

# Micro Focus The Theory of Relativity

A Primer

Micro Focus  
The Lawn  
22-30 Old Bath Road  
Newbury, Berkshire RG14 1QN  
UK  
<http://www.microfocus.com>

© Copyright 2018-2020 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Revised 2020-05-06 for version 12.17

# Contents

<b>The Theory of Relativity</b>	<b>1</b>
The Purpose.....	1
The Perspective .....	2
The Software Tools .....	2
ODBC-Enabled Application .....	2
Relativity DBMS.....	2
ODBC Drivers .....	3
DBMS Configurations .....	3
Relativity Designer.....	4
Relativity Schema Extractor .....	5
Relativity DBA.....	5
Relativity Server Administrator .....	5
Relativity Governor Administrator.....	5
Relativity Software Use Summary .....	6
The Concepts.....	6
Relativity Catalog.....	6
Data Source.....	6
Data Source Types .....	7
Data Source Administration Tools .....	9
Data Source Setup.....	10
Specifying Catalog Location for a Data Source .....	10
COBOL Source.....	11
Client/Server Example .....	11
Install Relativity Data Server and Create a Server Data Source .....	11
Install Relativity Data Client and Create a Client Data Source .....	12
Run the Relativity Schema Extractor (in Non-RM/COBOL Versions) .....	12
Import COBOL File Definitions .....	12
Augment the COBOL File Definitions on the FD Form .....	12
Define Relational Tables on the Table Form.....	13
Define Security for the Catalog .....	13
Distribute the Catalog to End Users and Set Up the End Users' Site .....	13

# The Theory of Relativity

Relativity is a product containing several related tools. To work successfully with any tool, you must understand how it operates and when to employ it. This document discusses the Relativity tools and their proper use.

---

## The Purpose

The purpose of Relativity is to provide access to data contained within files managed by COBOL applications. COBOL is an excellent language for storing and manipulating business data. However, many of the ad hoc query, reporting, executive support system, and decision support system programs available today are unable to access the COBOL data in any proficient way. Although COBOL programs could be written to analyze the data, it is not cost-effective for a programmer to spend weeks writing a program to perform an analysis that might be used only once.

Most modern, Windows-based analysis products provide access to external databases by using a programming interface called ODBC (Open Database Connectivity). Originally designed to access data from traditional relational databases such as Oracle or DB2, this same facility can be used to access COBOL data files if they are presented in the form of a relational database. That is precisely what Relativity does.

The ODBC interface presents data to a program in the form of tables. A table contains rows, which are similar to records in a COBOL file. A row contains columns, which are similar to fields in COBOL records.

Creating an ODBC-accessible relational database management system, however, is not as simple as reading records in the COBOL file, applying a few conversions, and returning the data disguised as rows. One difficulty is that COBOL programs frequently store more than one type of record in a file, so it becomes necessary to be able to distinguish one record from another.

A related problem happens when a single data item has different meanings depending on values stored in the record. For example, a discount field may mean a percentage discount or an absolute discount as a result of the setting of a "discount flag" stored in the same record. In addition, there are instances when one record contains multiple occurrences of a piece of data. This would normally need to be viewed as multiple rows in a table.

To be able to make these transformations requires knowledge of the COBOL application itself and the correct interpretation of this knowledge by the COBOL programmer.

---

## The Perspective

The Relativity tools are designed around the responsibilities of the people who will use them. A knowledge of their roles is helpful in understanding when each of the Relativity tools is to be employed.

The **End User** is the person currently using the COBOL application program, and who is extracting and manipulating information out of the data that the application program manages.

The **Database Administrator** is responsible for the security of the data managed by the COBOL application program. This is the person who installed the application, and assigned user names and privileges, if any, to the End User(s).

The **Programmer** is the person who maintains the COBOL application and has access to the COBOL source code. Even if the Programmer is not the original author, the Programmer ideally would be very familiar with the physical structure of the data within the data files.

The **System Administrator** is responsible for installing software on computer systems and maintaining these systems.

Clearly, not every site will have these jobs performed by separate people. At some locations, the Database Administrator and the System Administrator may be the same person. It also is not unusual for the Database Administrator and the Programmer roles to be performed by one individual. The larger and more secure a site is, however, the more likely that the jobs will be separated, if for no other reason than security.

See the table on page 5 for a summary of who uses the Relativity software tools.

---

## The Software Tools

Now that you have an understanding of the purpose of Relativity and a perspective of the people who will use this product, let's introduce the line of Relativity software tools.

### ODBC-Enabled Application

An ODBC-enabled application is a Windows application that has been designed to use the ODBC interface for access to data external to the application. The ODBC interface requires that a data source (see page 7) be configured first in order to allow data access. The End User will use these ODBC-enabled application programs to access the COBOL data through the Relativity DBMS. Such programs include Crystal Reports, Microsoft Access, Microsoft Excel, Microsoft Visual Basic, Microsoft Word, and Cognos Impromptu.

### Relativity DBMS

The Relativity DBMS (Database Management System) is software that implements an ODBC-compliant and relational database view of actual, operational COBOL data files. The Relativity DBMS also can take instructions from the ODBC-enabled

application to access and change data in the COBOL data files. The ODBC-enabled application issues a high-level command called an SQL (Structured Query Language) statement describing the data that the application wants to access. The Relativity DBMS performs the same actions a COBOL program would perform in order to extract that data and then returns the data to the ODBC-enabled application in the form of "rows".

Two further topics about the Relativity DBMS are of interest: the ODBC Drivers and the DBMS configurations.

### **ODBC Drivers**

As mentioned earlier, most modern, Windows-based applications provide access to external databases via a programming interface called ODBC. Originally, the databases to be accessed were established ones, such as Oracle or DB2. These databases had their own programming interfaces, although none of them had the same interface. Consequently, any Windows tool that wanted to access these databases had to write a separate routine for each database it wanted to access. Although ODBC solved this problem by providing a common database interface for all the Windows tools, there still needed to be a routine that converted the ODBC-enabled application interface into the interface for the actual database. ODBC calls this code a Driver.

The Relativity database was developed after Microsoft defined the ODBC interface. The Relativity DBMS uses the ODBC interface as its native programming interface. It implements the ODBC interface directly. Does Relativity have an ODBC Driver then? In the strictest sense, no. There is not a separate routine that converts ODBC calls into Relativity calls. However, to an ODBC-enabled application, this distinction is unimportant. The Relativity DBMS implements the ODBC interface. As far as the ODBC-enabled application is concerned, the Relativity DBMS has a driver, but that driver is integrated with the DBMS component.

### **DBMS Configurations**

The Relativity DBMS comes in two different configurations. In one configuration the complete Relativity DBMS resides on a single machine. In the other, the Relativity DBMS is split into two components so that the DBMS can reside on two machines. Even though Relativity has these two different configurations, the rest of this document will use the term Relativity DBMS to mean either configuration.

The single machine configuration is known as single-tier or non-client/server and is used when the data files are directly accessible from the Windows machine. When it is necessary to refer specifically to the single-tier configuration component of the Relativity DBMS, the following component name is used:

- **Relativity Data Manager.** The Relativity Data Manager is used in the single-tier configuration where the complete Relativity DBMS resides on a single Windows machine. This component implements the ODBC interface, the file to relational database transformations, and the application data file access routines. All the data files to be accessed must be accessible from the Windows machine, either because the files exist on disk drives attached to that machine or are accessible using the Windows network file system.

The two machine configuration is known as two-tier or client/server. In this configuration, one component of the Relativity DBMS resides on a remote machine, called the server, close to the data files that the DBMS accesses. Another

component of the Relativity DBMS resides on the Windows machine, called the client, so that the ODBC-enabled application can access the data. When it is necessary to refer specifically to either of the two-tier configuration components of the Relativity DBMS, the following component names are used:

- **Relativity Data Client.** The Relativity Data Client is the component of the Relativity DBMS that resides on the Windows client machine and provides the ODBC interface for the two-tier configuration of Relativity. ODBC-enabled application(s) running on the client machine access Relativity data on the server machine using the interface implemented in the Data Client. The Data Client is, essentially, the ODBC Driver for Relativity in this configuration.
- **Relativity Data Server.** The Relativity Data Server is the component that resides on the remote server machine and implements the file to relational database transformations and the application data file access routines. This component runs on the remote machine that has access to the COBOL data files, either directly or through the network file system of the remote machine.

In the two-tier configuration, the Relativity Data Client and Relativity Data Server communicate across the network that connects the client machine and the server machine. When an ODBC-enabled application using Relativity makes a request that requires participation of the server (not all ODBC requests require such participation), the Relativity Data Client forwards the request to the Relativity Data Server, which acts on the request and returns the results to the Relativity Data Client. The Relativity Data Client then provides the results to the ODBC-enabled application on the Windows machine.

## Relativity Designer

The Relativity Designer is a tool that is used by the Programmer to describe the details of the physical structure of data contained within the COBOL data files and the particular relational view of the data to be presented to ODBC-enabled applications. The Programmer must describe this structure using the Relativity Designer program so that the Relativity DBMS will have enough information to be able to carry out the requests from the ODBC-enabled application. To help achieve this, the Relativity Designer provides the ability to import a description of the data files from the COBOL application source itself and then to create tables containing columns that map to fields in the COBOL records.

As noted earlier, simply mapping COBOL records to table rows and COBOL fields to table columns is not sufficient. Consequently, the Relativity Designer allows the programmer to specify a proper mapping in ways including the following:

- The Relativity Designer allows multiple tables to be derived from a single file in order to more logically organize the data. The Relativity Designer can specify a selection of fields on which the columns of a specific table are based.
- If the COBOL data file contains multiple record types, the Relativity Designer can be used to name the record types and to indicate how to distinguish one record type from another. A table can then be associated with a single record type so that the columns of the table contain consistent data.
- If the COBOL data file contains repeating data, that is, data described with the OCCURS clause, the Relativity Designer can be used to specify the treatment of such data. Repeating data can generate multiple rows or be flattened into a single row using options in the Relativity Designer. Many other special cases related to repeating data also can be specified in the Relativity Designer.

- If there are COBOL fields containing data that need to be interpreted in a specific way, the Relativity Designer can be used to indicate the proper interpretation. For example, fields that represent dates can be indicated so that columns associated with the date field are provided as a date to the ODBC-enabled application. Field values that represent missing or invalid information represent another area requiring special interpretation. These field values must be presented as a special empty or NULL column to the ODBC-enabled application.

The information collected by the Relativity Designer is stored in a Relativity catalog (see page 6).

## Relativity Schema Extractor

The Relativity Schema Extractor utility program is used to provide the Relativity Designer with details about the COBOL application's data files. Similar to a compiler, the Relativity Schema Extractor utility is used to create a COBOL schema file from a COBOL source file (see the "COBOL Source" topic on page 11 for more details). This utility is used for IBM VS COBOL II, Micro Focus COBOL, DEC COBOL, and VAX COBOL, but is not necessary for RM/COBOL. The COBOL schema extracted by this utility is ultimately stored in a catalog when the Relativity Designer is used to import the schema from the schema file.

The schema extraction process extracts information from the file control entry (SELECT clause), file description entry (FD entry), and the associated data description entries for each file described in the COBOL source file and creates the COBOL schema file containing this information. The COBOL schema file is also called the Relativity Symbol Table file (or RST file).

## Relativity DBA

The Relativity DBA is a tool that is used by the Database Administrator to assign user names and passwords to the End Users. These security features can be used to restrict access to the COBOL data. The Relativity DBA provides the ability to create a group of users, to describe the tables and columns that the group may access, and even how they may access the tables and columns, such as the ability only to read them. Names of End Users are created and associated with the groups. With user names, there also may be associated passwords. The Relativity DBA also may be used by the Database Administrator to indicate where the Relativity DBMS can find the COBOL data files for a particular end-user site.

## Relativity Server Administrator

The Relativity Server Administrator utility program is used in two-tier (client/server) configurations of Relativity when the data being accessed reside on a remote server machine.

## Relativity Governor Administrator

The Relativity Governor Administrator utility program provides a mechanism to limit the resources used by queries that Relativity performs. The purpose of this limitation



is to allow the System Administrator to regulate the impact of Relativity on a production system.

## Relativity Software Use Summary

In summary, the following table illustrates "who uses what" in Relativity. As mentioned on page 2, even though each defined role tends to use a particular tool, some people fulfill multiple roles. Those intersections containing the words, "Probably" or "Maybe", indicate a tool that the person may use at some point.

	ODBC-Enabled Application	Relativity Designer	Relativity DBA	Relativity Server Administrator	Relativity Governor Administrator
End User	Yes	No	No	No	No
Database Administrator	No	No	Yes	Maybe	No
Programmer	Probably	Yes	No	Probably	No
System Administrator	No	No	Maybe	Yes	Yes

---

## The Concepts

This section discusses two essential elements of Relativity.

### Relativity Catalog

The Relativity catalog is a file in which the work of the Programmer, using the Relativity Designer to define the associations between table columns and COBOL data fields, is stored. The catalog allows Relativity to present the COBOL data files through the ODBC interface as a relational database of tables independent of the original COBOL data file's structure.

The catalog is accessed by the Relativity DBMS in order to read these associations and carry out the ODBC-enabled application's requests.

In addition, the Relativity DBA stores the privileges, groups, user names, and passwords in the catalog for the Relativity DBMS to enforce.

### Data Source

Data source is an ODBC concept with which the Relativity catalog is closely associated. When an ODBC-enabled application opens a data source, the appropriate ODBC driver is started and the data source is accessed. For Relativity, the Relativity DBMS starts, opens the catalog, verifies the named user has access to the catalog, and passes the names of the tables defined in the catalog for which the user has access privileges back to the ODBC-enabled application. When the ODBC-enabled application requests that a table be opened, the Relativity DBMS consults the catalog for the name of the appropriate COBOL data file, opens the file, extracts the data for the table, and returns the rows of the table to the ODBC-enabled application.

While it is possible to conclude that catalogs and data sources are one and the same, that oversimplifies the issue. Although several data sources can share the same catalog, there is no advantage to doing so. The combination of the Relativity DBMS, the Relativity catalog, and the COBOL data files implements a Relativity data source.

## **Data Source Types**

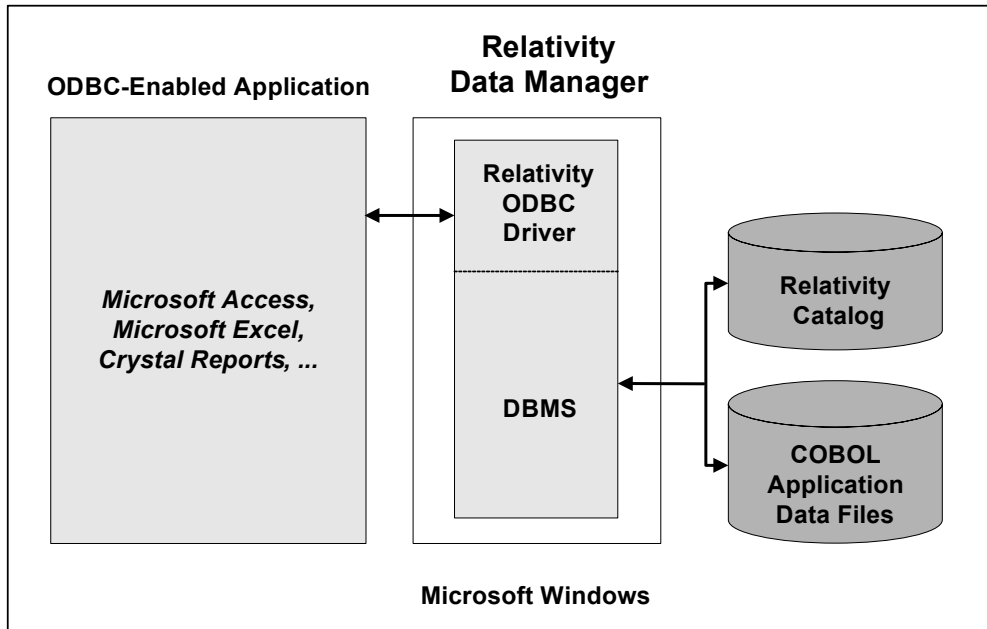
From the ODBC-enabled application's perspective, there is only the data source on the machine on which the application runs. Relativity, however, extends the definition of a data source to two types: single-tier and two-tier.

### **Single-Tier Data Source**

A **single-tier** (non-client/server) data source is one in which the data files are directly available from the Windows machine where the ODBC-enabled application is running. Even though the data files and the catalog can actually reside on another computer on the network, the Windows network file system makes the files appear to be on a local hard drive.

When an ODBC-enabled application opens a single-tier data source:

1. The Relativity Data Manager component of the Relativity DBMS obtains the Relativity catalog name and location from the single-tier data source.
2. The Relativity Data Manager presents a logon dialog box to obtain the End User's user name and password.
3. The Relativity Data Manager completes the data source open operation and validates the user name and password.
4. The Relativity Data Manager obtains from the catalog the names of the tables for which the End User has access privileges.
5. The Relativity Data Manager returns the names of the tables to the ODBC-enabled application.



## Single-Tier Configuration of Relativity

### Two-Tier or Client/Server Data Source

A **two-tier** (client/server) data source consists of two pieces: the server data source and the client data source.

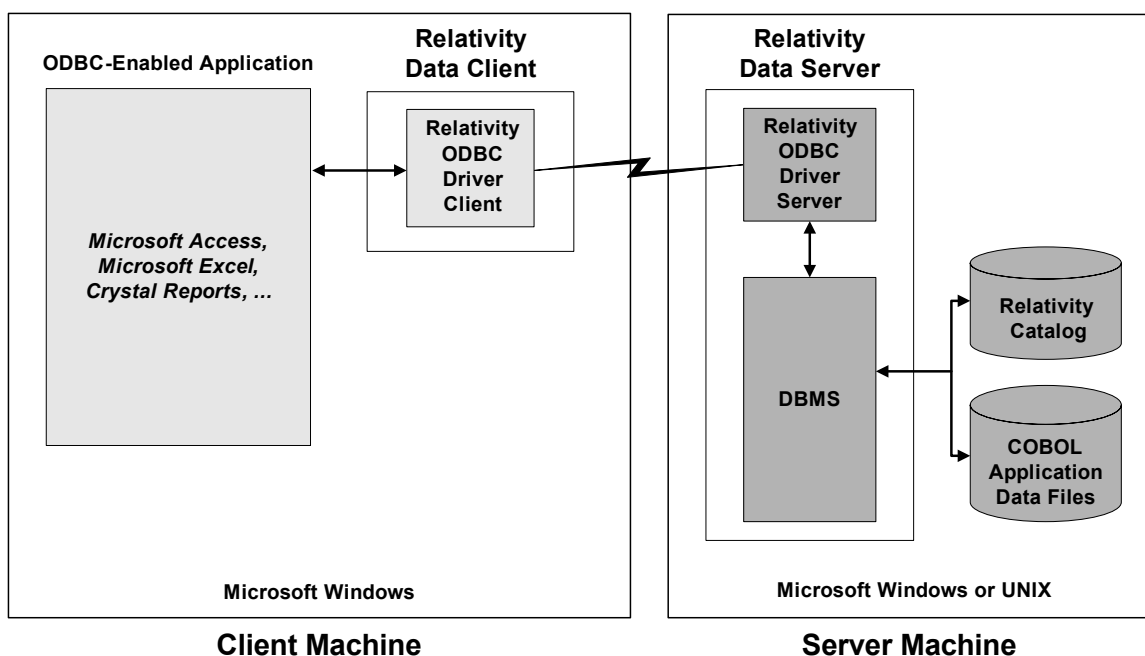
A **server data source** is the portion of a two-tier data source that resides on the remote machine. In many ways, a server data source resembles a single-tier data source in that both refer to the location of the catalog.

A **client data source** is the portion of a two-tier data source that resides on the Windows machine where the ODBC-enabled application is running. The client data source is the data source that the ODBC-enabled application will open. However, instead of containing the location of the catalog, the client data source contains the name of the server machine and the name of the server data source on that server machine.

When an ODBC-enabled application opens a client data source:

1. The Relativity Data Client component of the Relativity DBMS obtains the server name and server data source name from the client data source.
2. The Relativity Data Client presents a logon dialog box to obtain the End User's user name and password.
3. The Relativity Data Client uses the server name to contact the Relativity Data Server component of the Relativity DBMS.
4. The Relativity Data Client requests that the Relativity Data Server open the server data source using the server data source name on behalf of the current user.
5. The Relativity Data Server obtains the name and location of the Relativity catalog from the server data source.

6. The Relativity Data Server completes the data source open operation and validates the user name and password provided by the Relativity Data Client.
7. The Relativity Data Server obtains from the catalog the names of the tables for which the End User has access privileges.
8. The Relativity Data Server returns the names of the tables to the Relativity Data Client.
9. The Relativity Data Client returns the names of the tables to the ODBC-enabled application.



## Two-Tier or Client/Server Configuration of Relativity

### ***Data Source Administration Tools***

The ODBC Administrator is a standard Windows program provided by Microsoft that is accessible from the Windows Control Panel for creating and maintaining ODBC data sources. When requested to add a new data source, the ODBC Administrator will present a list of ODBC drivers that have been installed on the Windows machine. If Relativity has been installed on this machine, the Relativity DBMS will appear in the list. Selecting the Relativity DBMS causes Relativity to run its data source setup utility. Depending on the configuration of Relativity that is installed, either a single-tier or a client data source can be created.

The Relativity Server Administrator is a Relativity utility that is also accessible from the Windows Control Panel for creating and maintaining server data sources. The utility will request the name of the server. When requested to add a new data source, the Relativity Server Administrator will run a data source setup program to create a server data source.

## **Data Source Setup**

Either the ODBC Administrator or the Relativity Server Administrator can invoke a data source setup program to create a data source. In general, all data sources have a name and usually they contain the location of the Relativity catalog. There are differences in the data sources based on the type of COBOL used by the COBOL applications and the files to be accessed by the Relativity DBMS. Thus there are Relativity data source setup programs for the following:

- RM/COBOL (Windows and Windows/UNIX)
- Micro Focus (MF) COBOL (Windows and Windows/UNIX)
- DEC COBOL (Windows/VMS)
- IBM VS COBOL II (Windows/MVS)

The ODBC Administrator creates single-tier data sources for RM/COBOL (Windows) and MF COBOL (Windows) as well as client data sources for RM/COBOL (Windows/UNIX), MF COBOL (Windows/UNIX), DEC COBOL (Windows/VMS), and IBM VS COBOL II (Windows/MVS). Relativity supplies a separate DBMS for each of these types of COBOL. When creating single-tier data sources, the setup program will request the name of the data source, the location of the catalog, and other options that are specific to the type of COBOL that it supports. When creating client data sources, the setup program will request the name of the data source, the name of the server on which to find the server data source and, finally, the name of the server data source.

The Relativity Server Administrator creates server data sources for RM/COBOL, MF COBOL, DEC COBOL, and IBM VS COBOL II. A Relativity server can access only one dialect of COBOL file, thus the version of the server that is installed on the selected machine will dictate the type of server data sources that it will need. Therefore, after the Relativity Server Administrator has been requested to add a server data source, it will start the setup program for the correct COBOL dialect. This setup program will request the name of the data source, the location of the catalog, and other options specific to the dialect of COBOL that the Server Administrator supports.

## **Specifying Catalog Location for a Data Source**

When creating a single-tier data source or server data source in a two-tier (client/server) configuration, you must specify the location of the catalog. The placement of the catalog relative to the data files is important because it will affect how the Relativity DBMS locates the data files.

Essentially, the catalog can be installed anywhere on the machine containing the data files. As long as the Relativity DBMS has the ability to access both the files and catalog, the catalog can be located in any directory on any disk drive desired. However, it will be necessary to use the Relativity Designer or Relativity DBA to explicitly specify the complete name of the data files within the catalog. This has a distinct disadvantage: the location of the data files on the End User's machine may not be the same as on the Programmer's machine. Therefore, after the catalog is installed on the End User's machine, the End User or the Database Administrator must use the Relativity DBA program to change the location of the data files within the catalog before it can be used.

If, however, the catalog is installed in a location that is relative to the data files (either in the same directory, or in directories above or below), the Relativity DBMS can use

the location of the catalog to find the data files. All that needs to be done with the Relativity Designer is to set the location of the data files without a complete name, that is, missing the leading directory. On Windows, this means no leading drive letter or backslash. On UNIX, this means no leading slash. Other operating systems are similar. When the Relativity DBMS sees such omissions, it assumes the directory where the catalog is located is the leading directory for the data file and constructs the complete name to open. This has the advantage of having all the information about the data files and the catalog contained in the data source, and thus the catalog content remains independent of the location of the data files.

## COBOL Source

As mentioned on page 4, the Relativity Designer extracts much of the information it needs from the COBOL application source. This capability reduces human errors, automates what would otherwise be a tedious job, and facilitates the maintenance of the COBOL program and the catalog.

For RM/COBOL, the Relativity Designer can extract the information that it needs from the RM/COBOL application executable. For MF COBOL, DEC COBOL, and IBM VS COBOL II, a utility program called the Relativity Schema Extractor (see page 5) is used to “compile” an RST (Relativity Symbol Table) file from the COBOL source. The Relativity Designer can then import the information from the RST file.

For MF COBOL for Windows, MF COBOL for UNIX, and DEC COBOL, the Schema Extractor executes on the machine where the source resides. For UNIX and VMS, this means that the resulting RST file must be brought to the Windows machine where the Relativity Designer is to run. For IBM VS COBOL II, the COBOL source must first be copied to a Windows machine and the RST file compiled there.

The Relativity Designer includes the capability to merge the COBOL schema information into an existing catalog. Thus, if changes are made to the COBOL application that affect the record structure of files accessed by Relativity, the changes can be applied to the catalog in a mostly automated manner. The Programmer may still be required to make adjustments related to the changes, but will not have to redesign the tables and columns for the unchanged parts of the COBOL schema.

---

## Client/Server Example

The following overview describes the steps necessary to create and deploy a Relativity application in a mixed Windows and UNIX client/server environment.

### Install Relativity Data Server and Create a Server Data Source

The System Administrator installs the Relativity Data Server for UNIX on a machine that contains the COBOL application and data files. This is usually a Programmer's copy of the application containing test data.

The System Administrator then installs the Relativity Data Client on a Windows machine. This process also installs the Relativity Server Administrator. The System Administrator then uses the Relativity Server Administrator to create a server data source, creating a new catalog in a directory that is relative to the test data files.

## **Install Relativity Data Client and Create a Client Data Source**

The System Administrator then installs the Relativity Data Client and the Relativity Designer on the Programmer's workstation.

The System Administrator uses the ODBC Administrator to create a client data source on the Programmer's workstation and connects this client data source to the server data source created earlier.

## **Run the Relativity Schema Extractor (in Non-RM/COBOL Versions)**

If the COBOL application is written in MF COBOL, the Programmer logs on to the UNIX machine and places the location of the Relativity Data Server in their PATH. This will make the Schema Extractor compiler accessible. The Programmer changes to the directory containing the COBOL application and runs the Schema Extractor, specifying the name of the source file of the COBOL application.

## **Import COBOL File Definitions**

The Programmer returns to the Windows workstation and, using FTP (File Transfer Protocol, a common UNIX utility) or some other mechanism, brings down the COB file (if using RM/COBOL) or the RST file (if using MF COBOL) to the Windows machine. This must be a binary transfer.

The Programmer starts the Relativity Designer and opens the client data source created by the System Administrator. The Programmer clicks on the Files tab of the Catalog window and clicks the New button to specify that a new file definition is to be imported into the catalog. The Programmer specifies the location of the COB or RST file, the COBOL program within the COB file (the RST can contain information for only one program), and then the file-names for which the file definitions (COBOL schema) are to be imported.

## **Augment the COBOL File Definitions on the FD Form**

After the import process completes, the Relativity Designer lists the file-names for which it has definitions. The Programmer selects a file-name and clicks the Open button. The FD form opens, resembling a COBOL FD.

The Programmer manipulates the FD to add information that is not available from the COBOL application. For example, one common requirement is to modify the default name of columns later defined on the fields; the Relativity Designer initially establishes a default column name based on the data-name of the field. The Default Column Name column in the FD may be edited directly, or the form provides tools that may be used to perform string replacements on the default column names for all the FD fields.

Other changes that may be made in the FD form include specifying record types and the fields that are in them (including how to recognize the record type from the data within it), adding fields to the FD (especially condition fields to aid in the recognition

of record types), and adding transforms on the fields that contain dates and times to aid in the conversion of the data.

## **Define Relational Tables on the Table Form**

After the Programmer has made the necessary changes to the FD and saved the changes to the catalog, it is possible to create tables using that FD. This is done by returning to the Catalog window, selecting the Tables tab and clicking the New button. The Table form opens and the Programmer selects and drags fields from the FD form into the Table form. This action creates the columns in the table, and creates a connection between the columns and the fields in the FD. After all columns have been created, the Programmer changes, if necessary, the location of the data file that the table is to access, saves the table to the catalog (giving the table a name of the Programmer's choice), and then tests the table and its connection to the data file by pressing the Show button to display every row in the table.

## **Define Security for the Catalog**

After the Programmer has completed creating tables in the catalog, user group names can be created in the catalog, and permissions assigned to them. One of the created groups should be an administrator group, that is, a group with the Use Administrator privilege, to be used by the Database Administrator at the End User site. Finally, a user group name must be designated as the owner of the catalog.

## **Distribute the Catalog to End Users and Set Up the End Users' Site**

After the Programmer is finished with the catalog, it can be distributed to the End User(s). The System Administrator at the End User's site installs the catalog in the relative location that the Programmer used when designing the catalog. Then the System Administrator repeats the steps of installing the Relativity Data Server for UNIX and Relativity Data Client, and creating the server data source using the Relativity Server Administrator. This time, however, instead of creating a new catalog, the System Administrator selects the existing catalog that was installed earlier.

The Database Administrator installs the Relativity Data Client as well as the Relativity DBA on a secure machine. Using the ODBC Administrator, the Database Administrator creates a client data source and connects it to the server data source. Using the Relativity DBA, the Database Administrator opens the client data source, logs in with the user name of an administrator of the catalog, and creates user names in the various groups to reflect the desired access for the End User(s) at this site. Finally, passwords are assigned to the user names.

The Database Administrator installs the Relativity Data Client on the End User's machine and associates a client data source with the server data source. The End User is given a user name and password. Any desired ODBC-enabled applications are installed and the Database Administrator demonstrates how the client data source is accessed from the ODBC-enabled application. The End User is now ready to use Relativity.