INTEGRATED ETL AND MODELING

ORACLE 9*i* WAREHOUSE BUILDER

An Oracle White Paper
July 2002

INTRODUCTION

Many corporations face the complex task of converting their organizations' data into information. Making sense of massive data flows from dispersed systems is a Herculean task, which in many cases requires lengthy projects that tie up large number of resources. Even then, many efforts to create such a system fail or, at most, only partially succeed.

There are many reasons that such efforts fail, one of them being the wrong choice of software or software vendors. Organizations find themselves not only integrating their own data, but also integrating vendor's tools to create one application. When faced with the task of integrating disparate sources of data, the last thing an organization needs is the added complexity of integrating tools from disparate vendors.

The solution Oracle offers, with the database at the core, and Oracle Warehouse Builder, alleviates this issue by delivering an integrated suite of products. Based on its market-leading database, Oracle provides a set of tightly integrated tools capable of solving the core problem: the conversion of data into information.

The purpose of this paper is to introduce the core component of the tools suite, Oracle Warehouse Builder. This paper also explains why Warehouse Builder is the right tool for integrating data and deploying the much-desired information flow to executives and other knowledge workers.

After a high level overview of Warehouse Builder, the paper will discuss the important features of this tool and the product road map or future direction.

WHAT IS ORACLE 9*I* WAREHOUSE BUILDER?

Oracle Warehouse Builder solves the complex problem of data integration between various dispersed data sources and targets. In other words, Warehouse Builder is Oracle's tool to design and deploy enterprise data warehouses, data marts, and business intelligence applications. To top it off, Warehouse Builder provides all the necessary functionality to fully maintain the life cycle of the system you developed.

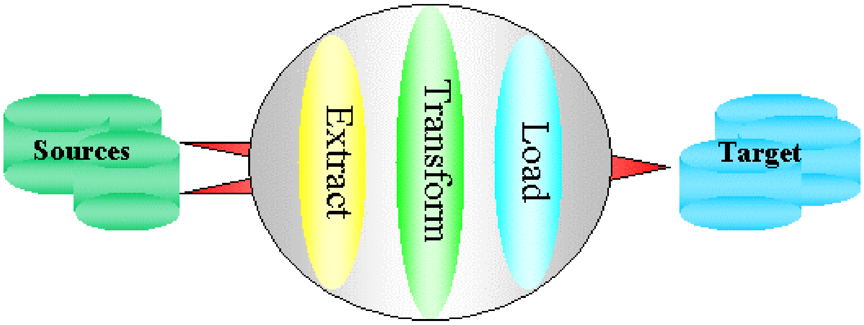


Figure 1 Warehouse Builder's Core Capabilities

In fact, Oracle Warehouse Builder is more than an ETL or a design tool; as it combines the best characteristics of both tools into one easy to use product.

In addition to integrating two major components of any Business Intelligence system design, Warehouse Builder leverages other investments in Oracle technology. It is the hub in Oracle's Business Intelligence tools suite. Warehouse Builder provides a tight integration with ad-hoc query tools, OLAP, and relational database features, thus leveraging the entire solution stack provided by Oracle.

HOW DOES WAREHOUSE BUILDER ACHIEVE ITS GOAL?

Creating a business intelligence application is a complex process. It involves various steps and phases, which may span a great number of systems, resources, and functional areas. Warehouse Builder enables you to:

- Import data source definitions
- Design and create the target database schema
- Define and create the data flows between sources and targets
- Manage and update the source definitions
- Manage, update, and upgrade the target schema
- Design and create the ad-hoc query tool environment
- Design and create the OLAP environment
- Leverage Oracle database features to ensure scalability, reliability and flexibility

To provide these capabilities, Warehouse Builder boasts the following components:

- A Repository – This set of relational tables and views in an Oracle database stores the metadata definitions.
- A Client Application – This java application provides you with an easy to use graphical interface that guides you through the definition and design of the application.
- A Code Generator – This component generates scripts based on the metadata in the repository, which you then apply to your target database schema. The generator is designed to utilize the full feature set of the Oracle database and the generated scripts provide optimal performance on the Oracle database.
- A Runtime Environment – The target environment not only stores the data loaded from the sources in a schema you design and create; it also holds audit information on the loading processes that move the data into this schema.
- A Runtime Audit Viewer – A Java application to inspect the audit data in the runtime environment.
- A Reporting Environment – The reporting capabilities of Warehouse Builder enable you to view your metadata from a web-based application and provide information to a larger audience. The information is aggregated for more business-oriented users.

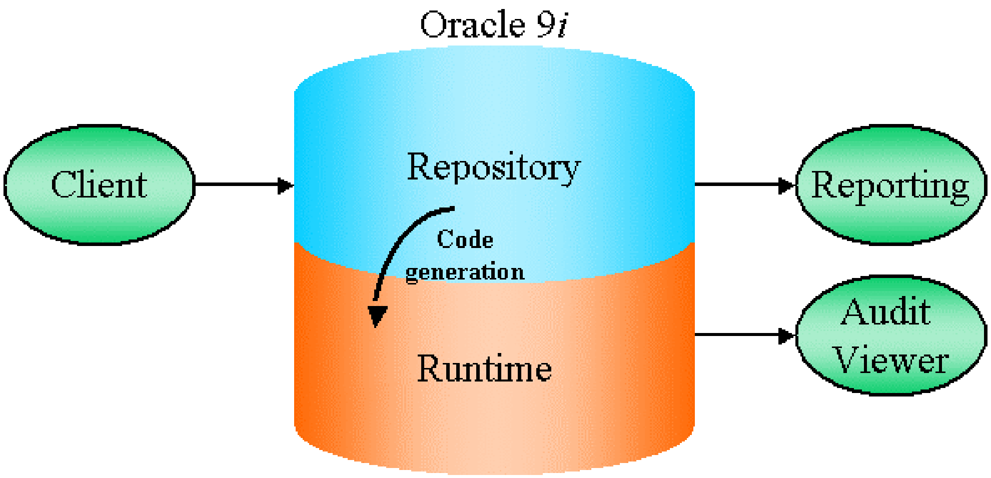


Figure 2 Components of Warehouse Builder

BENEFITS OF USING WAREHOUSE BUILDER

Warehouse Builder components translate into direct benefits for your organization:

- Rapid System Design – Warehouse Builder reduces the design time by providing visual models, wizard-driven user interfaces, and a library of predefined transformations.
- Centralized Design Information – Because of the repository, all information regarding the design of the system is stored and managed in one central place. This reduces duplication of metadata and creates a single source of truth, thus eliminating the extra cost of inconsistencies.
- Reduce Time to Incorporate Changes – Advanced Life Cycle management features based on the single repository enable a smooth maintenance process.

- Error-Free Code – Because it is generated from a design stored in one place, the code is not only error-free, but also easy to re-create, upgrade, and maintain.
- Leverage Technology Investments – By using the market-leading Oracle database as both a transformation engine and storage facility, Warehouse Builder leverages Oracle knowledge and technology, providing a higher return on investment to organizations.

BUSINESS INTELLIGENCE DESIGN

Key components in any business intelligence application are the data storage, data flows, and data query environment. Using Warehouse Builder, you can design all three of these components within one tool.

DESIGNING DATA STORAGE WITH WAREHOUSE BUILDER

There are many types of business intelligence systems: relational, multi-dimensional, OLAP, and any number of combinations of these. All serve specific needs and have unique characteristics. The typical problem is that these systems are built with separate tools on separate environments and cannot be combined. The ideal situation would be to combine an OLAP environment with pre-calculated data and with and ad-hoc query environment, thus giving people optimal flexibility. Choosing a toolset that provides this integration reduces the development costs and increases the users' satisfaction and productivity.

Warehouse Builder is ideally positioned to solve the design issue resulting from various best-of-breed solutions. It sits on top of the Oracle database and enables you to design all relevant objects in that database.

Warehouse Builder enables you to design the following relational objects in the database:

- Tables – including constraints and indexes
- Views – including the view query
- Materialized Views – including the refresh options
- Fact tables – including bitmap indexes and keys to the dimensions
- Dimensions – including the relational dimension and storage object

As of Oracle9*i*, the OLAP constructs are also available in the Oracle database. Warehouse Builder is uniquely positioned to make these constructs available from a design perspective. Within the same user interface, it enables you to design for the OLAP catalog, reusing some of the relational design objects:

- Dimensions – representing dimensions in OLAP
- Facts – representing cubes in OLAP

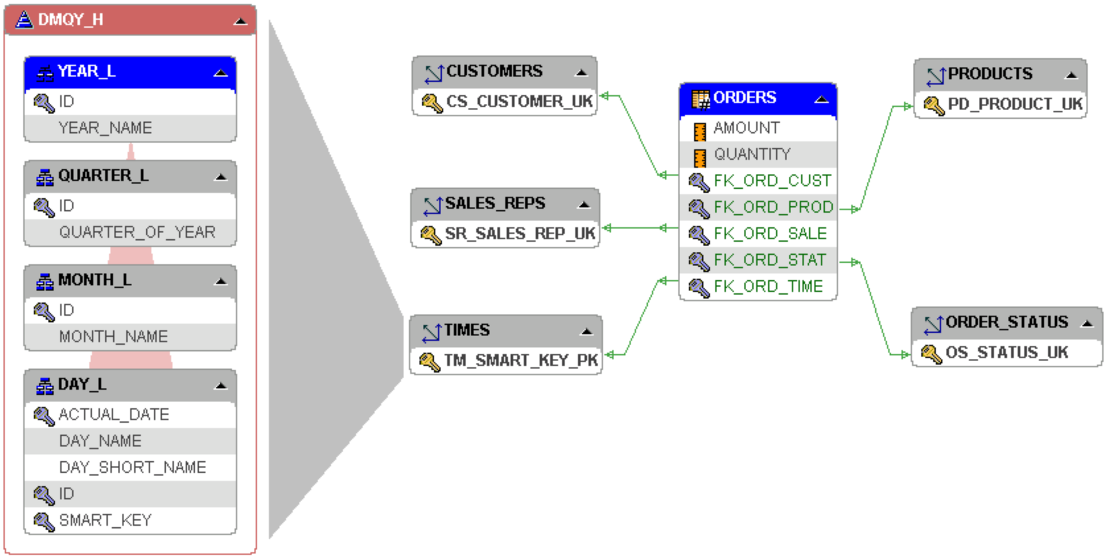


Figure 3 Dimensions surrounding a Fact in Warehouse Builder

With this, Warehouse Builder is one of the very few tools to have full design capabilities for both environments, even reusing the objects from one environment in the creation of the other. It is then obvious that Warehouse Builder saves time in designing the end user environment. An additional benefit is the user experience, which is now limited to one interface for both systems.

DESIGNING DATA FLOWS WITH WAREHOUSE BUILDER

Moving data from normalized structures to de-normalized structures is a tough job that typically involves a lot of coding. Most source structures are large and complex for the purposes of data integrity and insert/update needs, and it becomes quite an effort to merge this data with that of other sources into a newly designed environment that is optimized for query retrieval. To top it off, once the complex extraction routines are built, changes require a complete overhaul of the system, often suffering months of delay before its implementation.

Choosing the right tool reduces the time to delivery and the complexity of creating the system. Warehouse Builder is the right tool for the job. Why is this? Consider some of the key aspects of the mapping component:

- Graphical User Interface – The mapping[1] component makes design and documentation easy and accessible for people who are not necessarily familiar with coding
- Logical Design – Makes design a one-time exercise for multiple usages
- Re-Use of Pre-Built Components – Enabling standard activities to be performed with pre-built operators and functions
- Code Generation – Produces optimized code for the Oracle platform
- Open Standards – Using SQL that is verifiable and portable
- ABAP Code Generation – Allows native access to SAP packaged applications

[1] Extraction, Transforming and Loading (Data Flow) processes are called mappings in Warehouse Builder

- Extensibility – Allows re-use of existing code, which can now be managed from Warehouse Builder
- Pre-built Auditing – Automatically logs any information to a set of pre-built tables

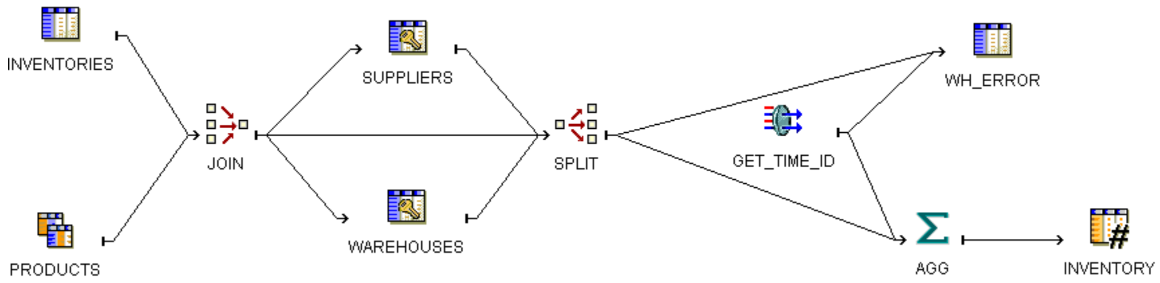
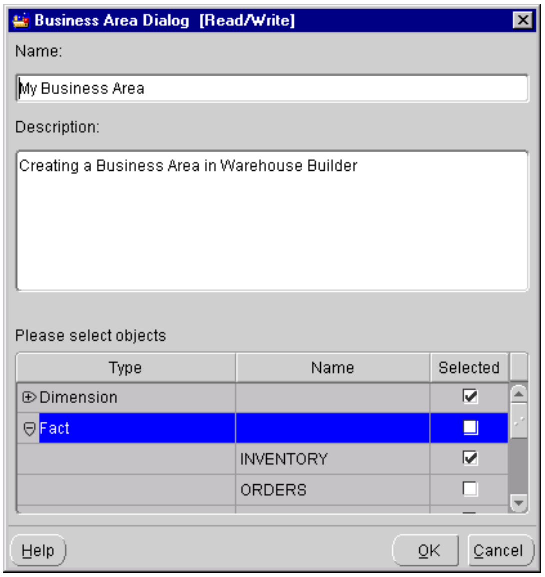


Figure 4 Data flow Diagram (Mapping) in Warehouse Builder

Armed with these features, Warehouse Builder reduces the complexities of the data flow design, greatly simplifying the creation and testing of these flows.

Designing Query Environments with Warehouse Builder

There are many data query tools on the market. Some tools are capable of querying a large number of different databases, while others are highly specialized. The biggest problem with all these stand-alone tools is the integration with the data storage design. Most tools read the design out of the database catalog, which results in a disparate repository with duplicate information. Refreshing the design of the data storage area causes a number of refreshes on a number of tools, costing time and money, and often leading to discrepancies.



Business Area Dialog [Read/Write]

Name:

My Business Area

Description:

Creating a Business Area in Warehouse Builder

Please select objects

Type	Name	Selected
⊕ Dimension		☑
⊖ Fact		□
	INVENTORY	☑
	ORDERS	□

Figure 5 Deriving a Query Tool Environment

With Warehouse Builder, you simply derive a representation with specific characteristics from the design environment you already have and publish it to the query tool. This process is much simpler and faster than reentering all this information in a separate tool. It also makes the integration tighter and easier to maintain.

SUMMARY

Warehouse Builder provides all three key design areas in an integrated manner. In data storage, Warehouse Builder builds in flexibility by enabling you to designing both the dimensional and the relational objects, including indexes. Based on this design, you can create the business intelligence application from Warehouse Builder, resulting in a fast and easy rollout of the end-user solution. Last but not least, the graphical mapper facilitates a scalable ETL solution maximizing the usage of the Oracle database engine.

Business Intelligence Metadata

The repository stores all metadata that is created via either the graphical user interface, import, or derivation. Metadata is the key aspect in Warehouse Builder's potential, which makes the repository a vital component.

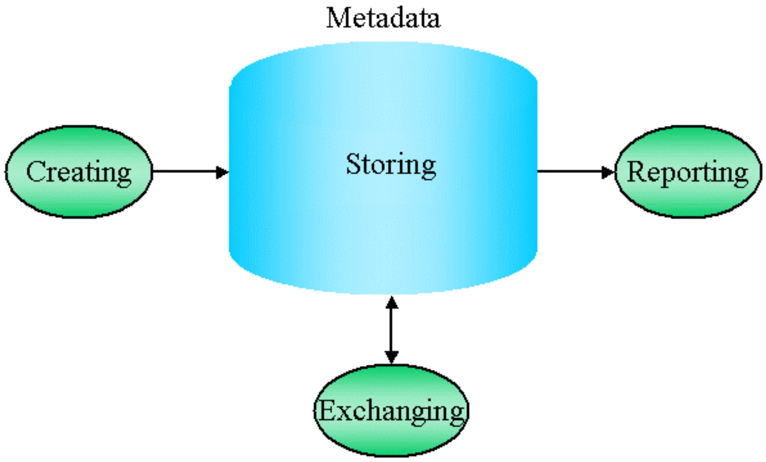


Figure 6 Utilizing Metadata from Warehouse Builder

In general, metadata is defined as "data about data," meaning that metadata describes data. Warehouse Builder utilizes information about data, of either a structural or business nature, and uses it in all its for design purposes.

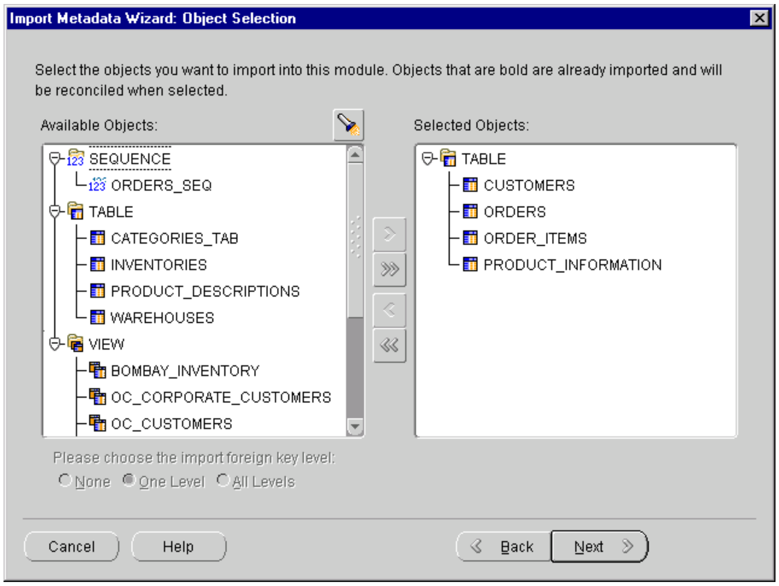
Creating Metadata

Metadata is stored in almost any information system. An important feature for Warehouse Builder is to capture or create this metadata. There are various ways of creating metadata in Warehouse Builder:

- Import Definitions –Warehouse Builder can read metadata definitions directly from the source and target system
- Import Design Definitions –Warehouse Builder can use existing design definitions to capture the definition of a source
- Import Stored Code –Warehouse Builder can incorporate previously written code
- Create Definitions – You can add additional metadata through either the user interface or the APIs

Importing Definitions

Warehouse Builder can capture definitions from many sources but will always represent them to the user in a similar way, hiding the complexity of the disparate sources. All relational systems are approached as if they were Oracle systems and their objects are represented as relational objects.



Import Metadata Wizard: Object Selection

Select the objects you want to import into this module. Objects that are bold are already imported and will be reconciled when selected.

Available Objects:

- SEQUENCE
  - ORDERS_SEQ
- TABLE
  - CATEGORIES_TAB
  - INVENTORIES
  - PRODUCT_DESCRIPTIONS
  - WAREHOUSES
- VIEW
  - BOMBAY_INVENTORY
  - OC_CORPORATE_CUSTOMERS
  - OC_CUSTOMERS

Selected Objects:

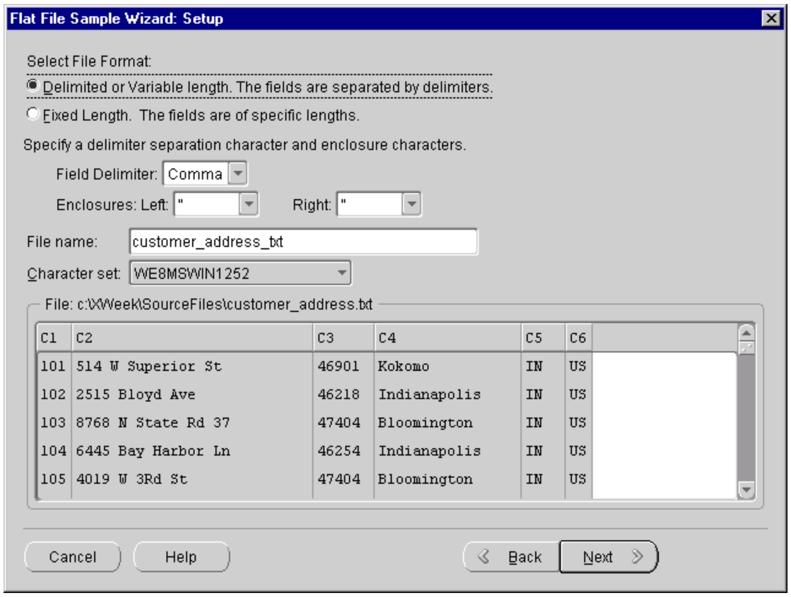
- TABLE
  - CUSTOMERS
  - ORDERS
  - ORDER_ITEMS
  - PRODUCT_INFORMATION

Please choose the import foreign key level:

○ None ● One Level ○ All Levels

Figure 7 Importing Relational Object Metadata

Files are also frequently used in business intelligence applications. Warehouse Builder provides file sampling in order to capture the file structure into the metadata repository.



Flat File Sample Wizard: Setup

Select File Format:

● Delimited or Variable length. The fields are separated by delimiters.

○ Fixed Length. The fields are of specific lengths.

Specify a delimiter separation character and enclosure characters.

Field Delimiter: Comma

Enclosures: Left: " Right: "

File name: customer_address_txt

Character set: WE8MSWIN1252

File: c:\XWeek\SourceFiles\customer_address.txt

C1	C2	C3	C4	C5	C6
101	514 W Superior St	46901	Kokomo	IN	US
102	2515 Bloyd Ave	46218	Indianapolis	IN	US
103	8768 N State Rd 37	47404	Bloomington	IN	US
104	6445 Bay Harbor Ln	46254	Indianapolis	IN	US
105	4019 W 3Rd St	47404	Bloomington	IN	US

Figure 8 Sampling a Flat File definition

Finally, Warehouse Builder can directly read from SAP through SAP's Business Object Layer. Even the definitions of complex objects, such as Pool and Cluster tables, can be imported into the repository and handled as any other data objects.

IMPORT DATA DESIGN DEFINITIONS

Some systems are designed in other tools, mainly CASE tools. In order to be able to read such metadata, Warehouse Builder provides import facilities for these tools. The main tools supported are:

- Oracle Designer
- CA ERWin

Importing metadata from Oracle Designer works much like importing metadata from a database catalog. A connection is made to the Designer repository and information is moved into Warehouse Builder. Metadata is exchanged with CA ERWin via the metadata exchange mechanism, in which case Warehouse Builder reads the ERWin export files into the repository.

In both cases, the objects appear as relational objects and can be used in modeling your business intelligence applications.

Import Code Definitions

In order to reuse code that has your organization may have already created, Warehouse Builder allows you to import PL/SQL code into transformation libraries. From here, you can maintain the code or periodically refresh the PL/SQL via re-import.

In this way, Warehouse Builder enables reuse of code, which saves time and effort by eliminating the need to redevelop the same structures again.

Creating Definitions

You can create new metadata using the client application that comes with Warehouse Builder (see the section on designing with Warehouse Builder for more details).

Storing Metadata

The Warehouse Builder repository is a set of relational tables in the Oracle database. The metadata you create through the various methods described is stored inside this repository. Using the Oracle database to store metadata leads to a number of advantages in an enterprise solution:

- Reliability
- Scalability
- Easy access through SQL

The repository has some advanced features that are required to support an enterprise scale repository:

- Support for multi user access
- Public Views for read-only 3rd party user access
- Public APIs and scripting for read/write access
- Extensibility of the repository with custom defined attributes
- Archive and Restore (versioning) mechanisms
- Export and Import facilities with Merge capabilities

In essence, the Warehouse Builder repository is the groundwork for your business intelligence application, housing all the metadata required to build and manage such a system. With its inherent and unique characteristics, the repository is the true foundation of your business intelligence application.

REPORTING ON METADATA

Metadata is accessible through a number of means. Technical users use the client application. Since the client application also provides read/write access to the metadata, it may not be advisable to give this type of access to all your users (e.g., business users in addition to technical users).

What you need is an easy-to-use, read-only access method that can be rolled out to a large number of users. The answer to this is web-based reporting. The benefits are an intuitive interface that is common to most users. As a method of deployment, Oracle 9*i*AS is ideal because it offers scalability, flexibility, and security to the application.

The Warehouse Builder browser solution is based on the following:

- Access via Public Views – Gives users a documented way of access, which they can extend if necessary
- Pre-built HTML reports – Supplying direct access to numerous reports including graphical data lineage and impact analysis reports
- Secure and scalable access via Oracle Portal – Enables many users to view metadata reports that can be browsed in a secure manner
- Extensibility – Enabling the addition of custom reports and links to the portal
- Integration – Using the Warehouse Builder portlets in conjunction with other applications, the organization can build its own business intelligence portal

Some of the pre-built reports, specifically the data lineage and impact analysis reports, provide extremely useful information during your system's life cycle. Using these reports, you can determine either the origin of data or the impact that changes may have on your application. The reports can save you a lot of time, not only during your development of the application, but also when you are verifying or tracing information at a later stage.

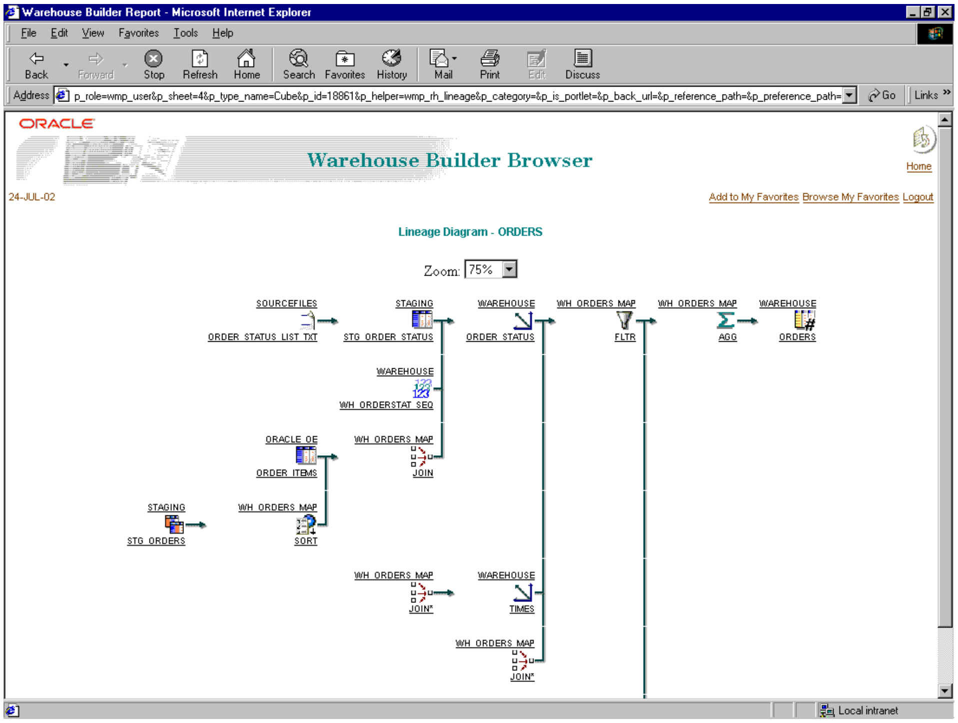


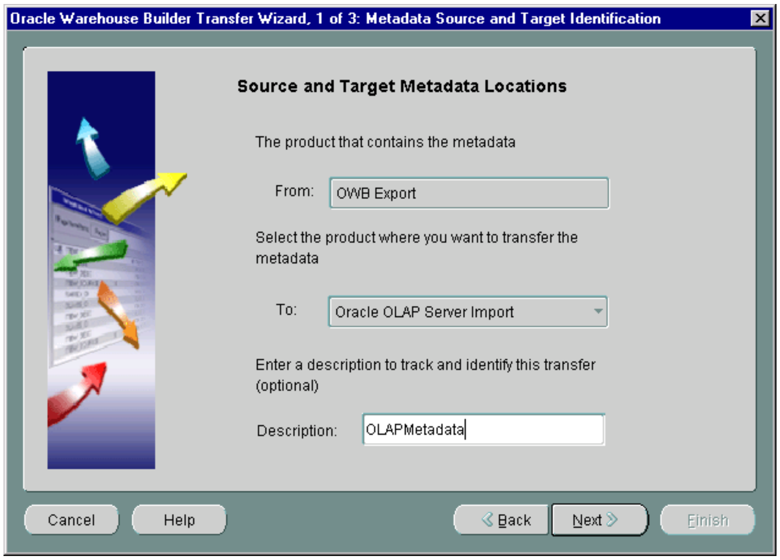
Figure 9 A Lineage Diagram from Warehouse Builder Browser

Exchanging Metadata

While Warehouse Builder is the central place for metadata, exchanging metadata with other tools or environments can be a desirable feature. Warehouse Builder supports various metadata exchange formats:

- OMG CWM[2] – An open standard endorsed by most major business intelligence vendors that allows Warehouse Builder to import from and export to other open tools in the market
- Oracle Discoverer Export Format – To export metadata to Oracle Discoverer, you can use Discoverer's export format
- Oracle OLAP – Allows Warehouse Builder to export metadata to and import metadata from the OLAP catalog
- Oracle Express – Allows Warehouse Builder to export metadata to RAM/RAA

[2] Object Management Group's Common Warehouse Model is an open standard specifically created to support warehouse tools and exchange of warehouse metadata. For more information visit http://www.omg.com



Oracle Warehouse Builder Transfer Wizard, 1 of 3: Metadata Source and Target Identification

Source and Target Metadata Locations

The product that contains the metadata

From: OWB Export

Select the product where you want to transfer the metadata

To: Oracle OLAP Server Import

Enter a description to track and identify this transfer (optional)

Description: OLAPMetadata

Figure 10 Exchanging Metadata with Warehouse Builder

SUMMARY

Warehouse Builder offers an enterprise-strength repository on a scalable and open platform. It provides various means of accessing the metadata, and supports all types of users and access specifications. It provides a number of important exchange formats that enable you to work with 3rd party or custom applications against the Warehouse Builder repository.

RUNTIME MANAGEMENT

In Warehouse Builder, you define a logical system and then deploy it into a runtime environment. The runtime environment is the environment in which you run, schedule, and audit the actual data loads. Warehouse Builder provides the following components to help you accomplish these tasks:

- Runtime Schema – This schema is installed from Warehouse Builder and contains the audit tables and packages
- Runtime Audit Viewer – This specialized tool allows you to review the audit information for your data loads
- Job Scheduling in Oracle Enterprise Manager – To allow you to schedule loads in a controlled fashion
- Process Flow Creation in Oracle Workflow – To allow you to create dependency management diagrams and notification services

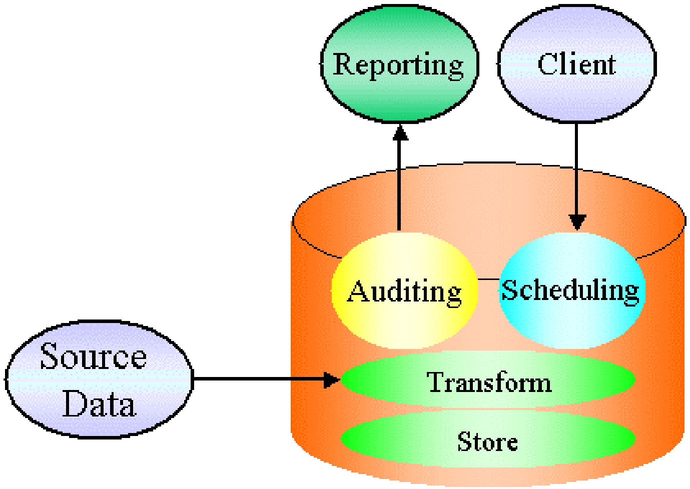


Figure 11 The Runtime Environment for Warehouse Builder

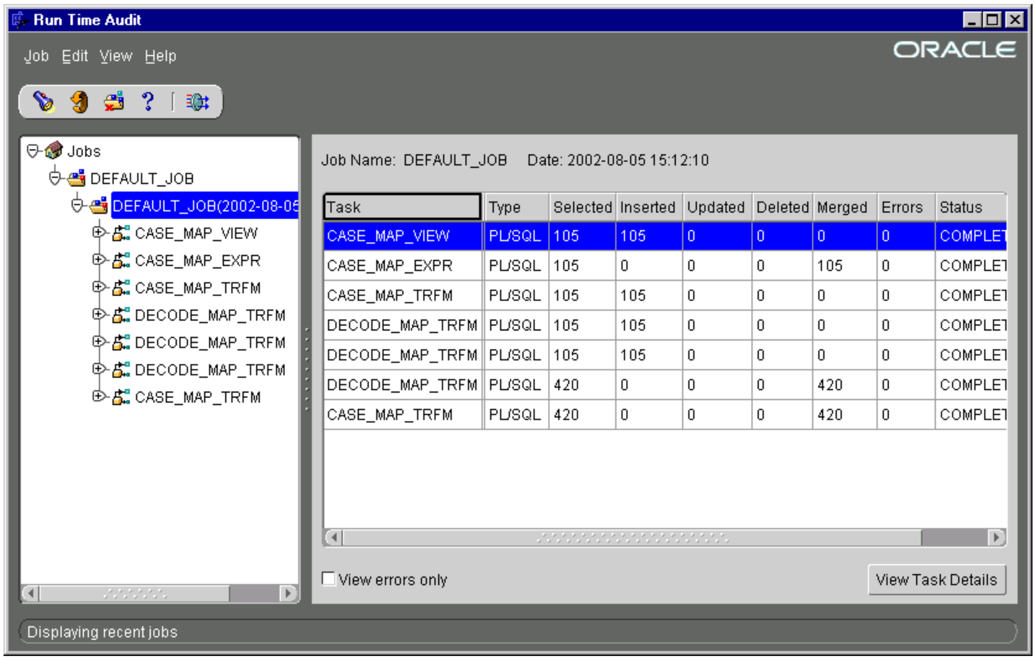
RUNTIME SCHEMA

The runtime schema is the actual physical location of the runtime information. You have the choice to install these objects into a target schema or to reserve a separate schema for this information.

The biggest advantage of the runtime schema is the time it saves you in developing a working system. In having a default runtime schema, Warehouse Builder provides you with an easy-to-use auditing system. There is no need to define anything except the level of auditing required, and Warehouse Builder then writes the necessary information to the runtime tables.

RUNTIME AUDIT VIEWER

The runtime audit viewer gives you access to the audit information generated from the mappings into the runtime schema.



Job Name: DEFAULT_JOB Date: 2002-08-05 15:12:10

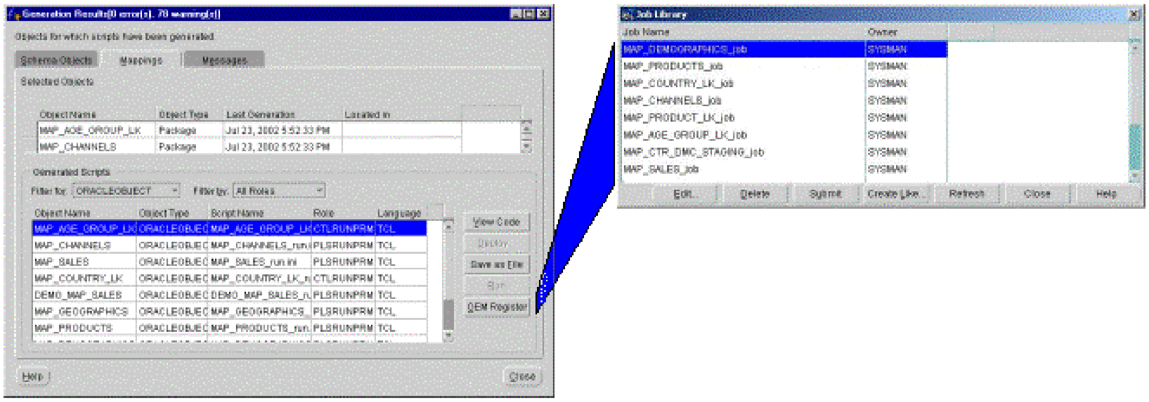
Task	Type	Selected	Inserted	Updated	Deleted	Merged	Errors	Status
CASE_MAP_VIEW	PL/SQL	105	105	0	0	0	0	COMPLET
CASE_MAP_EXPR	PL/SQL	105	0	0	0	105	0	COMPLET
CASE_MAP_TRFM	PL/SQL	105	105	0	0	0	0	COMPLET
DECODE_MAP_TRFM	PL/SQL	105	105	0	0	0	0	COMPLET
DECODE_MAP_TRFM	PL/SQL	105	105	0	0	0	0	COMPLET
DECODE_MAP_TRFM	PL/SQL	420	0	0	0	420	0	COMPLET
CASE_MAP_TRFM	PL/SQL	420	0	0	0	420	0	COMPLET

Figure 12 Runtime Audit Viewer

It is an easy-to-use tool that allows you to browse through the jobs that have been run on this runtime environment. The runtime audit viewer provides access to the error codes in the event of errors.

JOBS SCHEDULING

Warehouse Builder uses the standard scheduler available in Oracle Enterprise Manager (OEM). To use OEM, Warehouse Builder is integrated with OEM and, with the simple click of a button, the mappings created in Warehouse Builder appear in OEM's job library.



Objects for which scripts have been generated

Selected Objects

Object Name	Object Type	Last Generation	Located in
MAP_AGE_GROUP_LK	Package	Jul 23, 2002 5:52:33 PM	
MAP_CHANNELS	Package	Jul 23, 2002 5:52:33 PM	

Generated Scripts

Filter for: ORACLEOBJECT Filter by: All Roles

Object Name	Object Type	Script Name	Role	Language
MAP_AGE_GROUP_LK	ORACLEOBJEC	MAP_AGE_GROUP_LK	CTLRUNPRM	TCL
MAP_CHANNELS	ORACLEOBJEC	MAP_CHANNELS_run.	PLSRUNPRM	TCL
MAP_SALES	ORACLEOBJEC	MAP_SALES_run.ini	PLSRUNPRM	TCL
MAP_COUNTRY_LK	ORACLEOBJEC	MAP_COUNTRY_LK_r	CTLRUNPRM	TCL
DEMO_MAP_SALES	ORACLEOBJEC	DEMO_MAP_SALES_r	PLSRUNPRM	TCL
MAP_GEOGRAPHICS	ORACLEOBJEC	MAP_GEOGRAPHICS_	PLSRUNPRM	TCL
MAP_PRODUCTS	ORACLEOBJEC	MAP_PRODUCTS_run.	PLSRUNPRM	TCL

Job Name	Owner
MAP_DEMOGRAPHICS_job	SYSMAN
MAP_PRODUCTS_job	SYSMAN
MAP_COUNTRY_LK_job	SYSMAN
MAP_CHANNELS_job	SYSMAN
MAP_PRODUCT_LK_job	SYSMAN
MAP_AGE_GROUP_LK_job	SYSMAN
MAP_CTR_DMC_STAGING_job	SYSMAN
MAP_SALES_job	SYSMAN

Figure 13 Registering Jobs in OEM from Warehouse Builder

OEM provides all scheduling functionality. Because Oracle Warehouse Builder generates the underlying code, you are guaranteed to receive audit information as if the job were run manually.

PROCESS FLOW CREATION

In order to deal with dependencies between jobs (processes), Warehouse Builder leverages the power of Oracle Workflow. Once a job is registered in OEM, you can register the same job as a process in Workflow. From here you can create complex dependency diagrams.

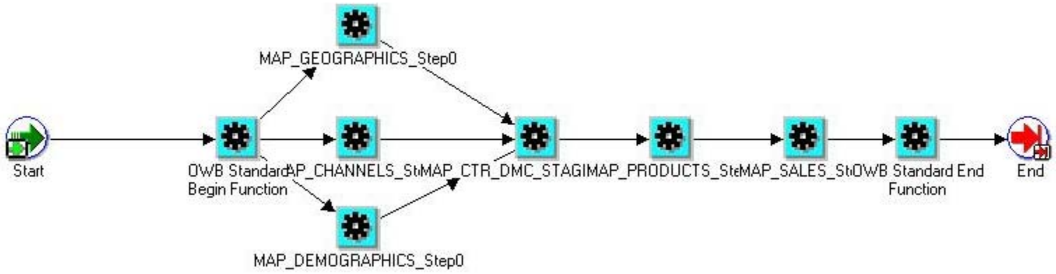


Figure 14 A Process Flow Diagram

The starting process is registered in OEM to allow for one job to act as the starting job from a scheduling perspective.

The main advantage of using the Oracle Workflow engine is that a rich set of features is available to you. Error loops, messaging via email or pagers, and other features are some highlights of Oracle Workflow.

SUMMARY

The runtime environment is a powerful asset to Warehouse Builder. It provides significant supporting functionality required for a business intelligence system, and brings it to the table out of the box. Using the standard runtime environment saves you a lot of effort. There is no need to develop such a system; it is pre-built and ready to use.

Data Quality

According to The Data Warehouse Institute[3], data quality problems cost US businesses more than $600 Billion per year. Addressing data quality when selecting an ETL tool is therefore essential.

Although data quality is not solely a business intelligence issue, it has a large impact on these systems. Low quality of data leads to incorrect information and consequently to bad decisions. Lack of correct information leads to distrust of the system, dramatically reducing the return on investment for the business intelligence system. Warehouse Builder provides an integrated data quality solution to mitigate data quality problems and contribute to a better decision support environment.

Data Quality Components

The functionality provided by Warehouse Builder rests on several components:

- Name & Address Client Component – Is available in the mapping tool and cleanses data while extracting and loading it into a target schema
- Name & Address Server Component – Is the database component that acts as a cleansing engine and is called from the code generated from the client
- Name & Address Libraries – Are the components that hold the data against which the customer data is validated

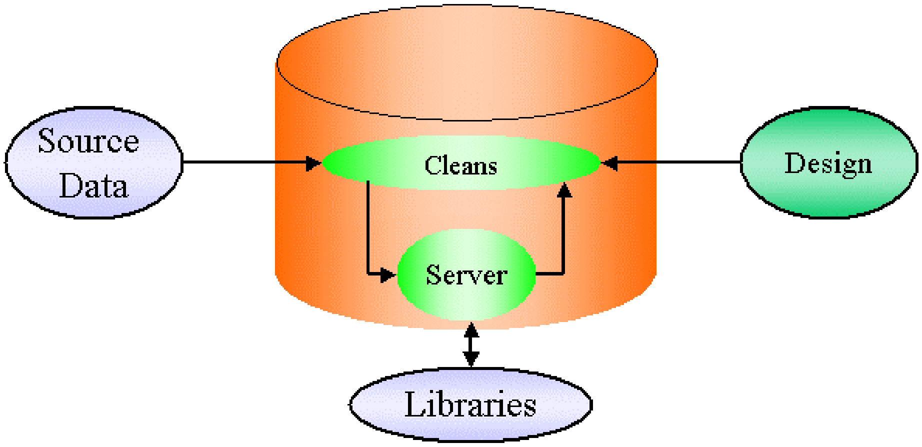


Figure 15 Name & Address components for Warehouse Builder

Name & Address Client

Designing data cleansing components is a function of the regular Warehouse Builder design client using the standard interface to which you quickly become accustomed during design. You can add a data-cleansing component to your mapping as part of the data flow diagram (mapping) design.

A mapping with an added data cleansing component behaves like any other mapping and Warehouse Builder generates the appropriate code to utilize the additional components.

[3] Source: The Data Warehouse Institute Report Series, www.dw-institute.com

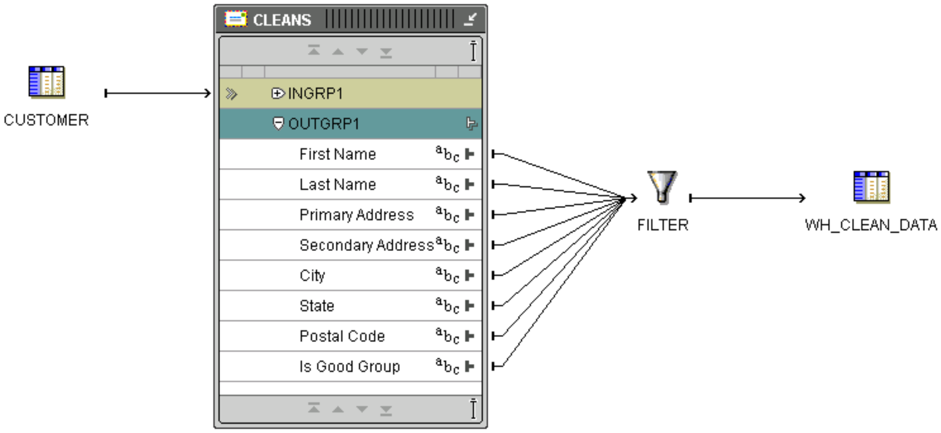


Figure 16 Data Cleansing Design in Warehouse Builder

In essence, the mapping calls the server and passes data to it. The server, in turn, performs a set of activities using the libraries and then returns a clean data record to the mapping, which processes it further if required. All cleansing takes place without any complex tasks.

The Name & Address Server

Cleansing data is a processing-intensive job and needs a powerful engine to clean data with high performance. Warehouse Builder uses the Oracle database as its engine for all processing, including processing data cleansing jobs. Because such processing constitutes a highly specialized function, a separate component is added to the database: the Name & Address Server.

The Name & Address Server performs these tasks:

- Parses Data – Decomposes the incoming data record in discrete components
- Standardizes Data – Applies a common format to common components to allow matching
- Corrects Data – Based on the information provided by the library
- Validates Data – Based on the information present in the library
- Augments Data – Based in the information present in the library

In addition to these tasks, the Name & Address Server has a Service Provider Interface that enables you to plug in other parties' libraries, resulting in an extensible and flexible architecture.

The Name & Address Libraries

Correcting and cleansing cannot be done without any reference to a common truth. When you use Warehouse Builder, this common truth is supplied by the Name & Address libraries, which are provided by Trillium[4]. These libraries provide international support covering the following regions:

- Europe, Middle East and Africa

[4] For more information about the integration with Trillium visit http://www.trilliumsoft.com/site/content/press_139.asp

- Latin America
- Asia Pacific
- North America – including Geo Coding

The libraries are updated frequently and available to Warehouse Builder customers who license the Name & Address Server.

SUMMARY

With integrated international data cleansing support, Warehouse Builder is a key component in any data quality initiative. It provides scalable and easy-to-use facilities that you can extend. Warehouse Builder requires a short learning curve and investment, with a high return in the form of clean and well-structured data that sets the foundation for better decision making.

LIFE CYCLE MANAGEMENT

Change is one of the major components responsible for the complexity of most business intelligence systems. For most systems, change is a constant and can have a severe impact on implementation if the wrong tool is chosen.

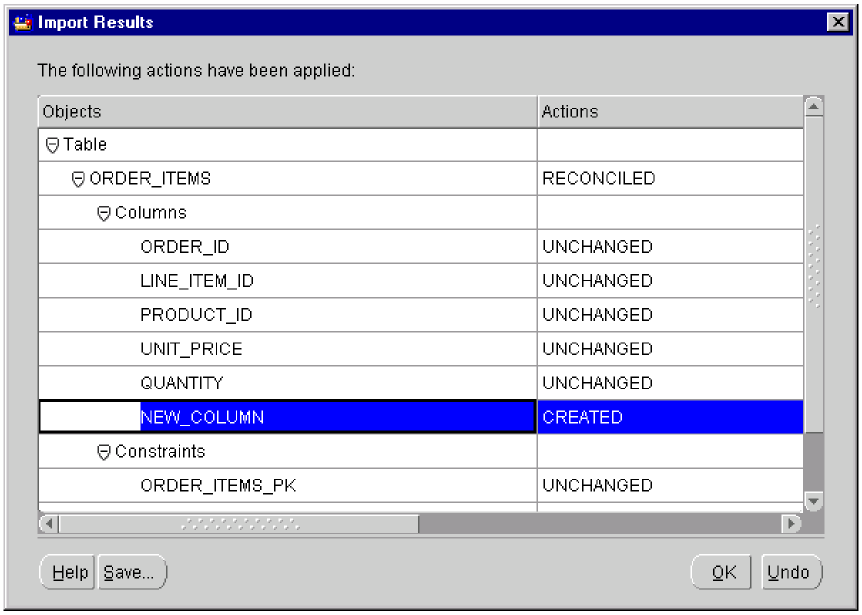
Warehouse Builder provides unique change management capabilities to keep your development on track even in the most dynamic environments. Some of the changes Warehouse Builder gracefully deals with include:

- Changes in Source Systems – Changes in source systems are reconciled into the repository
- Changes Impacting Mappings – Are reconciled using a sophisticated matching mechanism that allows changes to trickle through to the mappings
- Changes on Target Systems – Are converted in mutation scripts and applied to the target structures

In addition, Warehouse Builder prompts you with each change it detects so that you canevaluate the change's impact. You can choose whether or not to apply the change.

CHANGES IN SOURCE SYSTEMS

One of the most frequent types of modification is change, addition, or deletion of columns in source tables. Warehouse Builder deals with such changes by comparing the actual source definition with the existing metadata in the Warehouse Builder repository.



The following actions have been applied:

Objects	Actions
Table	
ORDER_ITEMS	RECONCILED
Columns	
ORDER_ID	UNCHANGED
LINE_ITEM_ID	UNCHANGED
PRODUCT_ID	UNCHANGED
UNIT_PRICE	UNCHANGED
QUANTITY	UNCHANGED
NEW_COLUMN	CREATED
Constraints	
ORDER_ITEMS_PK	UNCHANGED

Figure 17 Change Report for Warehouse Builder

When it detects differences in structure or in constraints, Warehouse Builder reports its findings. After inspecting the information, you can choose whether or not to apply any changes.

Once you have evaluated the changes to an object, you can determine their impact on the rest of the system by running an impact report on your metadata.

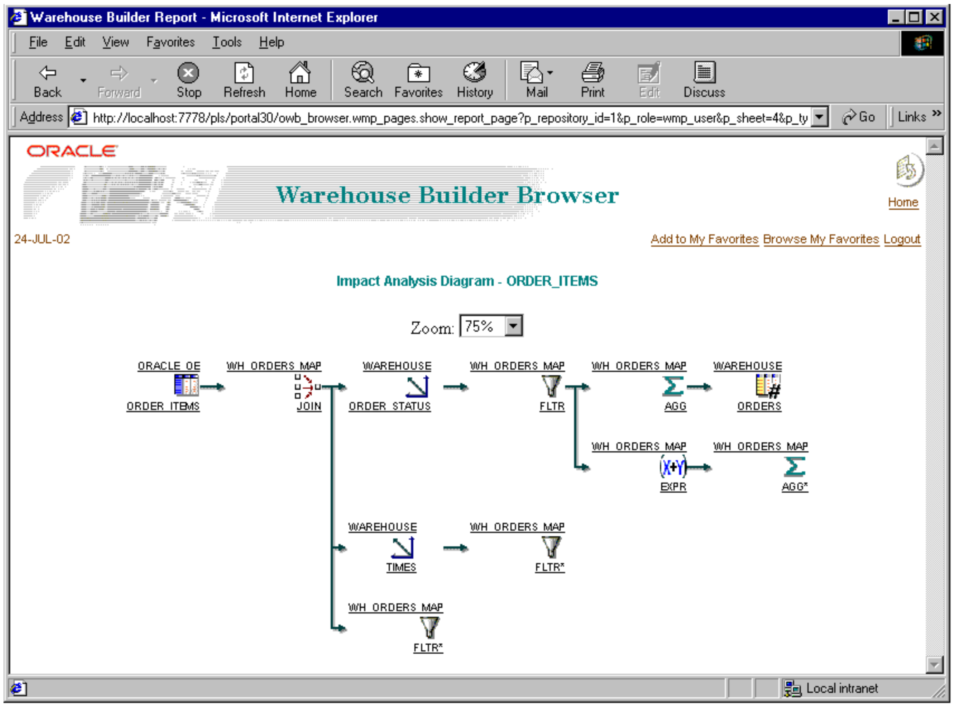


Figure 18 Judging Impacts of Changes in Warehouse Builder

Now you have a complete picture of the impact of a simple change on your system. You can now plan the work associated with this change and implement it in a controlled manner.

Changes Impacting Mapping

Once you identify all changes, you modify the components affected. Because the most complexity occurs in the mappings, you need a sophisticated mechanism to apply changes to modified objects. Warehouse Builder provides such a mechanism: (advanced) reconcile.

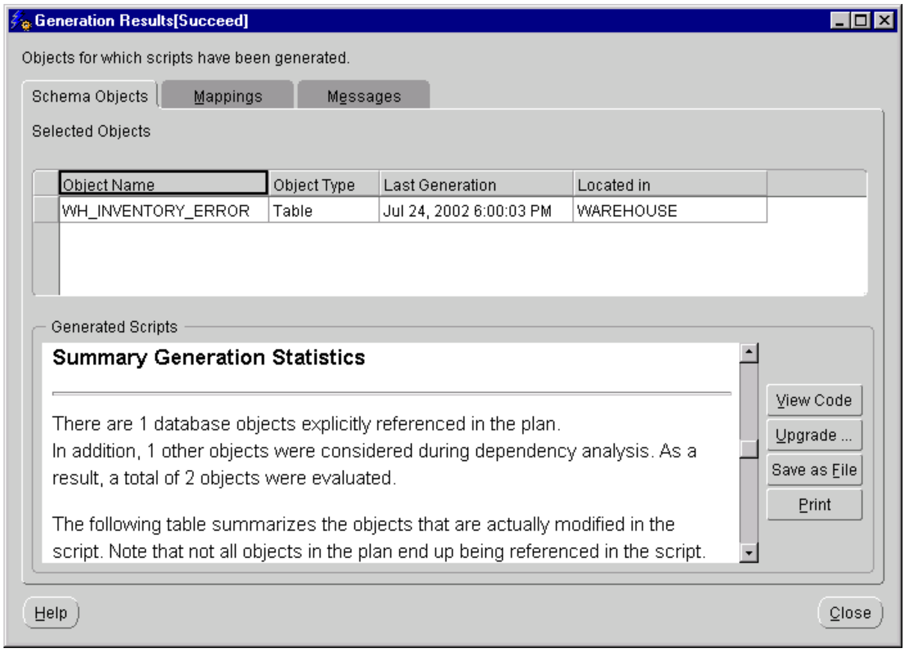
Reconcile means that the object that has been changed can be reconciled with the definition that is stored in mapping. This way, no existing mapping lines or expressions are broken; they are they are preserved as long as no items are deleted. Data type changes, additional columns, and even deleted columns are reflected in the mapping diagrams.

The major benefit to you is the preservation of your mappings, and, based on the impact diagram you only need to adjust existing mappings, not rebuild them. This is a tremendous reduction of time and effort.

Changes to Target Systems

Once all the metadata is adapted to reflect the changes applied to the sources, you may find that you have to modify the target tables or objects as well. Warehouse Builder generates the upgrade scripts for you.

Based on the metadata, Warehouse Builder performs a comparison with the currently deployed objects and reports any differences detected. Again, you can first investigate the impact and then act on it.



Generation Results[Succeed]

Objects for which scripts have been generated.

Selected Objects

Object Name	Object Type	Last Generation	Located in
WH_INVENTORY_ERROR	Table	Jul 24, 2002 6:00:03 PM	WAREHOUSE

Generated Scripts

Summary Generation Statistics

There are 1 database objects explicitly referenced in the plan.
In addition, 1 other objects were considered during dependency analysis. As a result, a total of 2 objects were evaluated.

The following table summarizes the objects that are actually modified in the script. Note that not all objects in the plan end up being referenced in the script.

Figure 19 Reporting on Target Changes in Warehouse Builder

Based on any identified differences, Warehouse Builder generates upgrade scripts that preserve the data in your target.

In order to allow you to roll back invasive upgrades even after applying the changes, the original target object is renamed before the upgrade takes place. This is Change Management in a secure, efficient, and easy way.

SUMMARY

An often-overlooked aspect of application design is the amount of change required during its life cycle. Warehouse Builder has capabilities to handle diverse changes without interrupting your entire system. It detects structural changes in sources, allows propagation through the mappings to the targets, and does so in a safe and easy way that enables you to evaluate the impact before applying the changes. These capabilities make dealing with any amount of change a manageable task.

API AND SCRIPTING

Warehouse Builder has a comprehensive user interface that allows the creation and manipulation of metadata in the repository as you have read in preceding sections. In some cases, you may wish to create or manipulate metadata from a different interface. Reasons for this can be:

- Building applications on top of the Warehouse Builder repository
- Providing metadata manipulation privileges to a large audience with a limited user interface
- Create a separate reporting environment with limited write back capabilities

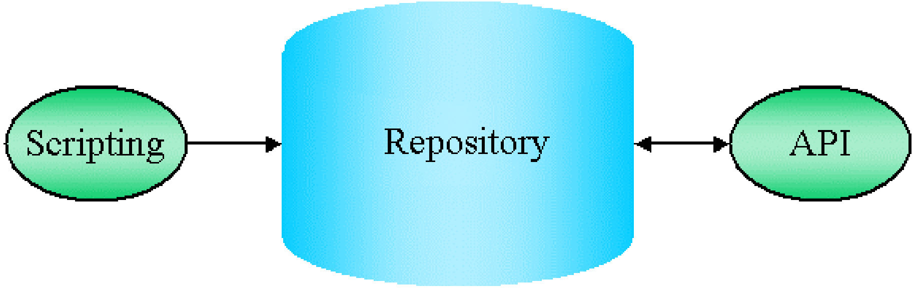


Figure 20 Scripting and APIs in Warehouse Builder

To this end, Warehouse Builder contains a published API and a scripting language on top of the repository[5].

PUBLIC APIS

Communicating with the repository is done via Java APIs, which are a set of published routines that you can call to perform actions on the repository. This is essentially an interface you use when you create or manipulate objects with the client application. Using the public APIs you can either create another front end or directly call these routines.

SCRIPTING

Based on the public APIs a scripting language is available. The scripting language shields the complexity of these APIs, as a user interface would do. A set of 4GL commands is available to call the API function via more understandable commands. The Warehouse Builder scripting is to the Warehouse Builder repository as SQL is to the Oracle database.

SUMMARY

By adding scripting and public APIs to such a rich tool as Warehouse Builder, Oracle arms you with a powerful component of a larger number of applications. You can embed Warehouse Builder in a custom application and roll out the application without using the Warehouse Builder user interface. This is a very lucrative alternative to Internet Service Vendors (ISVs).

[5] For more information on public APIs and Scripting please contact us via the OTN Forum on Warehouse Builder which is monitored by Product Management.

The Future of Warehouse Builder

One of the key points to consider when selecting a tool or vendor is the future outlook for both the vendor and the tool. Warehouse Builder is a strategic offering in Oracle's product portfolio. As we discussed in earlier sections, Warehouse Builder plays a key role in the business intelligence offering that provides the design and metadata components for the entire solution. Oracle as a vendor is a key player in the business intelligence market and covers the entire solution spectrum from database to end user query tools.

Future Aspects of Business Intelligence Design

One key area of focus for Warehouse Builder is the design for OLAP objects in the Oracle database. Since the release of Oracle 9*i*, both the relational and OLAP structures reside in the same database kernel.

Warehouse Builder contains metadata on dimensions and cubes that is presently utilized in the relational world. With the introduction of Oracle 9*i*, this metadata can now be used for two solution areas: both the relational and the OLAP components can make use of the same information.

The advantage is that you only have to design one system, but are free to deploy it in two very distinct styles, saving you not only development time and effort but also the time to learn both domains.

Oracle also constantly enhances the ETL processes to render them even more complete and compelling. As you can see, Warehouse Builder is becoming increasingly compelling to use in a business intelligence design environment.

Future Aspects of the Metadata Repository

One of the key features in today's world is security. Tightening security is one of the key enhancements planned for the upcoming versions of Warehouse Builder. Advanced security policies and user auditing will be incorporated in upcoming releases.

Another major feature is the further enhancement of repository extensibility. You will be able to add attributes to the Warehouse Builder repository, storing them side by side with the attributes that make up the standard repository. This means that you will be able to store more and more business metadata in the Warehouse Builder repository, making it suitable for a larger number of applications.

Versioning enhancements, or snapshotting, is another feature that will be added to the repository. Versioning exists as a feature in the current version, but future capability includes a mechanism to understand and view the differences between individual snapshots.

Future Aspects of the Runtime Environment

Remote access to systems can be interesting for system management groups. To accommodate it, a web-based Runtime Audit Viewer will replace the current java component . The useful and proven capabilities of the current reporting environment will be replicated, but now in a web format. The additional benefit is that this web-based interface can be added to corporate portals and allow all Warehouse Builder metadata to be viewed from one launch pad.

Another big improvement are the extended management capabilities of the runtime environment.

Making a complex environment more manageable is an important aspect of this addition, which will show the already deployed objects and even advise, based on design information, whether or not an object needs to be upgraded.

Finally, openness is will be addressed in future releases. Integrating with 3rd party scheduling and dependency tools will become easier in the near future. The new runtime environment will offer adapters to manage this interaction. External parties can use the specification for these adapters and create adapters to allow their tools receive information submitted by Warehouse Builder.

Future Aspects of Data Quality

Although the current solution solves the most critical part of data cleansing (name and address information), more data can be targeted with this solution in future. Essentially, Warehouse Builder will broaden the horizon to incorporate the full spectrum of data quality aspects.

As with the runtime environment, the data quality environment will become more open. It will allow you to choose from more parties' libraries using the Warehouse Builder Service Provider Interface.

Future Aspects of Life Cycle Management

As we have shown in preceding sections, life cycle management is already advanced in Warehouse Builder. However, it will become even more compelling in future releases. Warehouse Builder will provide you with "fix-it" capabilities throughout your repository. Based on the metadata you can allow Warehouse Builder to apply fixes by propagating changes to the repository from a certain starting point.

This functionality will reduce the amount of time required to address changes, making it dramatically easier to maintain and change your system.

Summary

Many more features are currently in development that will add great value to the already compelling offering.

It is clear that Warehouse Builder is a tool with an exciting future. Being part of one of the largest vendors in the software industry makes Warehouse Builder an offering to be noted for any data integration or business intelligence project.

SUMMARY

Oracle Warehouse Builder is an advanced ETL and Design tool for every business intelligence environment. It has a robust set of features for business intelligence applications.

Some key strengths and advantages are:

- Close integration with the leading data warehouse database, leveraging all the investments made in this database
- Close integration with the Oracle Business Intelligence toolset, allowing metadata to flow freely between these tools
- Based on Open Standards, such as ANSI SQL and OMG CWM standards
- An intuitive User Interface allowing a quick learning curve and fast creation of systems
- Extensive and easy to use metadata reports, available via a web browser
- Integrated Data Quality functionality that allows key data to be cleansed using a familiar user interface
- Extensive Life Cycle management capabilities, making changes to your environment easy to handle

The driving conclusion is that Warehouse Builder is a key player in the ETL tool market, and merits consideration as the tool for your data integration and business intelligence projects. It is a functionally rich but low-cost alternative to expensive niche tools and has a strong outlook for the future, from both a feature and a vendor sustainability perspective.

Integrated ETL and Modeling
Oracle 9i Warehouse Builder
July 2002

Author: Jean-Pierre Dijcks

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Warehouse Builder inquiries
http://otn.oracle.com/products/warehouse/content.html