossible to ascertain.

The price of poor data is illustrated by these examples:

- A data error in a bank causes 300 credit-worthy customers to receive mortgage default notices. The error costs the bank time, effort, and customer goodwill.
- A marketing organization sends duplicate direct mail pieces. A six percent redundancy in each mailing costs hundreds of thousands of dollars a year.
- A managed-care agency cannot relate prescription drug usage to patients and prescribing doctors. The agency's OLAP application fails to identify areas to improve efficiency and inventory management and new selling opportunities.

The source of quality issues is a lack of common standards for how to store data and an inconsistency in how the data is input. Different business operations are often very creative with the data values that they introduce into your application environments. Inconsistency across sources makes understanding relationships between critical business entities such as customers and products very difficult. In many cases, there is no reliable and persistent key that you can use across the enterprise to get all the information that is associated with a single customer or product.

Without high-quality data, strategic systems cannot match and integrate all related data to provide a complete view of the organization and the interrelationships within it. CIOs can no longer count on a return on the investments made in critical business applications. The solution calls for a product that can automatically re-engineer and match all types of customer, product, and enterprise data, in batch or at the transaction level in real time.

WebSphere QualityStage is a data re-engineering environment that is designed to help programmers, programmer analysts, business analysts, and others cleanse and enrich data to meet business objectives and data quality management standards.

Introduction to WebSphere QualityStage

WebSphere QualityStage comprises a set of stages, a Match Designer, and related capabilities that provide a development environment for building data-cleansing tasks called jobs.

Using the stages and design components, you can quickly and easily process large stores of data, selectively transforming the data as needed.

WebSphere QualityStage provides a set of integrated modules for accomplishing data re-engineering tasks:

- Investigating
- Conditioning (standardizing)
- Designing and running matches

- Determining which data records survive

The probabilistic matching capability and dynamic weighting strategies of WebSphere QualityStage help you create high-quality, accurate data and consistently identify core business information such as customer, location, and product throughout the enterprise. WebSphere QualityStage standardizes and matches any type of information. By ensuring data quality, WebSphere QualityStage reduces the time and cost to implement CRM, business intelligence, ERP, and other strategic customer-related IT initiatives.

Scenarios for data cleansing

Organizations need to understand the complex relationships that they have with their customers, suppliers and distribution channels. They need to base decisions on accurate counts of parts and products to compete effectively, provide exceptional service, and meet increasing regulatory requirements. Consider the following scenarios:

Banking: One view of households

To facilitate marketing and mail campaigns, a large retail bank needed a single dynamic view of its customers' households from 60 million records in 50 source systems.

The bank uses WebSphere QualityStage to automate the process. Consolidated views are matched for all 50 sources, yielding information for all marketing campaigns. The result is reduced costs and improved return on the bank's marketing investments. Householding is now a standard process at the bank, which has a better understanding of its customers and more effective customer relationship management.

Pharmaceutical: Operations information

A large pharmaceutical company needed a data warehouse for marketing and sales information. The company had diverse legacy data with different standards and formats, information that was buried in free-form fields, incorrect data values, discrepancies between field metadata and actual data in the field, and duplicates. It was impossible to get a complete, consolidated view of an entity such as total quarterly sales from the prescriptions of one doctor. Reports were difficult and time-consuming to compile, and their accuracy was suspect.

Most vendor tools lack the flexibility to find all the legacy data variants, different formats for business entities, and other data problems. The company chose WebSphere QualityStage because it goes beyond traditional data-cleansing techniques to investigate fragmented legacy data at the level of each data value. Analysts can now access complete and accurate online views of doctors, the prescriptions that they write, and their managed-care affiliations for better decision support, trend analysis, and targeted marketing.

Insurance: One real-time view of the customer

A leading insurance company lacked a unique ID for each subscriber, many of whom participated in multiple health, dental, or benefit plans. Subscribers who visited customer portals could not get complete information on their account status, eligible services, and other details.

Using WebSphere QualityStage, the company implemented a real-time, in-flight data quality check of all portal inquiries. WebSphere QualityStage and WebSphere MQ transactions were combined to retrieve customer data from multiple sources and return integrated customer views. The new

process provides more than 25 million subscribers with a real-time, 360-degree view of their insurance services. A unique customer ID for each subscriber is also helping the insurer move toward a single customer database for improved customer service and marketing.

Where WebSphere QualityStage fits in the overall business context

WebSphere QualityStage performs the preparation stage of enterprise data integration (often referred to as data cleansing), as Figure 36 shows. WebSphere QualityStage leverages the source systems analysis that is performed by WebSphere Information Analyzer and supports the transformation functions of WebSphere DataStage.

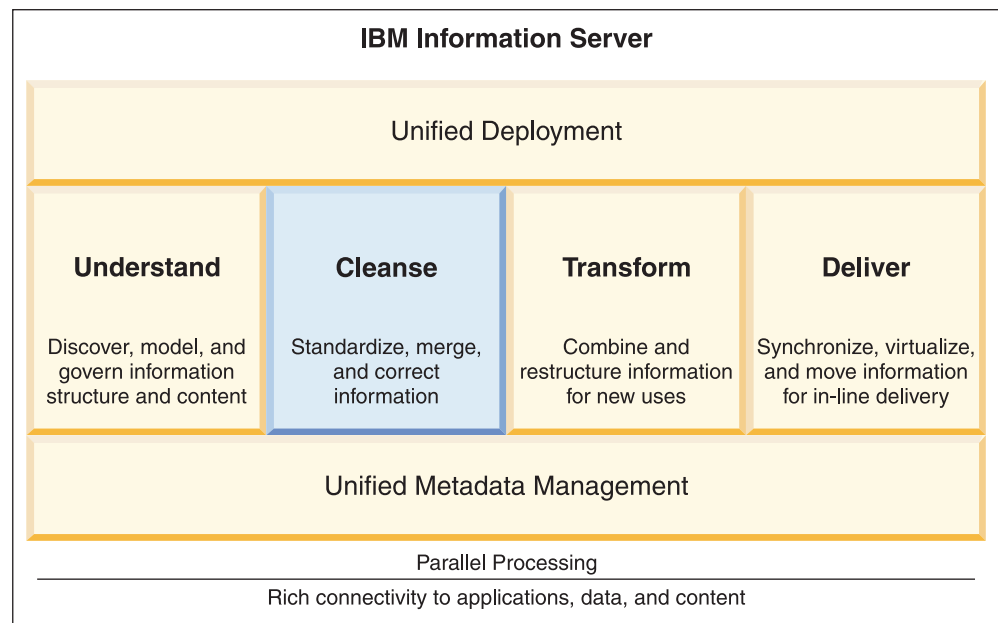


Figure 36. WebSphere QualityStage prepares data for integration

Working together, these products automate what was previously a manual or neglected activity within a data integration effort: data quality assurance. The combined benefits help companies avoid one of the biggest problems with data-centric IT projects: low return on investment (ROI) caused by working with poor-quality data.

Data preparation is critical to the success of an integration project. These common business initiatives are strengthened by improved data quality:

Consolidating enterprise applications

High-quality data and the ability to identify critical role relationships improves the success of consolidation projects.

Marketing campaigns

Strong understanding of customers and customer relationships cuts costs, improves customer satisfaction and attrition, and increases revenues.

Supply chain management

Better data quality allows better integration between an organization and its suppliers by resolving differences in codes and descriptions for parts or products.

Procurement

Identifying multiple purchases from the same supplier and multiple purchases of the same commodity leads to improved terms and reduced cost.

Fraud detection and regulatory compliance

Better reference data reduces fraud loss by quickly identifying fraudulent activity.

Whether an enterprise is migrating its information systems, upgrading its organization and its processes, or integrating and leveraging information, it must determine the requirements and structure of the data that will address the business goals. As Figure 37 shows, you can use WebSphere QualityStage to meet those data quality requirements with classic data re-engineering.

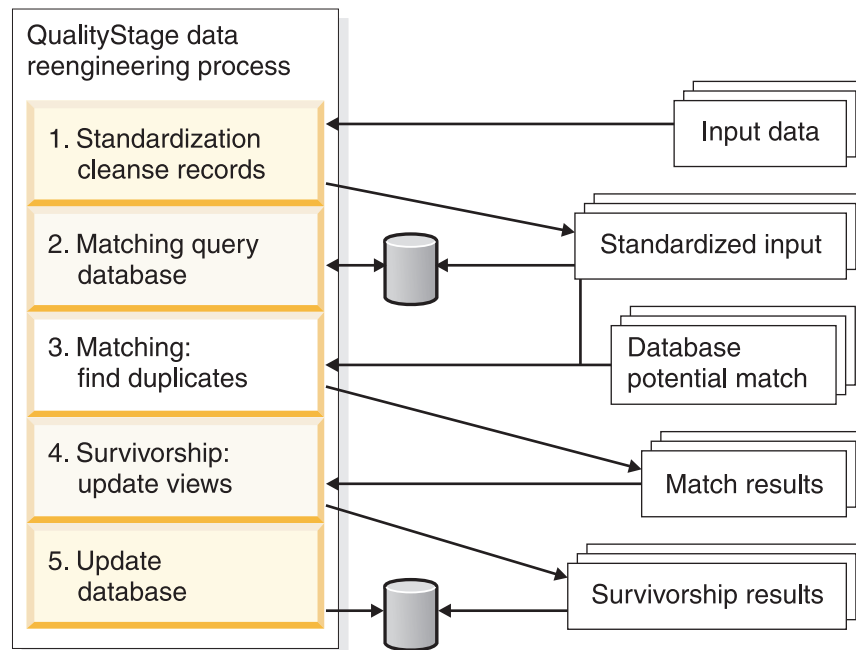


Figure 37. Classic data reengineering with WebSphere QualityStage

A process for reengineering data should accomplish the following goals:

- Resolve conflicting and ambiguous meanings for data values
- Identify new or hidden attributes from free-form and loosely controlled source fields
- Standardize data to make it easier to find
- Identify duplication and relationships among such business entities as customers, prospects, vendors, suppliers, parts, locations, and events
- Create one unique view of the business entity
- Facilitate enrichment of reengineered data, such as adding information from vendor sources or applying standard postal certification routines

You can use a data reengineering process in batch or real time for continuous data quality improvement.

A closer look at WebSphere QualityStage

WebSphere QualityStage uses out-of-the-box, customizable rules to prepare complex information about your business entities for a variety of transactional, operational, and analytical purposes.

WebSphere QualityStage automates the conversion of data into verified standard formats by using probabilistic matching, in which variables that are common to records (for example, given name, date of birth, or sex) are matched when unique identifiers are not available.

WebSphere QualityStage components include the Match Designer, for designing and testing match passes, and a set of data-cleansing operations called stages. Information is extracted from the source system, measured, cleansed, enriched, consolidated, and loaded into the target system.

At run time, data cleansing jobs consist of the following sequence of stages:

Investigate stage

Gives you complete visibility into the actual condition of data.

Standardize stage

Reformats data from multiple systems to ensure that each data type has the correct content and format.

Match stages

Ensure data integrity by linking records from one or more data sources that correspond to the same customer, supplier, or other entity. Matching can be used to identify duplicate entities that are caused by data entry variations or account-oriented business practices. Unduplicate match jobs group records into sets that have similar attributes. The Reference Match stage matches reference data to source data using a variety of match processes.

Survive stage

Ensures that the best available data survives and is correctly prepared for the target.

Business intelligence packages that are available with WebSphere QualityStage provide data enrichment that is based on business rules. These rules can resolve issues with common data quality problems such as invalid address fields across multiple geographies. The following packages are available:

Worldwide Address Verification and Enhancement System (WAVES)

Matches address data against standard postal reference data that helps you verify address information for 233 countries and regions.

Multinational geocoding

Used for spatial information management and location-based services by adding longitude, latitude, and census information to location data.

Postal certification rules

Provide certified address verification and enhancement to address fields to enable mailers to meet the local requirements to qualify for postal discounts.

Where WebSphere QualityStage fits in the IBM Information Server architecture

WebSphere QualityStage is built around a services-oriented vision for structuring data quality tasks that are used by many new enterprise system architectures. As part of the integrated IBM Information Server platform, it is supported by a broad range of shared services and benefits from the reuse of several suite components.

WebSphere QualityStage and WebSphere DataStage share the same infrastructure for importing and exporting data, designing, deploying, and running jobs, and reporting. The developer uses the same design canvas to specify the flow of data from preparation to transformation and delivery.

Multiple discrete services give WebSphere QualityStage the flexibility to match increasingly varied customer environments and tiered architectures. Figure 38 on page 69 shows how the WebSphere DataStage and QualityStage Designer (labeled "Development interface") interacts with other elements of the platform to deliver enterprise data analysis services.

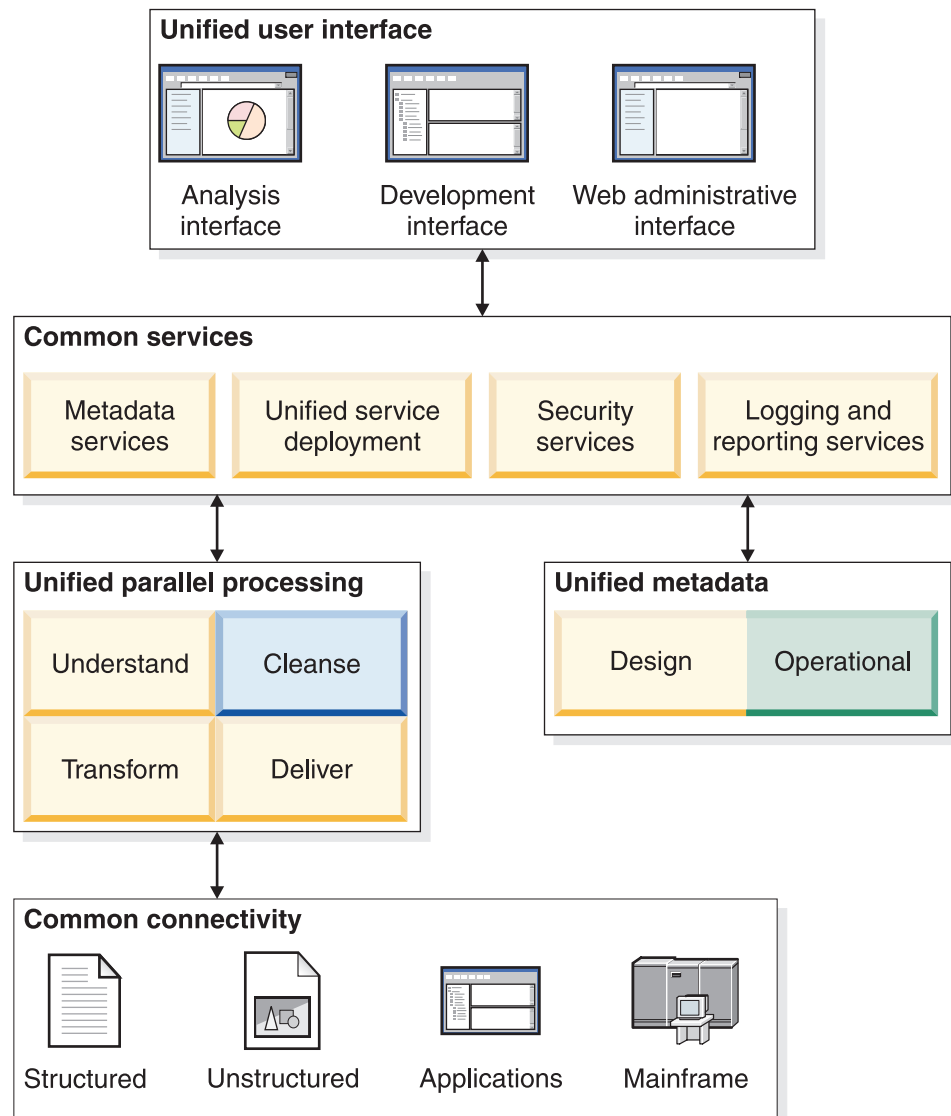


Figure 38. IBM Information Server product architecture

The following suite components are shared:

Common user interface

The WebSphere DataStage and QualityStage Designer provides a development environment. The WebSphere DataStage and QualityStage Administrator provides access to deployment and administrative functions. WebSphere QualityStage is tightly integrated with WebSphere DataStage and shares the same design canvas, which enables users to design jobs with data transformation stages and data quality stages in the same session.

Common services

WebSphere QualityStage uses the common services in IBM Information Server for logging and security. Because metadata is shared “live” across tools, you can access services such as impact analysis without leaving the design environment. You can also access domain-specific services for enterprise data cleansing such as investigate, standardize, match, and survive from this layer.

Common repository

The repository holds data to be shared by multiple projects. Clients can access metadata and results of data analysis from the respective service layers.

Common parallel processing engine

The parallel processing engine addresses high throughput requirements for analyzing large quantities of source data and handling increasing volumes of work in decreasing time frames.

Common connectors

Any data source that is supported by IBM Information Server can be used as input to a WebSphere QualityStage job by using connectors. The connectors also enable access to the common repository from the processing engine.

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

“Overview of the Designer, Director, and Administrator clients” on page 89

Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

WebSphere QualityStage tasks

WebSphere QualityStage helps establish a clear understanding of data and uses best practices to improve data quality.

As shown in Figure 39 on page 71, providing quality data has four stages:

Data investigation

To fully understand information.

Data standardization

To fully cleanse information.

Data matching

To create semantic keys to identify information relationships.

Data survivorship

To build the best available view of related information.

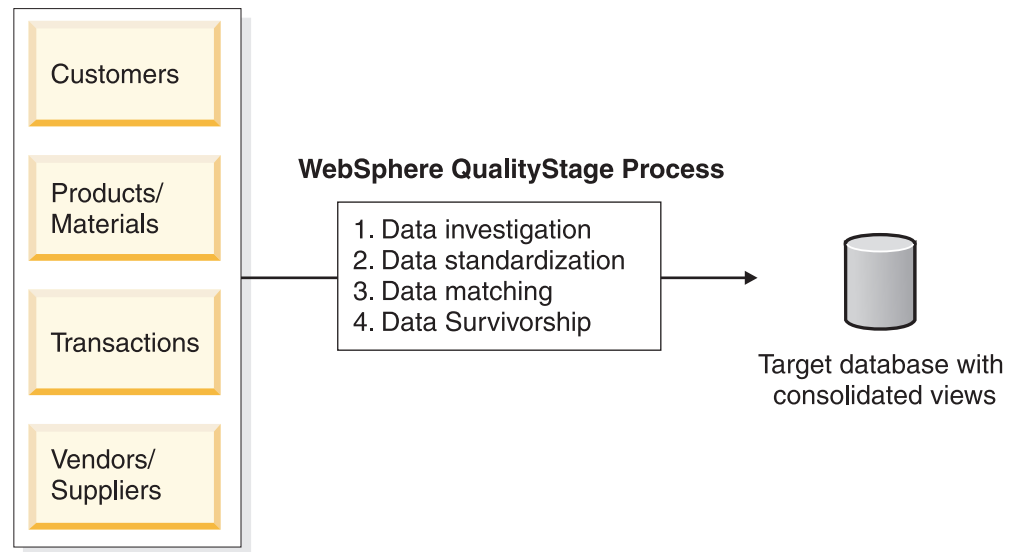


Figure 39. Steps in the WebSphere QualityStage process

Investigate stage

Understanding your data is a necessary precursor to cleansing. You can use WebSphere Information Analyzer to create a direct input into the cleansing process by using shared metadata, or use the Investigate stage to create this input.

The Investigate stage shows the actual condition of data in legacy sources and identifies and corrects data problems before they corrupt new systems. Investigation parses and analyzes free-form fields, counts unique values, and classifies or assigns a business meaning to each occurrence of a value within a field.

Investigation achieves these goals:

- Uncovers trends, potential anomalies, metadata discrepancies, and undocumented business practices.
- Identifies invalid or default values.
- Reveals common terminology.
- Verifies the reliability of fields proposed as matching criteria.

The Investigate stage takes a single input, which can be a link from any database connector that is supported by WebSphere DataStage, from a flat file or data set, or from any processing stage. Inputs to the Investigate stage can be fixed length or variable.

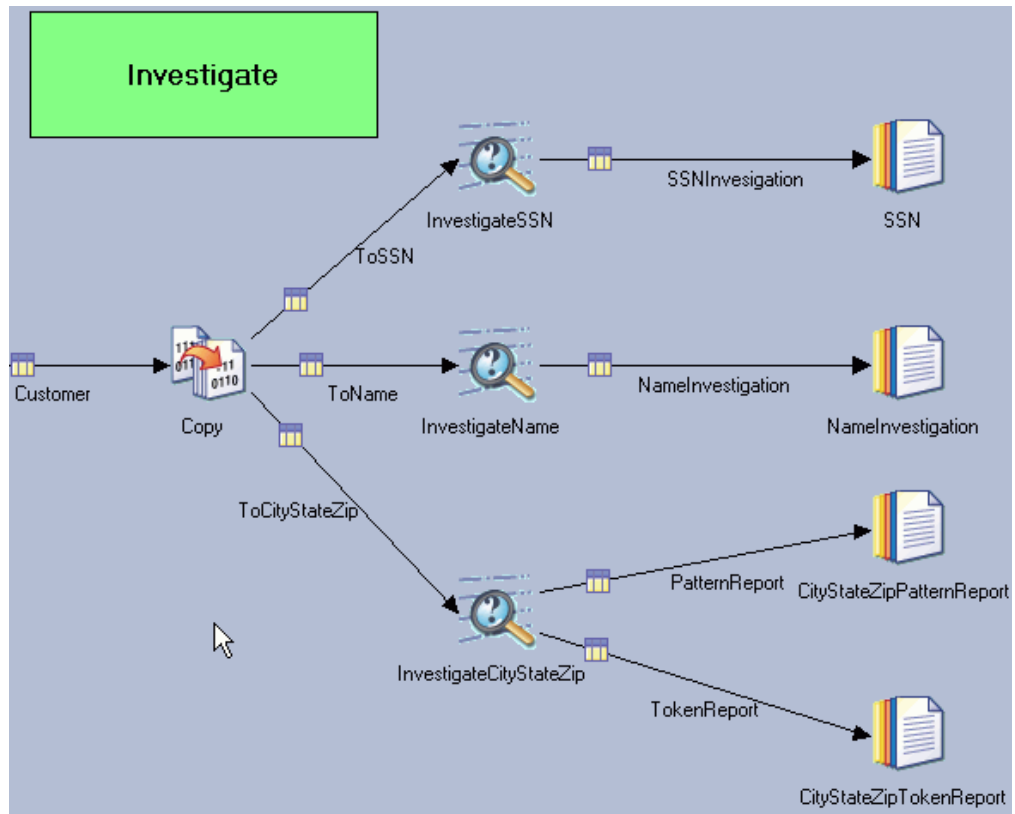


Figure 40. Designing the Investigate stage

As Figure 40 shows, you use the WebSphere DataStage and QualityStage Designer to specify the Investigate stage. The stage can have one or two output links, depending on the type of investigation that you specify.

The Word Investigation stage parses free-form data fields into individual tokens and analyzes them to create patterns. This stage also provides frequency counts on the tokens. To create the patterns in address data, for example, the Word Investigation stage uses a set of rules for classifying personal names, business names, and addresses. The stage provides pre-built rule sets for investigating patterns on names and postal addresses for a number of different countries. For example, for the United States the stage parses the following components:

USPREP

Name, address, and area if the data is not previous formatted

USNAME

Individual and organization names

USADDR

Street and mailing addresses

USAREA

City, state, ZIP code, and so on

The test field 123 St. Virginia St. is analyzed in the following way:

1. Field parsing would break the address into the individual tokens of 123, St., Virginia, and St.
2. Lexical analysis determines the business significance of each piece:
 - a. 123 = number

- b. St. = street type
 - c. Virginia = alpha
 - d. St. = Street type
3. Context analysis identifies the various data structures and content as 123 St. Virginia, St.
- a. 123 = House number
 - b. St. Virginia = Street address
 - c. St. = Street type

The Character Investigation stage parses a single-domain field (one that contains one data element or token, such as Social Security number, telephone number, date, or ZIP code) to analyze and classify data. The Character Investigation stage provides a frequency distribution and pattern analysis of the tokens.

A pattern report is prepared for all types of investigations and displays the count, percentage of data that matches this pattern, the generated pattern, and sample data. This output can be presented in a wide range of formats to conform to standard reporting tools.

Related concepts

“Overview of the Designer, Director, and Administrator clients” on page 89
 Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

Standardize stage

Based on an understanding of data from the Investigation stage, you can apply out-of-the-box rules with the Standardize stage to reformat data from multiple systems. This stage facilitates effective matching and output formatting.

WebSphere QualityStage can transform any data type into your desired standards. It applies consistent representations, corrects misspellings, and incorporates business or industry standards. It formats data, places each value into a single domain field, and transforms data into a standard format.

As Figure 41 on page 74 shows, you can select from predefined rules to apply the appropriate standardization for the data set.

Select the rule set, then specify the columns the rule set is to standardize

Rule Set:

Available Columns:

Column Name	Description
AddressLine1	Address Line 1
AddressLine2	Address Line 2
ApplicantSSN	Applicant SSN
City	City
State	State

Selected Columns:

Column Name
Name

Literal:

Optional NAMES Handling:

Figure 41. Standardize rule process

Match stages overview

Data matching finds records in a single data source or independent data sources that refer to the same entity (such as a person, organization, location, product, or material) even if there is no predetermined key.

To increase its usability and completeness, data can be consolidated or linked along any relationship, such as a common person, business, place, product, part, or event. You can also use matching to find duplicate entities that are caused by data entry variations or account-oriented business practices.

During the data matching stage, WebSphere QualityStage takes these actions:

- Identifies duplicate entities (such as customers, suppliers, products, or parts) within one or more data sources
- Creates a consolidated view of an entity according to business rules
- Provides householding for individuals (such as a family or group of individuals at a location) and householding for commercial entities (multiple businesses in the same location or different locations)
- Enables the creation of match groups across data sources that might or might not have a predetermined key
- Enriches existing data with new attributes from external sources such as credit bureau data or change of address files

Match frequency stage

The Match Frequency stage gives you direct control over the disposition of generated frequency data. This stage provides results that can be used by the Match Designer and match stages, but enables you to generate the frequency data independent of running the matches.

You can generate frequency information by using any data that provides the fields that are needed by a match. Then you can let the generated frequency data flow

into a match stage, store it for later use, or both. Figure 42 shows how Standardize stage and Match Frequency stage are added in the Designer client.

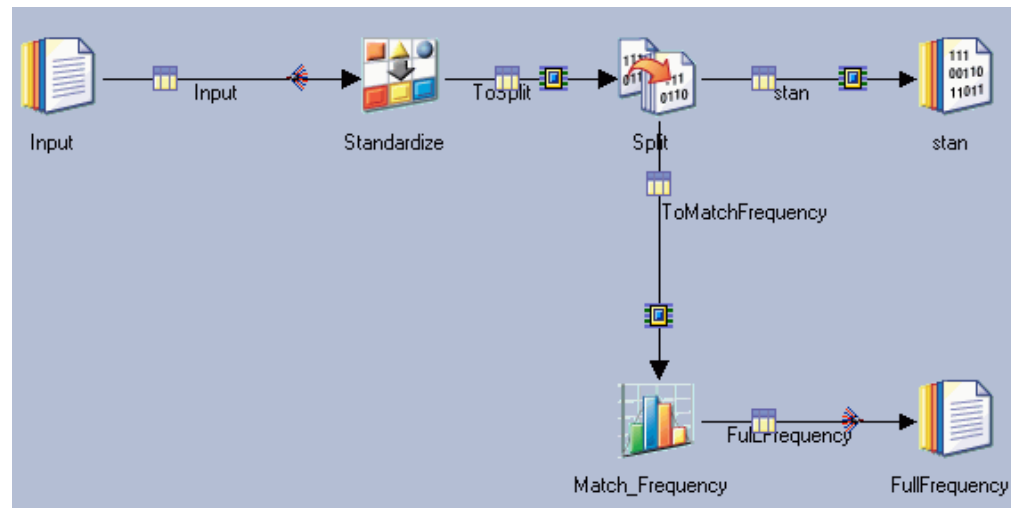


Figure 42. Designing a job with Standardize and Match Frequency stages

In this example, input data is being processed in the Standardize stage with a rule set that creates consistent formats. The data is then split into two data streams. One stream passes data to a standard output and the other passes data to the Match Frequency stage.

Related concepts

“Overview of the Designer, Director, and Administrator clients” on page 89
Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

Match stage

Matching is a two-step process: first you block records and then you match them.

Blocking step

Blocking identifies subsets of data in which matches can be more efficiently performed. Blocking limits the number of record pairs that are being examined, which increases the efficiency of the matching.

To understand the concept of blocking, consider a column that contains age data. If there are 100 possible ages, blocking partitions a source into 100 subsets. The first subset is all people with an age of zero, the next is people with an age of 1, and so on. These subsets are called blocks.

If the age values are uniformly distributed, 10 records out of the 1000-record source contain data for people of age 0 on each source, 10 records for people of age 1, and so on. The pairs of records to be compared are taken from records in the same block. The first block consists of all people of age 0 on each data source. This is 10 times 10 or 100 record pairs. The second block consists of all people on each data source with an age of 1.

When the process is complete, you compared 100 (blocks) x 100 (pairs in a block) = 10,000 pairs, rather than the 1,000,000 record pairs that are required without blocking. You can also combine multiple blocking variables into a single block for a single pass. For example, blocking on age

and gender divides the sources into sets of 0-year-old males, 0-year-old females, 1-year-old males, 1-year-old females, and so on.

Matching step

The strategy that you choose to match data depends on your data reengineering goals. There are two types of Match stage:

Reference match

Identifies relationships among records. This match can group records that are being compared in different ways:

One-to-one matching

Identifies all records in one data source that correspond to a record for the same individual, event, household, or street address in a second data source. Only one record in the reference source can match one record in the data source because the matching applies to individual events.

Many-to-one matching

Multiple records in the data file can match a single record in the reference file. For example, matching a transaction data source to a master data source allows many transactions for one person in the master data source.

Unduplicate Match

Locates and groups all similar records within a single input data source. This process identifies potential duplicate records, which might then be removed.

Related concepts

“Overview of the Designer, Director, and Administrator clients” on page 89
Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

Tasks in the Match Designer

The Match Designer is a tool for creating a match specification. The main area of the Match Designer is made up of two tabs. On the Compose tab, you design and fine tune the match passes. On the Total Statistics tab, you provide cumulative and individual statistics for match passes.

You can use the Match Designer to create multiple match specifications that include one or more passes. Each pass is separately defined and is stored in the repository to be reused. Within each pass of a match specification, you define the blocking fields and match commands.

You can run each pass on test data that is created from a representative subset of your production data and view the results in a variety of graphic displays. You can also use frequency information that is generated by the Match Frequency stage to help create your match specifications.

Compose

On the Compose tab, you design the Match passes and add them to the Match job. You can add, delete, and modify Match passes in this section. For each Match pass, you can specify blocking fields, matching fields, cutoff weights, and view the weight histogram, data results, and Match pass statistics.

The top pane in the Compose tab has two sections: the **Match Type** area (shown in Figure 43) and the **Match Pass Holding Area**.

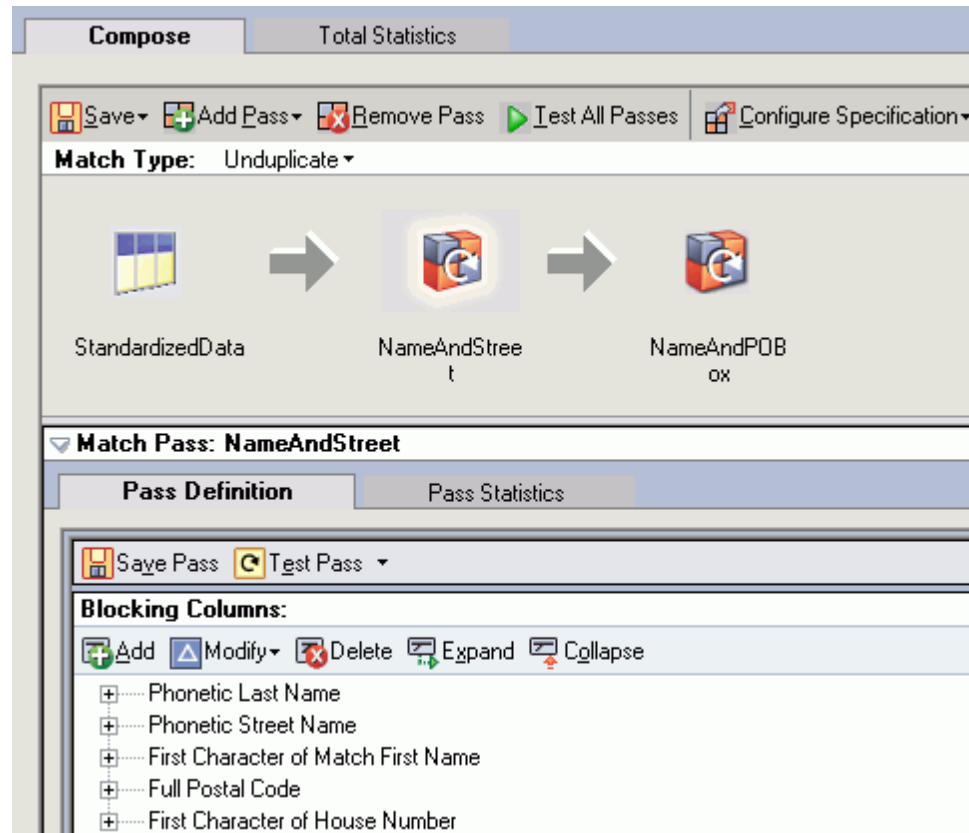


Figure 43. Compose tab of the Match Designer

The **Match Type** area is a kind of sandbox for designing jobs that displays the current Match job. In this area, you can rearrange the order in which the Match passes run in the Match job, add or remove passes, and create new passes.

The **Match Pass Holding Area** is used to keep iterations of a particular pass definition or alternate approaches to a pass. The passes in the holding area do not run as part of the match job, but can be tested in isolation. You can add any of the Match passes in the holding area to the Match job by moving the Match pass to the **Match Type** area. Also, you can remove any of the Match passes from the Match job by moving it from the type area into the **Match Pass Holding Area**. Any pass, whether in the type or holding areas, can be test run. This approach lets you perform trial runs of different pass definitions without needing to lose alternate definitions.

The **Blocking Columns** area designates the fields that must match exactly for records to be in the same processing group for the match. The right pane shows the histogram and data sections when the run is complete.

You can sort and search the data columns from the match results. You can also display weight comparisons of the selected records that are based on the last match run or the current match settings. Figure 44 on page 78 shows a pie chart that was built by using the results for pseudo matches, clerical pairs, and data residuals.

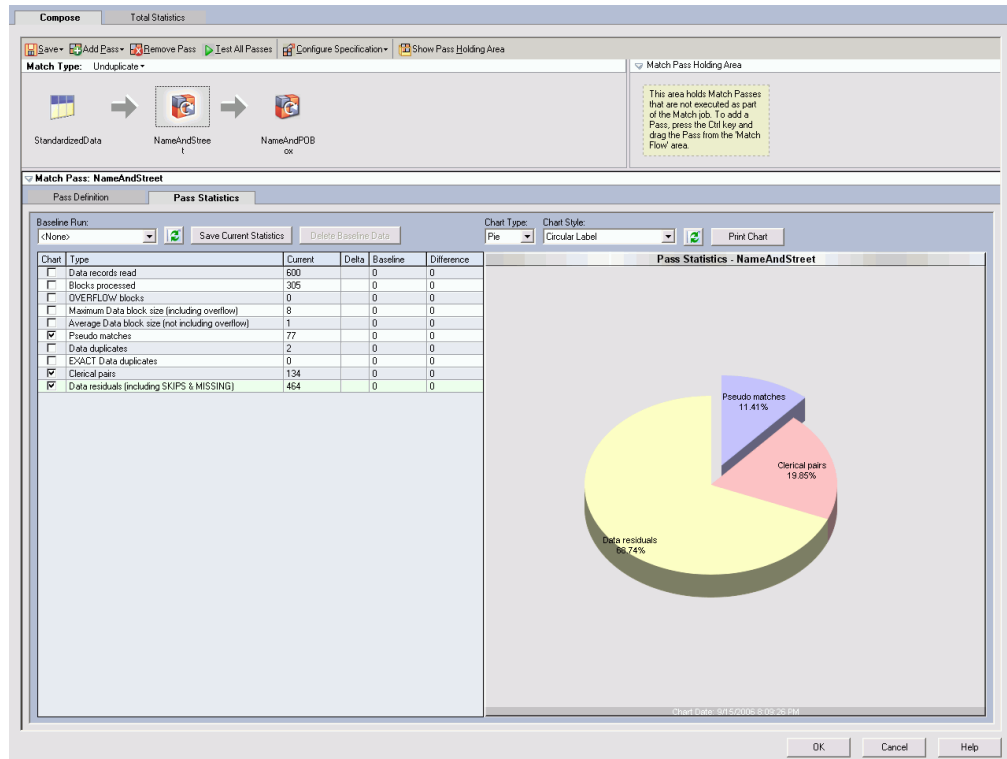


Figure 44. Pass Statistics tab of the Match Designer

The Total statistics tab displays cumulative statistics for the Match job and statistics for individual Match passes for the most recent run of the Match.

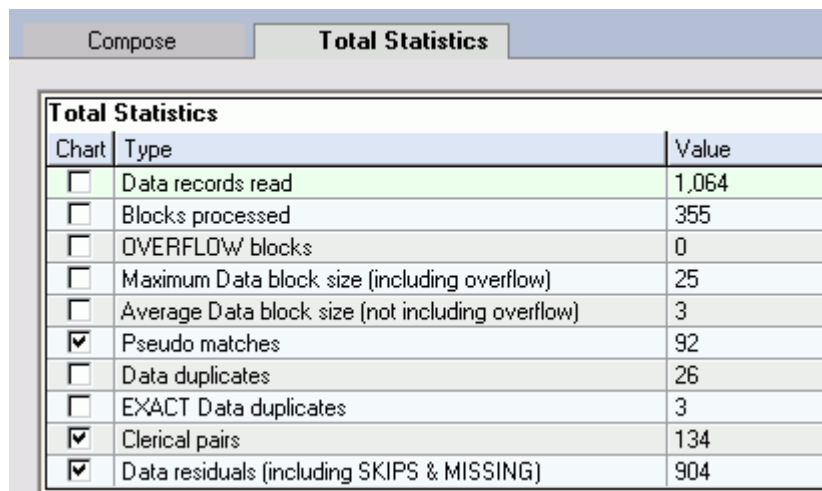


Figure 45. Total statistics tab

Related concepts

“Overview of the Designer, Director, and Administrator clients” on page 89
Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

Survive stage

The Survive stage consolidates duplicate records, which creates a best-of-breed representation of the matched data.

The Survive stage implements the business and mapping rules, creating the necessary output structures for the target application and identifying fields that do not conform to load standards.

Survivorship consolidates duplicate records, creating the best representation of the match data so companies can use it to load a master data record, cross-populate all data sources, or both. During the Survive stage, WebSphere QualityStage takes the following actions:

- Supplies missing values in one record with values from other records on the same entity
- Populates missing values in one record with values from corresponding records which have been identified as a group in the matching stage
- Enriches existing data with external data

Figure 46 shows a Survive stage called BTSURV. This stage is used as part of a job in which matched input data from a sequential file is acting as input, and the survived data is moved to a sequential file.

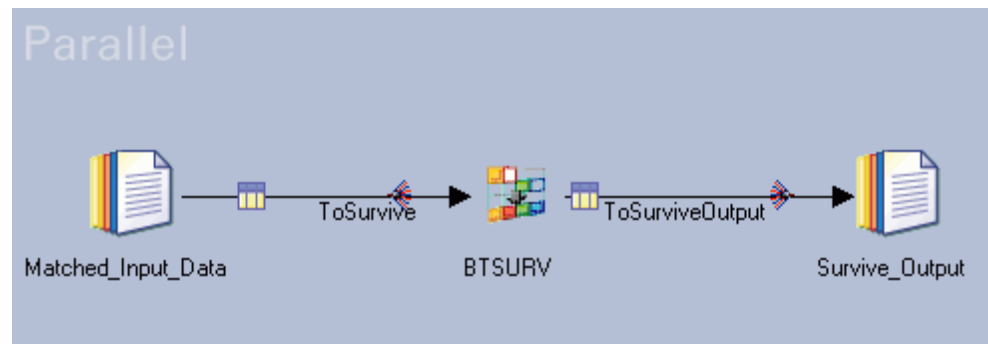


Figure 46. Designing the Survive stage

Accessing metadata services

WebSphere DataStage and WebSphere QualityStage users can access the WebSphere Metadata Server to obtain live access to current metadata about integration projects and enterprise data. Data generated by WebSphere MetaBrokers or WebSphere Information Analyzer is accessible from the WebSphere DataStage and QualityStage Designer.

The following services provide designers with access to metadata:

Simple and advanced find

Enables the WebSphere QualityStage user to search the repository for objects. Two functions are available: a simple find capability and a more complex advanced find capability.

Where used or impact analysis

Enables the WebSphere QualityStage user to show both “used by” and “depends on” relationships.

Job, Table, or Routine Difference

Enables the WebSphere QualityStage user to see difference reports that show change in integration processes or data.

Related concepts

Chapter 3, “Metadata services,” on page 17

When moving to an enterprise integration strategy, large organizations often face a proliferation of software tools that are built to solve identical problems. Few of these tools work together, much less work across problem domains to provide an integrated solution.

Information resources for WebSphere QualityStage

A variety of information resources can help you get started with WebSphere QualityStage.

Planning, installation, and configuration details for WebSphere QualityStage and other IBM Information Server suite components are available in the IBM Information Server information center at <http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp>.

Online help for the WebSphere QualityStage client interfaces is available in HTML format. The following documentation in PDF format is available from the Windows **Start** menu and the Quick Start CD:

- *WebSphere QualityStage Tutorial*
- *Migrating to WebSphere QualityStage Version 8*
- *WebSphere QualityStage User Guide*
- *WebSphere QualityStage Pattern - Action Reference*
- *WebSphere QualityStage Clerical Review Guide*
- *WebSphere QualityStage CASS Certified Stage Guide*

Planning, installation, and configuration details for IBM Information Server and its suite components are also available in the following PDFs that you can access from the Windows **Start** menu and the Quick Start CD:

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Chapter 7. WebSphere DataStage

Data transformation and movement is the process by which source data is selected, converted, and mapped to the format required by targeted systems. The process manipulates data to bring it into compliance with business, domain, and integrity rules and with other data in the target environment.

Transformation can take some of the following forms:

Aggregation

Consolidating or summarizing data values into a single value. Collecting daily sales data to be aggregated to the weekly level is a common example of aggregation.

Basic conversion

Ensuring that data types are correctly converted and mapped from source to target columns.

Cleansing

Resolving inconsistencies and fixing the anomalies in source data.

Derivation

Transforming data from multiple sources by using an algorithm.

Enrichment

Combining data from internal or external sources to provide additional meaning to the data.

Normalizing

Reducing the amount of redundant and potentially duplicated data.

Pivoting

Converting records in an input stream to many records in the appropriate table in the data warehouse or data mart.

Sorting

Sequencing data based on data or string values.

WebSphere DataStage supports the collection, transformation and distribution of large volumes of data, with data structures that range from simple to highly complex. WebSphere DataStage manages data that arrives and data that is received on a periodic or scheduled basis. WebSphere DataStage enables companies to solve large-scale business problems with high-performance processing of massive data volumes.

By leveraging the parallel processing capabilities of multiprocessor hardware platforms, WebSphere DataStage can scale to satisfy the demands of ever-growing data volumes, stringent real-time requirements, and ever-shrinking batch windows.

Introduction to WebSphere DataStage

WebSphere DataStage has the functionality, flexibility, and scalability that are required to meet the most demanding data integration requirements.

WebSphere DataStage has the following capabilities:

- Integrates data from the widest range of enterprise and external data sources

- Incorporates data validation rules
- Processes and transforms large amounts of data using scalable parallel processing
- Handles very complex transformations
- Manages multiple integration processes
- Provides direct connectivity to enterprise applications as sources or targets
- Leverages metadata for analysis and maintenance
- Operates in batch, real time, or as a Web service

Scenarios for data transformation

The following scenarios show how organizations use WebSphere DataStage to address complex data transformation and movement needs.

Retail: Consolidating financial systems

A leading retail chain watched sales flatten for the first time in years. Without insight into store-level and unit-level sales data, they could not adjust shipments or merchandising to improve results. With long production lead-times and existing large volume manufacturing contracts, they could not change their product lines quickly, even if they understood the problem. To integrate the company's forecasting, distribution, replenishment, and inventory management processes, they needed a way to migrate financial reporting data from many systems to a single system of record.

The company deployed IBM Information Server to deliver data integration services between business applications in both messaging and batch file environment. WebSphere DataStage is now the common companywide standard for transforming and moving data. The service-oriented interface allows them to define common integration tasks and reuse them throughout the enterprise. New methodology and reusable components for other global projects will lead to additional future savings in design, testing, deployment and maintenance.

Banking: Understanding the customer

A large retail bank understood that the more it knew about its customers, the better it could market its products, including credit cards, savings accounts, checking accounts, certificates of deposit, and ATM services. Faced with terabytes of customer data from vendor sources, the bank recognized the need to integrate the data into a central repository where decision-makers could retrieve it for market analysis and reporting. Without a solution, the bank risked flawed marketing decisions and lost cross-selling opportunities.

The bank used WebSphere DataStage to automatically extract and transform raw vendor data, such as credit card account information, banking transaction details and Web site usage statistics, and load it into its data warehouse. From there, the company can generate reports that let them track the effectiveness of programs and analyze their marketing efforts. WebSphere DataStage helps the bank maintain, manage, and improve its information management with an IT staff of three instead of six or seven, saving hundreds of thousands of dollars in the first year alone, and enabling it to use the same capabilities more rapidly on other data integration projects.

Where WebSphere DataStage fits in the overall business context

WebSphere DataStage enables an integral part of the information integration process: data transformation, as Figure 47 shows:

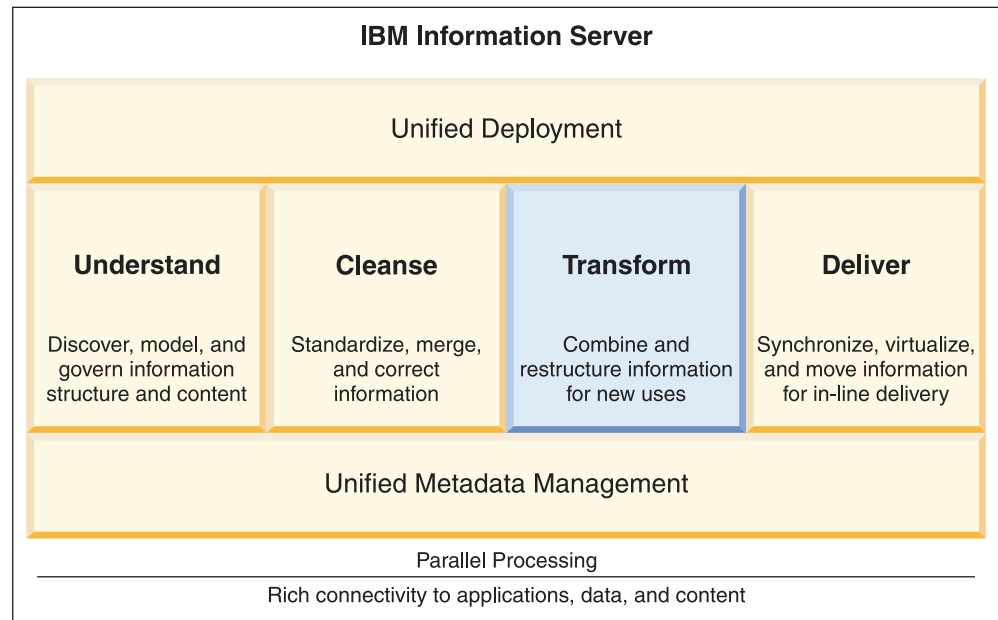


Figure 47. Transformation as part of the integration process

WebSphere DataStage is often deployed to systems such as enterprise applications, data warehouses, and data marts. WebSphere DataStage provides this functionality with extensive capabilities:

- Enables the movement and transformation of data between operational, transactional, and analytical targets
- Helps a company determine how best to integrate data, either in batch or in real time, to meet its business requirements
- Saves time and improves consistency of design, development, and deployment

Related concepts

Chapter 1, "Introduction," on page 1

Most of today's critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

A closer look at WebSphere DataStage

In its simplest form, WebSphere DataStage performs data transformation and movement from source systems to target systems in batch and in real time.

The data sources might include indexed files, sequential files, relational databases, archives, external data sources, enterprise applications, and message queues. Some of the following transformations might be involved:

- String and numeric formatting and data type conversions.

- Business derivations and calculations that apply business rules and algorithms to the data. Examples range from straightforward currency conversions to more complex profit calculations.
- Reference data checks and enforcement to validate customer or product identifiers. This process is used in building a normalized data warehouse.
- Conversion of reference data from disparate sources to a common reference set, creating consistency across these systems. This technique is used to create a master data set (or conformed dimensions) for data about products, customers, suppliers, and employees.
- Aggregations for reporting and analytics.
- Creation of analytical or reporting databases, such as data marts or cubes. This process involves denormalizing data into such structures as star or snowflake schemas to improve performance and ease of use for business users.

WebSphere DataStage can also treat the data warehouse as the source system that feeds a data mart as the target system, usually with localized, subset data such as customers, products and geographic territories.

WebSphere DataStage delivers four core capabilities:

- Connectivity to a wide range of mainframe, legacy, and enterprise applications, databases, and external information sources
- Prebuilt library of more than 300 functions
- Maximum throughput using a parallel, high-performance processing architecture
- Enterprise-class capabilities for development, deployment, maintenance, and high-availability

Where WebSphere DataStage fits within the IBM Information Server architecture

WebSphere DataStage is composed of client-based design, administration, and operation tools that access a set of server-based data integration capabilities through a common services layer. Figure 48 shows the clients that comprise the WebSphere DataStage user interface layer.

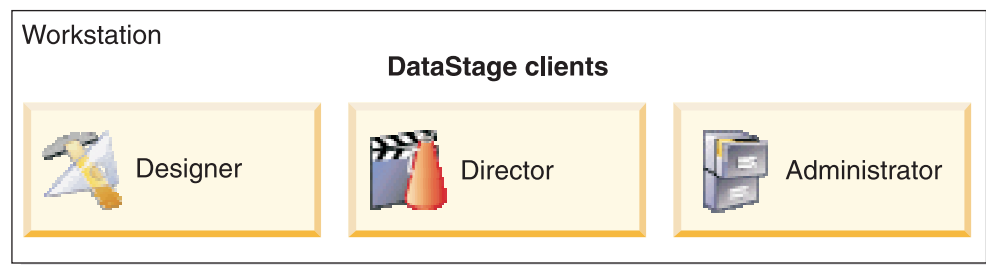


Figure 48. WebSphere DataStage clients

Figure 49 on page 85 shows the elements that make up the server architecture.

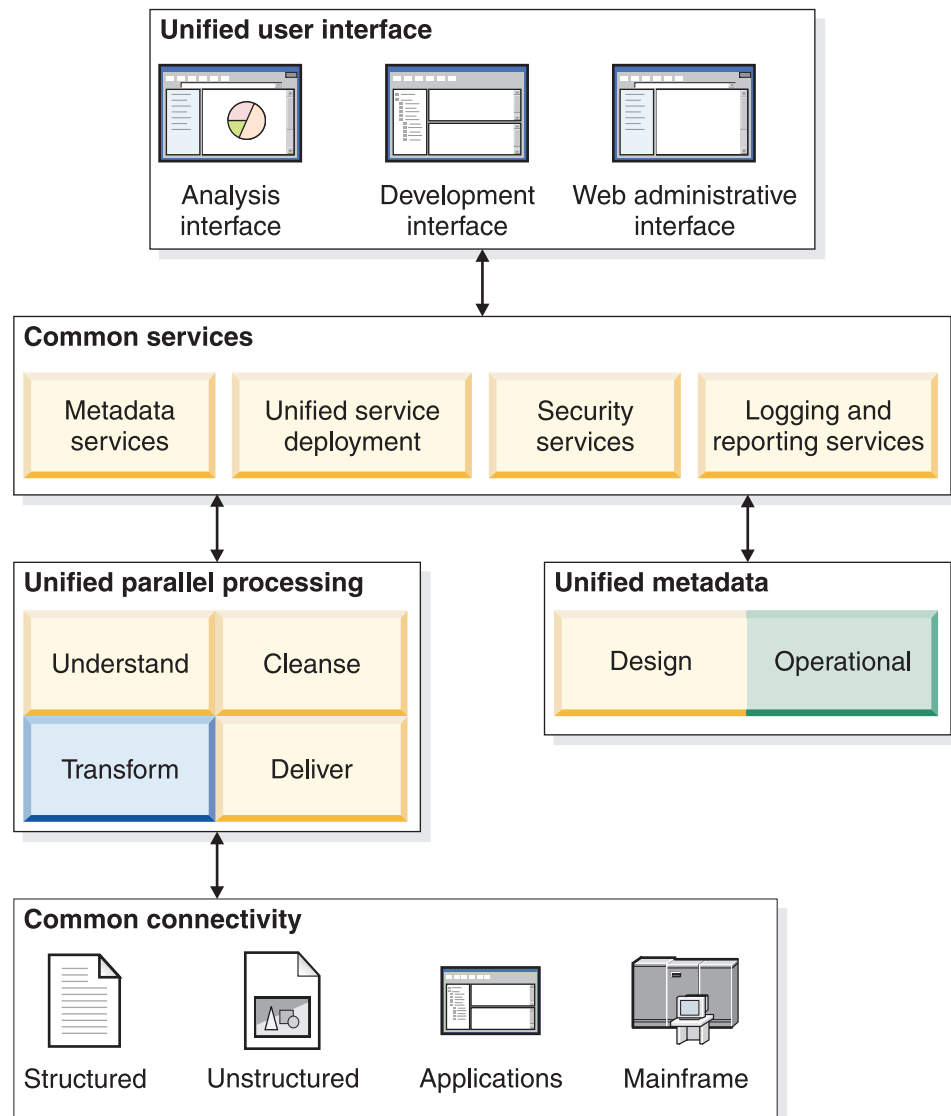


Figure 49. Server architecture

WebSphere DataStage architecture includes the following components:

Common user interface

The following client applications comprise the WebSphere DataStage user interface:

WebSphere DataStage and QualityStage Designer

A graphical design interface that is used to create WebSphere DataStage applications (known as jobs). Because transformation is an integral part of data quality, the WebSphere DataStage and QualityStage Designer is the design interface for both WebSphere DataStage and WebSphere QualityStage.

Each job specifies the data sources, the required transformations, and the destination of the data. Jobs are compiled to create executables that are scheduled by the WebSphere DataStage and QualityStage Director and run on the WebSphere DataStage server. The Designer client writes development metadata to the dynamic

repository while compiled execution data that is required for deployment is written to the WebSphere Metadata Server repository.

WebSphere DataStage and QualityStage Director

A graphical user interface that is used to validate, schedule, run, and monitor WebSphere DataStage job sequences. The Director client views data about jobs in the operational repository and sends project metadata to WebSphere Metadata Server to control the flow of WebSphere DataStage jobs.

WebSphere DataStage and WebSphere QualityStage Administrator

A graphical user interface that is used for administration tasks such as setting up IBM Information Server users; logging, creating, and moving projects; and setting up criteria for purging records.

Common services

The multiple discrete services of WebSphere DataStage give the flexibility that is needed to configure systems that support increasingly varied user environments and tiered architectures. The common services provides flexible, configurable interconnections among the many parts of the architecture:

- Metadata services such as impact analysis and search
- Execution services that support all WebSphere DataStage functions
- Design services that support development and maintenance of WebSphere DataStage tasks

Common repository

The common repository holds three types of metadata that are required to support WebSphere DataStage:

Project metadata

All the project-level metadata components including WebSphere DataStage jobs, table definitions, built-in stages, reusable subcomponents, and routines are organized into folders.

Operational metadata

The repository holds metadata that describes the operational history of integration process runs, success or failure of jobs, parameters that were used, and the time and date of these events.

Design metadata

The repository holds design time metadata that is created by the WebSphere DataStage and QualityStage Designer and WebSphere Information Analyzer.

Common parallel processing engine

The engine runs executable jobs that extract, transform, and load data in a wide variety of settings. The engine uses parallelism and pipelining to handle high volumes of work more quickly.

Common connectors

The connectors provide connectivity to a large number of external resources and access to the common repository from the processing engine. Any data source that is supported by IBM Information Server can be used as input to or output from a WebSphere DataStage job.

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all

types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

WebSphere DataStage tasks

The key elements of WebSphere DataStage are *jobs*, which define the sequence of transformation steps, and *stages*, the individual steps that make up jobs. Using WebSphere DataStage involves designing, executing, managing and deploying, and administering WebSphere DataStage jobs.

WebSphere DataStage elements

The central WebSphere DataStage elements are projects, jobs, stages, links, containers, and table definitions.

Projects

WebSphere DataStage is a project-based development environment that you initially create with the WebSphere DataStage Administrator. During installation, or when you start a WebSphere DataStage client tool (with the exception of the Administrator), you can create a project. Each project contains all of the WebSphere DataStage components including jobs and stages, table definitions, and links and containers.

Jobs and stages

Jobs define the sequence of steps that determine how IBM Information Server performs its work. After they are designed, jobs are compiled and run on the parallel processing engine. The engine runs functions such as connectivity, extraction, cleansing, transformation, and data loading based on the design of the job.

The individual steps that make up a job are called stages. IBM Information Server offers dozens of prebuilt stages for performing most common data integration tasks such as sort, merge, join, filter, transform, lookup, and aggregate. The stages include powerful components for high-performance access to relational databases for reading and loading, including parallel relational databases.

Stages typically provide 80 percent to 90 percent of the application logic that is required for most enterprise data integration applications. IBM Information Server also provides a number of stage types for building and integrating custom stages:

Wrapped stage

Enables you to run an existing sequential program in parallel

Build stage

Enables you to write a C expression that is automatically generated into a parallel custom stage.

Custom stage

Provides a complete C++ API for developing complex and extensible stages.

Figure 50 on page 88 shows a simple job that consists of a data source, a Transformer (conversion) stage, and the target database. The links between the stages represent the flow of data into or out of a stage.



Figure 50. Simple example of a WebSphere DataStage job

WebSphere DataStage provides a wide variety of stages. Table 2 describes some representative examples.

Table 2. Examples of stages

Icon	Stage	Description
	Transformer stage	Performs any required conversions on an input data set, and then passes the data to another processing stage or to a stage that writes data to a target database or file.
	Sort stage	Performs complex high-speed sort operations.
	Aggregator stage	Classifies data rows from a single input data set into groups and computes totals or aggregations for each group.
	Complex Flat File stage	Extracts data from a flat file containing complex data structures, such as arrays or groups.
	DB2 stage	Reads data from or writes data to IBM DB2.

Each stage has properties that tell it how to perform or process data. Properties might include the file name for the Sequential File stage, the columns to sort, the transformations to perform, and the database table name for the DB2 stage.

The WebSphere DataStage plug-in architecture makes it easy for WebSphere software and vendors to add stages, such as additional connectivity.

Table definitions

Table definitions are the record layout (or schema) and other properties of the data that you process. Table definitions contain column names, data type, length, and other column properties including keys and null values. You can import table definitions from databases, COBOL copybooks, and other sources, using the Designer client. These table definitions are then used within the links to describe the data that flows between stages.

Links and containers

In WebSphere DataStage, links join the various stages in a job that describe the flow of data and the data definitions from a data source through the processing stages to the data target. Input links that are connected to the stage generally carry data to the stage. Output links carry data that is processed by the stage.

Containers hold user-defined groupings of stages, or links that you can reuse. Containers make it easier to share a workflow. There are two types of containers:

Shared

Reusable job elements that typically comprise a number of stages and links

Local

Elements that are created within a job and are accessible only by that job.

A local container, edited in a tabbed page of the job's diagram window, can be used to "clean up" the diagram to isolate areas of the flow.

Overview of the Designer, Director, and Administrator clients

Three interfaces simplify the task of designing, executing, managing and deploying, and administering WebSphere DataStage jobs: the WebSphere DataStage and QualityStage Designer, WebSphere DataStage and QualityStage Director, and WebSphere DataStage and QualityStage Administrator.

WebSphere DataStage and QualityStage Designer

The WebSphere DataStage and QualityStage Designer helps you create, manage, and design jobs. You can also use the Designer client to define tables and access metadata services.

Table definitions

You can import, create, and edit table definitions from many sources (for example, one source of table definitions is metadata from WebSphere Information Analyzer). When you edit or view a table, the Table Definitions window opens, as Figure 51 on page 90 shows.

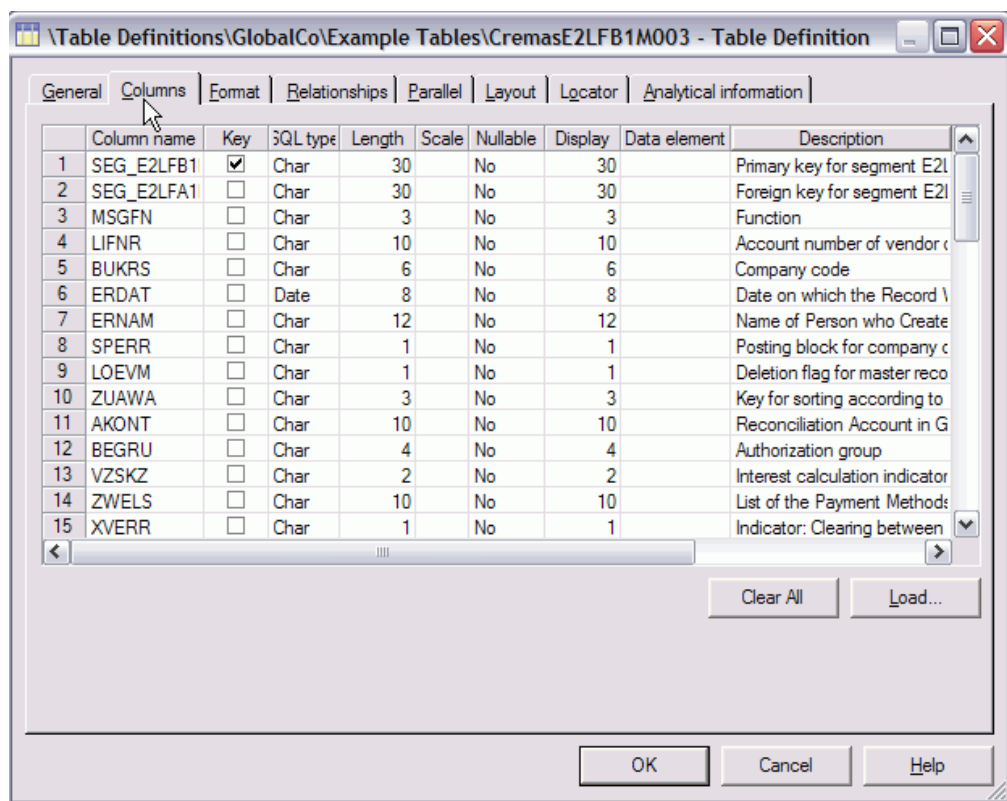


Figure 51. Table Definitions window

This window has the following pages:

General

Contains data source and description information.

Columns

Contains information about the columns including key values, SQL type, and length.

Format

Contains information that describes data format when the data is read from or written to a sequential file.

Relationships

Provides foreign key information about the table.

Parallel

Shows extended properties for table definitions that you can use in parallel jobs.

Layout

Shows the schema format of the column definitions in a table.

Locator

Enables you to view and edit the data resource locator that is associated with the table definition. The data resource locator describes the real-world object.

NLS (if installed)

Shows the current character set map for the table definitions.

Analytical information

Shows metadata that WebSphere Information Analyzer generated.

Accessing metadata services

WebSphere DataStage and WebSphere QualityStage access WebSphere Metadata Server to obtain live access to current metadata about integration projects and your organization's enterprise data. You access data that is generated by WebSphere MetaBrokers or WebSphere Information Analyzer by using the Designer client. The following services provide designers with access to metadata:

Simple and advanced find service

Enables you to search the repository for objects

Where used or impact analysis service

Shows both "used by" and "depends on" relationships

An option in the WebSphere DataStage and QualityStage Designer shows differences between jobs or table definitions in a WebSphere DataStage context. Figure 52 shows a textual report with links to the relevant editor in the Designer client. You can also view differences for subsets of jobs such as shared containers and routines. This report can optionally be saved as an XML file.

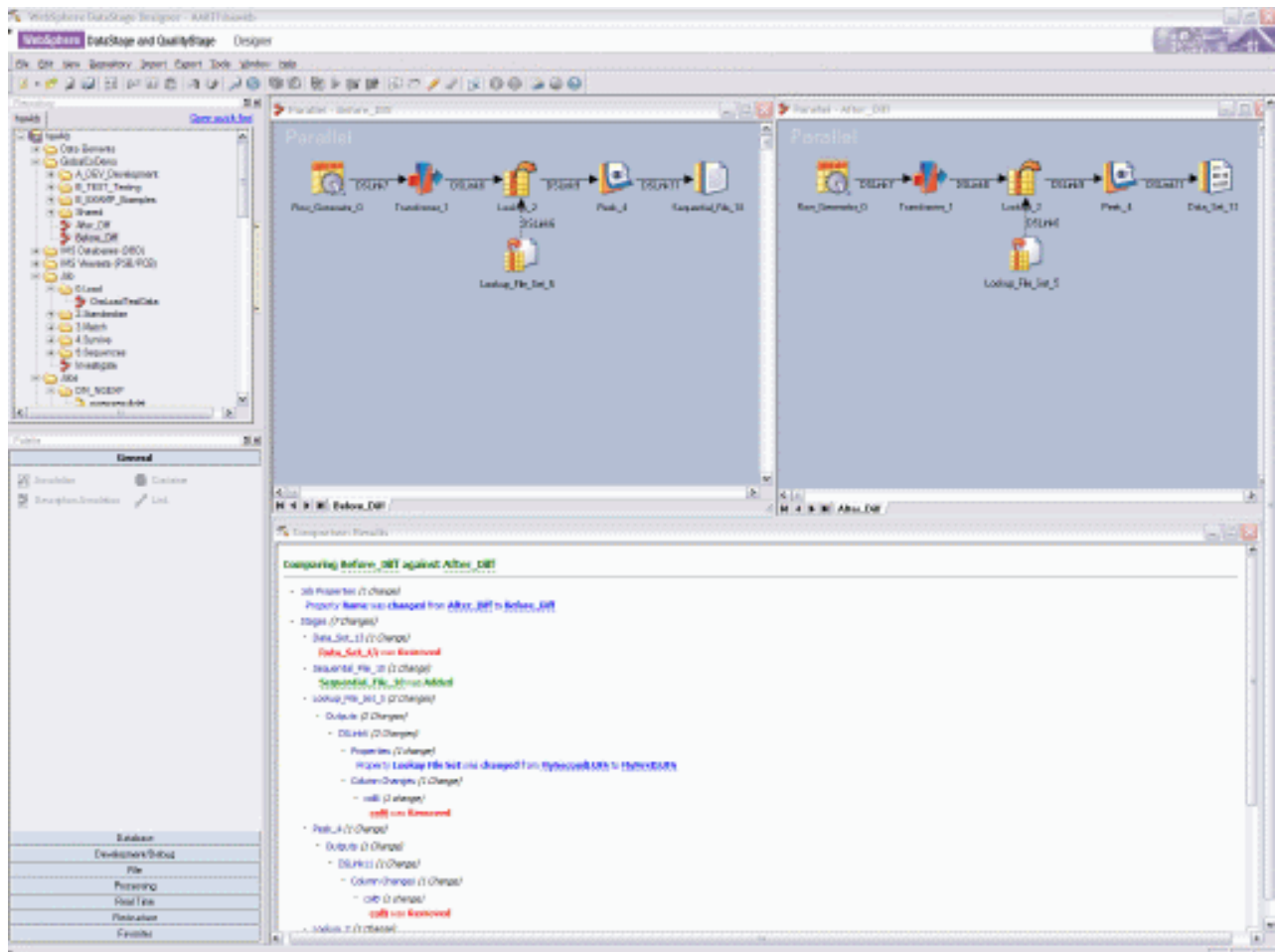


Figure 52. Job difference report

Creating jobs

When you use the Designer client, you choose the type of job to create and how to create it, as Figure 53 shows.

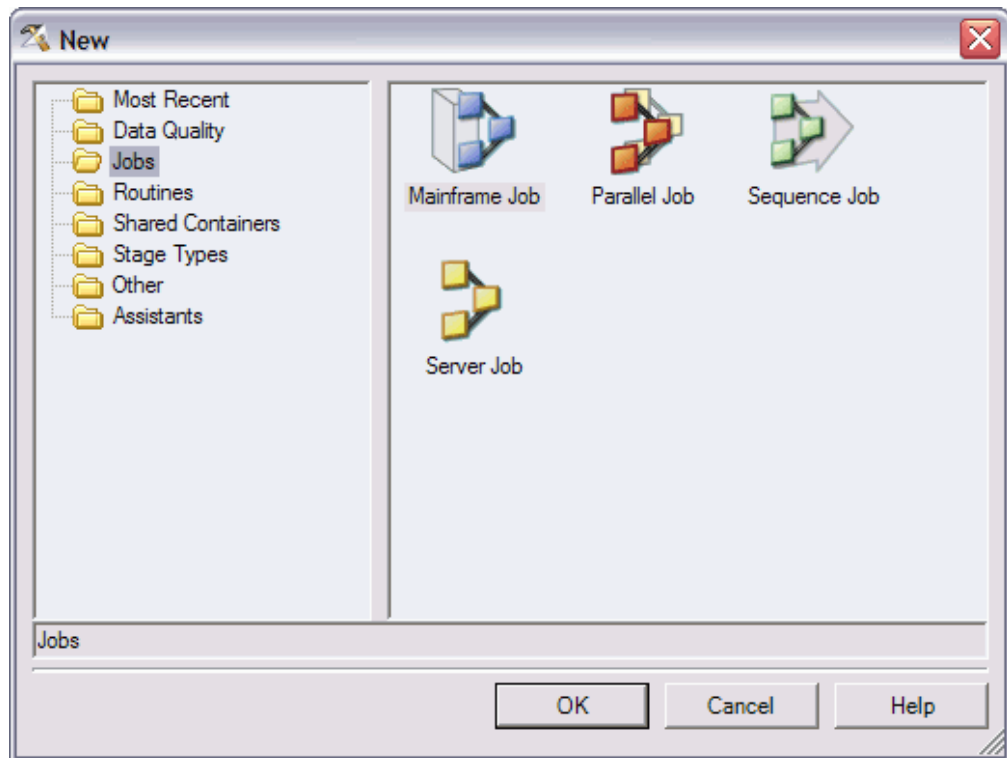


Figure 53. Choosing a job type

Different job types include parallel, mainframe, and job sequences. Job templates help you build jobs quickly by providing predefined job properties that you can customize. Job templates also provide a basis for commonality between jobs and job designers.

You use the design canvas window and tool palette to design, edit, and save the job, as shown in Figure 54 on page 93.

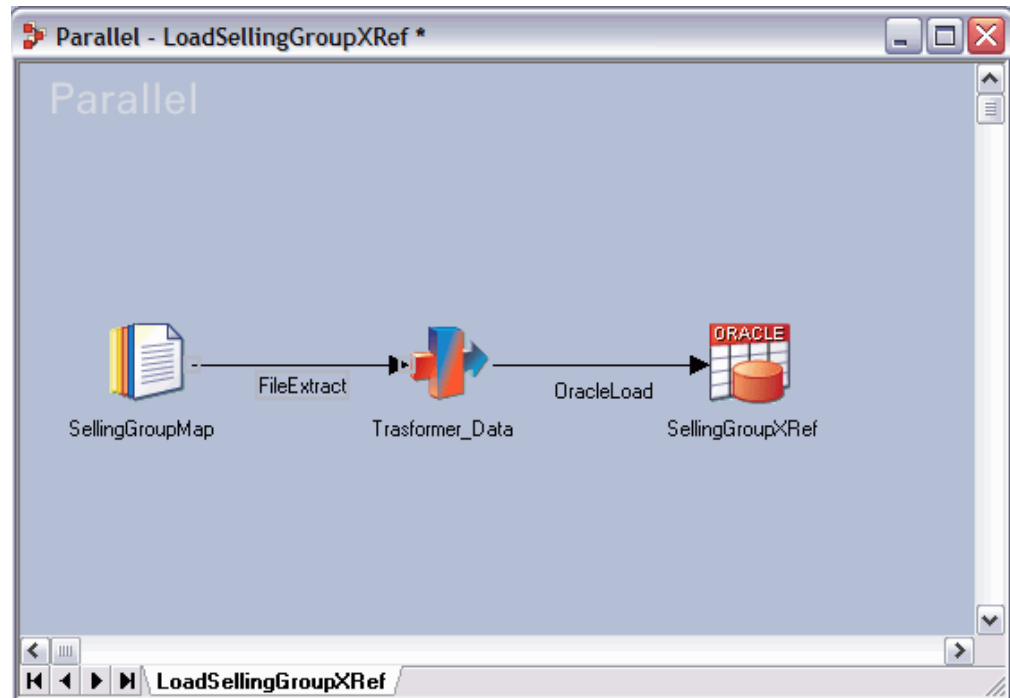


Figure 54. Simple WebSphere DataStage job

Figure 54 shows the most basic WebSphere DataStage job, which contains three stages:

- Data source (input) stage
- Transformation (processing) stage
- Target (output) stage

WebSphere DataStage jobs can be as sophisticated as required by your company's data integration needs. Figure 55 on page 94 is an example of a more complex job.

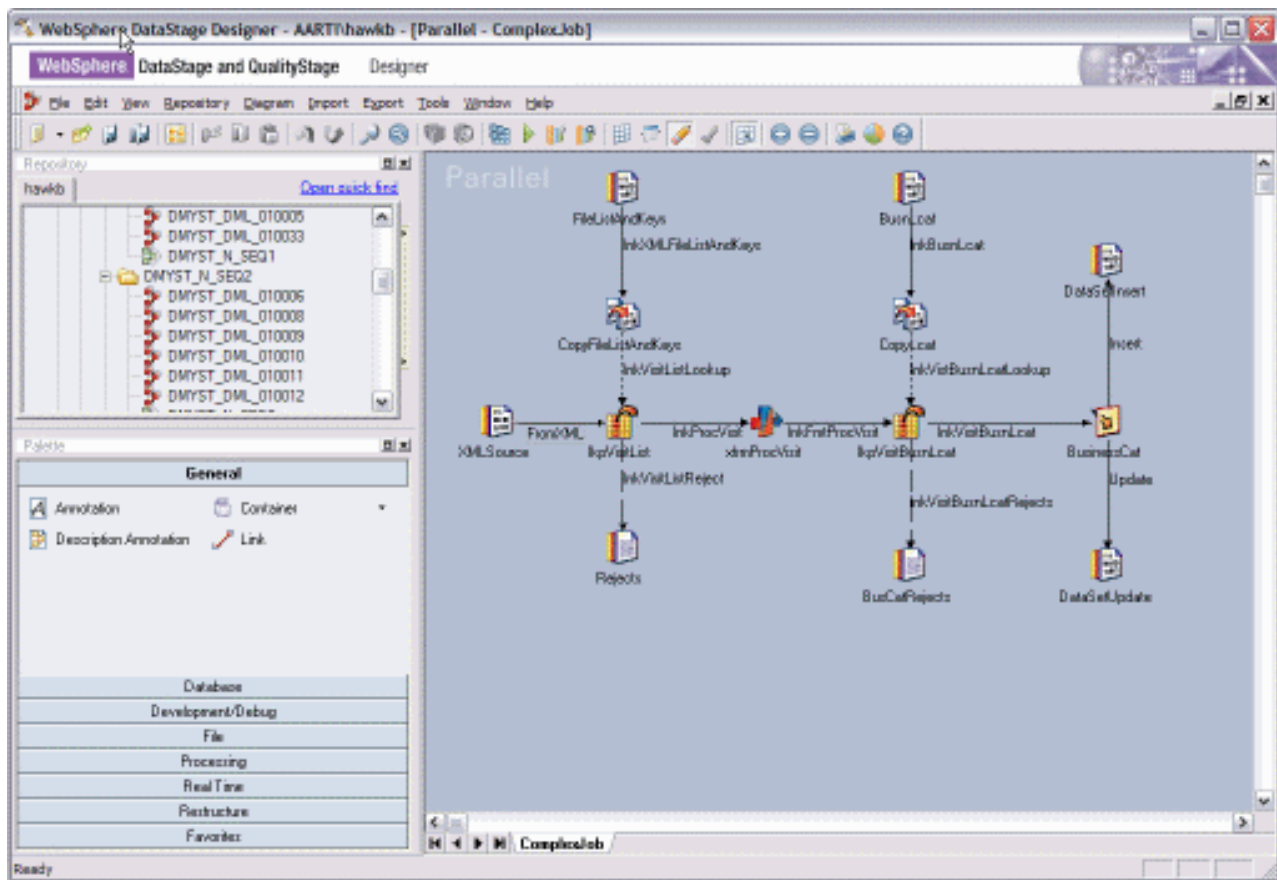


Figure 55. More complex WebSphere DataStage job

Designing jobs

With the Designer client, you draw the integration process and then add the details for each stage. This method helps you build and reuse components across jobs. The Designer client minimizes the coding that is required to define even the most difficult and complex integration process.

Each data source and each processing step is a stage in the job design. The stages are linked to show the flow of data. You drag and drop stages from the tool palette to the canvas. This palette contains icons for stages and groups that you can customize to organize stages, as shown in Figure 56 on page 95.

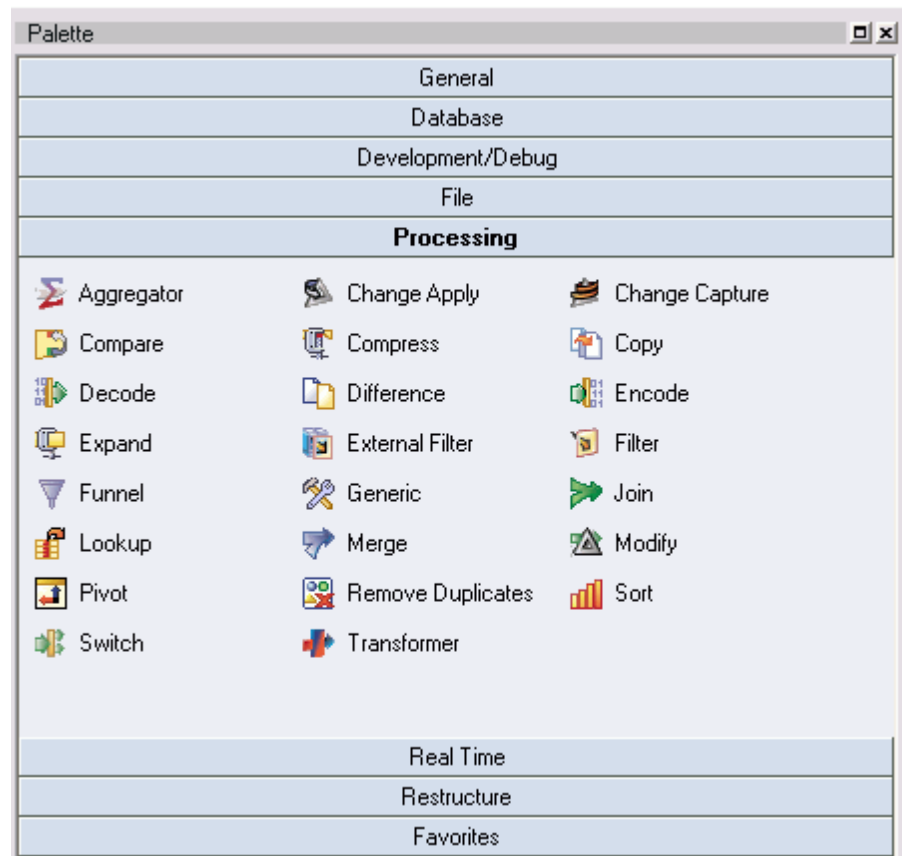


Figure 56. Tool palette

After stages are in place, they are linked together in the direction that the data will flow. For example, in Figure 54 on page 93, two links were added:

- One link between the data source (Sequential File stage) and Transformer stages
- One link between the Transformer stage and the Oracle target stage

You load table definitions for each link from a stage property editor, or select definitions from the repository and drag them onto a link.

Stage properties

Each stage in a job has properties that tell the stage how to perform or process data. Stage properties include file name for the Sequential File stage, columns to sort and the ascending-descending order for the Sort stage, database table name for a database stage, and so on. Each stage type uses a graphical editor.

Complex Flat File stage

The Complex Flat File (CFF) stage allows easy sourcing of data files that contain numerous record formats in a single file. Figure 57 on page 96 shows a three-record join. This stage supports both fixed and variable-length records and provides an easy way to join data from different record types in a logical transaction into a single data record for processing. For example, you might join customer, order, and units data.

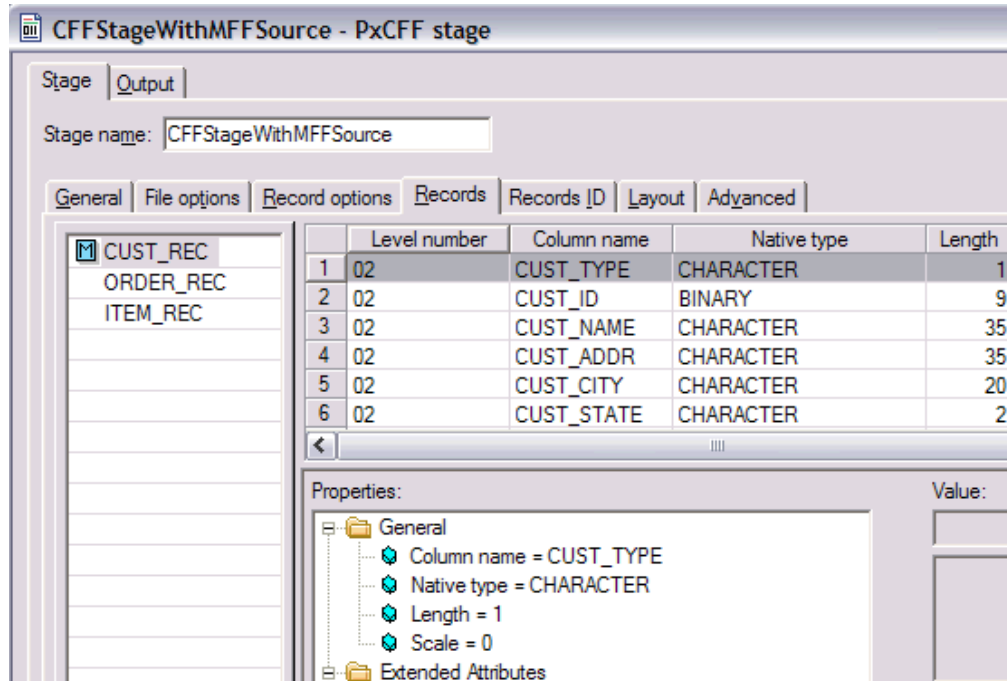


Figure 57. Complex Flat File stage window

The CFF stage and Slowly Changing Dimension stage offer a Fast Path concept for improved usability and faster implementation. The Fast Path walks you through the screens and tables of the stage properties that are required for processing the stage. Help is available for each tab by hovering the mouse over the "i" in the lower left.

Transformer stage

Transformer stages can have one primary input link, multiple reference input links, and multiple output links. The link from the main data input source is designated as the primary input link. You use reference links for lookup operations, for example, to provide information that might affect the way the data is changed, but not supplying the actual data to be changed.

Input columns are shown on the left and output columns are shown on the right. The upper panes show the columns with derivation details. The lower panes show the column metadata.

Some data might need to pass through the Transformer stage unaltered, but it is likely that data from some input columns need to be transformed first. You can specify such an operation by entering an expression or selecting a transform to apply to the data, called a derivation. WebSphere DataStage has many built-in functions to use inside the derivations. You can also define custom transform functions that are then stored in the repository for reuse.

You can also specify constraints that operate on entire output links. A constraint is an expression that specifies criteria that data must meet before it can pass to the output link.

Slowly Changing Dimension Stage

A typical design for an analytical system is based on a dimensional database that consists of a central fact table that is surrounded by a single layer of smaller dimension tables, each containing a single primary key. This design is also known as a star schema.

Star schema data is typically found in the transactional and operational systems that capture customer information, sales data, and other critical business information. One of the major differences between a transactional system and an analytical system is the need to accurately record the past. Analytical systems often must detect trends to enable managers to make strategic decisions. For example, a product definition in a sales tracking data mart is a dimension that will likely change for many products over time but this dimension typically changes slowly. One major transformation and movement challenge is how to enable systems to track changes that occur in these dimensions over time. In many situations, dimensions change only occasionally.

Figure 58 shows a typical primary key, the product sales keeping unit (PRODSKU).

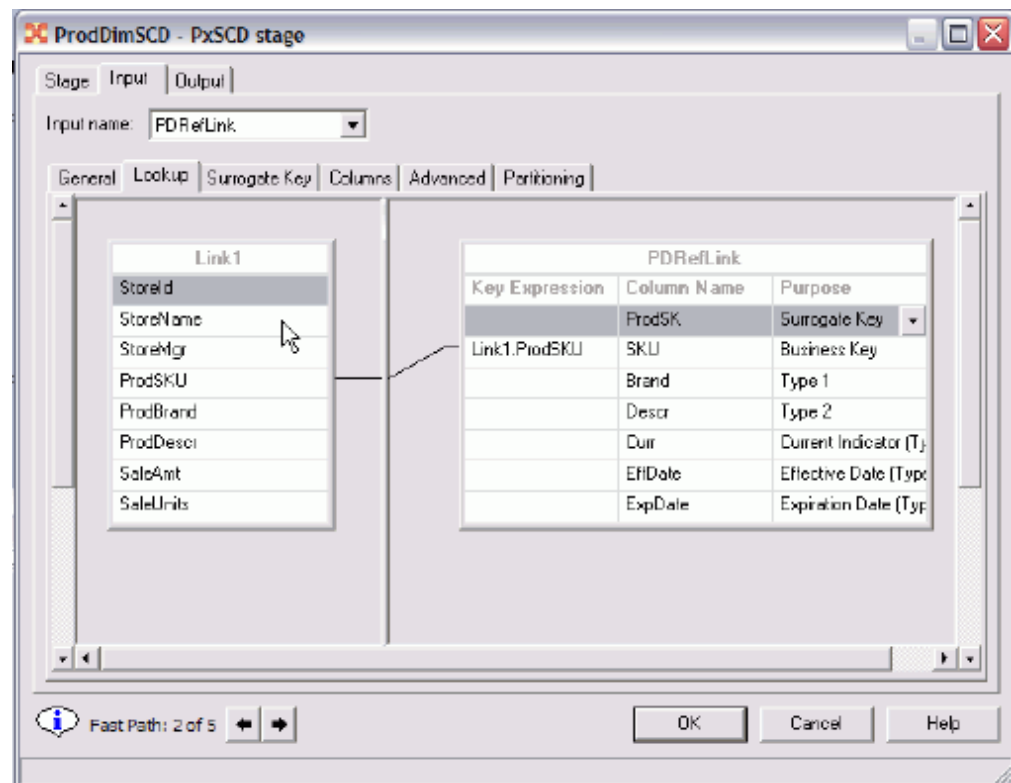


Figure 58. Looking up primary key for a dimension table

The Slowly Changing Dimension (SCD) stage processes source data for a dimension table within the context of a star schema database structure. The stage lets you overwrite the existing dimension (known as a Type-1 change), update while preserving rows (known as Type 2), or have a hybrid of both types. To prepare data for loading, the SCD stage performs the following process for each changing dimension in the star schema:

1. Business keys from the source are used to look up a surrogate key in each dimension table. Typically the dimension row is found.
2. If a dimension row is not found, a row must be created with a surrogate key.
3. If a dimension row is found but must be updated (Type-1), the update must be done.
4. For preserving history (Type-2), a new row is added and the original row is marked. A surrogate key is added to the source data and non-fact data is deleted.

In a Type-2 update, a new row with a new surrogate primary key is inserted into the dimension table to capture changes. All the rows that describe a dimension contain attributes that uniquely identify the most recent instance and historical dimensions. Figure 59 shows how the new product dimension is redefined to include the data that goes into the dimension table and also contains the surrogate key, expiry date, and the currency indicator.

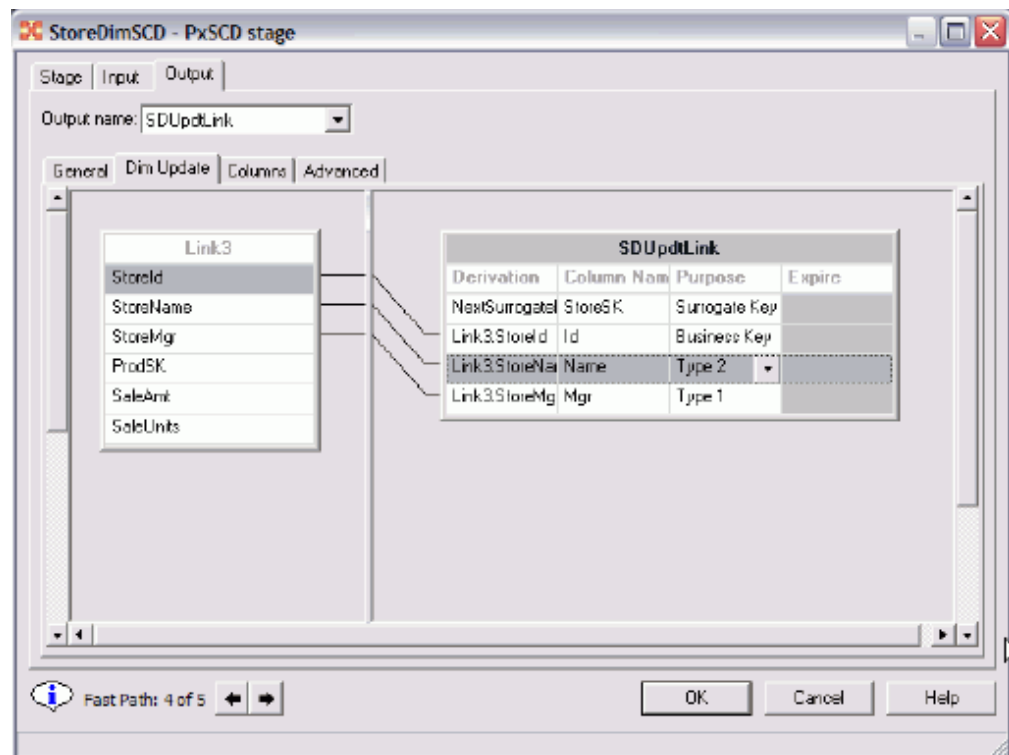


Figure 59. Redefining a dimension table

Finally, the new record is written into the dimension table (with all surrogate keys), reflecting the change in product dimension over time. Although the product sales keeping unit has not changed, the database structure enables the user to identify sales of current versions versus earlier versions of the product.

Dynamic Relational Stage

While WebSphere DataStage provides specific connectivity to virtually any database management system, the Dynamic Relational stage allows the binding of the type (for example, Oracle, DB2, or SQL Server) to be specified at run time rather than design time. The Dynamic Relational stage reads data from or writes data to a database. Figure 60 on page 99 shows the general information about the

database stage including the database type, name, user ID, and password that is used to connect. Passwords can be encrypted.

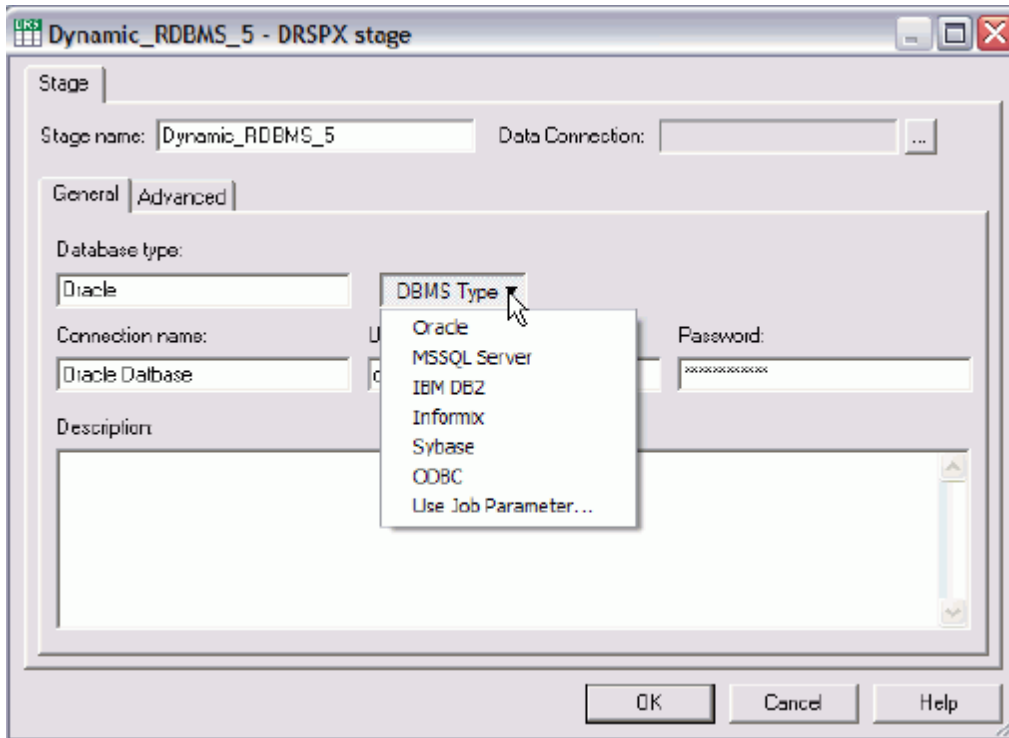


Figure 60. Designing for the Dynamic Relational stage

SQL builder

For developers who need to use SQL expressions to define database sources, the SQL builder utility provides a graphical interface for building simple-to-complex SQL query statements. The SQL builder supports DB2, Oracle, SQL Server, Teradata and ODBC databases. Although ODBC can be used to build SQL that will work for a broad range of databases, the database-specific parsers help you take advantage of database-specific functionality. Figure 61 on page 100 shows how the SQL builder guides developers in creating well-formed SQL queries.

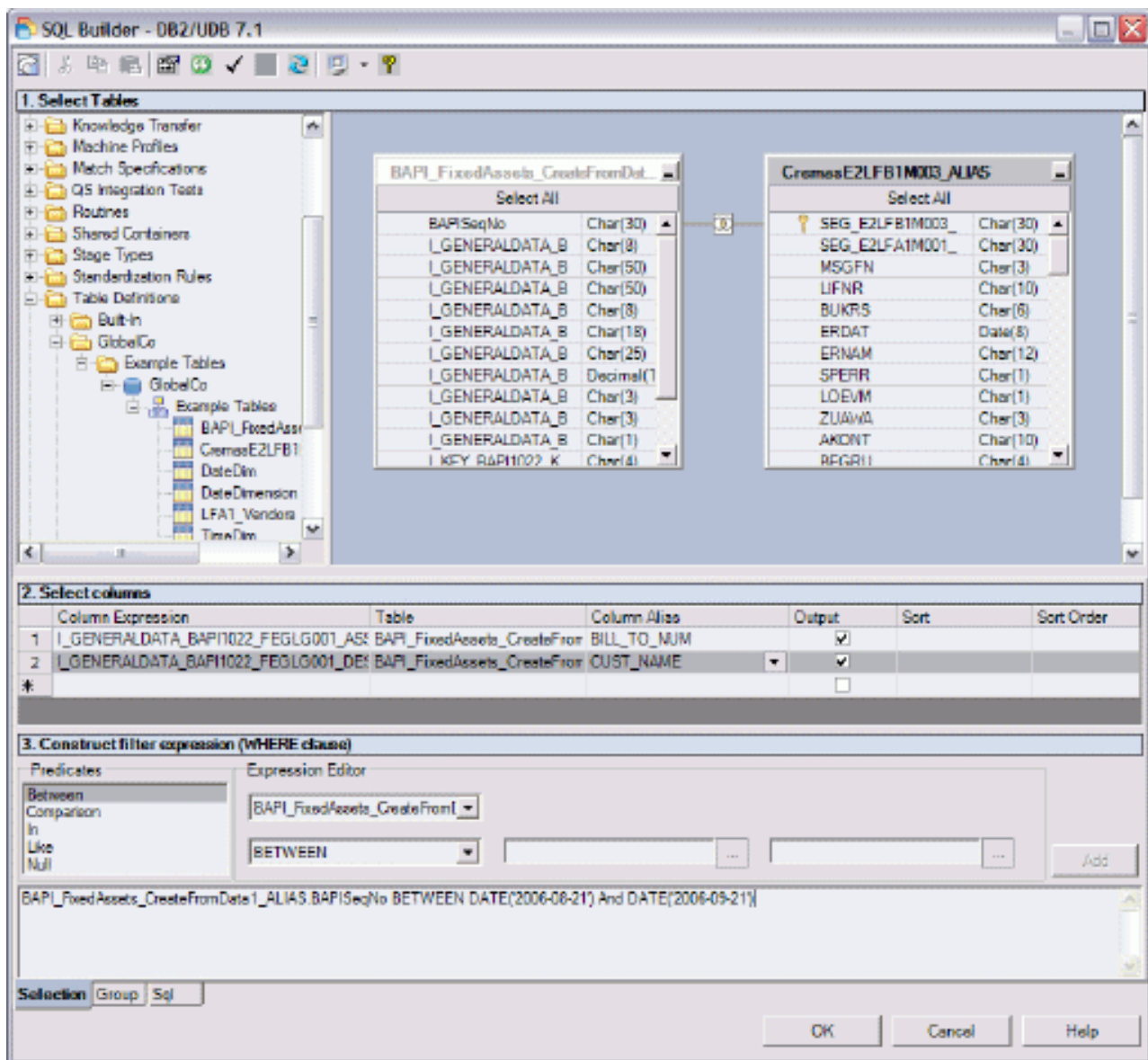


Figure 61. SQL builder utility

Job sequences

WebSphere DataStage provides a graphical job sequencer in which you can specify a sequence of jobs to run. The sequence can also contain control information. For example, the sequence might indicate different actions depending on whether a job in the sequence succeeds or fails. After you define a job sequence, you can schedule and run the sequence by using the Director client, the command line, or an API. The sequence appears in the repository and in the Director client as a job.

Designing a job sequence is similar to designing jobs. You create the job sequence in the WebSphere DataStage and QualityStage Designer, and add activities (rather than stages) from the tool palette. You then join activities with triggers (rather than links) to define control flow. Each activity has properties that can be tested in trigger expressions and passed to other activities farther down the sequence. Activities can also have parameters, which supply job parameters and routine arguments.

The job sequence has properties and can have parameters that can be passed to the activities that it is sequencing. The sample job sequence in Figure 62 shows a typical sequence that is triggered by an arriving file. The job also contains exception handling and with looping and flow control.

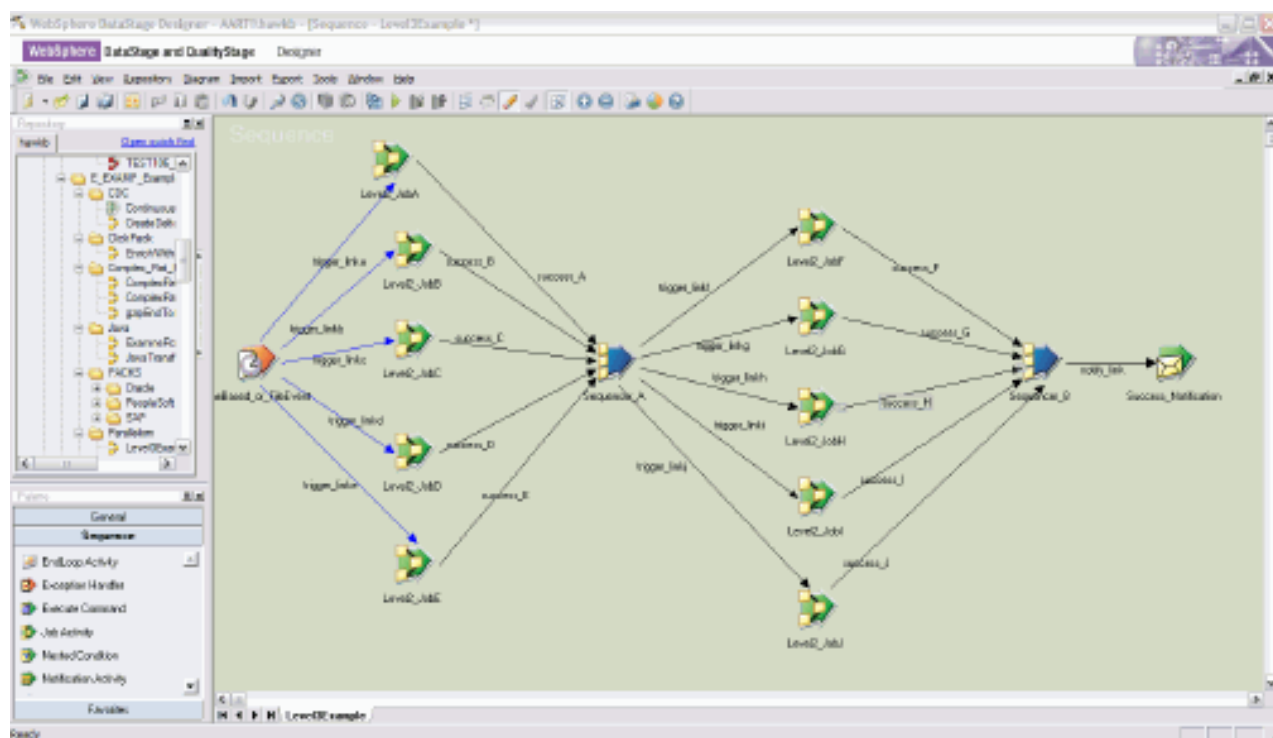


Figure 62. Sample job sequence

The job sequence supports the following types of activities:

Job Specifies a WebSphere DataStage job.

Routine

Specifies a routine.

ExecCommand

Specifies an operating system command to run.

E-mail notification

Specifies that an e-mail notification should be sent at this point of the sequence by using Simple Mail Transfer Protocol (SMTP). This method is often used in exception and error handling.

Wait-for-file

Waits for a specified file to appear or disappear. This activity can send a stop message to a sequence after waiting a specified period of time for a file to appear or disappear.

Run-activity-on-exception

Only one run-activity-on-exception is allowed in a job sequence. This activity runs if a job in the sequence fails to run. (Other exceptions are handled by triggers.)

Checkpoint, restart option for job sequences:

The checkpoint property on job sequences allows a sequence to be restarted at the failed point.

Looping stages

StartLoop and EndLoop activities make the job sequencer more flexible and give you more control.

User expressions and variables

Enables you to define and set variables. You can use these variables to evaluate expressions within a job sequence flow.

Abort-activity-on-exception

Stops job sequences when problems occur.

Job management

The Designer client manages the WebSphere DataStage project data, enabling you to view and edit items that are stored in WebSphere Metadata Server. This functionality enables you to import and export items between different WebSphere DataStage systems and exchange metadata with other tools. You can request reports on items in the metadata server.

The Designer client provides the following capabilities:

- Importing and exporting DSX and XML files
- EE configuration file editor
- Table definitions import
- Message Handler Manager
- MetaBroker import and export
- Importing Web service definitions
- Importing IMS™ definitions
- JCL templates editor

Figure 63 on page 103 shows the Designer client window for importing table definitions.

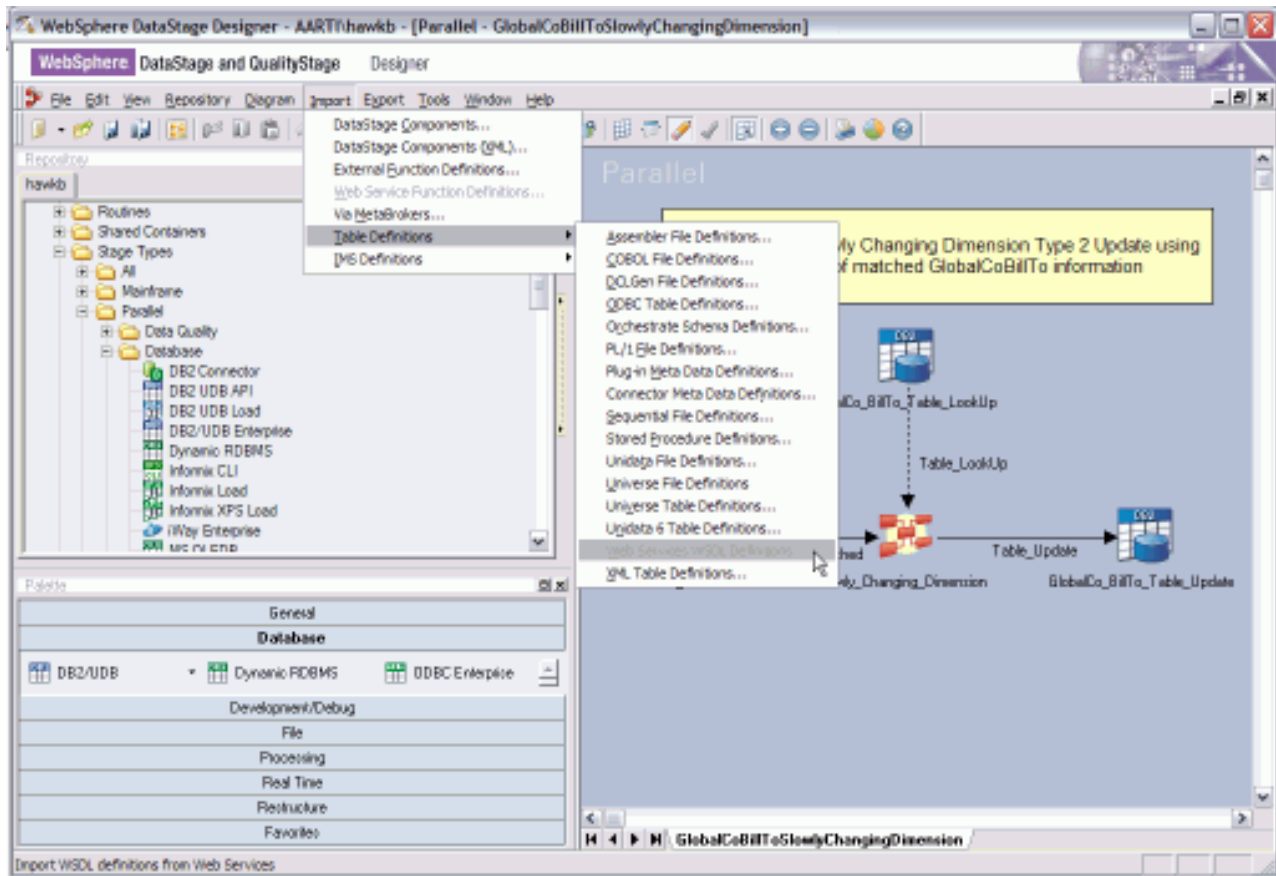


Figure 63. Importing table definitions

Importing and exporting jobs

The WebSphere DataStage and QualityStage Designer enables you to import and export components for moving jobs between WebSphere DataStage development, test, and production environments. You can import and export any component in the repository, including a job.

The export facility is also valuable for generating XML documents that describe objects in the repository. You can use a Web browser to view these documents. The Designer client also includes an import facility for importing WebSphere DataStage components from XML documents.

Related concepts

“Tasks in the Match Designer” on page 76

The Match Designer is a tool for creating a match specification. The main area of the Match Designer is made up of two tabs. On the Compose tab, you design and fine tune the match passes. On the Total Statistics tab, you provide cumulative and individual statistics for match passes.

WebSphere DataStage and QualityStage Director

The WebSphere DataStage and QualityStage Director is the client component that validates, runs, schedules, and monitors jobs that are run by the WebSphere DataStage server.

Running jobs

Running jobs with the WebSphere DataStage and QualityStage Director includes the following tasks:

Setting job options

Each time that a job is validated, run, or scheduled, you can set options to change parameters, override default limits for row processing, assign invocation IDs, and set tracing options.

Validating jobs

You can validate jobs before you run them for the first time and after any significant changes to job parameters.

Starting, stopping, or resetting a job run

A job can be run immediately or scheduled to run at a later date.

Creating multiple job invocations

You can create multiple invocations of a WebSphere DataStage server job or parallel job, with each invocation using different parameters to process different data sets.

Monitoring jobs

The Director client includes a monitoring tool that displays processing information. As Figure 64 shows, the Monitor Job Status window displays the following details:

- Name of the stages that are performing the processing
- Status of each stage
- Number of rows that were processed
- Time to complete each stage
- Rows per second

Stage/Link name	Link type	Status	Num rows	Started at	Elapsed time	Rows/sec
 Generator_2		Running 	5000000	:06:10 PM	00:01:12	69444
 Generator		Running 	5000000	:06:10 PM	00:01:12	69444
 Generator_3		Running 	5000000	:06:10 PM	00:01:12	69444
 Funnel_1		Running 	5000000	:06:10 PM	00:01:12	69444
 DSLink2	<<Pri		5000000			69444
 DSLink7	<<Pri		5000000			69444
 DSLink8	<<Pri		5000000			69444
 DSLink3	>Out		15000000			208333
 Sort		Running 	15000000	:06:11 PM	00:01:35	157895
 RemDup		Running 	4684681	:06:11 PM	00:01:35	49312
 Switch		Running 	1592137	:06:11 PM	00:01:35	16759

Figure 64. Monitor Job Status window

A monitor window is available before a job starts, while it is running, or after it completes. You can monitor multiple jobs at the same time with multiple monitor windows.

Reviewing job log files

The job log file is updated when a job is validated, run, or reset. The log file is valuable for troubleshooting jobs that fail during validation or that end abnormally.

Each log file describes events that occurred during the last (or previous) runs of the job. Figure 65 shows a graphical view of the log. The most recent or current run is shown in black, the previous run is shown in dark blue, and the others are in light blue. Entries are written to the log at these intervals:

- A job or batch starts or finishes
- A stage starts or finishes
- Rejected rows are output
- Warnings or errors are generated

The screenshot shows the 'WebSphere DataStage Director - AARTI\hawkb' window. The 'Job Log' tab is selected, displaying a table of events for the job 'samplejob'. The table has columns for 'Occurred', 'On date', 'Type', and 'Event'. The events are listed chronologically, showing the progress of the job from 8/20/2006 4:18:41 PM to 8/21/2006 4:47:09 PM. The most recent event is highlighted in black.

Occurred	On date	Type	Event
4:18:41 PM	8/20/2006	Info	GlobalCo_billTo_flat.0: Progress: 60 percent.
4:18:41 PM	8/20/2006	Info	GlobalCo_billTo_flat.0: Progress: 70 percent.
4:18:42 PM	8/20/2006	Info	GlobalCo_billTo_flat.0: Progress: 80 percent.
4:18:42 PM	8/20/2006	Info	GlobalCo_billTo_flat.0: Progress: 90 percent.
4:18:44 PM	8/20/2006	Info	GlobalCo_billTo_flat.0: Import complete; 11968 records imported successfully, 0 rejected.
4:18:45 PM	8/20/2006	Info	main_program: Step execution finished with status = OK.
4:18:45 PM	8/20/2006	Info	main_program: Startup time, 0:10; production run time, 0:17.
4:18:49 PM	8/20/2006	Info	Parallel job reports successful completion.
4:18:49 PM	8/20/2006	Control	Finished Job samplejob.
4:45:25 PM	8/21/2006	Control	Starting Job samplejob (...)
4:45:47 PM	8/21/2006	Info	Environment variable settings: (...)
4:45:47 PM	8/21/2006	Info	Parallel job initiated.
4:46:04 PM	8/21/2006	Info	main_program: IBM WebSphere DataStage Enterprise Edition 8.0.0 (...)
4:46:13 PM	8/21/2006	Info	main_program: orchgenext: loaded (...)
4:46:18 PM	8/21/2006	Info	main_program: APT configuration file: C:/IBM/WDIS/Server/Configurations/default.apt (...)
4:46:26 PM	8/21/2006	Info	main_program: Resource Tracker was not running. Starting resource_tracker.
4:46:26 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 10 percent.
4:46:28 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 20 percent.
4:46:38 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 30 percent.
4:46:43 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 40 percent.
4:46:46 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 50 percent.
4:46:48 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 60 percent.
4:46:50 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 70 percent.
4:46:57 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 80 percent.
4:46:58 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Progress: 90 percent.
4:47:02 PM	8/21/2006	Info	GlobalCo_billTo_flat.0: Import complete; 11968 records imported successfully, 0 rejected.
4:47:02 PM	8/21/2006	Info	main_program: Step execution finished with status = OK.
4:47:02 PM	8/21/2006	Info	main_program: Startup time, 0:26; production run time, 0:39.
4:47:07 PM	8/21/2006	Info	Parallel job reports successful completion.
4:47:09 PM	8/21/2006	Control	Finished Job samplejob.

Log for job: samplejob | 42 entries filtered | Server time: 8/21/2006 04:47 PM

Figure 65. Job log view

When an event is selected from the job log, you can view the full message in the Event Detail window, as Figure 66 on page 106 shows. This window contains a summary of the job and event details.

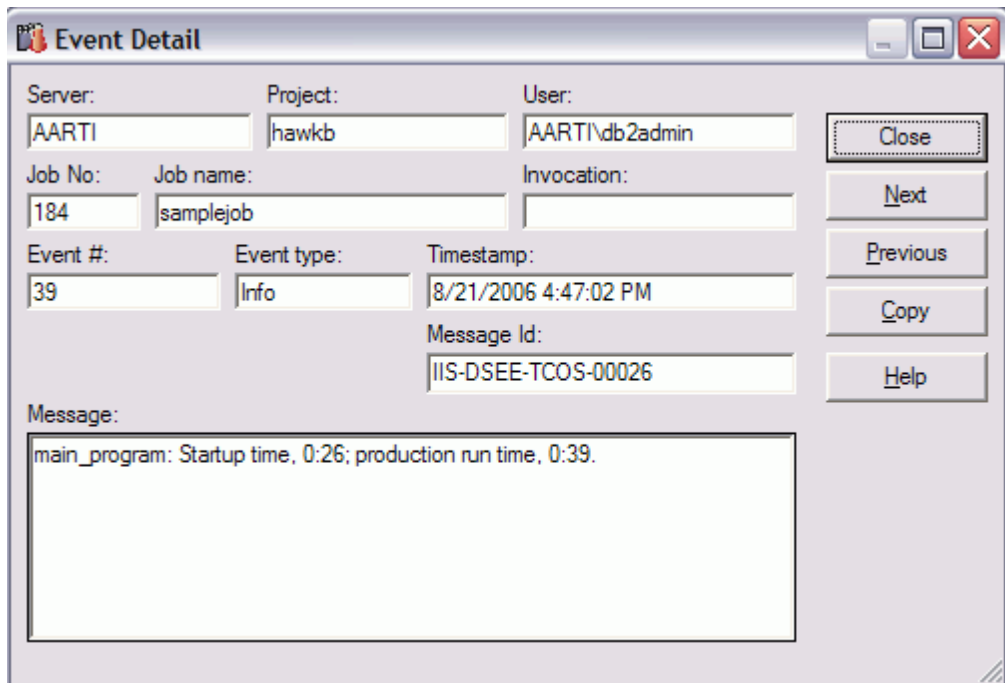


Figure 66. Event Detail window

You can use the window to display related jobs. You can also filter items in the log by time and event types, such as warnings.

Command-line interfaces

You can start, stop, and monitor WebSphere DataStage jobs from the command line and by using an extensive API.

The Command stage is an active stage that can run various external commands, including WebSphere DataStage engine commands, programs, and WebSphere DataStage jobs from anywhere in the WebSphere DataStage data flow.

You can run any command, including its arguments, by using the native command window (shell) of the operating system, such as Windows NT® or UNIX®. Examples include Perl scripts, DOS batch files, UNIX scripts, and other command-line executable programs that you can call if they are not interactive.

Command line, API, and Web service interfaces also exist to return job monitoring information, as text or XML.

WebSphere DataStage and QualityStage Administrator

WebSphere DataStage and QualityStage Administrator provides tools for managing general and project-related tasks such as server timeout and NLS mappings.

The Administrator client supports the following types of tasks:

- Adding new projects
- Deleting projects
- Setting project-level properties
- Setting and changing NLS maps and locales

- Setting permissions and user categories to enable only authorized users to edit components in the project or run jobs
- Setting mainframe and parallel job properties and default values

Data transformation for zSeries®

To integrate data throughout the enterprise, companies must access and respond to all the information that affects them. A significant amount of corporate data continues to reside in mainframes.

Mainframes play a key role in many enterprises. In some cases, the volume of data is too large to be moved off the mainframe, for example the data stored in very large databases (VLDB). In other cases, there are no migration paths, such as from IBM IMS. Mainframes can also be the most reliable platform upon which to run corporate data for day-to-day business functions.

Some data integration efforts, such as decision support, occur off mainframe systems to avoid tying up mainframe resources and to provide the fastest possible response times. The mainframe-connectivity tools in IBM Information Server are designed to help companies transmit data between mainframe systems and their data warehouse systems.

WebSphere DataStage MVS Edition

WebSphere DataStage MVS™ Edition enables integration of mainframe data with other enterprise data. You can integrate data between applications and databases on the mainframe, or between the mainframe and UNIX, Red Hat Enterprise Linux®, SUSE Enterprise Linux and Windows.

Introduction to WebSphere DataStage MVS Edition

WebSphere DataStage MVS Edition consolidates, collects, and centralizes information from various systems and mainframes by using native execution from a single design environment.

WebSphere DataStage MVS Edition generates COBOL applications and the corresponding custom JCL scripts for processing mainframe flat files and data from VSAM, DB2, IMS, and Teradata. Users can also integrate custom in-house applications into the design.

WebSphere DataStage MVS Edition includes the following features:

- Native COBOL support
- Support for complex data structures
- Multiple source and target support
- Complete development environment
- End-to-end metadata management

Figure 67 on page 108 shows the data transformation process that this edition uses.

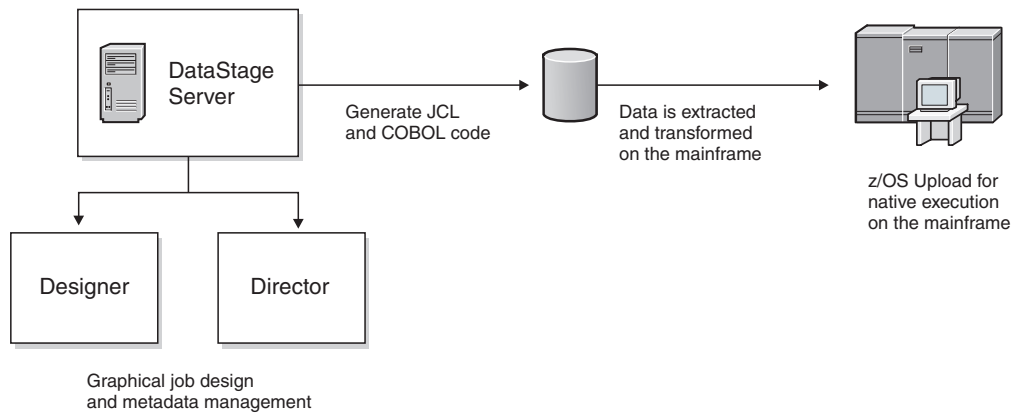


Figure 67. Data transformation process used by WebSphere DataStage MVS Edition

WebSphere DataStage MVS Edition complements existing infrastructures and skill sets by processing directly on the mainframe, where sophisticated security, access, and management already exist.

Without using WebSphere DataStage MVS Edition, development time can take 16 to 20 times longer, and maintenance can take 10 to 20 times longer.

WebSphere DataStage MVS Edition tasks

WebSphere DataStage MVS Edition provides a broad range of metadata import functions.

- COBOL file descriptions that enable you to import copybooks or definitions from COBOL programs
- DB2 table definitions that enable you to import a DCLGEN report or connect to DB2
- IMS Database Definition (DBD) and Program Specification Block (PSB)
- PL/I file descriptions that enable you to import table definitions that were written using PL/I language constructs to describe a record
- Assembler DSECT import function
- Metadata from any of the WebSphere MetaBrokers or metadata bridges

Figure 68 shows a sample mainframe job.

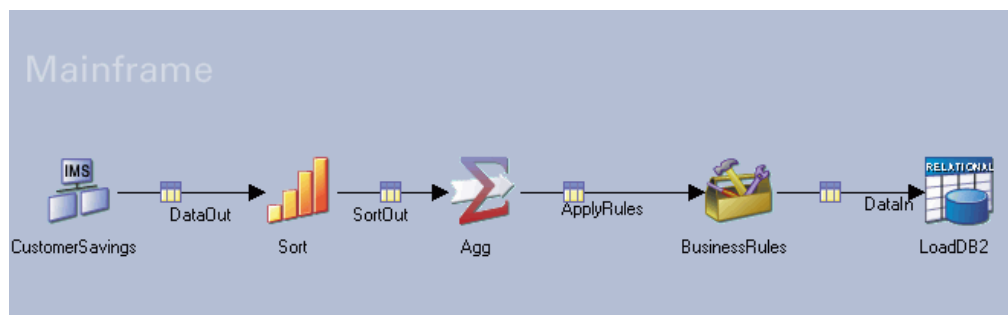


Figure 68. Sample WebSphere DataStage MVS Edition job

With WebSphere DataStage MVS Edition, a job is generated into:

- A single COBOL program
- Compiled JCL with end-user customization capabilities

- Run JCL for application execution and other steps as needed, based on job design (for example, bulk load, FTP, presort)

After WebSphere DataStage MVS Edition generates the COBOL and JCL, it uploads the files to the mainframe for compilation and execution. After the job runs, operational metadata is sent to the repository.

WebSphere DataStage MVS Edition connectivity

WebSphere DataStage MVS Edition provides mainframe connectivity to DB2, IMS, VSAM, QSAM, ISAM, flat files, and Teradata on Windows or MP-RAS systems. A sophisticated editor is provided for hierarchical and multiformat files, which are common to mainframe environments. Teradata connectivity supports Teradata FastLoad, MultiLoad, TPump, FastExport, and a relational stage for custom SQL statements.

IMS connectivity includes a graphical editor to specify details about the IMS database, view, segments, and fields. WebSphere DataStage MVS Edition generates DLI and BMP programs for accessing IMS data.

WebSphere DataStage Enterprise for z/OS

WebSphere DataStage Enterprise for z/OS® enables WebSphere DataStage to run under UNIX Systems Services (USS) on the mainframe.

You develop USS jobs by using a Windows-based WebSphere DataStage client that is connected to a WebSphere DataStage server on UNIX. The same parallel jobs that run on Linux, UNIX, and Windows run in parallel under USS.

You compile and run jobs by using one of two modes:

Tightly coupled

Allows jobs to be designed, compiled, and run under the control of the WebSphere DataStage clients. Logging and monitoring information is available in WebSphere DataStage. Remote shell (rsh) and FTP are used to automatically connect to the mainframe.

Loosely coupled

Does not require a remote shell server to be enabled on the mainframe. You can send job scripts to the mainframe automatically by using FTP or manually. Jobs that contain transformers, lookups, or buildups are then compiled on the mainframe. All jobs might be run using command-line interfaces or a mainframe scheduler. Job logging and monitoring information is not returned to the WebSphere DataStage server in this mode.

All of the base parallel stages on Linux, UNIX, and Windows are available in WebSphere DataStage Enterprise for z/OS, which can work with the following data:

- z/OS UNIX files (read and write)
- QSAM data sets (read only)
- VSAM (ESDS, KSDS, RRDS) data sets (read only)
- DB2 (read, write, load, lookup, upsert)
- Teradata

Information resources for WebSphere DataStage

A variety of information resources can help you get started with WebSphere DataStage.

Planning, installation, and configuration details for WebSphere DataStage and other IBM Information Server suite components is available in the IBM Information Server information center at <http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp>.

Online help for the WebSphere DataStage client interfaces is available in HTML format. The following documentation in PDF format is available from the Windows **Start** menu and the Quick Start CD:

WebSphere DataStage

- *WebSphere DataStage Server Job Tutorial*
- *WebSphere DataStage Parallel Job Tutorial*
- *WebSphere DataStage Administrator Client Guide*
- *WebSphere DataStage Designer Client Guide*
- *WebSphere DataStage Director Client Guide*
- *WebSphere DataStage BASIC Reference Guide*
- *WebSphere DataStage Parallel Engine Message Reference*
- *WebSphere DataStage Mainframe Job Developer Guide*
- *WebSphere DataStage National Language Support Guide*
- *WebSphere DataStage Parallel Job Advanced Developer Guide*
- *WebSphere DataStage Parallel Job Developer Guide*
- *WebSphere DataStage Server Job Developer Guide*

IBM Information Server and suite components

Planning, installation, and configuration details are also available in the following PDFs that you can access from the Windows **Start** menu and the Quick Start CD:

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Chapter 8. WebSphere Federation Server

IBM Information Server provides industry-leading federation in its WebSphere Federation Server suite component to enable enterprises to access and integrate diverse data and content, structured and unstructured, mainframe and distributed, public and private, as if it were a single resource.

Because of mergers and acquisitions, hardware and software improvements, and architectural changes, organizations often must integrate diverse data sources into a unified view of the data and ensure that information is always available, when and where it is needed, by people, processes, and applications.

WebSphere Federation Server is central to the Deliver capability of IBM Information Server, as Figure 69 shows.

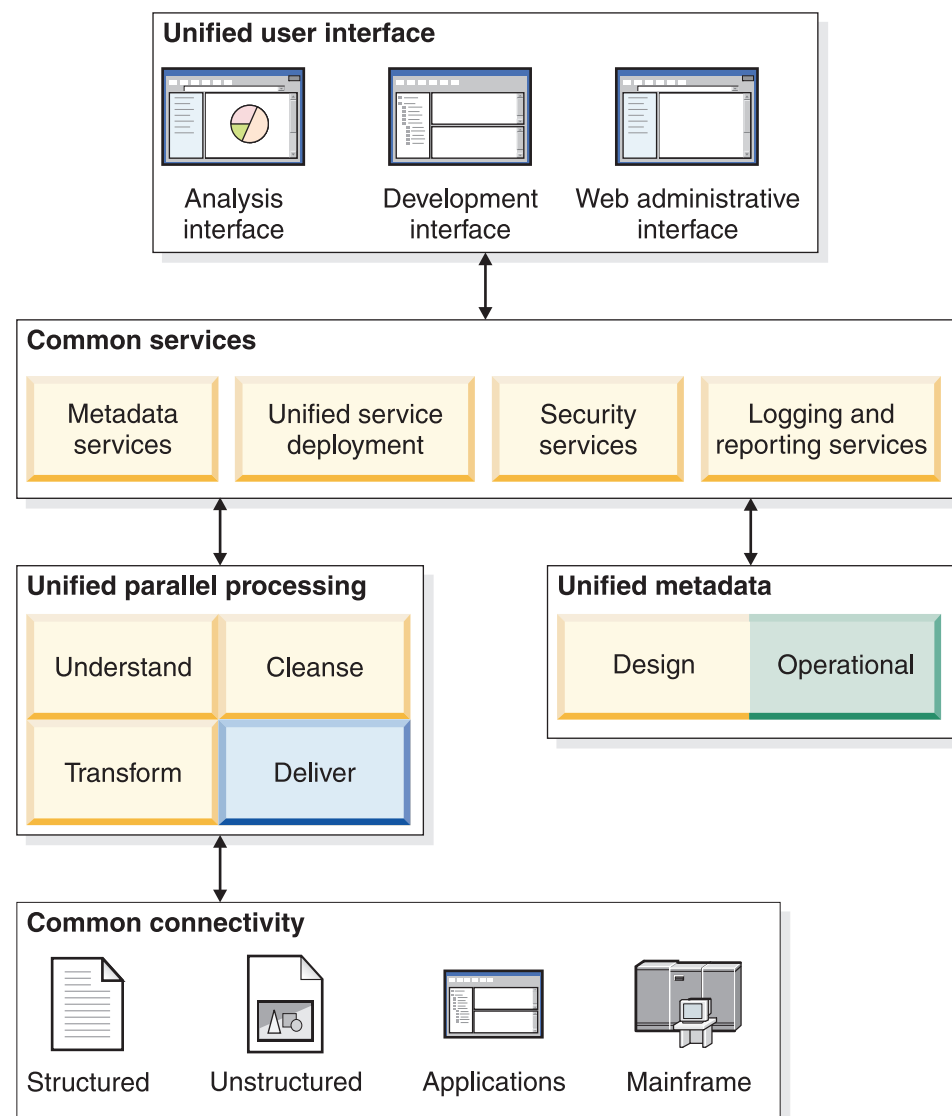


Figure 69. IBM Information Server architecture

Data federation aims to efficiently join data from multiple heterogeneous sources, leaving the data in place and avoiding data redundancy. The source data remains under the control of the source systems and is pulled on demand for federated access.

A federated system has several important advantages:

Time to market

Applications that work with a federated server can interact with a single virtual data source. Without federation, applications must interact with multiple sources by using different interfaces and protocols. Federation can help reduce development time significantly.

Reduced development and maintenance costs

With federation, an integrated view of diverse sources is developed once and leveraged multiple times while it is maintained in a single place, which allows a single point of change.

Performance advantage

By using advanced query processing, a federated server can distribute the workload among itself and the data sources that it works with. The federated server determines which part of the workload is most effectively run by which server to speed performance.

Reusability

You can provide federated data as a service to multiple service consumers. For example, an insurance company might need structured and unstructured claims data from a wide range of sources. The sources are integrated by using a federated server, and agents access claims data from a portal. The same federated access can then be used as a service by other consumers such as automated processes for standard claims applications, or client-facing Web applications.

WebSphere Federation Server offers two complementary federation capabilities. One capability offers SQL-based access across a wide range of data and content sources. A second capability offers federation of content repositories, collaboration systems, and workflow systems with an API optimized for the business needs of companies that require broad content federation solutions.

Introduction to WebSphere Federation Server

WebSphere Federation Server allows organizations to virtualize their data and provide information in a form that applications and users need while hiding the complexity of the underlying sources. Data virtualization allows information to be accessed through a common interface that centralizes the control of data access.

Federation is also known as enterprise information integration. It provides an optimized and transparent data access and transformation layer with a single relational interface across all enterprise data.

With a federated system, you can send distributed requests to multiple data sources within a single SQL statement. For example, you can join data that is in a DB2 table, an Oracle table, a Web service, and an XML tagged file in a single SQL statement.

Figure 70 on page 113 shows the components of a federated system and a sample of the data sources that you can access.

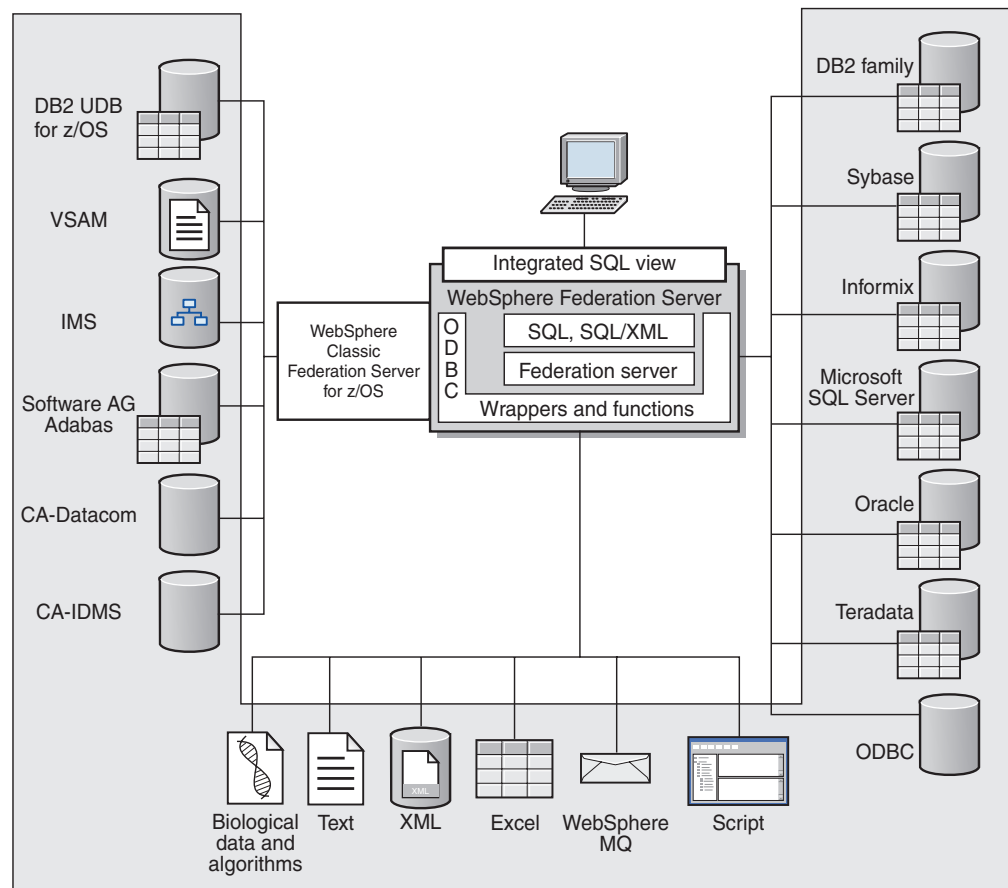


Figure 70. Components of a federated system and sample of data sources

WebSphere Federation Server leverages the metadata of sources systems to automate the building and compiling of federated queries. Metadata also enables traceability and auditability throughout the federation process. Federated queries can easily scale to run against any volume of information by leveraging IBM Information Server's powerful parallel processing engine.

You can deploy federation logic as real-time services within a SOA, as event-driven processes triggered by business events, or on-demand within self-service portals.

A federated system has the following abilities:

- Correlate data from local tables and remote data sources, as if all the data is stored locally in the federated database
- Update data in relational data sources, as if the data is stored in the federated database
- Move data to and from relational data sources
- Use data source processing strengths by sending requests to the data sources for processing
- Compensate for SQL limitations at the data source by processing parts of a distributed request at the federated server
- Access data anywhere in your enterprise, regardless of what format it is in or what vendor you use, without creating new databases and without disruptive changes to existing ones, using standard SQL and any tool that supports JDBC or ODBC.

WebSphere Federation Server delivers all of these core federation capabilities, plus the following features:

- Visual tools for federated data discovery and data modeling
- Industry-leading query optimization with single sign-on, unified views, and function compensation
- Federated two-phase commit for updating multiple data sources simultaneously within a distributed system, while maintaining data integrity across distributed sources
- Remote stored procedures to avoid unnecessary development costs by leveraging previously developed procedures within heterogeneous data sources

Scenarios for data federation

The following scenarios show how organizations use WebSphere Federation Server to solve their integration needs.

Financial services: Risk management

A major European bank wanted to improve risk management across its member institutions and meet deadlines for Basel II compliance. The bank had different methods of measuring risk among its members.

The solution is a database management system that stores a historical view of data, handles large volumes of information, and distributes data in a format that enables analysis and reporting. WebSphere Federation Server enables reporting systems to view data in operational systems that are spread across the enterprise, including vendor information. Risk-calculation engines and analytical tools in the IBM solution provide fast and reliable access to data. The new solution will enable compliance with Basel II while using a single mechanism to measure risk.

Manufacturing: defect tracking

A major automobile manufacturer needed to quickly identify and remedy defects in its cars. Traditional methods, such as data queries or reporting, were too complex and too slow to pinpoint the sources of problems.

By installing WebSphere Federation Server, the company was able to quickly and easily identify and fix defects by mining data from multiple databases that store warranty information and correlating warranty reports with individual components or software in its vehicles.

Government: emergency response

An agriculture department in a U.S. state needed to eliminate storage of redundant contact information and simplify maintenance. The department had very limited resource for any improvements (one DBA and a manager).

The department chose WebSphere Federation Server for its emergency response system. WebSphere Federation Server joins employee contact information in a human resources database on Oracle with information about employee skills in a DB2 database. The information is presented to emergency personnel through a portal that is implemented with WebSphere Application Server. The small staff was able to accomplish this project because all they needed to learn to use federation was SQL.

Related concepts

Chapter 4, “Service-oriented integration,” on page 29

IBM Information Server simplifies the creation of shared data integration services by enabling integration logic to be used by any business process.

Chapter 1, “Introduction,” on page 1

Most of today’s critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

A closer look at WebSphere Federation Server

The components of WebSphere Federation Server include the federated server and database, wrappers, nicknames, and other federated objects. Capabilities of WebSphere Federation Server that provide performance and flexibility for integration projects include compensation, the query optimizer, and two-phase commit.

Related concepts

Chapter 2, “Architecture and concepts,” on page 5

IBM Information Server provides a unified architecture that works with all types of information integration. Common services, unified parallel processing, and unified metadata are at the core of the server architecture.

“SOA and data integration” on page 40

Enabling an IBM Information Server job as a Web service enables the job to participate in various data integration scenarios.

The federated server and database

Central components of a federated system include the federated server and the federated database.

The federated server

In a federated system, the server that receives query requests and distributes those queries to remote data sources is referred to as the *federated server*. A federated server embeds an instance of DB2 to perform query optimization and to store statistics about remote data sources.

Application processes connect and submit requests to the database within the federated server. A federated server is configured to receive requests that might be intended for data sources. The federated server distributes these requests to the data sources.

A federated server uses the native client of the data source to access the data source. For example, a federated server uses the Sybase Open Client to access Sybase data sources and an Microsoft SQL Server ODBC Driver to access Microsoft SQL Server data sources.

The federated database

To users and client applications, data sources appear as a single relational database. Users and applications interface with the *federated database* that is managed by the federated server.

The federated database system catalog contains entries that identify data sources and their characteristics. The federated server consults the information that is stored in the federated database system catalog and the data source connector to determine the best plan for processing SQL statements.

The federated system processes SQL statements as if the data from the data sources were ordinary relational tables or views within the federated database.

- The federated system can correlate relational data with data in nonrelational formats. This is true even when the data sources use different SQL dialects, or do not support SQL at all.
- The characteristics of the federated database take precedence when the characteristics of the federated database differ from the characteristics of the data sources. Query results conform to DB2 semantics, even if data from other non-DB2 data sources is used to compute the query result.

Wrappers and other federated objects

Within a federated server and federated database, you use connectors (referred to as wrappers in the federated system), server definitions, user mappings, and nicknames to configure connections to a data source and to reference objects within the data source.

Wrappers

Wrappers are a type of connector that enable the federated database to interact with data sources. The federated database uses routines stored in a library called a *wrapper module* to implement a wrapper.

Wrapper modules enable the federated database to perform operations such as connecting to a data source and retrieving data. Typically, the federated instance owner uses the CREATE WRAPPER statement to register a wrapper in the federated database.

You create one wrapper for each type of data source that you want to access. You use the server definitions and nicknames to identify the details (name, location, and so on) of each data source object.

A wrapper performs many tasks:

- Connecting to the data source by using the data source's standard connection API
- Submitting queries to the data source in SQL or the native query language of the source
- Receiving results sets from the data source by using the data source standard APIs
- Gathering statistics about the data source

Wrapper options are used to configure the wrapper or to define how WebSphere Federation Server uses the wrapper.

Server definitions and server options

After you create a wrapper for a data source, you supply a name to identify the data source to the federated database.

The name and other information that the instance owner supplies to the federated server are collectively called a *server definition*. Data sources answer requests for data and as such are also servers.

For example, a DB2 family data source can have multiple databases. The server definition must specify which database the federated server can connect to. In

contrast, an Oracle data source has one database, and the federated server can connect to the database without knowing its name. The database name is not included in the server definition of an Oracle data source.

Some of the information in a server definition is stored as server options. Server options can be set to persist over successive connections to the data source, or set for the duration of a single connection.

User mappings

You can define an association between the federated server authorization ID and the data source user ID and password. This association is called a *user mapping*.

In some cases, you do not need to create a user mapping if the user ID and password that you use to connect to the federated database are the same as those that you use to access the remote data source. You can create and store the user mappings in the federated database, or you can store the user mappings in an external repository, such as LDAP.

Nicknames

After you create the server definitions and user mappings, you create nicknames. A *nickname* is an identifier that refers to an object at the data sources that you want to access. The objects that nicknames identify are referred to as *data source objects*.

Nicknames are pointers by which the federated server references the nickname objects. Nicknames are mapped to specific objects at the data source. These mappings eliminate the need to qualify the nicknames by data source names. The location of the data source objects is transparent to the end user and the client application.

For example, if you define the nickname DEPT to represent an Informix® database table called NFX1.PERSON, you can use the SQL statement `SELECT * FROM DEPT` from the federated server. However, the statement `SELECT * FROM NFX1.PERSON` is not allowed from the federated server (except in a pass-through session) unless there is a local table on the federated server named NFX1.PERSON.

When you create a nickname for a data source object, metadata about the object is added to the global catalog. The query optimizer uses this metadata, and the information in the wrapper, to facilitate access to the data source object. For example, if the nickname is for a table that has an index, the global catalog contains information about the index.

Query optimization

The federated database optimizes the performance of SQL queries against heterogeneous data sources by leveraging the DB2 query optimizer and by determining when it is faster to process a query on the data source or on the federated database.

Compensation

The process of compensation determines where a federated query will be handled. The federated database compensates for lack of functionality at the data source in two ways:

- It can request that the data source use one or more operations that are equivalent to the DB2 function in the query.
- It can return the set of data to the federated server, and perform the function locally.

For relational data sources, each type of relational database management system supports a subset of the international SQL standard. If an SQL construct is found in the DB2 SQL dialect but not in the relational data source dialect, the federated server can implement this construct on behalf of the data source.

Even data sources with weak SQL support or no SQL support will benefit from compensation.

The query optimizer

As part of the SQL compiler process, the query optimizer analyzes a query. The compiler develops alternative strategies, called access plans, for processing the query.

Access plans might call for the query to be processed by the data source, the federated server, or partly by each.

The query optimizer uses information in the wrapper and global database catalog to evaluate query access plans. The optimizer decomposes the query into segments that are called query fragments. Typically it is more efficient to push down a query fragment to a data source if the data source can process the fragment. However, the query optimizer evaluates other factors:

- Amount of data that needs to be processed
- Processing speed of the data source
- Amount of data that the fragment will return
- Communication bandwidth
- Whether a usable materialized query table on the federated server represents the same query result

The query optimizer generates access plan alternatives for processing a query fragment. The plan alternatives perform varying amounts of work locally on the federated server and on the remote data sources. The query optimizer chooses the plan with the least resource consumption cost.

Two-phase commit for federated transactions

A federated system can use two-phase commit for transactions that access one or more data sources. Two-phase commit can safeguard data integrity in a distributed environment.

Consider these differences between one-phase commit and two-phase commit:

One-phase commit

Multiple data sources are updated individually by using separate commit operations. Data can lose synchronization if some data sources are successfully updated and others are not.

Two-phase commit

Commit processing occurs in two phases: the prepare phase and the commit phase. During the prepare phase, a federated server polls all of the federated two-phase commit data sources that are involved in a

transaction. This polling verifies whether each data source is ready to commit or roll back the data. During the commit phase, the federated server instructs each two-phase commit data source to either commit the data or to roll back the transaction.

For example, if a transaction withdraws funds from one account and deposits them in another account using one-phase commit, the system might successfully commit the withdraw operation and unsuccessfully commit the deposit operation. The deposit operation can be rolled back, but the withdrawal operation cannot because it already successfully committed. The result is that the funds are virtually "lost."

In a two-phase commit environment, the withdrawal and deposit transactions are prepared together and either committed or rolled back together. The result is that the integrity of the fund amounts remains intact.

Rational Data Architect

Rational Data Architect is a companion product to the WebSphere Federation Server component of IBM Information Server that helps you design databases, understand information assets and their relationships, and streamline integration projects.

Rational Data Architect provides tools for the design of federated databases that can interact with WebSphere DataStage and other IBM Information Server components.

With Rational Data Architect, you can discover, model, visualize and relate heterogeneous data assets. The product combines traditional data modeling capabilities with unique mapping capabilities and model analysis, all organized in a modular, project-based manner. Figure 71 shows how Rational Data Architect helps map four tables that contain employee information from the source database into a single, denormalized table in a target data warehouse.

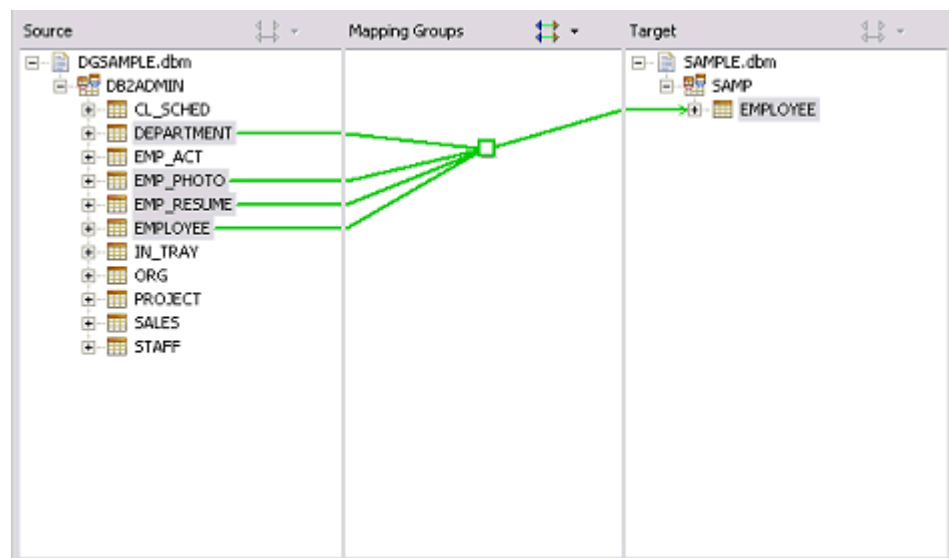


Figure 71. Using Rational Data Architect to map source tables to a target table

Rational Data Architect discovers the structure of heterogeneous data sources by examining and analyzing the underlying metadata. Rational Data Architect requires only an established JDBC connection to the data sources to explore their structures using native queries.

Rational Data Architect includes these key features:

- An Eclipse-based graphical interface for browsing the hierarchy of data elements to understand their detailed properties and visualize tables, views, and relationships in a contextual diagram.
- Ability to represent elements from physical data models by using either Information Engineering (IE) or Unified Modeling Language (UML) notation, enabling data architects to create physical data models from scratch, from logical models by using transformation, or from the database using reverse engineering.
- Rule-driven compliance checking that operates on models or on the database. Rational Data Architect can analyze for first, second and third normal form, check indexes for excessive use, and perform model syntax checks.

WebSphere Federation Server tasks

WebSphere Federation Server includes IBM DB2 9 relational database management system, which enables you to interact with a federated system by using the DB2 Control Center, DB2 commands, APIs, Web services providers, and other methods.

Federated objects

WebSphere Federation Server uses a wizard-driven approach that simplifies the tasks of setting up, configuring, and modifying the federated system.

The DB2 Control Center is a graphical interface that you can use to perform the essential data source configuration tasks:

- Create the wrappers and set the wrapper options
- Specify the environment variables for your data source
- Create the server definitions and set the server options
- Create the user mappings and set the user options
- Create the nicknames and set the nickname options or column options

You can also use the DB2 Control Center to configure access to Web services, WebSphere Business Integration, and XML data sources.

Figure 72 on page 121 shows the Wrapper page of the Create Federated Objects wizard with a NET8 wrapper selected to configure access to Oracle.

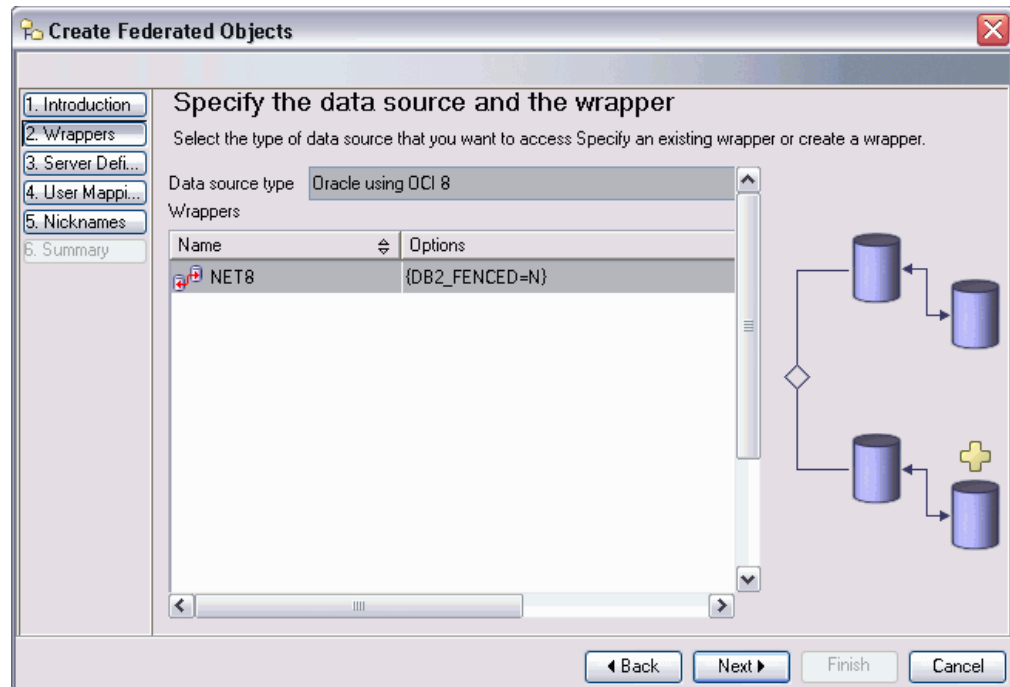


Figure 72. Create Federated Objects wizard

The wizard provides a fast and flexible discovery mechanism for finding servers, nicknames, and other federated objects. You can specify filter criteria in the Discover window to narrow your choices, as Figure 73 shows.

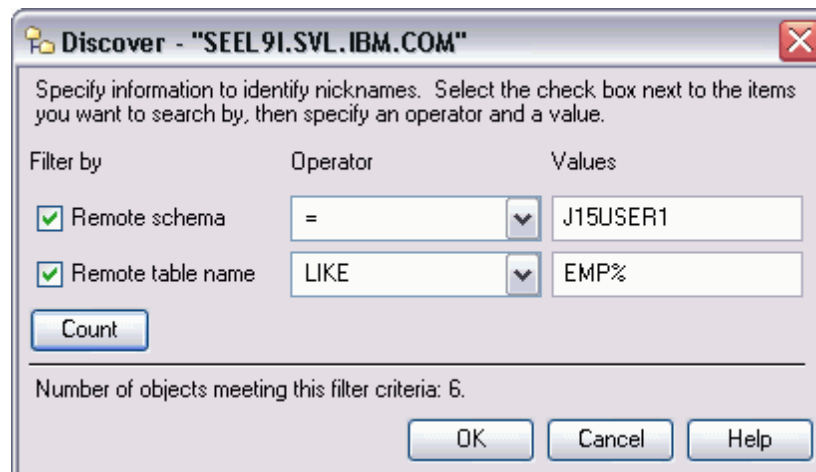


Figure 73. Using the Discover function to find nicknames on a federated server

Cache tables for faster query performance

A cache table can improve query performance by storing the data locally instead of accessing the data directly from the data source.

A cache table consists of the following components:

- A nickname on the federated database system
- One or more materialized query tables that you define on the nickname

- A replication schedule to synchronize the local materialized query tables with your data source tables

You use the Cache Table wizard in the DB2 Control Center to create the components of a cache table. Figure 74 shows the wizard page where you specify details to create a materialized query table.

The wizard automatically indicates when required settings for creating the materialized query table are missing. In this example, the wizard validates the settings after the EMPNO column was selected as a unique index for the table.

Cache Table Wizard

Specify the information for the materialized query table

The name of the cache table is based on the nickname that you selected. The materialized query table can be either a full replica or a subset of the remote data source table. To cache a subset of the columns, select the columns and the index. To cache a subset of the rows specify a WHERE clause. To cache a full replica of the rows omit the WHERE clause. Click Advanced MQT Settings to specify the columns, the indexes, the table space information, and the partitioning key information.

Name of the cache table: DB2ADMIN.EMPLOYEE

MQT schema: DB2ADMIN

MQT name: EMPLOYEE_MQT

MQT columns:

Name	Data Type	Length	Scale	Precision	Nullable
EMPNAME	CHARACTER	20	-	-	Yes
SALARY	INTEGER	-	-	-	Yes
EMPNO	CHARACTER	6	-	-	Yes
COMMISSION	DECIMAL	3	1	3	Yes

WHERE clause:

Table space: USERSPACE1

Valid information

Advanced MQT Settings...

< Back Next > Finish Cancel

Figure 74. Cache Table wizard

The DB2 Control Center also provides simple and intuitive controls for these tasks:

- Routing queries to cache tables
- Enabling and disabling the replication cache settings
- Modifying the settings for materialized query tables
- Dropping materialized query tables from a cache table

Monitoring federated queries

To see how your federated system is processing a query, you can get a snapshot of the remote query.

The snapshot monitor tracks two aspects of each query:

- The entire federated query as submitted by the application, which references nicknames, local tables, or both.

- For queries that use nicknames, one or more remote fragments. Remote fragments are the statements that are automatically generated and submitted to remote data sources in their native dialects on behalf of the federated query.

To monitor federated queries, you look at the work done at the federated server and the work done at remote servers in response to remote query fragments.

You can use a simple command to see the snapshot monitor results in text form, or you can direct the results of the snapshot monitor to a table that contains one row per query (federated or non-federated) and one row per query fragment.

Federated stored procedures

A *federated procedure* is a federated database object that references a procedure on a data source. Federated procedures are sometimes called federated stored procedures.

A federated procedure is to a remote procedure what a nickname is to a remote table. With a federated procedure, you can call a data source procedure. You can create a federated procedure by using the DB2 Control Center or from the command line.

WebSphere Federation Server provides the same powerful discovery functions for federated stored procedures as it does for servers, nicknames, and other objects.

Figure 75 shows the Create Federated Stored Procedures window after the Discovery window was used to generate a list of potential data source procedures where Name is like %EMP%. You then select the procedure that you want to create and the DB2 Control Center populates the fields and settings based on information from the data source procedure.

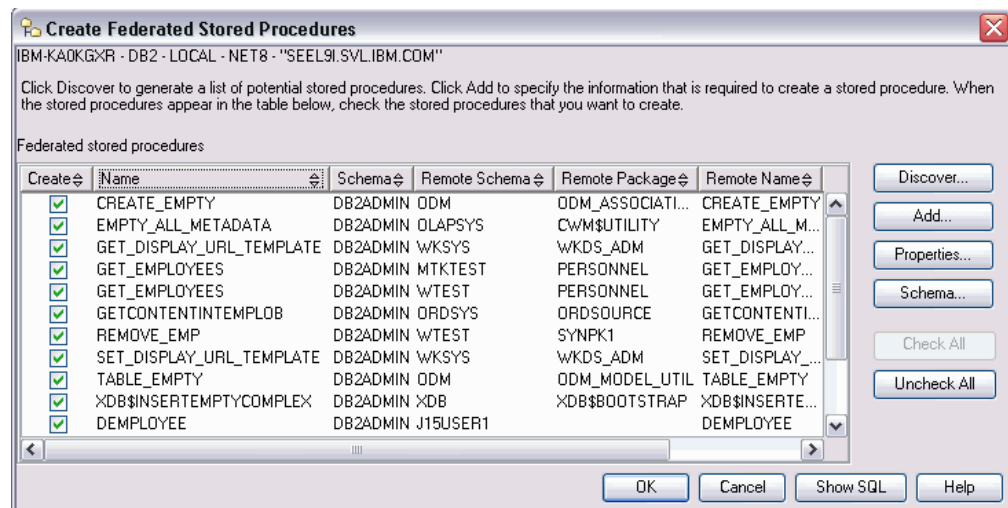


Figure 75. Create Federated Stored Procedures window

Related concepts

“SOA and data integration” on page 40

Enabling an IBM Information Server job as a Web service enables the job to participate in various data integration scenarios.

Information resources for WebSphere Federation Server

A variety of information resources can help you get started with WebSphere Federation Server.

The following publications and Web sites are available:

- WebSphere Information Integration Information Center (<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>)
- *Installation Guide for Federation, Replication, and Event Publishing on Linux, UNIX, and Windows* (GC19-1017-00)
- *Migrating to Federation Version 9* (SC19-1019-00)
- System requirements for WebSphere Federation Server (www.ibm.com/software/data/integration/federation_server/requirements.html)
- *Application Development Guide for Federated Systems* (SC19-1021-00)
- *Configuration Guide for Federated Data Sources* (SC19-1034-00)
- *Administration Guide for Federated Systems* (SC19-1020-00)
- WebSphere Federation Server product information (www.ibm.com/software/data/integration/federation_server/)
- *Data Federation with IBM DB2 Information Integrator V8.1* (www.redbooks.ibm.com/abstracts/sg247052.html?Open)
- *Performance Monitoring, Tuning and Capacity Planning Guide* (www.redbooks.ibm.com/abstracts/sg247073.html?Open)
- "IBM Federated Database Technology" (www.ibm.com/developerworks/db2/library/techarticle/0203haas/0203haas.html)
- "Using data federation technology in IBM WebSphere Information Integrator: Data federation design and configuration (Part 1 in a series introducing data federation)" (www-128.ibm.com/developerworks/db2/library/techarticle/dm-0506lin/)
- "Using data federation technology in IBM WebSphere Information Integrator: Data federation usage examples and performance tuning (Part 2 in a series introducing data federation)" (www-128.ibm.com/developerworks/db2/library/techarticle/dm-0507lin/)

Chapter 9. Companion products

Companion products for IBM Information Server provide extended connectivity for enterprise applications, change data capture, and high-speed, event-based replication and publishing from databases.

WebSphere DataStage Packs provide connectivity to widely used enterprise applications such as SAP and Oracle. These prebuilt packages enable companies to integrate data from existing enterprise applications into new business systems.

WebSphere DataStage change data capture products help you transport only the insert, update, and delete operations from a variety of commercial databases such as Microsoft SQL Server and IBM IMS.

WebSphere Replication Server provides a high-volume, low-latency data replication solution that uses WebSphere MQ message queues for high availability and disaster recovery, data synchronization, and data distribution.

WebSphere Data Event Publisher detects and responds to data changes in source systems, publishing changes to subscribed systems, or feeding changed data into other modules for event-based processing.

WebSphere DataStage Packs

WebSphere DataStage Packs enable a company to use WebSphere Information Analyzer, WebSphere QualityStage, WebSphere DataStage, and SOA-based capabilities to create a complete data integration solution.

The WebSphere DataStage Packs enable enterprise applications to benefit from the following capabilities of IBM Information Server:

- Support for complex transformations
- Automated data profiling
- Best-in-class data quality
- Integrated metadata management

The following products provide WebSphere DataStage connectivity for enterprise applications:

- WebSphere DataStage Pack for SAP BW
- WebSphere DataStage Pack for SAP R/3
- WebSphere DataStage Pack for Siebel
- WebSphere DataStage Pack for PeopleSoft Enterprise
- WebSphere DataStage Pack for Oracle Applications
- WebSphere DataStage Pack for JD Edwards Enterprise One
- WebSphere DataStage Pack for SAS

Where WebSphere DataStage Packs fit within the IBM Information Server architecture

To provide a complete data integration solution, WebSphere DataStage Packs perform the following functions:

- Manage connections to application source systems
- Import metadata from source systems
- Integrate design and job control in WebSphere DataStage
- Use WebSphere DataStage to load data to target applications, including other enterprise applications and data warehouses or data marts
- Allow bulk extract and load and delta processing

Figure 76 shows how WebSphere DataStage Packs fit within the IBM Information Server architecture.

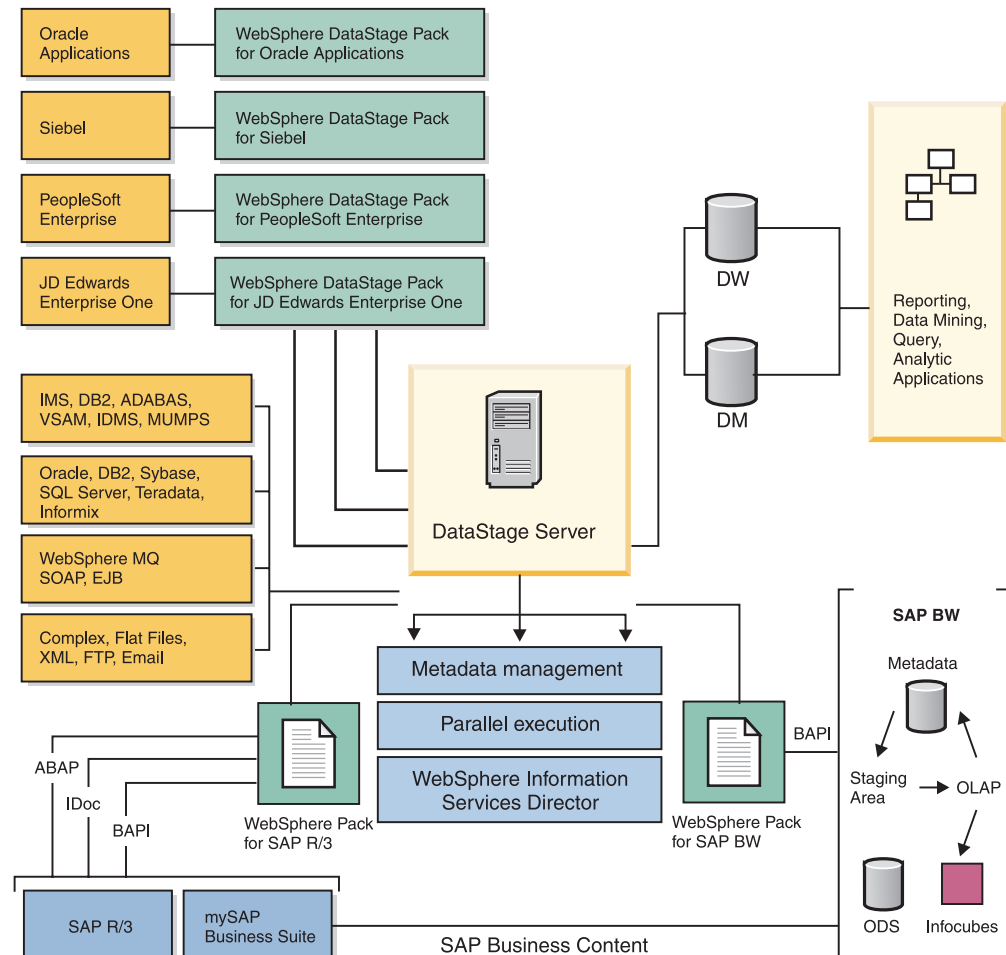


Figure 76. Architectural overview

Scenarios for IBM Information Server companion products

The following scenarios demonstrate WebSphere DataStage Packs in a business context:

Life science: Integrating around SAP BW

A global leader in life science laboratory distribution implemented SAP BW for sales, supply chain, and contact center metrics, but still has a huge amount of data on non-SAP systems in areas of enterprise resource planning, financials, and custom applications.

The IT department needs to support the business by delivering key sales and revenue status reports and an analytical workspace to corporate and

field staff in a timely way, meeting scalability requirements, easily managing metadata, and moving data from sources to targets.

The company uses the WebSphere DataStage Pack for Oracle Applications to access financial and accounts receivable data, WebSphere DataStage to transform it, and the WebSphere DataStage Pack for SAP BW to load the transformed data into SAP BW.

The company now quickly assembles data sources, performs data transformations, and enforces referential integrity before loading data into SAP BW. Data is ready faster and the process is easy to use. The business users can react more quickly to changes in their marketplace.

Lawn care and gardening company and SAP

One of the world's leading producers and marketers of lawn care and gardening products needed to minimize inventory costs at its 22 distribution hubs. But without knowing their customers' forecasted demand, the company was forced to carry excess inventory to protect against running out of stock and creating customer dissatisfaction because of lost sales.

Managers realized that if they could collect retail customer point-of-sale and sales forecast data from outside of their SAP applications, compare it to their internal SAP R/3 data, and properly load it into SAP BW, they could get a customized view of the data and properly plan shipments and inventory to meet demand.

The company implemented WebSphere DataStage, WebSphere Information Analyzer, and WebSphere DataStage Packs for SAP R/3 and SAP BW to collect sales data from customers and then cleanse, match, and load that data into SAP BW.

The resulting information helped the company lower inventory by 30 percent or \$99.4 million in one year, which contributed to a \$158 million increase in free cash flow. The project also significantly reduced distribution and holding costs.

Related concepts

Chapter 1, "Introduction," on page 1

Most of today's critical business initiatives cannot succeed without effective integration of information. Initiatives such as single view of the customer, business intelligence, supply chain management, and Basel II and Sarbanes-Oxley compliance require consistent, complete, and trustworthy information.

A closer look at WebSphere DataStage Packs

WebSphere DataStage Packs provide high-speed connectivity to packaged enterprise applications that uses the metadata capabilities of IBM Information Server to help companies integrate data and create consistent, trustworthy information.

Using WebSphere DataStage Packs to connect enterprise application data with IBM Information Server provides the following benefits:

- Faster deployment and reduced integration costs
- Faster integration of enterprise data and metadata
- Improved decision support by presenting aggregated views of the business
- Improved reporting and analysis

- Better use of enterprise applications by connecting to certified, vendor-optimized APIs

WebSphere DataStage Pack for SAP BW

The WebSphere DataStage Pack for SAP BW integrates non-SAP data into SAP Business Information Warehouse.

This pack populates the SAP warehouse with data from any source system:

- Enterprise data warehouses, complex flat files, mainframe legacy systems, customer systems, and supplier systems.
- Other enterprise applications, through the WebSphere DataStage Packs for Siebel, PeopleSoft Enterprise and JD Edwards EnterpriseOne, and Oracle Applications

Using SAP's standard business APIs, the WebSphere DataStage Pack for SAP BW automates the process of connecting to an SAP source and selecting source data through metadata integration, data mapping, and direct data load into SAP BW.

The pack also extracts information from SAP BW for use in other data marts, data warehouses, reporting applications, and targets. Using SAP BW OpenHub, the pack assists you in invoking BW Process Chains and collecting the output from OpenHub targets.

This pack provides direct access to, and creation of, SAP BW metadata from the WebSphere DataStage user interface. You can browse, select, create, and change SAP BW metadata objects such as Source Systems, InfoSources, InfoObjects, InfoCatalogs, and InfoPackages.

The pack also helps you develop BW integration jobs from a single environment. WebSphere DataStage Pack for SAP BW does not require pre-work in SAP BW before you can set up integration jobs.

The WebSphere DataStage Pack for SAP BW includes the following interfaces:

Staging Business API (BAPI) interface

The BW load plug-in uses SAP staging BAPIs to load data from any source into SAP's Business Information Warehouse (BW). Load events can be initiated from BW or WebSphere DataStage. You can stream data into SAP BW without writing the data to disk during the process.

OpenHub Interface

The BW extract plug-in works with SAP's OpenHub architecture to extract data from BW. InfoSpokes are activated by Process Chains to populate a relational table or flat file. The BW extract plug-in can initiate Process Chains or be called from an active Process Chain started from BW.

The WebSphere DataStage Pack for SAP BW is certified by SAP.

WebSphere DataStage Pack for SAP R/3

The WebSphere DataStage Pack for SAP R/3 helps you extract data from and load data into SAP R/3 and all mySAP Business Suite application modules.

The pack enables you to generate native SAP Advanced Business Application Programming (ABAP) code that eliminates manual coding while speeding deployment. The pack also enables you to capture incremental changes and produce event-triggered updates with SAP's Intermediate Documents (IDoc) functionality.

WebSphere DataStage IDoc extract interface retrieves IDoc metadata and automatically translates the segment fields into WebSphere DataStage for real-time SAP data integration. This interface helps you load quality data into SAP R/3 and the mySAP Business Suite, by using the built-in validations in SAP IDoc. Finally, the SAP Business Application Program Interface (BAPI) enables you to work with a business view, eliminating the need for knowledge of SAP application modules to move data into and out of SAP.

The WebSphere DataStage Pack for SAP R/3 includes the following interfaces:

- ABAP** Provides flexibility in constructing the data set to extract. This interface should be used for extracting large volumes of data when you have an understanding of the functional area from which to extract.
- IDoc** Typically used to move data between SAP instances within an enterprise. This interface should be used primarily for bulk data transfers when the desired data set is already represented by an available IDoc.
- BAPI** Most suited to transactional environments where the efficiency of mass data transfers is not a requirement.

The WebSphere DataStage Pack for SAP R/3 is certified by SAP.

WebSphere DataStage Pack for Siebel

The WebSphere DataStage Pack for Siebel enables you to extract data from and load data into Siebel applications so that you can leverage customer relationship management (CRM) information throughout the enterprise.

This pack includes interfaces to Siebel's Data Integration Manager (EIM) and Business Component layers, enabling you to work through familiar business views without understanding the underlying base tables. The WebSphere DataStage Pack for Siebel also includes an interface that makes it easy to select, identify, and extract data from the Siebel hierarchies.

With this pack installed, you can use WebSphere DataStage to customize extractions and automatically create and validate EIM configuration files. You can then launch EIM and use Business Components to map business objects from Siebel for use in other applications.

The corresponding Siebel data and metadata are easily extracted and loaded to a target such as SAP BW or any open environment, enabling decision support and CRM insight.

The WebSphere DataStage Pack for Siebel includes the following interfaces:

- EIM** EIM moves data back and forth between Siebel tables by using intermediate interface tables. This interface is most often used for bulk transfers and is most useful for initial loading of a Siebel instance. It blends the benefits of the Business Component and Direct Access interfaces at the expense of a more complicated job design process.

Business Component

This interface works with Siebel by using the Siebel Java Data Bean. This interface exposes the Siebel Business Object model, which corresponds directly with the objects that users are familiar with from working with Siebel client applications. Business Component is better suited to transactional operations than high-volume throughput.

Direct Access

Using an intelligent metadata browser, this interface enables WebSphere DataStage developers to define complex queries from their Siebel application data. Because this interface bypasses the Siebel application layer to protect the integrity of underlying data, it does not support load operations.

Hierarchy

This interface migrate hierarchies from Siebel to SAP BW. It uses database connectivity to extract data in a format compatible with SAP BW.

This Pack is validated by Seibel.

WebSphere DataStage Pack for Oracle Applications

The WebSphere DataStage Pack for Oracle Applications enables you to extract data from the entire Oracle E-Business Suite of applications, including Oracle Financials, Manufacturing, CRM, and others.

Like the other WebSphere DataStage Packs, the Oracle Pack simplifies integration of Oracle Applications data in the diverse target environments that are supported by WebSphere DataStage, such as SAP BW or any other business intelligence environment.

The pack extracts data from Oracle flex fields by using enhanced processing techniques. It extracts data from the complex reference data structures with the Hierarchy Access component.

WebSphere DataStage Pack for PeopleSoft Enterprise

The WebSphere DataStage Pack for PeopleSoft Enterprise is designed to extract data from PeopleSoft Enterprise application tables and trees.

This pack helps you select and import PeopleSoft Enterprise metadata into WebSphere DataStage, where the metadata can be managed with other enterprise information. A metadata browser enables searches by table name and description or by business view from the PeopleSoft Enterprise Panel Navigator.

WebSphere DataStage Pack for JD Edwards EnterpriseOne

Organizations that implement Oracle's JD Edwards EnterpriseOne product can use the data extraction and loading capabilities of the WebSphere DataStage Pack for JD Edwards EnterpriseOne.

This pack speeds integration from EnterpriseOne applications by using standard ODBC calls to extract and load data. The pack enables you to extract business views, which are the pre-joined database tables that are constructed with user-defined metadata.

The WebSphere DataStage Pack for JD Edwards EnterpriseOne also enables JD Edwards EnterpriseOne data to be used in other applications, and the pack loads JD Edwards EnterpriseOne with important legacy, flat file, and other source data.

WebSphere DataStage Change Data Capture

Data integration tasks typically involve transforming and loading data from source systems on a regular basis. When you move data from large databases, you often want to move only the data that has changed in the source system since the previous extract and load process.

The ability to capture only the changed source data is known as change data capture (CDC). IBM Information Server provides CDC capability in addition to its ability to move all the data from a source to a target system in batch or real time. Capturing changes reduces traffic across your network, enables shorter batch windows, and enables you to use events on your source systems to initiate data integration processes.

The following methods are commonly used to capture database changes:

- Read the database recovery logs and extract changes to the relevant tables
- Use the replication functions provided by the database
- Use database triggers

CDC can be delivered in two ways:

Event driven

Called the *push* model. Updates are applied in response to an event on the data source. Change capture agents identify and send changes to the target system as soon as the changes occur. This model enables customers to update their analytical applications on-demand with the latest information.

Interval-driven

Called the *pull* model. Updates are applied at regular intervals in response to requests from the target. Requests might occur every five minutes or every five days. The size of the interval is usually based on the volatility of the data and the latency requirements of the application.

CDC uses the native services of the database architecture, adheres to the database vendor's documented formats and APIs, and minimizes the invasive impact on any operational systems.

The following CDC companion products are available to work with IBM Information Server:

- IBM WebSphere DataStage Changed Data Capture for Microsoft SQL Server
- IBM WebSphere DataStage Changed Data Capture for Oracle
- IBM WebSphere DataStage Changed Data Capture for DB2 for z/OS
- IBM WebSphere DataStage Changed Data Capture for IMS

WebSphere Replication Server

WebSphere Replication Server distributes, consolidates, and synchronizes data for high availability, high throughput, and business continuity.

Two types of replication, Q replication and SQL replication, support a broad range of business scenarios:

Q replication

A high-volume, low-latency replication solution that uses WebSphere MQ message queues to transmit transactions between source and target databases or subsystems. A capture process reads the DB2 recovery log for changes to source tables and sends transactions as messages over queues, where they are read and applied to targets. Q replication offers the following advantages:

Minimum latency

Changes are sent as soon as they are committed at the source and read from the log.

High-volume throughput

The capture process can keep up with rapid changes at the source, and the multithreaded apply process can keep up with the speed of the communication channel.

Minimum network traffic

Messages are sent using a compact format, and data-sending options enable you to transmit the minimum amount of data.

Asynchronous

The use of message queues enables the apply process to receive transactions without needing to connect to the source database or subsystem. If either of the replication programs is stopped, messages remain on queues to be processed whenever the program is ready. Because the messages are persistent, the source and target remain synchronized even if a system or device fails.

Q replication supports DB2 for z/OS and DB2 for Linux, UNIX, and Windows as source platforms. Q replication supports the following target platforms: DB2 for z/OS, DB2 for Linux, UNIX, and Windows, Informix, Microsoft SQL Server, Oracle, and Sybase.

SQL replication

SQL replication captures changes to source tables and views and uses staging tables to store committed transactional data. The changes are then read from the staging tables and replicated to corresponding target tables. SQL replication offers the following advantages:

Capture once

With staging tables, data can be captured and staged once for delivery to multiple targets, in different formats, and at different delivery intervals. Both tables and views are supported as sources.

Flexibility

You can replicate continuously, at intervals, or for one time only. You can also trigger replication with database events. You can replicate a subset of the table by excluding columns and filtering rows. You can also use expressions and other functions to transform data before it is applied.

Hub-and-spoke configurations

You can replicate data between a master data source and one or more replicas of the source. Changes that are made to the master source are propagated to the replicas, and changes that are made to the replicas are also propagated to the master source. Whenever a conflict occurs between the data that is sent from the master source and data this is sent from a replica, the data from the master source takes precedence.

SQL replication supports the following source and target platforms: DB2 for z/OS, DB2 for Linux, UNIX, and Windows, DB2 for iSeries™, Informix, Microsoft SQL Server, Oracle, and Sybase. In addition, SQL replication supports Teradata targets.

Related concepts

“Introduction to WebSphere Federation Server” on page 112

WebSphere Federation Server allows organizations to virtualize their data and provide information in a form that applications and users need while hiding

the complexity of the underlying sources. Data virtualization allows information to be accessed through a common interface that centralizes the control of data access.

WebSphere Data Event Publisher

WebSphere Data Event Publisher captures changed-data events and publishes them as WebSphere MQ messages that can be used by other applications to drive subsequent processing.

Changes to source tables, or events, are captured from the log and converted to messages in an Extensible Markup Language (XML) format. This process provides a push data integration model that is ideally suited to data-driven enterprise application-integration (EAI) scenarios and change-only updating for business intelligence and master-data management.

Each message can contain an entire transaction or only a row-level change. Messages are put on WebSphere MQ message queues and read by a message broker or other applications. You can publish subsets of columns and rows from source tables so that you publish only the data that you need.

You can use event publishing for a variety of purposes that require published data, including feeding central information brokers and Web applications, and triggering actions based on insert, update, or delete operations at the source tables. Source tables can be relational tables in DB2 for z/OS and DB2 for Linux, UNIX, and Windows.

Related concepts

“Introduction to WebSphere Federation Server” on page 112

WebSphere Federation Server allows organizations to virtualize their data and provide information in a form that applications and users need while hiding the complexity of the underlying sources. Data virtualization allows information to be accessed through a common interface that centralizes the control of data access.

Information resources for IBM Information Server companion products

A variety of information resources can help you get started with IBM Information Server companion products.

HTML help is available for all of the following connectivity features and packs. The following publications are available in PDF format:

WebSphere DataStage connectivity products

- *WebSphere DataStage Connectivity Guide for the Dynamic Relational Stage*
- *WebSphere DataStage Connectivity Guide for Teradata Databases*
- *WebSphere DataStage Connectivity Guide for Sybase Databases*
- *WebSphere DataStage Connectivity Guide for Stored Procedures*
- *WebSphere DataStage Connectivity Guide for SAS*
- *WebSphere DataStage Connectivity Guide for IBM Red Brick Warehouse*
- *WebSphere DataStage Connectivity Guide for Oracle Databases*
- *WebSphere DataStage Connectivity Guide for ODBC*
- *WebSphere DataStage Connectivity Guide for Netezza Performance Server*

- *WebSphere DataStage Connectivity Guide for Microsoft SQL Server and OLE DB Data*
- *WebSphere DataStage Connectivity Guide for iWay Servers*
- *WebSphere DataStage Connectivity Guide for IBM Informix Databases*
- *WebSphere DataStage Connectivity Guide for IBM WebSphere MQ Applications*
- *WebSphere DataStage Connectivity Guide for IBM UniVerse and UniData*
- *WebSphere DataStage Connectivity Guide for IBM DB2 Databases*
- *WebSphere DataStage Connectivity Guide for IBM WebSphere Information Integrator Classic Federation Server for z/OS*

WebSphere Replication Server and WebSphere Data Event Publisher

- *Introduction to Replication and Event Publishing (GC19-1028-00)*
- *ASNCLP Program Reference for Replication and Event Publishing (SC19-1018-00)*
- *Replication and Event Publishing Guide and Reference (SC19-1029-00)*
- *SQL Replication Guide and Reference (SC19-1030-00)*

IBM Information Server and suite components

- *IBM Information Server Planning, Installation, and Configuration Guide*
- *IBM Information Server Quick Start Guide*

Accessing information about the product

IBM has several methods for you to learn about products and services.

You can find the latest information on the Web:
www.ibm.com/software/data/integration/info_server/

To access product documentation, go to publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order.
- To order publications by telephone in the United States, call 1-800-879-2755.

To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide.

Providing comments on the documentation

Please send any comments that you have about this information or other documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- Send your comments using the online readers' comment form at www.ibm.com/software/awdtools/rcf/.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, please include the location of the text (for example, a title, a table number, or a page number).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM trademarks and certain non-IBM trademarks are marked at their first occurrence in this document.

See www.ibm.com/legal/copytrade.shtml for information about IBM trademarks.

The following terms are trademarks or registered trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel[®], Intel Inside[®] (logos), MMX and Pentium[®] are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names might be trademarks or service marks of others.

Index

A

- accessibility 135
- Administrator client 106
- Aggregator stage 87
- architecture, IBM Information Server 5
- asset rationalization 45

B

- baseline comparison 57
- benchmarks
 - WebSphere DataStage MVS Edition 108
- blocking step 75
- business initiatives aided 1

C

- cache tables, creating 121
- Character Investigation 71
- column analysis 54
- common connectivity 5
- common services 5
- companion products
 - information resources 133
 - WebSphere Data Event Publisher 125
 - WebSphere DataStage Change Data Capture 125
 - WebSphere DataStage Packs 125
 - WebSphere Replication Server 125
- compensation by federated database 117
- Complex Flat File stage 87
- Compose tab 76
- content, use in SOA 40
- cross-domain analysis 54
- cross-platform services 13
- customer scenarios 45, 63
 - WebSphere DataStage Packs 125
 - WebSphere Information Analyzer 45
 - WebSphere QualityStage 63

D

- data investigation 70
- data matching 70
- data monitoring and trending 45, 57
- data partitioning 8
- data pipelining 8
- data profiling and analysis 54
- data quality assessment 45
- data quality issues 63
- data rules and metrics 57
- data standardization 70
- data stewardship 21
- data survivorship 70
- data transformation
 - zSeries 107
- data transformation, types 81
- DB2 Control Center 120, 121

- DB2 stage 87
- Designer interface 89
- Director interface 104
- documentation
 - accessible 135
 - companion products 133
 - WebSphere Business Glossary 28
 - WebSphere DataStage 110
 - WebSphere Federation Server 124
 - WebSphere Information Analyzer 61
 - WebSphere Information Services
 - Director 42
 - WebSphere MetaBrokers and bridges 28
 - WebSphere QualityStage 80
- domain analysis 54
- dynamic repartitioning 8

E

- Enterprise Java Beans (EJB) 35

F

- federated database 115
- federated objects, creating 120
- federated queries, monitoring 122
- federated server 115
- federated stored procedures 123
- federated system, definition 112
- federated two-phase commit
 - overview 118
- field parsing 71
- foreign key analysis 54

G

- geocoding 67
- grid computing 12

H

- Holding area 76

I

- IBM Information Server
 - architecture and concepts 5
 - capabilities 1
 - companion products 125
 - cross-platform services 13
 - logging services 13
 - metadata services 17
 - overview 1
 - reporting services 15
 - scalability 11
 - scheduling services 13
 - security services 13

- IBM Information Server (*continued*)
 - service-oriented architecture (SOA) 29
 - characteristics 29
 - Service-Oriented Architecture (SOA)
 - benefits 33
 - run-time components 35
 - support for grid computing 12
 - Web console 21
 - information providers 36
 - integrated find 23
 - Investigate stage 71

J

- J2EE
 - as an SOA component 35
- JD Edwards EnterpriseOne, WebSphere DataStage Pack for 130

L

- legal notices 137
- lexical analysis 71
- logging services 13

M

- mapping table 57
- Massively Parallel Processing (MPP), exploiting 11
- Match Designer 74
 - using 76
- Match Frequency stage 74
- Match passes 76
- Match stage 75
- Match stages
 - overview 74
- match statistics 76
- matching step 75
- metadata interchange 23
- metadata repository 23
- metadata services
 - overview 17
 - scenarios 17
- metamodels 23
- multiple binding support for SOA 35
- multisite update, federated
 - see federated two-phase commit 118

N

- nicknames 116

O

- Oracle Applications, WebSphere DataStage Pack for 130

overview
federated two-phase commit 118

P

parallel processing
basics 8
overview 8
pattern report 71
PeopleSoft Enterprise, WebSphere
DataStage Pack for 130
performance
WebSphere DataStage MVS
Edition 108
poor data, results 63
postal certification 67
primary key analysis 54
probabilistic matching 63

Q

Q replication 131
query optimizer 117

R

range checking 57
Rational Data Architect 119
reference match 75
reporting services 15
repository, metadata 23

S

SAP
WebSphere DataStage Pack for SAP
BW 128
WebSphere DataStage Pack for SAP
R/3 128
scalability 11
scenarios
metadata services 17
service-oriented architecture
(SOA) 29
WebSphere DataStage 81
WebSphere Federation Server 112
scheduling services 13
screen readers 135
search, metadata 23
security
WebSphere Information Analyzer 60
security services 13
server definitions, federated 116
service-oriented architecture (SOA) 29
scenarios 29
service-ready integration 33
services
batch jobs 33
creating 36
logging 13
overview 29
reporting 15
scheduling 13
security 13
topologies 33

Siebel, WebSphere DataStage Pack
for 129
SOA 29
WebSphere Information Services
Director 36
with WebSphere Information
Integration 40
SOAP over HTTP (Web services) 35
Sort stage 87
SQL replication 131
Standardize stage 73, 74
statistics, WebSphere QualityStage
matches 76
stewardship 21
stored procedures, federated 123
Survive stage 79
survivorship rules 79
Symmetric Multiprocessing (SMP),
exploiting 11

T

threshold-balanced parallelism, in
SOA 35
trademarks 139
Transformer stage 87
two-phase commit for federated
transactions
see federated two-phase commit 118

U

Unduplicate Match 75
unified metadata 5
unified parallel processing engine 5
unified user interfaces 5
UNIX Systems Services (USS) 109
user mappings 116

V

validity checking 57

W

WAVES 67
WebSphere AuditStage 48, 52, 57
WebSphere Business Glossary
information resources 28
overview 17, 20
using 21
WebSphere Data Event Publisher 125,
133
WebSphere DataStage
accessing metadata services 89
architecture 83
command-line interfaces 104
Complex Flat File stage 89
concepts 87
creating jobs 89
designing jobs 89
dynamic relational stage 89
importing and exporting jobs 89
information resources 110
job sequences 89

WebSphere DataStage (*continued*)

jobs 87
jobs, defined 87
links and containers 87
managing jobs 89
metadata exchange 89
monitoring jobs 104
overview 81
overview of user interfaces 89
projects 87
reviewing job log files 104
running jobs 104
scenarios 81
slowly changing dimension stage 89
SQL builders 89
stage properties 89
stages, defined 87
stages, examples 87
table definitions 87, 89
transformations 83
transformer stage 89
using 87
WebSphere DataStage and QualityStage
Administrator 83, 106
WebSphere DataStage and QualityStage
Designer 71, 83, 89
accessing metadata 79
WebSphere DataStage and QualityStage
Director 83, 104
WebSphere DataStage Change Data
Capture 125, 131
WebSphere DataStage Enterprise for
z/OS 109
WebSphere DataStage MVS Edition 107
overview 108
WebSphere DataStage Pack for JD
Edwards EnterpriseOne 130
WebSphere DataStage Pack for Oracle
Applications 130
WebSphere DataStage Pack for PeopleSoft
Enterprise 130
WebSphere DataStage Pack for SAP
BW 128
WebSphere DataStage Pack for SAP
R/3 128
WebSphere DataStage Pack for
Siebel 129
WebSphere DataStage Packs 125
benefits 127
customer scenarios 125
list 125
WebSphere Federation Server
cache tables 121
compensation 117
information resources 124
monitoring queries 122
nicknames 116
overview 111
query optimizer 117
Rational Data Architect 119
scenarios 112
stored procedures 123
wrappers 116
WebSphere Information Analyzer 45
architecture 48
dashboard view 48
data monitoring and trending 57

- WebSphere Information Analyzer
 - (continued)*
 - data profiling and analysis 54
 - information resources 61
 - overview 45
 - Project view 52
 - security 60
 - using 52
- WebSphere Information Analyzer Workbench 54
- WebSphere Information Services Director
 - information resources 42
 - using 36
- WebSphere MetaArchitect 23
- WebSphere MetaBrokers and bridges 23
 - information resources 28
 - overview 17
- WebSphere Metadata Server
 - overview 17, 23
- WebSphere MQ 63
- WebSphere QualityStage 63, 67
 - accessing metadata 79
 - architecture 67
 - components 67
 - information resources 80
 - Investigate stage 71
 - Match Frequency stage 74
 - Match stage 75
 - Match stages 74
 - methodology 70
 - overview 63
 - Standardize stage 73, 74
 - Survive stage 79
 - using 70
- WebSphere Replication Server 125, 131
- Word Investigation, example 71
- Worldwide Address Verification and Enhancement System 67
- wrappers 116

Z

- zSeries, data transformation 107



Printed in USA

SC19-1049-01



Spine information:

IBM Information Server Version 8.0.1

IBM Information Server Introduction

