



Relate

Data Maintenance for NonStop Servers

Product Introduction Guide

Ascert

taking systems
to the edge

This document contains proprietary and confidential information belonging to Ascertain, LLC. This information may not be used, duplicated, or disclosed without the prior written consent of Ascertain. Do not remove, deface or otherwise obscure this notice.

Whilst Ascertain makes every effort to ensure the accuracy of the information contained within this document, no warranty, implied or otherwise, can be made as to its contents.

The information contained is subject to regular change and revision. Therefore, before using this publication please check with Ascertain, or its representative, that this edition is applicable and current.

Copyright © 2012 by Ascertain, LLC. All rights reserved.

For further information:

Ascertain, LLC
759 Bridgeway
Sausalito, CA 94965

Telephone: (415) 339-8500
Email: info@ascertain.com
Web: <http://www.ascertain.com>

Table of Contents

CHAPTER ONE: INTRODUCTION	1
WHAT IS RELATE?	1
WHY BE RELATED?	1
CHAPTER TWO: RELATE IN USE	2
INTRODUCTION	2
SUPPORTING DEVELOPMENT DEPARTMENTS	2
REGRESSION TESTING	2
DATABASE CONVERSION	2
RECONCILING SOURCE AND PRODUCTION PROGRAM LIBRARIES	3
RESETTING THE PRODUCTION DATABASE	3
CHAPTER THREE: A QUICK TOUR	4
INTRODUCTION	4
GETTING STARTED	4
APPLYING RECORD LAYOUTS	5
SPECIFYING FIELDS TO DISPLAY	7
COMPARING FILES	8
VARIABLY FORMATTED RECORDS	9
CHAPTER FOUR: IN CONCLUSION	11
POWER BRINGS RESPONSIBILITY	11
WHERE NEXT	11

Chapter One

INTRODUCTION

What is RELATE?

RELATE is a tool that assists Information Technology Departments with the development and maintenance of databases and database access programs within an HP NonStop Integrity environment.

RELATE provides intelligent manipulation and comparison functions on all Enscribe files on your server. By accessing the definition of your files within your DDL dictionary or COBOL copy library, RELATE can:

- Compare sets of records from differing databases down to the field level, using a number of strategies, even where the layouts may be different.
- Display the contents of nominated fields within selected records, intelligently applying group and field redefinitions.
- Globally modify individual fields of selected records within your database.
- Globally delete selected records from the database, and insert new records
- Copy selected records between databases, optionally re-sizing and reformatting the fields within the records.
- Convert databases as part of a data migration project.

Why be RELATED?

The need for a product such as RELATE was recognized following several years of marketing and supporting testing and simulation products in the NonStop marketplace.

Conventional testing tools tend only to look at the application under test from the perspective of the tool. These test tools generally simulate a user sitting at a terminal and then verify that the data displayed on the terminal is correct, but they cannot verify that the data entered by the user has the correct impact on the application's database. Furthermore, any such analysis of the tests, if performed at all, will typically be based on the manual examination of database reports, or will involve the creation of custom code.

With RELATE, the user is able to verify the application's affect on the database by pinpointing problems in a timely manner without writing custom code. Additionally, RELATE automates the analysis of the various tests, thereby removing the tedious and error prone nature of manual test review

Some development shops may currently use file comparison products, but the tools on the market today appear to be derived from text-based comparison utilities. As such, these tools tend to have the following characteristics:

- They only compare complete files;
- They only compare records sequentially;
- They know nothing about the record structure; and
- They compare whole records but they cannot ignore parts of the record, like timestamps; and
- They may reside on another host platform, resulting in the need to transfer data between systems.

This severely limits their usefulness, and will still necessitate the use of costly manual intervention.

RELATE is not subject to these limitations because it uses standard DDL or COBOL record descriptions to perform data-specific comparison algorithms that can identify data differences down to the field level. It is therefore, an excellent alternative to the standard file comparison products.]

Chapter Two

RELATE IN USE

Introduction

RELATE has relevance across all industries, and throughout IT departments.

Supporting Development Departments

In development departments, RELATE can be used to populate a test database. Following a test run, the user can reset the modified records back to the initial settings without having to purge the entire file. This is particularly useful when several people have to share the files, and it is not feasible to shut down the application whilst the data is reset.

As manual or automated unit and integration testing is performed, there will be changes made to the underlying database. Whilst Ascert's VersaTest product can generate and process test scripts at the transaction processing level, RELATE is the first tool available to verify the underlying database. Traditionally, any verification of the updates to the database would be performed either manually, or by writing one-off inquiry programs. Both of these methods are time consuming, prone to error, and unless considerable effort is expended, will not provide verification results in a straightforward manner. RELATE, on the other hand, can be used to automatically analyze, down to the field level, the modifications made to the application database. In its use of verifying a post-test database against a base-line database, RELATE will provide the development staff with immediate results that are both comprehensive and accurate.

RELATE's command-line operation also makes the product eminently suitable for automated execution from within managed test environments, such as VersaTest Automator. This enables database results checking to be performed as part of every test case, enabling rapid detection of emerging problems.

Regression Testing

Regression testing of new releases or amended modules within an application is necessary to ensure that the functionality of the new release conforms to specifications, and that any database changes are consistent. RELATE provides the Quality Assurance engineers with the ability to automatically compare the before and after databases and to highlight any unexpected changes, thus allowing them the opportunity to verify the changes or correct any underlying problems.

Additionally, as developers' reliance on third party application software increases, so too does their exposure to immature and poorly tested software releases. By utilizing RELATE in the certification of third party software, developers can ensure that any new database changes introduced will conform to specifications.

Database Conversion

As users develop and re-develop their applications utilizing the latest technology, the problem of converting and reconfirming legacy databases becomes more challenging.

RELATE provides an ideal solution to this by being able to copy records to new files, whilst simultaneously re-sizing and re-formatting the data fields within the records. Database conversion, like all functionalities within RELATE, is based around direct access to the underlying DDL. In order to perform the conversion, RELATE will access the DDL definition of the old and new formats of the record. A simple 'COPY CORRESPONDING' command is all that is then required by the user to start the conversion. RELATE even

allows the operator to override the default correspondence to allow for the case where field names change between records.

Since RELATE allows the user to specify the manner in which dates and times are stored within DDL fields, RELATE can also perform date/time conversions and aging. RELATE supports a virtually unlimited number of date and time storage formats, and even allows the operator to specify a year to use as the start of a century window when processing 2-digit years.

RELATE can also assist with the verification of converted databases, regardless of how they were converted. This is possible because RELATE can compare records of differing formats, which simplifies the analysis of dissimilar data.

Reconciling Source and Production Program Libraries

RELATE can compare both text and binary data files, which means it can be used to verify source program versions. This is important because it identifies differences in the source code, and it can be used to ensure that the latest version is being used in production.

Resetting the Production Database

In technical support departments, RELATE can be used to selectively display file contents and to globally update files automatically that may previously have required custom programs. It can also be effective in managing configuration changes between systems and between application releases.

When a production database fails due to operating system problems, application errors or hardware faults, it is critical that the organization reset the database as quickly as possible. Prior to the introduction of RELATE, the only options available to the operations staff were to develop one-off programs to reset the database or to restore the database from the last security copy.

Since creating new programs is error prone and carries with it the possibility of compounding the database failure by introducing new errors, RELATE provides the operations staff with a proven method of accurately restoring failed databases to a predetermined baseline. This means that the production environment is made available to the users in a timely and accurate fashion, and the risk associated with restoring the latest security copy and potentially losing any new updates is reduced.

Chapter Three

A QUICK TOUR

Introduction

This chapter provides, by way of example, a quick introduction to RELATE, which will enable readers to feel comfortable with the basic operation of the product.

Getting Started

RELATE is a standalone server-based program, run directly from the TACL command line. It provides intelligent manipulation and comparison of Enscribe files in an HP NonStop Integrity environment.

RELATE uses a conversational command interface, utilizing syntax with which NonStop users will already be familiar. Users can enter RELATE commands on the startup command line; interactively in response to RELATE prompts; or programmatically within a command file, which facilitates the integration of RELATE into users' own environments.

In order to use RELATE, the operator first 'OPEN's a file he wishes to work on - the *current* file. He can then immediately perform various actions on either the entire opened file, or on a subset of records in the file by qualifying the actions with a 'WHERE' clause, similar to Enform / SQL inquiries. Available actions include the following:

- List record contents.
- Update / delete existing records.
- Copy existing records between files.
- Insert new records
- Compare records from different files

An example of this follows (note that the user input has been underlined):

```
TACL 1 > RUN RELATE
RELATE (Rev 2.X - 15MAR201X)
Relate> HELP

The following commands are available:
CLEAR          COMPARE          COPY
COUNT        DELETE           EXIT
FC            HELP           INFO
INSERT        LIST           OBEY
OPEN          OUT           SET
SHOW         UPDATE          VOLUME
X

Relate> HELP OPEN
Syntax: OPEN [<file> [<open-options>]]
```

Note: Opens the Enscribe file specified, and makes it the current file. If no file is specified, then the name of the file currently open will be displayed, together with its open mode.

By default, when a file is opened it will be opened for shared

```

        read-only access. This can be overridden using the <open-options>.
Relate> HELP <OPEN-OPTIONS>
Syntax: <open-options> ::= { FORUPDATE }
                          { FORXUPDATE }

Relate> OPEN EGDDL
Relate> OPEN
Current file: $DATA1.REL8ISV.EGDDL (Read-only)
Relate> LIST WHERE LINENUM < 5
  1. : * This DDL source file is used to generate the dictionary
  2. : * and files used in the examples contained within the
  3. : * RELATE User Guide
  4. : *
  5. : ?DICTIN !, COMMENTS
5 Records listed
Relate> OPEN EGFKS FORUPDATE
Relate> SET PRESENTATION HEX
Relate> DELETE
PKEY:...      39 32 30 30 30 31 63 00          | 920001c.
Delete (Y/N/A/Q)? Y
PKEY:...      39 32 30 30 30 32 63 00          | 920002c.
Delete (Y/N/A/Q)? A
2 Records deleted
Relate> EXIT
TACL 2 >

```

In this example, the user first requested some help information. Note that the list of commands displayed includes the standard set of NonStop subsystem commands (for example ‘OPEN’, ‘SHOW’, ‘FC’, ‘EXIT’, etc.). Following display of more detailed help text regarding the ‘OPEN’ command, the edit-type file EGDDL was opened for read-only access, and the first 5 lines ‘LIST’ed. The default display options were in effect, therefore all of the first five lines of the file were displayed in ASCII.

A new key-sequenced file EGFKS was then opened for update access. The user wanted to interactively delete all of the records in the file. Since he would be prompted as to whether to delete the individual records by displaying (in this default case) the primary key fields, he set the default method of presentation to be ‘HEX’ to allow binary data to be displayed.

The user then deleted all of the remaining records. He did this executing the ‘DELETE’ command to perform the interactive deletions, responded ‘Y’ to RELATE’s prompt after the primary key of the first record was displayed, then responded ‘A’ to the prompt after the second record was displayed.

Applying Record Layouts

RELATE is most effective where the file on which the user wishes to operate has its definition stored in a DDL dictionary. RELATE is able to access that dictionary in order to ascertain the format of the current file. Users can then reference group or field elements in any of the actions they wish to perform. Additionally, comparisons between files can now be made at the field level, rather than the record level, meaning that discrepancies can be more readily identified.

An example of this follows:

```

Relate> CLEAR ALL
Relate> OPEN EGFKS FORUPDATE
Relate> SET RECORD FORMAT DDL EGFKS-REC
Relate> INFO RECORD
01 EGFKS-REC.
  02 PKEY.
    03 EMPLOYEE-ID.
      04 JOIN-YEAR          PIC 9(2).
      04 SEQ-NUM           PIC 9(4).
    03 COMPANY-CODE      BINARY 16 UNSIGNED.
  02 NAME                PIC X(16).
  02 DEPT                 PIC X(8).
  02 CHARACTERISTICS.
    03 FULL-TIME          PIC X.
    03 PART-TIME-INFO.

```

```

    04 HOURS-PER-WEEK          PIC 9(2).
    03 FULL-TIME-INFO          REDEFINES PART-TIME-INFO.
    04 HOLIDAY-ENTITLEMENT     PIC 9(2).
    02 FILLER                   PIC X.
    02 LAST-UPDATE             BINARY 64.
    02 NUM-SKILLS              BINARY 16 UNSIGNED.
    02 SKILLS-MATRIX           PIC X(8)
                               OCCURS UPTO 8 TIMES
                               DEPENDING ON NUM-SKILLS.
Relate> INSERT JOIN-YEAR = 92, SEQ-NUM = 1, COMPANY-CODE=99, NAME= "FRED", FULL-TIME = "Y"
Relate> INSERT JOIN-YEAR = 92, SEQ-NUM = 2, COMPANY-CODE=99, NAME= "BILL", FULL-TIME = "N"
Relate> UPDATE HOURS-PER-WEEK = 5 WHERE NAME BEGINS-WITH "BILL"
1 Records updated
Relate> LIST FIELDS JOIN-YEAR, SEQ-NUM, NAME DETAIL
PKEY:...   Join-year = 92,  Seq-num = 1,  Company-code = 99
  Data:    JOIN-YEAR =          92
           SEQ-NUM =           1
           NAME =              "FRED "
PKEY:...   Join-year = 92,  Seq-num = 2,  Company-code = 99
  Data:    JOIN-YEAR =          92
           SEQ-NUM =           2
           NAME =              "BILL "
2 Records listed

```

In this example, the user first ‘CLEARED ALL’ of the current environment settings back to their default values and then opened the file EGFKS, as in the previous example. He then specified that the format of the file be obtained by reading the definition for ‘EGFKS-REC’ from the DDL dictionary on the current subvolume.

Having associated a format with the current file, there was no need to use the ‘SET PRESENTATION HEX’ command (as in the previous example), because when a record is displayed, all fields are output in a format consistent with their type. The actual retrieved format was verified by the user entering the ‘INFO RECORD’ command.

Two records were then inserted by the user, with default values being entered for unsupplied fields. The first record was then updated with some new field values.

Finally the user displayed the contents of some of the fields in all of the records within the file. Note how the output format of each field displayed was dependent on the field type.

Note also that some fields were displayed twice by the ‘LIST’ command. This is because by default, RELATE will output primary key values for all records displayed which is controlled by the setting ‘INDEX’. In the case of key-sequenced files such as EGFKS, the primary key is formed from actual field contents, so these fields can be automatically be displayed.

The user then continued with a different file, EGFKSB, as follows:

```

Relate> OPEN EGFKSB FORUPDATE
Relate> SET RECORD FORMAT DDL EGFKS-REC
Relate> DELETE WHERE JOIN-YEAR > 90
PKEY:...   Join-year = 92,  Seq-num = 1,  Company-code = 99
Delete (Y/N/A/Q)? A
2 Records deleted

```

Here the user, though working on a different file, could still use the same format that was used by EGFKS. This format had to be re-specified to RELATE, however, since all record settings are cleared whenever a new file is opened.

Lastly, the user requested to interactively delete a selection of records from this second file. This time, when prompted with the first record's primary key, he opted to delete ‘ALL’ of the records. Note though, that this time RELATE displayed the actual component fields of the primary key. The next section details how the user can further customize this output so that fields appropriate to his own decision-making process are shown.

Specifying Fields to Display

In the previous example, the user explicitly specified the fields where the contents were to be displayed. However, by default, the 'LIST' command (among others) displays what are known as *summary* fields. These are fields in the record format which have been 'tagged' with a summary *attribute*. This is achieved through the 'SET RECORD SUMMARY' command.

An example of this follows:

```
Relate> CLEAR ALL
Relate> OPEN EGFKS FORUPDATE
Relate> SET RECORD FORMAT DDL EGFKS-REC
Relate> SET RECORD SUMMARY (JOIN-YEAR, SEQ-NUM, NAME)
Relate> INFO RECORD , PKEY
  02 PKEY.
    03 EMPLOYEE-ID.
      04 JOIN-YEAR                               PIC 9(2)
                                               SUMMARY.
    04 SEQ-NUM                                   PIC 9(4)
                                               SUMMARY.
    03 COMPANY-CODE                             BINARY 16 UNSIGNED.
Relate> LIST DETAIL
=====
PKEY:...   Join-year = 92,  Seq-num = 1,  Company-code = 99
  Data:    01 Egfks-rec.
           02 Pkey.
             03 Employee-id.
               04 Join-year =                92
               04 Seq-num =                  1
           02 Name =                          "FRED "
=====
PKEY:...   Join-year = 92,  Seq-num = 2,  Company-code = 99
  Data:    01 Egfks-rec.
           02 Pkey.
             03 Employee-id.
               04 Join-year =                92
               04 Seq-num =                  2
           02 Name =                          "BILL "
=====
2 Records listed
Relate>
Relate> SET RECORD SUMMARY *
Relate> LIST
=====
PKEY:...   Join-year = 92,  Seq-num = 1,  Company-code = 99
  Data:    Join-year = 92,  Seq-num = 1,  Company-code = 99
           Name = "FRED           ", Dept = " ", Full-time = "Y"
           Hours-per-week = 0, Last-update = 0, Num-skills = 0
=====
PKEY:...   Join-year = 92,  Seq-num = 2,  Company-code = 99
  Data:    Join-year = 92,  Seq-num = 2,  Company-code = 99
           Name = "BILL           ", Dept = " ", Full-time = "N"
           Hours-per-week = 5, Last-update = 0, Num-skills = 0
=====
2 Records listed
```

In this example, the user opened and set the record format as before. This time, however, he set the summary attribute on the fields where the contents were previously explicitly displayed. This was partially verified by the user in his next command: 'INFO RECORD, PKEY'. This command resulted in the display of the record format information for the record subgroup 'PKEY', which showed the summary attribute set on two of the fields.

Next, the user entered the 'LIST DETAIL' command. This time, since no field list was specified, the default action taken was to display all 'SUMMARY' fields. Note that this time the 'DETAIL' option resulted in the full field hierarchy being displayed. Contrast this with the next 'LIST' command entered.

In this case the user nominated all fields to be summary fields using the ‘SET RECORD SUMMARY *’ syntax, and then listed all of the fields in all of the records. Because no ‘DETAIL’ option was given, the display of the field contents was compressed into as few screen lines as possible.

If you refer back to the record format of the EGFKS file (obtained through the earlier ‘INFO RECORD’ command), you will notice that it contained a group ‘FULL-TIME-INFO’, which was a *redefinition* of the group ‘PART-TIME-INFO’. In the last example you may also have noticed that RELATE did not display any of the fields contained in ‘FULL-TIME-INFO’. Instead, RELATE assumed that the first definition it encountered was the correct redefinition to apply to the data. RELATE can, in fact, be told how to determine which of many redefinitions to apply to the record data as it is being processed. This knowledge (which is achieved using the ‘SET RECORD REDEFINITION’ command) permits the intelligent display of data in its correct format based on data content.

Comparing Files

RELATE can compare any two files regardless of whether or not the formats have been defined. Obviously more meaningful information can be displayed if the formats are defined, however, the ideal scenario would be to display only the fields that differ between the files.

An example of this follows:

```

Relate> CLEAR ALL
Relate> OPEN EGFKS FORUPDATE
Relate> SET RECORD FORMAT DDL EGFKS-REC
Relate>
Relate> COPY TO EGFKSB
2 Records copied
Relate> UPDATE HOLIDAY-ENTITLEMENT = 10, DEPT = "ADMIN" WHERE NAME BEGINS-WITH "FRED"
1 Records updated
Relate> COMPARE AGAINST EGFKSB
RELATE FILE COMPARISON
  Current File:      $DATA1.REL8ISV.EGFKS
  Baseline File:    $DATA1.REL8ISV.EGFKSB
  Time of Comparison: 18-OCT-201X, 13:14:20.250
===== Mismatched Record =
PKEY:      Join-year = 92,  Seq-num = 1,  Company-code = 99
- - - - - (Baseline Version)-
Diff:      Dept = "      ",  Hours-per-week = 0
+ + + + + (Current Version) +
Diff:      Dept = "ADMIN ",  Hours-per-week = 10
=====
Comparison Results:-
  Current File:.....$DATA1.REL8ISV.EGFKS
  Baseline File:.....$DATA1.REL8ISV.EGFKSB
  Synchronization:.....KEY

  Records compared from Baseline file...2
  Records compared from Current file...2
  Extra records in Current file.....0
  Records missing from Current file....0
  Mismatched records.....1
  Total records differing .....1

```

In this example, the user re-opened the original key-sequenced file EGKFS and copied all of the records within it to the *baseline* file EGFKSB.

The user then modified some of the fields in one of the records in the current file before comparing it against the *baseline* file. Note that RELATE was able to pick out exactly those fields that were modified because of the types of files that were compared and the comparison strategy that RELATE chose to adopt when comparing the files.

The user can specify various comparison settings in order to improve the accuracy of the comparison process. These settings are made based on the type of data to be compared. For instance, when an application creates log files, the order in which the log entries are created may differ between test runs. RELATE can be

instructed to reconcile these "out-of-order" records, thus eliminating the need for the operator to perform the same task manually. RELATE comparisons can also be operated from within TACL routines. In this situation, RELATE will return a completion code back to its creating process, enabling the automated processing and coordination of test and comparison activities.

Variably Formatted Records

Sometimes records do not follow a strict fixed format specifiable by a DDL or COBOL data definition. Instead the records consist of fields (or groups of fields) that are either variable in length, or perhaps not even present in the record at all. Either situation will impact the placement of the later fields in the record.

Such data formats are common in data-communication applications, where it is desirable to reduce the physical amount of data transferred in order to optimize transmission speeds and reduce transmission costs. A common example of this is the financial services industry ISO 8583 standard, which uses a bit-map to indicate the presence of absence of data elements.

Variable format data is also found in applications with a large install-base. In such applications, data required by one installation may not be required by another. However the costs of storing and processing the unrequired data by the second installation may be huge. If the data is not present, the problem goes away. ACI's BASE24 product is an example of an application that has reduced the amount of data to be processed through the use of *segmented records*; these segmented records use message segments of variable length and identify themselves using an ID field. Both the ID and length indicators are placed at a fixed position (relative to the start of the segment) to enable them to be processed (or ignored) by all application modules.

Some product versions of RELATE support the use of variable-format records containing fields with the following characteristics:

- Conditionally present, based on one or more other field values
- Variable length, as controlled by a length indicator field
- Variable occurrences, as controlled by an index field
- Truncated fields, due to the removal of trailing spaces

The following example shows RELATE being used to compare two BASE24 files containing segmented data.

```
Relate> OPEN ACAFON

RELATE> SET RECORD FORMAT COBOBOL SEC RECDEF OF COBLIB
RELATE> SET FIELD ACCTCAF VARIABLE (PRESENT IF ACCTCAF.ID = 1, LENGTH IN ACCTCAF.SEG-LGTH.LGTH)
RELATE> SET FIELD PREAUTH VARIABLE (PRESENT IF PREAUTH.ID = 2, LENGTH IN PREAUTH.SEG-LGTH.LGTH)
RELATE> COMPARE AGAINST Q60BREL8.ACAFON

RELATE FILE COMPARISON
  Current File:      \VAN.$D2.Q60ATEST.ACAFON
  Baseline File:    \VAN.$D2.Q60BREL8.ACAFON
  Time of Comparison: 16-JUL-20x1, 14:26:41.750

===== Mismatched Record =
INDEX:      Num = "7411400132009902  ", Mbr-num = 1
----- (Baseline Version)-
Diff:      01 Caf.
           02 Seg1.
           03 Used-prd =                12
           03 Grp-prd.
           04 Ttl-wdl-prd =            53227
+++++ (Current Version) +
Diff:      01 Caf.
           02 Seg1.
           03 Used-prd =                10
           03 Grp-prd.
           04 Ttl-wdl-prd =            38227
=====

Comparison Results:-
```

```
Current File:.....\VAN.$D2.Q60ATEST.ACAFON  
Baseline File:.....\VAN.$D2.Q60BREL8.ACAFON  
Synchronization:.....KEY
```

```
Records compared from Baseline file...403  
Records compared from Current file...403  
Extra records in Current file.....0  
Records missing from Current file....0  
Mismatched records.....1  
Total records differing .....1
```

Chapter Four

IN CONCLUSION

Power Brings Responsibility

This guide has provided a quick introduction to the RELATE product. We've shown how its method of operation will be familiar to all NonStop Users, and how its use can benefit many different departments, from Development to Support. RELATE's proficiency in quickly and accurately determining field-level database changes allow the impact of change to be assessed and audited. RELATE's ability to make targeted changes throughout a file can get an errant application back into production in minimal time – but must be used judiciously. RELATE is a powerful tool that should be an essential part of every NonStop toolbox, but just keep in mind that with great power comes great responsibility.

Where Next

If you'd like to take a closer look at RELATE, then please contact us to arrange a free evaluation. This can be achieved via our website at the following URL:

<http://www.ascert.com/contact/request-information>