RESEARCH REPORT
OCCASIONAL PAPER

Criteria for Evaluating General Database Migration Tools

Bin Wei and Tennyson X. Chen

October 2012

RTI Press

RTI INTERNATIONAL

About the Authors

Bin Wei, MS, is a senior researcher at the Pacific Islands Fisheries Science Center, a National Oceanic and Atmospheric Administration (NOAA) research branch within the Joint Institute for Marine and Atmospheric Research, a NOAA Cooperative Institute at the University of Hawaii at Manoa.* His areas of expertise include designing systems for remote data collection and data transformation.

Tennyson X. Chen, MS, is a senior research analyst and software system architect in RTI International's Research Computing Division. His main focus is the National Survey of Drug Use and Health (NSDUH) project, for which he is a key system designer and database manager.

* Although Mr. Wei is employed by the NOAA Joint Institute for Marine and Atmospheric Research, this paper was produced independently and is not a work product of NOAA.

RTI Press publication OP-0009-1210

Suggested Citation

Wei, B., and Chen, T. X. (2012). *Criteria for evaluating general database migration tools*. RTI Press publication No. OP-0009-1210. Research Triangle Park, NC: RTI Press. Retrieved from http://www.rti.org/rtipress.

This publication is part of the RTI Research Report series. Occasional Papers are scholarly essays on policy, methods, or other topics relevant to RTI areas of research or technical focus.

RTI International
3040 Cornwallis Road
PO Box 12194
Research Triangle Park, NC
27709-2194 USA

Tel: +1.919.541.6000
Fax: +1.919.541.5985
E-mail: rtipress@rti.org
Web site: www.rti.org

http://dx.doi.org/10.3768/rtipress.2012.op.0009.1210

www.rti.org/rtipress

Criteria for Evaluating General Database Migration Tools

Bin Wei and Tennyson X. Chen

Abstract

Software updates often involve data migration, especially when converting legacy software implemented to interface with outdated relational database management systems or other nonrelational database electronic files. Moreover, many software applications rely on data migration to import data from a variety of platforms. Usually, database migrations are time consuming and error prone. Many independent general data migration products are available, but what features does such a tool need to provide to be effective? Based on our experience designing and implementing custom utilities to convert a large number of legacy databases and files in different platforms, we developed five criteria that need to be considered when evaluating a data migration tool (DMT). These criteria can help users and software development project managers make informed decisions in data conversion tasks, help software developers assess design and implementation considerations for future DMT products, and provide guidelines for database administrators to evaluate a general DMT.

Contents

RTI INTERNATIONAL

Introduction

One of the largest problems organizations face is ensuring quality database administration.[1] As many legacy software applications age, programmers are upgrading them to the most up-to-date technologies. Because many of these applications interface with old database platforms, one of the most important issues in a software upgrading effort is data conversion, that is, transferring the data from the old systems to new relational database management system (RDBMS) platforms or converting data between RDBMS platforms. Converting data from one database platform to another, whether it has the same or a different data structure or format, is called data migration.

Data migration is a specific implementation of a broader concept called data extract, transform, and load (ETL). A properly designed ETL system extracts data from the source systems, enforces data quality and consistency standards, conforms data so that separate sources can be used together, and delivers the data in a presentation-ready format.[2] The criteria we discuss in this paper contribute to the body of knowledge and experience of ETL.

According to an industry analyst's estimate in 1999,[3] at any given time, roughly two-thirds of the Fortune 1000/Global 2000 companies are engaged in some form of data conversion project, including migrating data from legacy systems to packaged applications, consolidating data, improving data quality, or creating data warehouses and data marts. Although this estimate is somewhat outdated, we believe that data conversion is still a very common task among today's database administrators. In addition, businesses may face situations in which their success relies on data migration, such as transferring data in different platforms into a consolidated database during a company merger or moving data between contractors' databases when a federal contract changes contractors.

Unfortunately, the cost and difficulties of a data conversion effort are almost always underestimated. Problems with data can result from missing information and mismatches between the data in the original platform (which is often less clearly documented) and the new platform (which is usually more explicitly documented).[4]

Most RDBMS, such as Oracle, My SQL, and SQL Server, have their own database transferring utilities for some database migration tasks. However, these transferring utilities are often based on two important assumptions. First, users must assume the content of the source data is correct, especially with respect to domain integrity, entity integrity, and referential integrity, which are discussed in the section Support for Data Integrity Checking. If the source data have problems, as most source data do, data transfers using the built-in utilities will often fail. Second, these RDBMS utilities often assume that the data in the destination database have a structure identical to that of the source database. If database administrators want to modify the structure in the destination database to accommodate requirement changes, they have to change the structure of the source data as well. In reality, it is rare for a data migration project to move data from one system to another with identical data structure. Consequently, database administrators often must fix the source data manually or by writing a customized program. Customized programming increases the cost of a data migration project and could be a source of errors as well.

These limitations mean that the RDBMS built-in data transfer utilities are difficult to use in large-scale data migration projects. Quite a few third-party products have been developed and are commercially available—such as Altova MapForce 2012,[5] Astera Centerprise Data Integrator 5.0,[6] DBConvert data migration products,[7] and SwisSQL Data Migration Tool 6.5[8]—each with its own strengths and limitations.

In a project for the US National Oceanic and Atmospheric Administration (NOAA), we implemented a robust database migration tool (DMT) to transform about 500 FoxPro flat tables and other external files into Oracle and MySQL. The data in these tables and files dated as far back as 1948. This DMT is also currently used to convert about 100 flat tables per month from different sources on a flow basis. We designed this DMT to allow data to be migrated from a large variety of platforms into the most current RDBMS platforms. For this project, we considered the possibility of purchasing an existing DMT product or developing a customized tool or utility. We thoroughly researched user manuals and experimented with trial versions of many commercial DMT products. For each product, we examined several fundamental questions, such as whether the tool would be suitable for the task at hand or it would be more cost-effective to build a custom tool. After in-depth evaluation, we concluded that none of the products that we had investigated could meet our requirements without significant customized programming. Therefore, we decided to build a new DMT tool to perform our task. As a result of this experience, we developed a set of five criteria that are useful in evaluating a DMT:

- types of databases the DMT supports;
- user interface configurability, maintainability, and reusability;
- support for data integrity checking;
- customization adaptability of the DMT; and
- data correctness verification.

In the following section, we discuss these criteria in detail to help users and software development project managers evaluate general DMTs and make informed decisions when facing data conversion tasks. Software developers might also use these criteria as design and implementation considerations for future DMT products. We believe these criteria can benefit other data migration projects and can evolve into best practice guidelines in DMT product evaluation and development.

Evaluating Database Migration Tools

The complexity of database migration varies from project to project. Therefore, depending on the data conversion task, database administrators may be interested in different features of a DMT, and they may not consider each criterion with equal weight. We offer the following criteria as a general guideline to help users decide whether a DMT product is viable based on their particular needs.

Types of Databases the DMT Supports

The first criterion relates to whether the DMT supports the source and destination data platforms involved in the desired data conversion. Although most of the current DMT products on the market, such as Astera and SwisSQL, support the most popular databases, such as Oracle, SQL Server, DB2, Sybase, MySQL, and MS Access, users cannot assume that all DMT products have the same functionality. As a matter of fact, in our NOAA project, certain source database types were not supported by any of the tools mentioned above. Also, because today's computer applications are increasingly likely to interface with multiple operating systems and database platforms, users may find it necessary to evaluate whether a DMT supports databases on mobile devices such as smartphones and tablets. In addition, they may need to consider other software packages such as SAS if the data conversion task involves interfacing with these mobile devices or software packages and the data formats they use.

If a DMT does not support the database platforms the user is working on, the second question is how easily the DMT can be adapted to expand its support to the desired database platforms. The evaluation question is if a new database platform can be added and how much effort this addition will be involved. If this problem can be resolved without significant extra costs, the DMT passes this first test.

In some special situations, like our recent project for NOAA that required converting data from DBase, FoxPro, and data files dated back to the 1940s, none of the DMT products that we evaluated were able to successfully migrate all the data. This is one of the main reasons behind the decision to build a customized DMT. In fact, in our DMT design, because of the need to read data from so many old database platforms, we focused on the diversity of the databases that the tool supports. As a result, the tool we developed is very robust in migrating data between a variety of electronic platforms. If we need to add a new platform as a source or destination database, we only need to add the corresponding database driver to the application package and do not need to change the migration application itself. This approach is unique, compared with commercial DMT products currently available.

User Interface Configurability, Maintainability, and Reusability

Data migration involves a certain level of risk. If not executed carefully and correctly, the process may erase part or all of the data in either the source or the destination database. For example, if a data migration interface allows user to mistakenly switch the specification of source and destination database, the source data may be overwritten, hence destroyed, by the data in the destination database. This consideration brings us to the second evaluation criterion: the user interface of the DMT product. A good DMT needs to provide an interface that allows users to easily and unambiguously specify a data transfer configuration, such as the source and destination databases, tables and columns, and ranges of the data to be included in the transfer.

Moreover, although most of the popular products, such as Astera, DBConvert, and SwisSQL, provide common capabilities in their interfaces for users to input the source and destination databases or file information, more advanced levels of customization should be considered as well.

Does the DMT allow users to do the following?

- Select specific tables and columns to transfer.
- Add, change, or remove column name, type, or other properties when transferring.
- Add, change, or remove constraints like primary key, foreign key reference, and other properties.
- Add, change, or remove viewers, functions, or other utilities in the destination database when transferring data.

Another significant factor to keep in mind is that data migration is usually not a one-time process. For a large data transfer, users usually will need to set up experimental environments to test the procedure and correct potential errors before officially running the process to transfer real data. Once a configuration works correctly in experimental trials, the DMT should be able to use that configuration to run the real data transfer. In addition, data migration can be an ongoing process in which users are required to run and convert new data with that same configuration into production databases on a continuous basis. Therefore, maintaining and reusing a configuration are important issues.

Most DMT products, like the ones mentioned previously, use graphical user interface (GUI) wizards to steer users through the data-transferring process. They provide the interfaces for users to enter source and destination database platform information and select tables and columns, for example. Although this method works well most of the time for one-time operations, our experience indicates that a configuration defined in Extensible Markup Language (XML) specification will work better in terms of maintaining and reusing a configuration. XML is common knowledge among computing professionals, is easily understood among different users, and is easy to modify. A user can modify it in a text editor without going through the entire GUI. Another benefit of using XML specification is that the XML file itself documents how the data transfers are performed. This document can help users trace problems if errors occur during or after a transfer. The GUI method does not have this provision. Furthermore, a DMT that works with an XML configuration can be scheduled to run at certain times of day without human intervention—another advantage of XML over GUI interfaces when undertaking an ongoing and repetitive data migration task.

One hybrid approach is to take advantage of the strengths of both GUI and XML. Having the capacity to save a GUI configuration into XML format that allows future modification provides flexibility. Users who need to run the same or similar data transfer multiple times during the course of a project should check whether the DMT has the capability to handle XML configuration specifications. This feature can be a real benefit in large, ongoing, and complex data migration tasks.

Support for Data Integrity Checking

The third criterion relates to database integrity and its impact on the data migration process. Quite often, a database transfer attempt fails because the data fail to comply with the database integrity constraints in the destination database. Understanding database integrity and how it can affect data migration is critical as well as beneficial.

In essence, three types of integrity constraints are important components of a relational data model:

Entity integrity: Every table must have a primary key, and the column or columns chosen to be the primary key should be unique and not null.[9]

Referential integrity: No record with a foreign key referencing column is allowed to exist if the corresponding referenced record does not exist.[10]

Domain integrity: Domain integrity controls the validation, such as the type of data and the range of the data, of values for a column.[11]

These database integrities play an important role in preserving the accuracy and consistency of the data in a database. However, these constraints require a certain level of agreement between the source and destination databases when attempting to transfer data between them. If the source data do not meet the integrity requirements in the destination database, the data transfer will not succeed.

Many DMT products, especially those like SQL Server and Oracle utilities that are built into the RDBMS, proceed with the data transfer without checking database integrity. Consequently, the data transfer fails and the operation is halfway completed. The incomplete process causes additional hours of clean-up work. Usually, undoing a failed database transferring operation on a large scale is risky and time-consuming.

A DMT tool that does not offer integrity checking prior to a database transfer can introduce another potential error, in which data may be lost after the transfer, even when the migration appears to be "successful." For example, suppose all character fields of a table in the destination database are specified with a certain length limitation. If the source data designated to be migrated into a character field contain strings that exceed this field's length limitation, a DMT tool may just truncate the strings to make them suitable for the destination database. This approach results in loss of data.

A better way is for a DMT product to check the source data first to ensure that all data comply with the database integrity requirements in the destination database. If such a check determines that the transfer will not succeed because of the database integrity requirements, the DMT should not proceed with the transfer. Rather, it should prompt the user to fix the data or make some adjustments in the databases before proceeding. We discuss this issue in more detail in the following section, Customization Adaptability of the DMT.

Database integrity should be checked at two levels. The first level is with respect to domain integrity. For each column, users need to verify that the source data can be correctly situated in the destination database. Two simple examples of issues are as follows:

- For a numeric column in the destination database, do the source data contain only numeric values in the corresponding column?
- For a character column in the destination database, does the length requirement of the corresponding column in the source database match?

The second level of database integrity checking is more complex. Users need to check the entity integrity and referential integrity among tables. Two examples are as follows:

- When the destination database is empty, the DMT should obtain the primary key and foreign key definitions in the destination database and verify that the data in the source database tables comply with the defined constraint requirements.
- When the destination database is not empty, the DMT first needs to perform the above check to validate the source data. Then the DMT needs to check whether the data in the source database present primary key conflicts with the existing data in the destination database.

If either check detects missing values in primary key columns or foreign key referred columns or if inserting the new data would cause key violations in the destination database, the DMT should generate a report to alert the user before transferring the data. These checks are critical because they will avoid failed transfers and the loss of data.

Customization Adaptability of the DMT

The fourth criterion relates to how easily the DMT can be customized. In a complex data migration task, rarely can a set of data be converted directly into another set in a totally different structure with different constraint requirements. When transferring data, users often need to correct the data in the source database or make adjustments in the destination database, especially when the data fail to pass the database integrity check discussed in the previous section, Support for Data Integrity Checking. For example, if a foreign key reference depends on a value that is missing in the referred table, the user needs to either add the value in the source database or relax the referential constraint in the destination database prior to the data transfer. Otherwise, the data transfer will fail.

If the amount of data to be transferred is huge, this task of changing data or making database adjustments is too great to perform manually. Therefore, designing a DMT product that includes a mechanism to change data or database constraints greatly simplifies the data migration task. Furthermore, when converting data from one platform to another, quite often there is no direct mapping of data type and content. In this case, some special rules need to be implemented. For example, in our project work, we found that some older database files, such as DBase and FoxPro, have "Boolean" or "Logic" as data type, but SQL Server does not have an equivalent data type. Therefore, when we tried to transfer data from these older databases into SQL Server, we had to decide between either converting a column with Boolean data type into a "Bit" field in the SQL Server and translating true and false values into 1s and 0s or converting the Boolean field into a "Character" field that takes a value of "Y" or "N." Either way, a special rule had to be applied during the data transfer.

In addition, in a relational database, the complexity of the interrelationship of dependencies as a result of the database integrity constraints among tables grows exponentially when the number of interrelated tables grows. Sometimes the dependencies are so complex that converting all the data automatically all at one time is impossible. Quite often, the order of the tables and columns to be transferred needs to be specified. For example, a table with a column that is referred to by a foreign key in another table must be transferred prior to the latter table. This problem is common with all DMT products. Callan summarized this problem nicely: consider "break[ing] a dependency chain before it breaks you and the migration process."[12]

In summary, users can evaluate three important factors when assessing how well a DMT product can be customized:

- Does the DMT product allow users to change data in the source database and make necessary adjustments in the destination database?
- Does the DMT product allow users to write customized code to be incorporated into the data transferring process to implement special business rules when writing the data into the destination database? These rules include data type conversion, data value translation, and default value setting, for example.
- Does the DMT product allow users to analyze dependencies among tables in the source and destination databases and arrange the data transfer in the correct sequence? Does it allow users to specify the sequence of the data transfer? Ideally, the DMT could generate a report suggesting the data transferring sequence with regards to database integrity, and let the user verify the validity of transferring sequence.

A complete and useful DMT product should address these three issues to provide flexibility in the data migration process. Among the popular DMT products we evaluated, Astera had the most flexibility for customization. Unfortunately, many other DMT products do not allow users to customize the data migration process. These products require users to migrate the data into the exact table structure in the destination databases. This level of inflexibility greatly limits the scope of application of these products.

Data Correctness Verification

The final criterion relates to verifying the data's correctness. Even if a data transfer request is executed successfully by a DMT, users cannot assume all the data are transferred and situated correctly in the destination database. Most DMT products provide only a simple report stating the number of records that have been migrated. Although this record count provides an important clue to any records missed during the data transfer, it does not ensure the correctness of the data.

Currently, none of the DMT products we evaluated provides a comprehensive database comparison function to verify the correctness of a data transfer. Therefore, after each data transfer, users must perform this validation either by inspecting the database visually or by writing special processes.

Ideally, future DMT products would incorporate a database comparison utility to help users verify the correctness of a data transfer. When comparing databases, one approach is to perform a record-by-record and column-by-column assessment. But this method is time-consuming and is not feasible for a large amount of data. The other approach is to perform spot and summary checks of the data. A few examples of database comparison checks that we implemented in our own DMT are as follows:

- For columns with "Numeric" data type, we compared the maximum, minimal, average, and summary values between the two databases.
- For columns with "String" data type, we compared the string length and checksum values between the two databases.
- For columns with "Date/Time" data type, we converted the values into the numeric representation and compared the databases with the method applied on Numeric columns.

Implementing these checks in our DMT tool was instrumental to the success of our data migration project. These checks can catch the error in loss of data discussed in the section Support for Data Integrity Checking, and they provide assurance in assessing the successfulness of a data migration operation. This data correctness verification feature saves time and lowers costs and, hence, should be an attractive feature for future DMT products.

Comparing Data Migration Tools

Based on the products' user manuals and our experience with the available trial version of the products, we used the five criteria discussed above to compare the commercial DMT tools mentioned in this paper. The results are shown in Table 1.

Table 1. Database migration tool (DMT) products compared according to the criteria discussed

DMT Product	Types of Databases the DMT Supports	User Interface Configurability, Maintainability, and Reusability	Support for Data Integrity Checking	Customization Adaptability of the DMT	Data Correctness Verification
Altova (MapForce)	SQL Server, DB2, Oracle, MySQL, PostgreSQL, Microsoft Access	Good user interface with a GUI that allows users to define database tree and create migration data flows graphically.	Allows users to set primary and foreign keys during data migration but does not support integrity check prior to migration.	Allows users to create functions based on the product's own function library, but no outside function library is allowed.	Does not provide data correctness verification.
Astera (Centerprise Data Integrator 5)	SQL Server, Oracle, DB2, Microsoft Access, Sybase, MySQL	Good user interface with a GUI that allows users to define data flows graphically.	Allows users to define data integrity check rules before migration.	Users can only use the predefined functions in the product's function library.	Allows users to create profile to obtain data migration statistics, but manual comparisons are required.
SwisSQL (Data Migration Tool 6.5)	Oracle, DB2, SQL Server, Sybase, MySQL, PostgreSQL	Poor user interface with no GUI to define data flows. Users need to follow a step-by-step wizard to perform a data migration.	Provides a database migration verifier to verify the integrity of the data being migrated.	Customization is not allowed.	Does not provide data correctness verification.
DBConvert	Oracle, DB2, SQLite, MySQL, PostgreSQL, Microsoft Access, FoxPro	Poor user interface with no GUI to define data flows. Users need to follow a step-by-step wizard to perform a data migration.	Allows users to set primary and foreign keys during data migration but does not support integrity check prior to migration.	Customization is not allowed.	Does not provide data correctness verification.
Oracle (SQL Developer 3.1 Data Dump)	Oracle	Provides a simple GUI user interface, with command line and API supports.	Does not support integrity check prior to migration.	Customization is not allowed.	Does not provide data correctness verification.
SQL Server 2008 (Import and Export)	SQL Server, MySQL, Oracle (need .Net Framework Data Provider)	Provides a simple GUI user interface, with command line and API supports.	Does not support integrity check prior to migration.	Customization is limited.	Does not provide data correctness verification.

API = application programming interface; GUI = graphical user interface.

Table 1 shows that although some criteria this paper proposes may sound trivial, they are often not implemented in DMT products. Thoroughly evaluating the product according to our proposed criteria will have a significant effect on the success of a data migration project. Data migration project managers need to perform a comprehensive check before selecting a DMT product.

Future Work

While the criteria outlined above are adequate for the complex project that we have developed for NOAA, the complexity of a general extract, transform, and load (ETL) system may go beyond what these criteria can evaluate. For example, in our application, we have not dealt with complex data like image, audio, and video files. Investigations on how to evaluate the migration of such data will be helpful and are desired. In addition, some other criteria, such as data migration performance and cost, can be important to project managers and DMT developers. Further studies in these areas will help develop a complete set of data migration evaluation standards.

Conclusion

Database migration is a common task that most database administrators need to tackle. Knowing how to evaluate and choose the right DMT can be vital to the fate of a software project that might directly contribute to the success of a business operation. Yet there are few guidelines in how to evaluate the usefulness and effectiveness of a general DMT. In this paper, we illustrate five criteria that can serve as standards for current and future DMT products.

The first and foremost criterion relates to the types of database the DMT supports. If a DMT does not support the databases from or to which users need to perform the migration, the DMT will not do a good job regardless of how perfectly other functions are performed.

The second criterion addresses how the database transfer is configured through the DMT's interface. This configuration will determine whether the data transfer can be executed repeatedly. An XML configuration specification is superior to GUI-only interfaces in most cases.

The third criterion of a good DMT product is whether it checks database integrity before executing a data transfer. Quite often a database migration operation fails because of database integrity violations among the data. Unfortunately, not all DMT products available today provide such a check. A good DMT product should enforce this check rather than functioning in a trial-and-error manner.

The fourth criterion is how well the DMT incorporates customized data transfer requirements. Does the DMT provide a mechanism that allows data to be modified prior to or during the data transfer? Does the DMT allow users to specify the sequence of the data transfer? This kind of customization is usually necessary for a successful data transfer, especially when a large amount of data is involved.

The fifth criterion is ensuring the correctness of a completed database migration operation. Because none of the current DMTs contain a mechanism to compare the source and destination databases to ensure the correctness of a database migration, this feature is a wish list item for future DMT products.

Our experience reveals that these criteria will serve users well in evaluating DMT products and helping design future tools.

References

1. Mullins CS. Database administration: the complete guide to practices and procedures. Reading (MA): Addison-Wesley Professional; 2002.
2. Kimball R, Caserta J. The data warehouse ETL toolkit. Indianapolis (IN): Wiley Publishing; 2004.
3. Shepherd JB. Data migration strategies. Information Management Magazine [Internet]. 1999 Jun. Available from: http://www.information-management.com/issues/19990601/996-1.html
4. Brown CV, Topi H. IS management handbook. 8th ed. Boca Raton (FL): Auerbach Publications; 2003.
5. Altova. Altova MapForce 2012. Database conversion/migration tool. 2012. Available from: http://www.altova.com/download-database-conversion.html?gclid=COjhg--AqqECFRdvswodDnYMGA
6. Astera. Centerprise Data Integrator 5.0. 2010. Available from: http://www.astera.com/Centerprise/?gclid=CLGL_8nOrKECFQqF7QodSxppHw
7. DBConvert. Smart database conversion tools. No date. Available from: http://dbconvert.com
8. SwisSQL. SwisSQL–Data Migration Tool 6.5. 2009. Available from: http://www.swissql.com/products/datamigration/data-migration.html
9. ITL Education Solutions Limited. Introduction to Database Systems. Pearson Education India; 2008.
10. Beynon-Davies P. Database systems. 3rd ed. Basingstoke, UK: Palgrave; 2004.
11. Date CJ. The relational database dictionary. Sebastopol (CA): O'Reilly Media; 2006.
12. Callan S. Database migration—a planned approach. Database Journal [Internet]. 2006 Feb 22. Available from: http://www.databasejournal.com/features/oracle/article.php/3586516/Database-Migration—A-Planned-Approach.htm

Acknowledgments

We owe Michael Quach of the Pacific Islands Fisheries Science Center of the National Oceanic and Atmospheric Administration a debt of gratitude for his wholehearted support during the implementation of our data migration tool. We would also like to thank Craig Hollingsworth and Sharon Barrell of RTI International for making substantial editorial contributions to this paper.

RTI International is an independent, nonprofit research organization dedicated to improving the human condition by turning knowledge into practice. RTI offers innovative research and technical solutions to governments and businesses worldwide in the areas of health and pharmaceuticals, education and training, surveys and statistics, advanced technology, international development, economic and social policy, energy and the environment, and laboratory and chemistry services.

The RTI Press complements traditional publication outlets by providing another way for RTI researchers to disseminate the knowledge they generate. This PDF document is offered as a public service of RTI International.

www.rti.org/rtipress **RTI Press publication OP-0008-1210**