

September 2009

Welcome to the September 2009 edition of IBM InfoSphere Information Server Developer's Notebook. This month we answer the question;

In prior editions of this document, you discussed data discovery using the Information Analyzer (IA) component to Information Server; you covered the 5 (then) core functions of IA. Can you go farther, and show some of the power user type functions to IA?

Excellent question! This is the second in a two part series on Information Analyzer (IA); IA features we had not previously covered, and features that are new to IA release 8.1.1. In this edition of IISDN, we give focus solely to Data Rules and related topics.

In this edition of IISDN, we detail how to create Data Rules, Rule Sets, and related topics. Also, we use this edition of IISDN to perform a bit of review on the topic of IA.

Software versions

All of these solutions were *developed and tested* on (IBM) InfoSphere Information Server (IIS) version 8.1.1, using the Microsoft Windows XP/SP3 platform to support IIS client programs, and a RedHat Linux Advanced Server 5 (RHAS 5) 32 bit SMP server (Linux kernel version 2.6.9-67.EL-smp) to support the IIS server side components.

IBM InfoSphere Information Server allows for a single, consistent, and accurate view of data across the full width of the corporate enterprise, be it relational or non-relational, staged or live data. As a reminder, the IBM InfoSphere Information Server product contains the following major components;

WebSphere Business Glossary Anywhere™, WebSphere Information Analyzer™, WebSphere Information Services Director™, WebSphere DataStage™, WebSphere QualityStage™, WebSphere Metadata Server and Metabridges™, WebSphere Metadata Workbench™, InfoSphere Federation Server™, Classic Federation™, Event Publisher™, Replication Server™, InfoSphere Data Architect™, DataMirror Transformation Server™, and others.

Obviously, IBM InfoSphere Information Server is a large and capable product, addressing many strategic needs across the enterprise, and supporting different roles and responsibilities.

32.1 Terms and core concepts

As mentioned above, the February/2009 and March/2009 editions of IBM InfoSphere Information Server Developers Notebook (IISDN) detailed the (then 5) core areas of functionality inside the Information Analyzer (IA) component to IBM InfoSphere Information Server (IIS). Those two editions of IISDN serve as a primer to IA.

The most recent edition of IISDN (August/2009) covered all of those IA features we previously left uncovered, including some from version 8.1.1 of IA; a part 1 of 2, if you will. And then this edition of IISDN is part 2 of 2, and covers Data Rules and related topics in IA version 8.1.1.

Data Rules are such a capable and large new topic, it gets its own edition of IISDN. (And with the addition of Data Rules, we would now state that Information Analyzer has 6 main areas of functionality.)

Sample data and tables

In the February/2009 edition of this document, we detailed a number of sample tables and data we have used throughout this discussion on Information Analyzer. In this edition of this document, we will use 1 primary table, and 1 table for lookup (data validation).

The Order Header table, our primary table in these examples, is displayed in Figure 32-1. The Order Header table, Customer Number column will also be joined to the Customer (not pictured) table for a dynamic validation test.

Select Data Sources to Work With							
Column Analysis							
Name	Sequence	Column Status	Column Analysis Statu.	Data Type	Length	Precision	Scale
▼ RHHOST.GRID							
▶ FlatFile_IA_DS							
▼ ids_stores_ia_ds							
▶ base							
▶ group_1							
▶ group_2							
▶ group_3							
▶ group_delta							
▼ group_quality							
▼ order_header_q			100.00 %				
◊ order_num	1	Analyzed	Analyzed	INT32	4	4	--
◊ order_date	2	Analyzed	Analyzed	DATE	10	10	--
◊ customer_num	3	Analyzed	Analyzed	INT32	4	4	--
◊ ship_instruct.	4	Analyzed	Analyzed	STRING	40	--	--
◊ backlog	5	Analyzed	Analyzed	STRING	1	--	--
◊ po_num	6	Analyzed	Analyzed	STRING	10	--	--
◊ ship_date	7	Analyzed	Analyzed	DATE	10	10	--
◊ ship_weight	8	Analyzed	Analyzed	DECIMAL	8	8	2
◊ ship_charge	9	Analyzed	Analyzed	DECIMAL	6	6	2
◊ paid_date	10	Analyzed	Analyzed	DATE	10	10	--

Figure 32-1 Order Header table used in these examples.

The Order Header table has specific data formatting issues with its PO Num column, as displayed in Figure 32-2.

Overview

Frequency Distribution

Data Class

Properties

Domain & Completeness

Format

View the frequency of formats in the column, view the distinct values that are associated with the formats for the column, and mark a format invalid.

Number of Formats:

8

Conforming Count:

23

General Format

General Form	Count	Percent	Status	Do
AA999999	1	4.35	Conform	
AA9999	2	8.7	Conform	
AA999	2	8.7	Conform	
A99999	7	30.43	Conform	
A9999	3	13.04	Conform	
999A	1	4.35	Conform	
999999	2	8.7	Conform	
9999	5	21.74	Conform	

Distinct Values

Distinct Value	Count	Percent	
B77836	1	4.3478	
B77890	1	4.3478	
B77897	1	4.3478	
B77930	1	4.3478	
Q13557	1	4.3478	
S22942	1	4.3478	
Z55709	1	4.3478	

Format Violations

Distinct Value	General Form

Domain Values Status

Drill Down

Figure 32-2 Open Column Analysis -> View Details -> Format TAB.

Figure 32-2 is the result of a, (highlight the PO Num column) -> Open Column Analysis -> View Details -> Format TAB.

Other tests we will detail below include:

- Checking for the presence of data in a column, and its value.
- Checking column format.
- Checking the total volume of valid data in a given data set (a given data load).

Review of IA 8.1.1 Menus

Version 8.1.1 of Information Analyzer adds a number of new menu items to the IBM InfoSphere Information Server (IIS) Console. We'll use Figure 32-3, below, to overview all of Information Analyzer's menus.

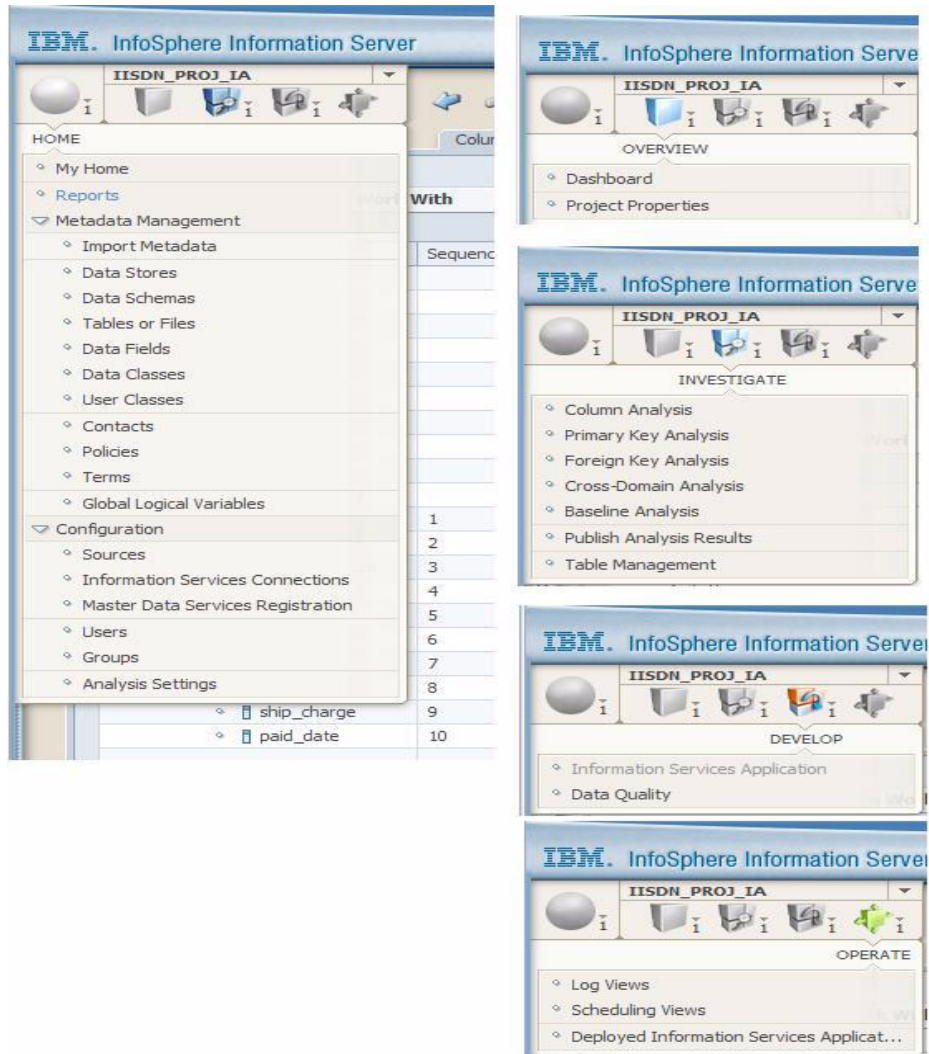


Figure 32-3 All IA Sub-Menus from the Menu Bar.

A code review related to Figure 32-3 includes;

- We will discuss these menu items in chronological sequence; meaning, the order in which you would normally use them.

5 Menus are displayed above; Home, Overview, Investigate, DEvelop, and Operate.

- We use 3 or more sub-menus from the Home menu.

- Home -> Configuration -> Analysis Settings

This menu item leads to 3 TABs where we specify the run time connectivity to the DataStage Engine, the IADB database, and certain system wide analysis default settings.

Note: The Information Analyzer (IA) component to IBM InfoSphere Information Server (IIS) runs on the IIS Parallel Framework as a series of (DataStage) Jobs.

- Home -> Configuration -> Sources

This menu item is where we define system wide data sources; in effect, the configuration data for a series of ODBC drivers.

Each of these major entries is referred to as a Data Store.

- Home -> Metadata Management -> Import Metadata

This menu item is where we call to import the actual schema, table and column names, column data types and more.

Again this is a system wide collection of data.

Note: On a lower menu item, we add the object tables and other objects to a given Information Analyzer (IA) Project.

- Home -> Metadata Management -> Global Logical Variables

This menu item is new with release 8.1.1 of Information Analyzer. This menu item allows us to create, etcetera. Global Logical Variables are defined below.

- We use one sub-menu from the Overview sub-menu.

This menu item is where we add the system wide tables and columns to a given Information Analyzer Project.

- We use all sub-menus from the Investigate sub-menu.

These 7 menu items used to form the bulk of Information Analyzer functionality; now that functionality is shared with the menu item that follows.

This menu item could still be characterized as data discovery.

- On the Develop sub-menu, we use the Data Quality menu item.

The Data Quality menu item is where all Data Rules, Rule Sets, and other related objects are housed, maintained, other.

- And lastly, the Operate sub-menu is where we access logging function to Information Analyzer and more.

How to use the log with Information Analyzer is detailed in the prior edition of this document, August/2009.

IA 8.1.1 Data Rules, object hierarchy

As with any (new) environment, Information Analyzer Data Rules introduce a number of new terms. These include;

- Global Logical Variable

An Information Analyzer (IA) Global Logical Variable is like a DataStage Job Parameter; it serves as a macro, a string, that is used (expanded) later.

The advantage to using an IA Global Logical Variable is that you can set its value one time, reference it throughout IA, and if you need to change its value, do so in only one place.

A Global Logical Variable is defined system wide, and can be shared across numerous Information Analyzer Projects.

- Data Rule Definition

A Data Rule Definition and all of the terms that follow exist inside a standard Information Analyzer Project, right beside all of the data discovery you have already been performing. And, you benefit from all of the analysis work you performed earlier, when you start working with Data Rule Definitions.

An example Data Rule Definition would resemble,

(column) > 17

A Data Rule Definition may be bound to an explicit data source column, or it may remain logical; not bound to any specific source column, so that the given Data Rule Definition may be used over and over.

Data Rule Definitions form the bulk of an *Information Analyzer Data Quality Application*. (There is no Information Analyzer Data Quality Application object per se; this term refers to the functionality of an IA Project that contains and executes Data Rule Definitions, etcetera.)

Note: You can test a Data Rule Definition interactively, live on your screen. Eventually, the Data Rule Definition becomes instantiated as a Data Rule, and enters the run time environment proper, as an object that can be run from the command line or scheduled Data Quality Jobs.

- Data Rule

There is little distinction at first between a Data Rule Definition and a Data Rule. At some point you finish testing and defining a given Data Rule Definition, and it enters the run time environment as a Data Rule, or as a member of a Rule Set.

- Rule Set

Where a Data Rule Definition generally evaluates a single source data column, these definitions can be combined into more powerful grouping of (rules) called Rule Sets.

And as a larger object, a Rule Set adds further testing criteria that can be defined/applied.

- Metric

A Metric is an object that can be used to examine the Results of a Data Rule or Rule Set Job run. For example, you can measure and report if more than 5% of the data records processed met or failed a given set of criteria.

- (Base lining), and more

Much like the Data Discovery functionality within this and earlier versions of Information Analyzer, you can set a variety of base lines measurements, evaluate trends over time, etcetera.

32.2 Complete the following examples

1. Log on to the IBM InfoSphere Information Server (IIS) Console, and Connect to an existing Information Analyzer (IA) Project.

Its better if you have followed the examples throughout this series of documents on IA, but any existing IA Project with 2 tables, where you are familiar with the data will suffice.

Minimally, you need to have completed Column Analysis on your primary table used in these examples.

2. Create a Global Logical Variable.

We are going to create a Global Logical Variable entitled, `glv_MinimumShipWeight`, and use the definition later when evaluating the quality of our inbound data.

- a. From the Menu Bar, Select, Home -> MetaData Management -> Global Logical Variables.
- b. In the Tasks panel, click, Create New.
- c. Manage your display to equal that as shown in Figure 32-4.

You will need to use the Set Bindings button at some point.

Note: We don't detail use of the Attachment TAB anywhere in this edition of IISDN, but its use is highly powerful. There you can associate Global Logical Variables, Rule Set Definitions, etcetera with Terms from the business Glossary and more.

The screenshot shows the 'Create New Global Logical Variable' dialog box. The 'Name' field contains 'glv_MinimumShipWeight'. The 'Data Type' is set to 'Numeric'. The 'Binding' field contains 'S.0'. The 'Short Description' and 'Long Description' fields are empty. The 'Example' field is empty. The 'Created By' field is empty. The 'Created On' field is empty. The 'Last Modified' field is empty. The 'Data Steward' field contains 'iisadmin'. The 'Status' field is set to 'Candidate'. The 'Literal' field contains 'S.0'. A 'Set as Binding' button is located next to the 'Literal' field.

Figure 32-4 Creating a Global Logical Variable.

- d. When your display equals Figure 32-4, Click, Save -> Save and Close.
3. Create a number of Folders to contains all that you are about to create.

We are about to create a number of distinct objects, and wish to keep them organized.

- a. From the Menu Bar, Select, Develop -> Data Quality -> (Task Panel) -> Manage Folders -> Create New.

Created new Folders entitled; drd_ColumnLevel, ColumnCheckConstraints, ColumnFormatConstraints, ColumnLookups, and rs_TableLevel.

Example as shown in Figure 32-5.

Note: Some of these Folders are attached to (Root), while other folders are nested.

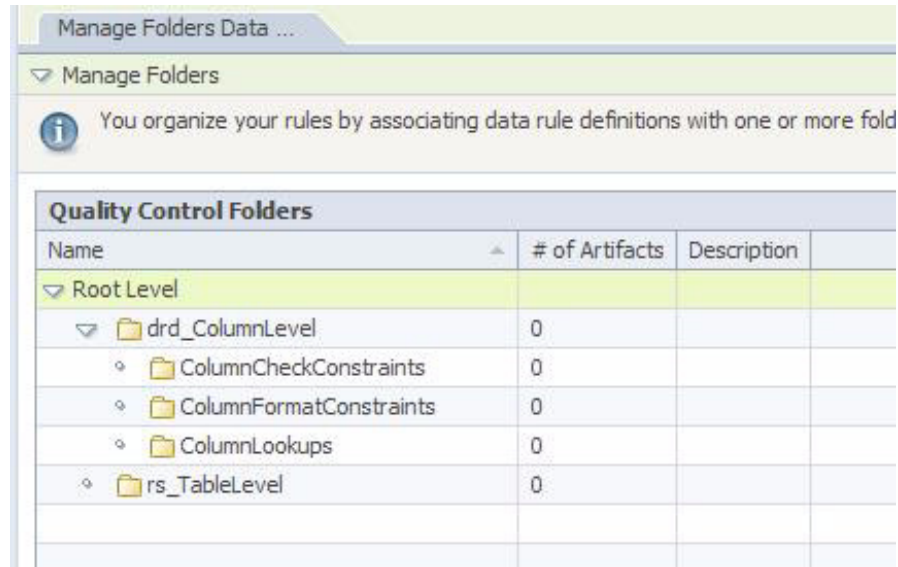


Figure 32-5 Managing Folder, common practice

- b. You are done when your display equals that as shown in Figure 32-5.
Click Close.
4. Create a New Data Rule Definition
 - a. From the Tasks panel, Select, New Data Rule Definition.
 - b. Under the Overview TAB -> Overview view, name this Data Rule Definition, IfColumnIsNotNULL.
 - c. Under the Overview TAB -> Folders view, Click Add, and complete the steps to add this object to the drd_ColumnLevel -> ColumnCheckConstraints Folder.
 - d. Under the Rule Logic TAB, manage your display to equal that as displayed in Figure 32-6.

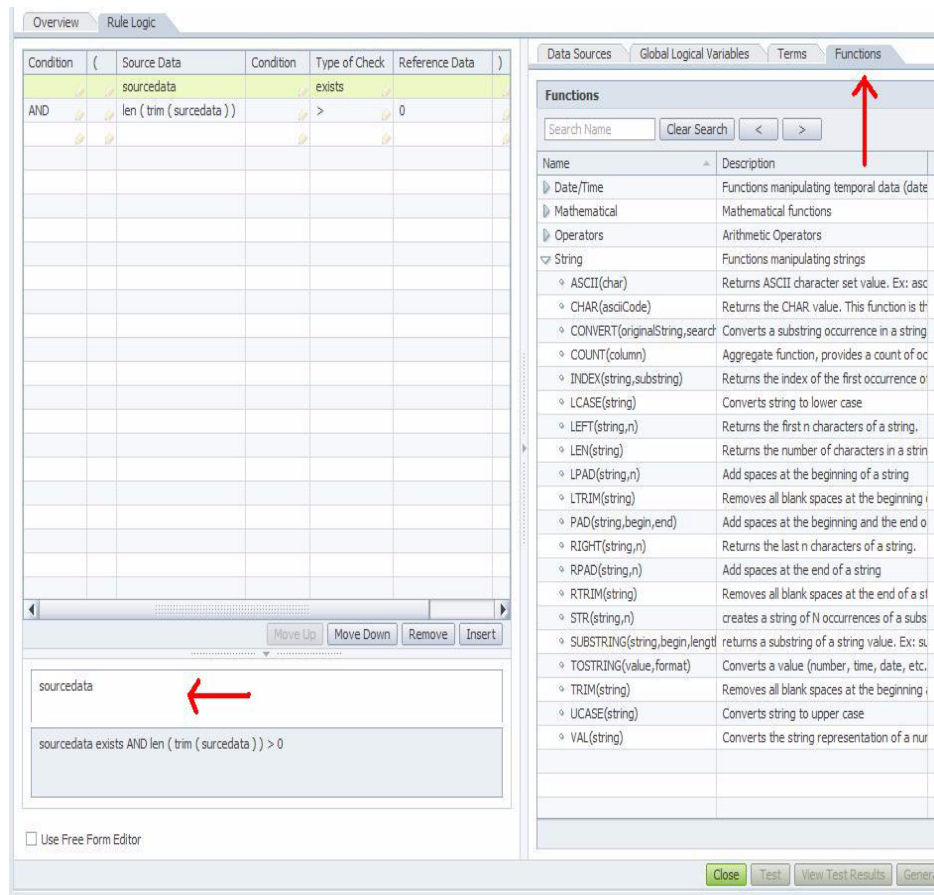


Figure 32-6 Specifying Data Rule Definition Logic.

Related to Figure 32-6, the following is offered;

- Two red arrows are painted in the display in Figure 32-6. The bottom left area showcases an area where you assemble Rule Logic.

You can type in this area.

In fact, this is your primary work area on this screen.

- “sourcedata” is a Data Rule Definition keyword. It is a place holder for a source data column name to be supplied later, (bound to later).

Using sourcedata is what will make this Data Rule Definition more powerful; able to be used for numerous actual source data columns later.

You have to type the word, sourcedata. There is no button or picker for it.

- The upper right red arrow in Figure 32-6 showcases the Function TAB. this is where we discovered the built in functions, trim() and ucase().

Note: We are building a Data Rule Definition to check for NULL or empty character strings. that is where we use the Exists operator, *and* check the length of this column.

Strings are tricky that way.

- e. Click the Validate button when your display matches that as shown in Figure 32-6.
- f. Click Save -> Save (do not Save and Close), and Click, Test.

Generally, running a Test of a Data Rule Definitions has 4 Steps, 4 TABs that you have to complete.

At this point, the system will remind you that you used the sourcedata as your column binding (a logical binding), and you now need to supply a real data source column for the duration of this testing.

Manage your display to equal that as shown in Figure 32-7.

And Click, Set as Bindings.

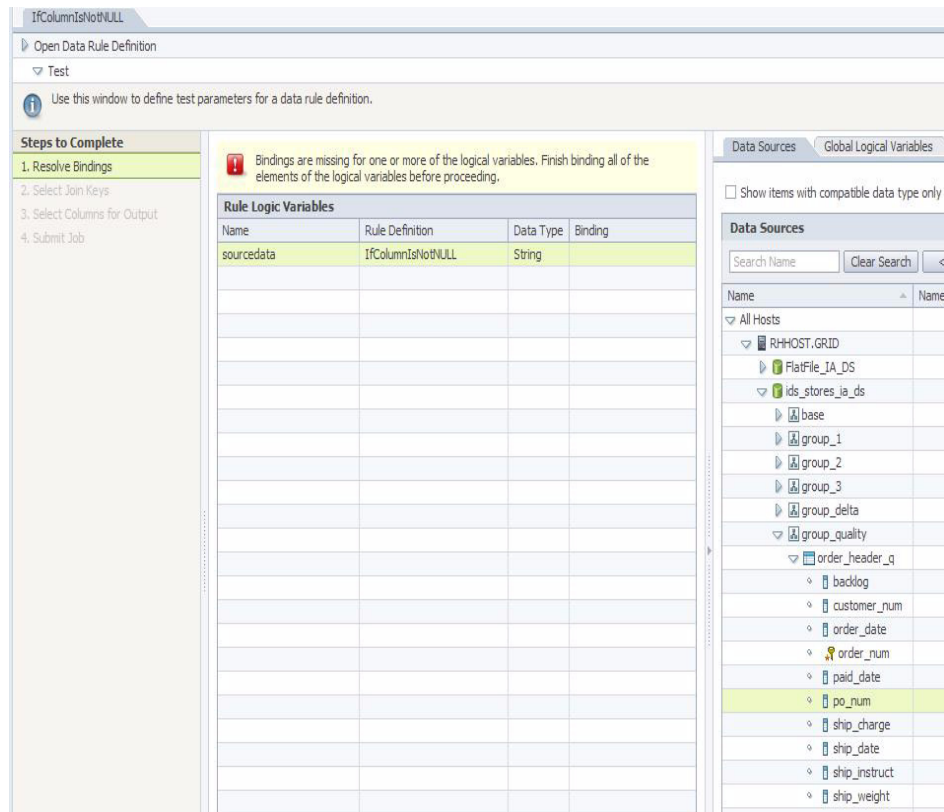


Figure 32-7 Specifying settings for Test, Step 1 of 4.

A completed Binding appears in Figure 32-8.

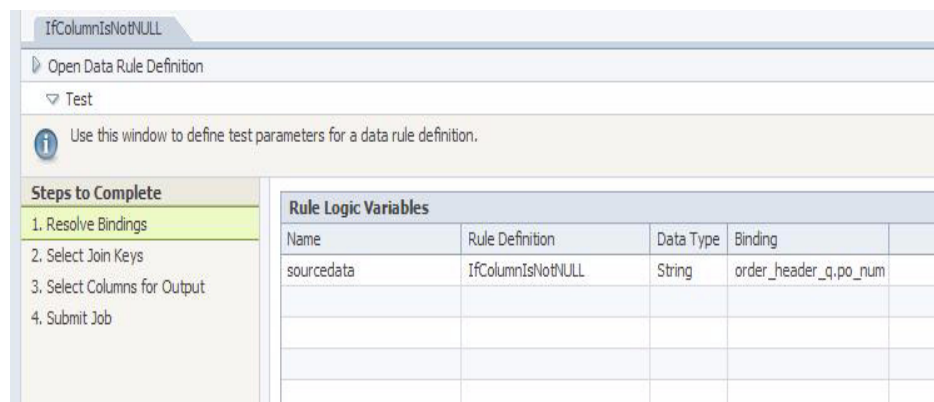


Figure 32-8 Completed Binding to source data column.

- g. While there are 4 Steps (TABs) to Testing a Data Rule Definitions, this is a simple Data Rule Definition and we can skip Step 2.

Click Next twice, or use the panel on the left (Steps to Complete) to skip to Step 3.

- h. Under Step 3 (Select Columns to Output), manage your display to equal that as shown in Figure 32-9.

The screenshot shows the 'Selected Columns' panel on the left and the 'Rule Logic Variables' panel on the right. The 'Selected Columns' panel has a dropdown menu for 'Output Records' set to 'All Records'. Below it is a table with columns: Output Name, Source, Binding, and Alias. The 'Rule Logic Variables' panel has a table with columns: Name, Data Type, and Binding.

Output Name	Source	Binding	Alias
sourcedata	sourcedata	order_header_q.po_num	

Name	Data Type	Binding
sourcedata	String	order_header_q.po_num

Figure 32-9 Specifying output columns and conditions.

- i. Click Next, then Finish.

This Data Rule Definition Test runs as a (DataStage component) Job in the background. when this Job is completed, you can Click, View Test Results.

Example as shown in Figure 32-10.

Note: The very first time you Test a Data Rule Definition can take a while; be patient.

IfColumnIsNotNull	
Open Data Rule Definition	
View Test Results	
The Overview tab shows summary information about the data rule inclu	
Overview	Result
Summary	
	Run
Rule Definition	
Rule Definition	IfColumnIsNotNull
Statistics	
Table 1	group_quality.order_header_q
Total Records	23
# Met	22
% Met	95.6522 %
# Not Met	1
% Not Met	4.3478 %
Job Details	
Start	1/2/2010 9:54:37 PM
End	1/2/2010 9:54:57 PM
Elapsed	0 minutes, 20 seconds
Sample Details	
Data Sample	No
Sample Size	
Sample Type	

Figure 32-10 23 Rows tested, 1 Row was NULL or blank.

- j. Click Save Data Rule (we called ours, IfColumnIsNotNull_rule).
And Click Close, and Click Close again.

Note: The above details the basics towards making and testing a Data Rule Definition.

5. Make 4 additional Data Rule Definitions. At this point, we will only detail what is entirely new.
 - a. Add a rule using the Global Variable Definition you created earlier to check the Ship Weight column.
 - i. Name this first Data Rule Definition, IfIsValidShipWeight.

- ii. Add it under the ColumnCheckConstraints Folder.
- iii. The Rules TAB to this Data Rule Definition is displayed in Figure 32-11.

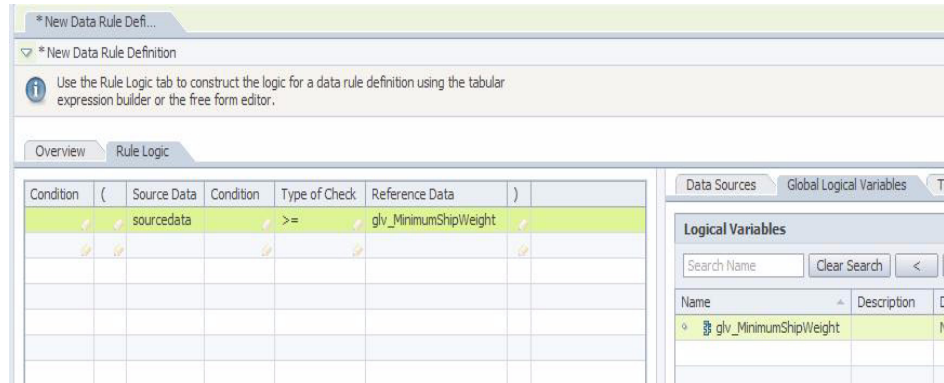


Figure 32-11 Data Rule Definition referencing Global Variable Definition.

- iv. You are done with this Data Rule Definition when it correctly evaluates Ship Weight.
- b. Add a rule to check is the Back Log column equals the value; "N", "n", "Y", "y".
 - i. Name this Data Rule Definition IfValuels_YN.
 - ii. Add it under the ColumnCheckConstraints Folder.
 - iii. The Rules TAB to the Data Rule Definition is displayed in Figure 32-12.

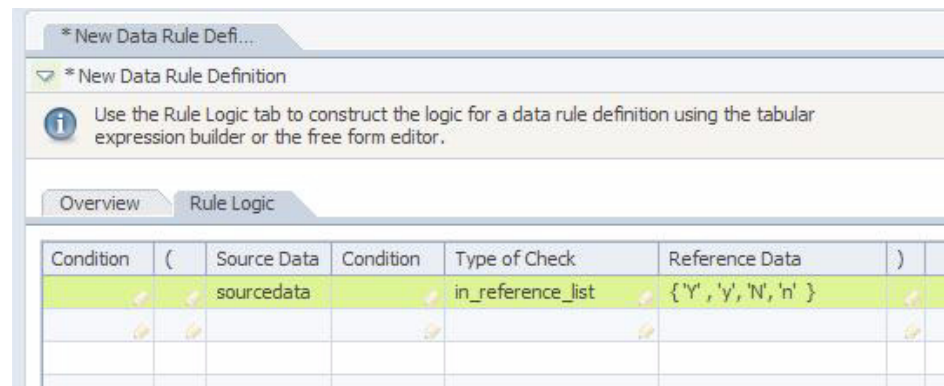


Figure 32-12 Data Rule Definition using Reference List.

- iv. You are done with this Data Rule Definition when it correctly evaluates Back Log.

- c. Add a rule to check if the PO Num is formatted correctly.

Information Analyzer Data Rule Definitions can use the extremely powerful and standard RegEx (Regular Expressions Language) to validate data domain and formatting.

We wish to validate PO Number formatting as they appeared in Figure 32-2, above.

Note: RegEx is an industry standard.

A good url to develop and check RegEx expressions is located at,

http://www.codehouse.com/webmaster_tools/regex/

And a good primer on RegEx is located at,

<http://www.addedbytes.com/download/regular-expressions-cheat-sheet-v2/pdf/>

- i. Name this Data Rule Definition IsIsValidPONum.
- ii. Add it under the ColumnFormatConstraints Folder.
- iii. The Rules TAB to the Data Rule Definition is displayed in Figure 32-13.

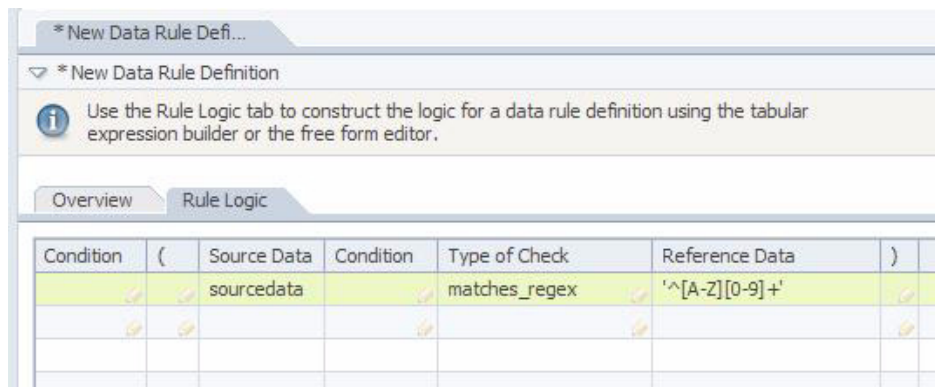


Figure 32-13 Data Rule Definition using RegEx.

The RegEx expression above validates PO Numbers in the format, A99, as shown in the example below, Figure 32-14.

The screenshot shows a web-based 'Regular Expression Calculator'. It has several sections: 'Flags' with checkboxes for 'global', 'insensitive', and 'multiline'; 'Operation' with radio buttons for 'test' (selected), 'match', 'search', and 'replace'; 'Regular Expression:' with an example '\w' and the input '^ [A-Z] [0-9]+'; 'Source Text:' with the input 'A1234'; buttons for 'Evaluate', 'Clear Output', and a checked 'Auto clear' checkbox; and a 'Result:' section showing the output '/^[A-Z][0-9]+/.test(A1234) = true;'. The interface is light gray with red icons for each section header.

Figure 32-14 From, http://www.codehouse.com/webmaster_tools/regex/

- iv. You are done with this Data Rule Definition when it correctly evaluates PO Num.
- d. This last Data Rule Definition accesses an additional database table, to determine if a given value is found to be valid (contained) there. This Data Rule Definition also requires us to specify additional properties.
 - i. Name this Data Rule Definition IfIsValidCustomer.
 - ii. Add it under the ColumnLookups Folder.
 - iii. The Rules TAB to the Data Rule Definition is displayed in Figure 32-15.

The screenshot displays the 'Rule Logic' tab in the IBM InfoSphere Information Server Developer's Notebook. The main area contains a table for defining conditions, with columns for 'Condition', 'Source Data', 'Condition', 'Type of Check', and 'Reference Data'. The first row is highlighted in yellow and contains the following data:

Condition	(Source Data	Condition	Type of Check	Reference Data)
		sourcedata		in_reference_column	customer_num	

To the right of the table is a 'Data Sources' panel. It has a search bar and a list of data sources. The list is expanded to show the following sources:

- ▼ All Hosts
 - ▼ RHHOST.GRID
 - FlatFile_IA_DS
 - ▼ ids_stores_ia_ds
 - base
 - ▼ group_1
 - catalog_1
 - cust_calls_1
 - customer_1
 - address1
 - address2
 - city
 - company
 - customer_num
 - fname

Figure 32-15 Data Rule Definition using In Reference Column, a Lookup.

- iv. This is your first Binding, with 2 or more columns. See Figure 32-16.

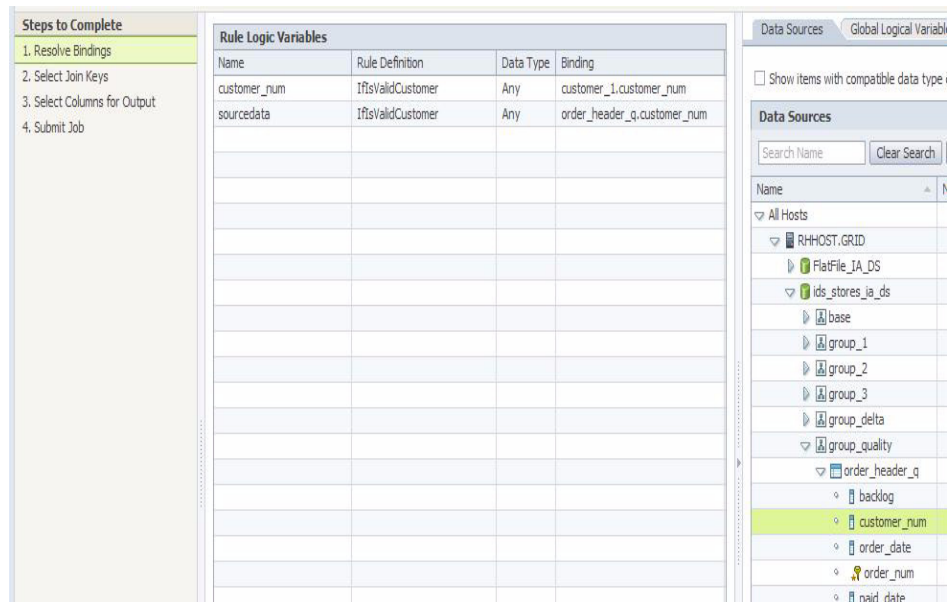


Figure 32-16 Biding with multiple columns.

- v. You are done with this Data Rule Definition when it correctly evaluates Customer Num.

At this point we have created 5 new Data Rule Definitions and tested same, including;

- 3 Column level check constraints of various complexity.
- A column format check constraint.
- And a lookup into another table for data validation.

Now its time to package these rules together into one unit, that can be executed collectively on a single or set of tables. This is accomplished by creating a Rule Set Definition.

6. Create a new Rule Set Definition.
 - a. From the Tasks panel, Select, New Rule Set definition.
 - b. Name this rule Set Definition, OrderHeaderTable_CheckBeforeLoad.
 - c. And specify a, Validity Benchmark, Monitor Records.

We called to monitor for load volumes greater than 500 rows.

Example as shown in Figure 32-17.

*** New Rule Set Definition**

Use this workspace to define constraints and conditions of multiple data rules that are to be applied to a physical data source and verify the quality of the data values in that data source. Use the Overview tab to define basic information about a rule set; to associate a rule set definition with folders, terms, policies, and contacts; and to test the rule set definition. You can also view the components included in the rule set definition, the objects using the rule s...

Select View

- Overview
- Folders
- Attachments
- Usage
- Audit Trail

Name: *

OrderHeaderTable_CheckedBeforeLoad

Short Description:

Long Description:

Created By:

Created On:

Last Modified:

Data Steward:

isadmin

Owner:

isadmin

Status:

Candidate

Validity Benchmark

☒ Monitor records that do not meet 1 or more rules

Benchmark:

Met All Rules # > 500

Confidence Benchmark

☐ Monitor records according to number of rules not met

Rules Not Met Limit (Maximum Acceptable Rules Not Met Per Record)

10.0000 %

Max Records Allowed Over Not Met Limit:

10.0000 %

Baseline Comparison Benchmark

☐ Monitor Records According to Difference from Baseline

Benchmark:

Degradation <= 0.0000 %

Baseline Date:

Figure 32-17 Creating a New Rule Set Definition.

- Under the Quality Controls TAB, add all 5 of the previous Data Rule Definitions we just created and tested.
- Click Save, Click Test.
- After the above job completes, Click, View Test Results.

With this Test, you will be prompted for the Join Pair of the Lookup into the Customer table. Example as shown in Figure 32-18.

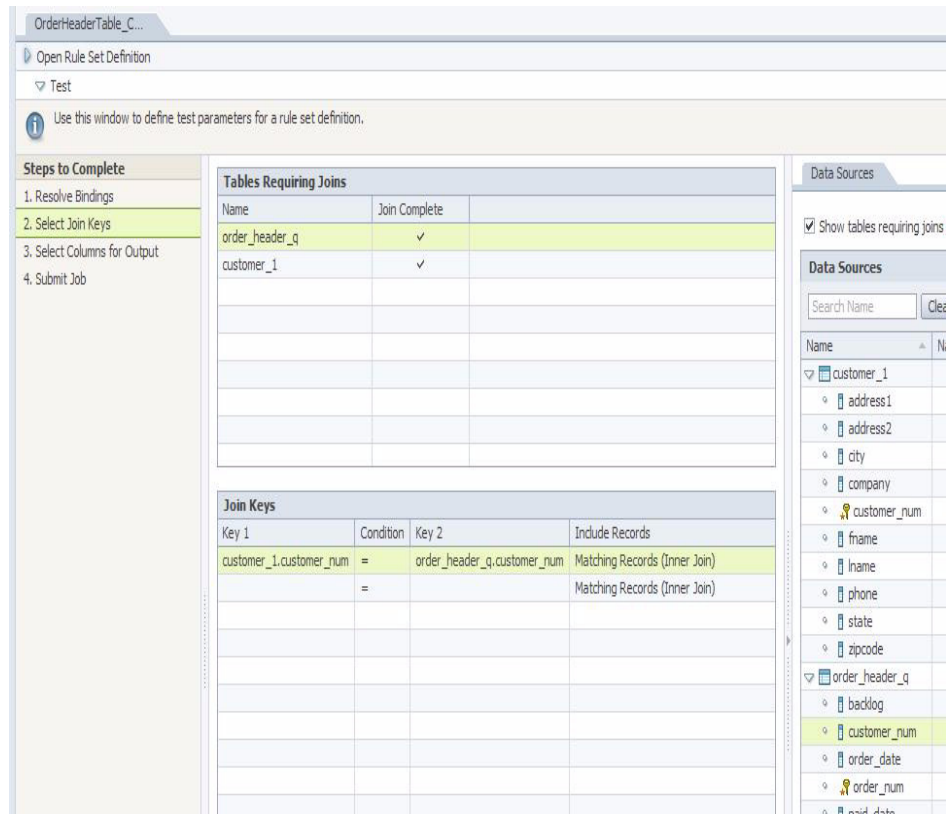


Figure 32-18 Specifying Join Keys for Rule Set Definition.

Note: In each subsequent test, you are inheriting the column binding you tested with in the previous iteration. You can over ride these (default) bindings if you wish, as is the power of a Data Rule Definition, etcetera, that is not hard bound to any given source data column.

g. Sample results are displayed in Figure 32-19 and Figure 32-20.


OrderHeaderTable_C...	
Open Rule Set Definition	
View Test Results	
Overview Result	
Summary	
	Run
Rule Definition	
Rule Definition	OrderHeaderTable_CheckBeforeLoad
Statistics	
Table 1	group_quality.order_header_q
Table 2	group_1.customer_1
Total Records	22
# Met All Rules	14
% Met All Rules	63.6364 %
# Did Not Meet 1 or More Ru	8
% Did Not Meet 1 or More Ru	36.3636 %
Mean #	0.4091
Mean %	8.1818 %
Standard Deviation #	0.5768
Standard Deviation %	11.5351 %
Validity Benchmark	
Benchmark Status	
Benchmark	Met > 500
Variance %	2,209.0909 %
Variance #	486
Trend	
Job Details	
Start	1/2/2010 11:35:31 PM
End	1/2/2010 11:35:45 PM
Elapsed	0 minutes, 14 seconds
Sample Details	
Data Sample	No
Sample Size	
Sample Type	

Figure 32-19 Sample Results, page 1.

Rules Composed in the Rule Set							
Rule Definition	Description	Rule Logic	Met		Not Met		
			#	%	#	%	
IfValueIsYN		sourcedata in_reference_list {'Y','y','N','n'}	22	100.0000 %	0	0.0000 %	
IfIsValidShipWeight		sourcedata >= glv_minimumshipweight	21	95.4545 %	1	4.5455 %	
IfIsValidPONum		sourcedata matches_regex '^[A-Z][0-9]	15	68.1818 %	7	31.8182 %	
IfColumnIsNotNull		sourcedata exists and len(trim(sourcedata)) > 0	21	95.4545 %	1	4.5455 %	
IfIsValidCustomer		sourcedata in_reference_column custom	22	100.0000 %	0	0.0000 %	

Figure 32-20 Sample Results, page 2.

- From the Tasks panel, highlight the Rule Set Definition we just created, and Select, Generate Data Rule or Rule Set.

Complete the steps; naming, other.

From the Tasks Panels, Select Run, and then View Output.

Experiment with Make Baseline, and Chart View. (Chart view is a button in the bottom left areas of the display.)

Note: This definition of 'baseline', is just like that from BaseLine analysis; a comparison between 2 recorded moments in time.

- Not covered in this document is the topic of Metrics.

Metrics allow you to define and also weight the various observed results from a Data Rule or Rule Set Job run.

In this manner you could specify things like; I wish to record a missing column value here, but that is of lower important than, for example, an outright incorrect Customer Number, etcetera.

- Optional: Create an IBM InfoSphere Information Server DataStage component Job to run any of the above Data Rules or Rule Sets from the command line.

Generally, the Data Rules and Rule Sets you Generated above can be exported as an XML encoded property file, that is then used to configure a standard runner.

- a. From the Tasks panel, highlight the Rule Set Definition we created above, and Select, Run.
- b. Under the Scheduler TAB, Select 'Create External Bundle Location'.
Enter a directory name here with no trailing slash.
- c. And Click, Submit.

An XML properties file will be saved on the Client Workstation with the name, {Rule Set}.xml, where "Rule Set" is the name of the rule Set you selected to Run.

Note: So obviously we would like a network mounted drive here to facilitate file location on the DSEngine Server tier.

- d. On a Linux Server, this XML properties file is run with a command equal to (one line below, split across multiple lines for readability),

```
/opt/IBM/InformationServer/ASBNode/bin/IAJob.sh
-user {username} -password {password}
-isHost {Linux hostname with DSEngine tier}
-port 9080 -runBunble
{Absolute path name to XML Properties File created above}
```

That's a lot of typing, so we put our command invocation inside a DataStage component Job, as displayed in Figure 32-21.

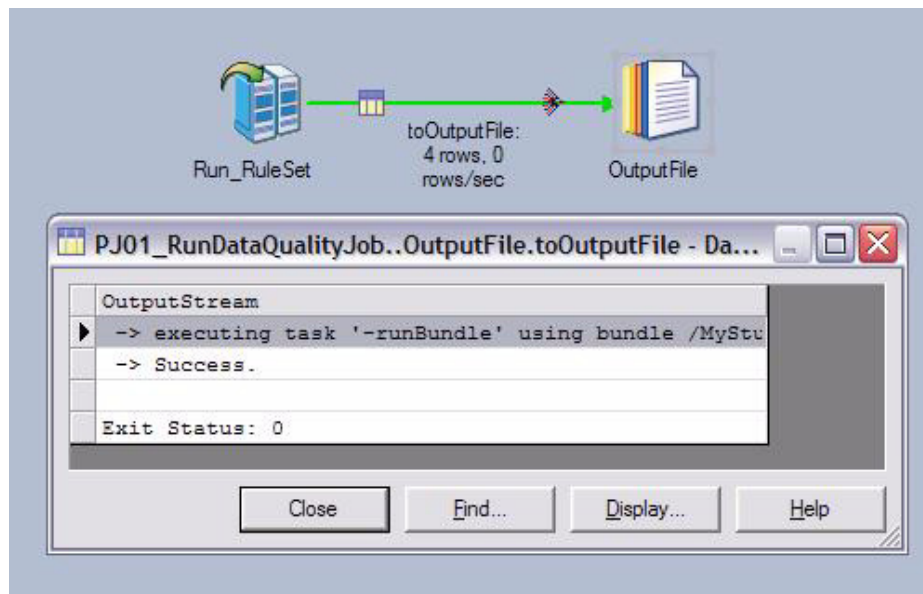


Figure 32-21 DataStage Job to Run and results from above.

32.3 In this document, we reviewed or created:

We detailed the Information Analyzer version 8.1.1 new functional area of Data Quality; including, Data Rule Definitions, Rule Set Definitions, Global Logical Variables, Metrics, Rule Set Expressions and much more.

With this new area of functionality inside IBM InfoSphere Information Server, end users can truly monitor the quality of their input data feeds, and more.

Persons who help this month.

Steve Fazio, Robert Dickson.

Additional resources:

RegEx Expression checker,

http://www.codehouse.com/webmaster_tools/regex/

RegEx Primer, documentation,

<http://www.addedbytes.com/download/regular-expressions-cheat-sheet-v2/pdf/>

Legal statements:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating trademarks that were owned by IBM at the time this information was published. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product or service names may be trademarks or service marks of others.

Special attributions:

The listed trademarks of the following companies require marking and attribution:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are 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.

Microsoft trademark guidelines

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel trademark information

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Other company, product, or service names may be trademarks or service marks of others.