

**Oracle Warehouse Builder 10g Release 2**

**Business Rule Driven Data Integration**

*January 2007*

**Note:**

This document is for informational purposes. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

# Oracle Warehouse Builder 10g Release 2

## Business Rule Driven Data Integration

### INTRODUCTION

We've heard it before, but can you really do enterprise class data integration driven by business rules?

In this paper you will see how you can create business rules in Warehouse Builder and then apply them to real world ETL problems. You will see how changing a business rule will change the ETL logic with a few simple clicks. You will see how you can create error handlers completely driven by business rules.

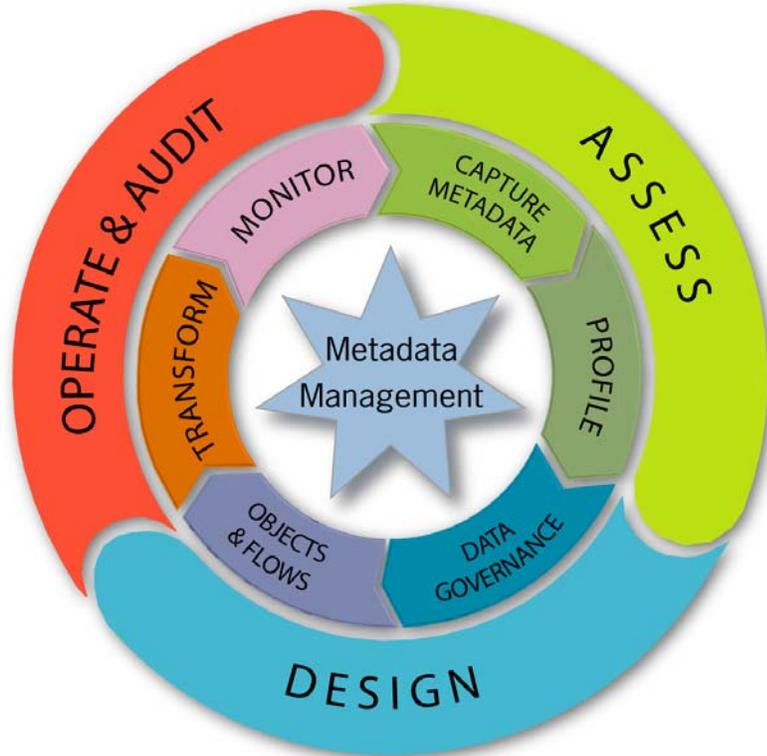
As an extension of the rules in ETL we will show you how to validate your rules on your actual source data, to ensure the rules are in fact correct and supported by your data.

This paper will show you how business rules can have a real impact on a developer's life. Maintenance of ETL logic becomes simpler; error routines can be modified in minutes, saving you valuable time and effort.

So, can you drive an enterprise class data integration solution with business rules? With Warehouse Builder 10gR2 the answer is a definite yes.

## BUSINESS RULES AND WAREHOUSE BUILDER

With the Warehouse Builder 10g Release 2 release, Oracle offers the use of business rules in the wider context of the data integration life cycle. Data Governance, a distinct part in the design phase of data integration encompasses Data Rules.



**Figure 1 Data Integration Life Cycle**

In data integration, traditionally design of data flows is a labor-intensive job and data governance is a newly created step in most of the data integration processes. Data governance is probably conceived as additional work to the rest of the cycle, but in reality it might help make the design of both storage objects and data flows simpler and more efficient.

This simplification and higher efficiency is certainly something that, with introducing business rules into Warehouse Builder, is achieved. To understand more about how this is achieved we need to discuss how business rules are present in Warehouse Builder.

### Data Rules

When opening Warehouse Builder you will see that the product names business rules data rules. This is done to get a more accurate description of what the rules describe. So not any and all business rules – for example business processes like an order process with order statuses – can be modeled in Warehouse Builder. Data

rules aim to model rules around data such as verification and validation of data and compliance of data to the set rules. These data rules are stand-alone objects in Warehouse Builder and can be shared across projects within a repository or can be confined to a single project.

### **Flexibility**

The main important objective of data rules is to allow you flexibility to write business logic into data integration logic. By documenting the data rules into separate objects, diverse sets of people can create and maintain the logic. The changes that need to be made are then at the metadata level, rather than in code embedded deep in the data integration programs.

Changing a data rule changes the data integration logic without requiring the developers to directly change their code.

### **Language of Data Rules**

Even if we allow data rules to be a individual component, extracting it from the main coding logic, there still needs to be a language to capture the semantics in.

In Warehouse Builder the data rules are implemented in two main “languages”, one is SQL and PL/SQL and the other is called regular expressions. SQL is mostly used to determine data or filters; PL/SQL is used to do more complex operations, which might require inter-row work.

Regular expressions are used for patterns and for string matching, where the content of a field is probed for a certain pattern or where certain patterns need to be enforced.

## SOLVING DATA INTEGRATION PROBLEMS WITH BUSINESS RULES

This example shows you how to use data rules to generate error handling based on data rules. This way the design contains the error handling rules and directly progresses these rules into the ETL design. A change in the data rules directly changes the ETL error handling, without costly redesign.

### Creating a Data Rule

To create a regular (e.g. not a shared one) data rule in Warehouse Builder you simply go into any project and create data rule. In this case we create them in a new folder to organize the work.

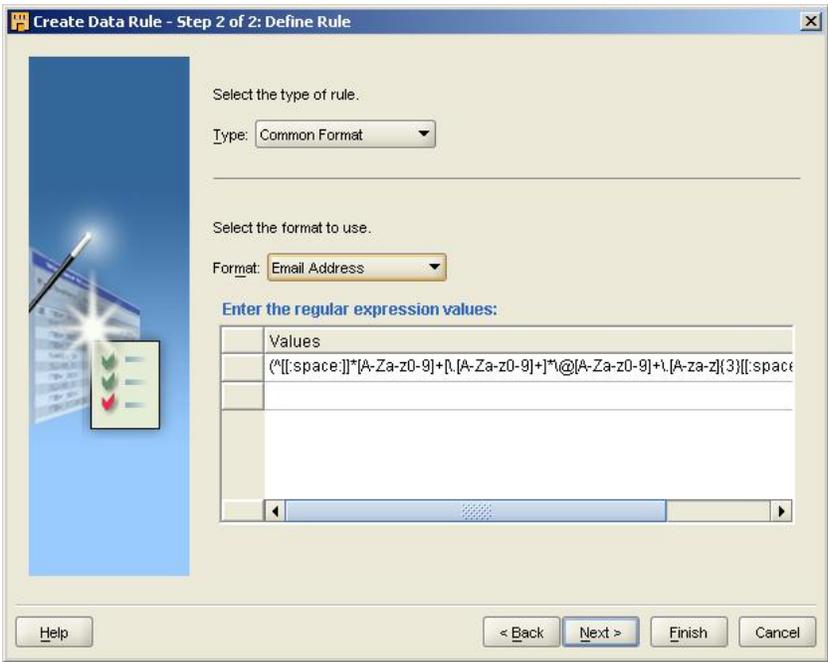


**Figure 2 Creating a data rule folder**

Within these folders, which are a purely organizational construct, you create your data rules. In this particular example the data rule uses regular expressions to verify the pattern of a string.

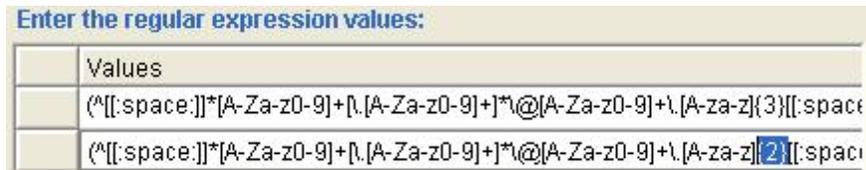
For that example, create a new data rule that will verify a specific pattern in a field. It will not be a simple string search for an @ sign and then hope that it is an email address, but instead it will be a complete pattern description.

To start this rule, start with the built-in data rule for verifying email addresses. The pattern for that rule looks like the one shown in Figure 3. As you can see this pattern is described in a regular expression, and the type of rule is a common format rule.



**Figure 3 A built-in data rule for email address patterns**

The built-in rule has some limitations, first of all it verifies whether the last bit of the string is a 3 character long string. This is common in the US where most domains are .com, .org, .gov or something along these lines. For European domains that might not be correct. Many countries add their own country to domains to create things like oracle.nl. For this reason add a second pattern that will allow testing for rules like this one.



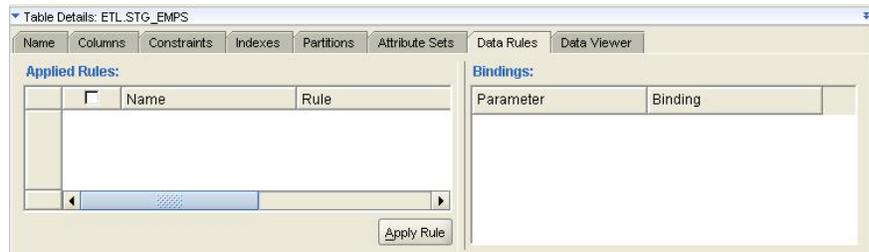
**Figure 4 A second pattern to detect 2 character strings**

The pattern rules allow you to simply add rules and these are verified in a logical OR. So the first rule or the second rule or the third gives a match. The outcome of the rules is either true or false. In other words the pattern either matches one of the rules or does not match any of the rules.

### Using Data Rules

The data rule is not directly used in the ETL process as a standalone component, it is used as a part of the table that is used as a record source. This way the logic is tied as much to the schema as to the actual data flow.

To add a rule, open the table editor on the table that will hold the data that is being verified (see Figure 5).



**Figure 5 Apply a data rule in the data object editor**

The important step here is to match the table column to the parameter of the rule, this is done in the Bindings part of the wizard that guides you through applying the data rule. The end result shows that the parameter is bound and that the rule is added to the desired table.



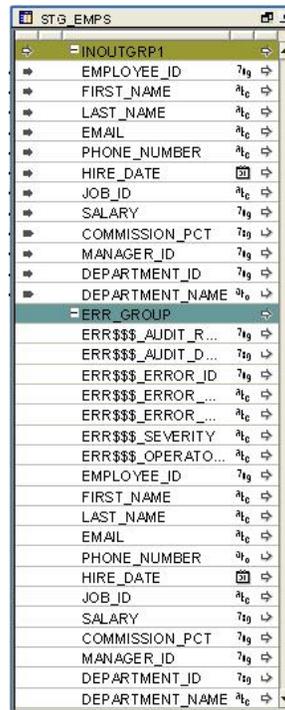
**Figure 6 An applied rule on a table**

The big benefit of adding the data rule to the actual table rather than to a data flow, is that in any situation where this table is used the data rule is automatically applied. The only thing required from the data flow designer is to decide how (if at all) to use the rule.

### Data Rules and Data Flows

When you add a table to a data flow that has one or more data rules applied to it, the operator in the mapping editor has some additional features. The first is that the table operator shows an error group, as shown in Figure 7.

This error group will hold data that does not comply with the data rules, and is implemented in a separate table. This table is generated by Warehouse Builder and the binding between the error group and that error table is done automatically.



**Figure 7 Mapping table operator with error group**

To control the usages of the data rules applied and the error group and table, the operator has properties per data rule.

<b>STG_EMPS</b>	
Target Load Order	
Data Rules	{ETL_EMAIL_VERIFICATION}
0	ETL_EMAIL_VERIFICATION
Data Rule Name	EMAIL_VERIFICATION
Rule Action	IGNORE
Data Rule Type	IGNORE
Key	MOVE TO ERROR
Bound Name	REPORT
Loading Type	INSERT
Primary Source	No

**Figure 8 Mapping properties per data rule**

For each data rule and its usage on the table the mapping allows a specific usage. For the data rule in Figure 8 you can choose to either ignore the rule, which essentially does nothing to the data flow.

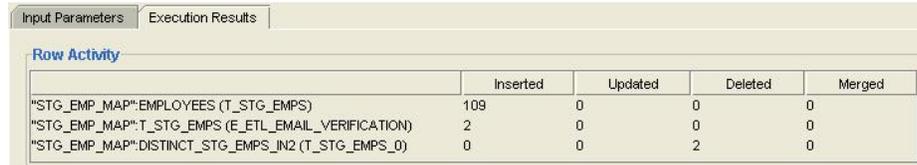
If you choose to MOVE TO ERROR (as is being done here), all rows in the table are verified against the rule and then removed from the table and moved to the error table. This means that the bad data is removed from the data flow and will not go any further than the error table. From the error table the user can transform the data and if desired route it back into the rest of the data flow.

If you choose to REPORT the faulty row, the error table will hold the row identifier and the reason for reporting. It will however not remove the row from

the data flow and the faulty rows will flow without transformation through the remainder of the flow.

### Understanding Results

A mapping with a data rule in it, as shown above, gives a specific audit pattern in the Control Center Manager. When choosing the MOVE TO ERROR action for the data rule, Warehouse Builder actually deletes the rows. This is logged in the mapping run audit information and looks like this:



	Inserted	Updated	Deleted	Merged
"STG_EMP_MAP":EMPLOYEES (T_STG_EMPS)	109	0	0	0
"STG_EMP_MAP":T_STG_EMPS (E_ETL_EMAIL_VERIFICATION)	2	0	0	0
"STG_EMP_MAP":DISTINCT_STG_EMPS_IN2 (T_STG_EMPS_0)	0	0	2	0

**Figure 9 Audit results with a data rule**

As you can see in Figure 9 two rows are deleted from the data flow. This is shown in the last row of the audit details.

### Changing the Data Flow

With the data rule a separate entity but tightly coupled to the data flow, it is very simple to change the rules as to what email addresses are valid. Either one of the current regular expressions is changed, or you add another regular expression. The mapping will then (after redeploying) verify the data with the new logic and allow the ETL design to not change any of the core processing logic.

## **CONCLUSION**

With the right toolset and the right approach to data integration, developers and project managers can quickly produce integrated data of high quality. To stay flexible and avoid a lot of hard coding of logic, using metadata and business rules makes perfect sense.

Warehouse Builder 10g Release 2, with its extensive business rule logic and embedding of this logic into the data flows allows for complex logic to be distilled and maintained outside the mainstream data integration code base, allowing more people access to the logic. This flexibility brings business rules and data integration together in an enterprise class data integration tool.



Oracle Warehouse Builder 10gR2 – Business Rule driven Data Integration

January 2007

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](http://www.oracle.com)

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.