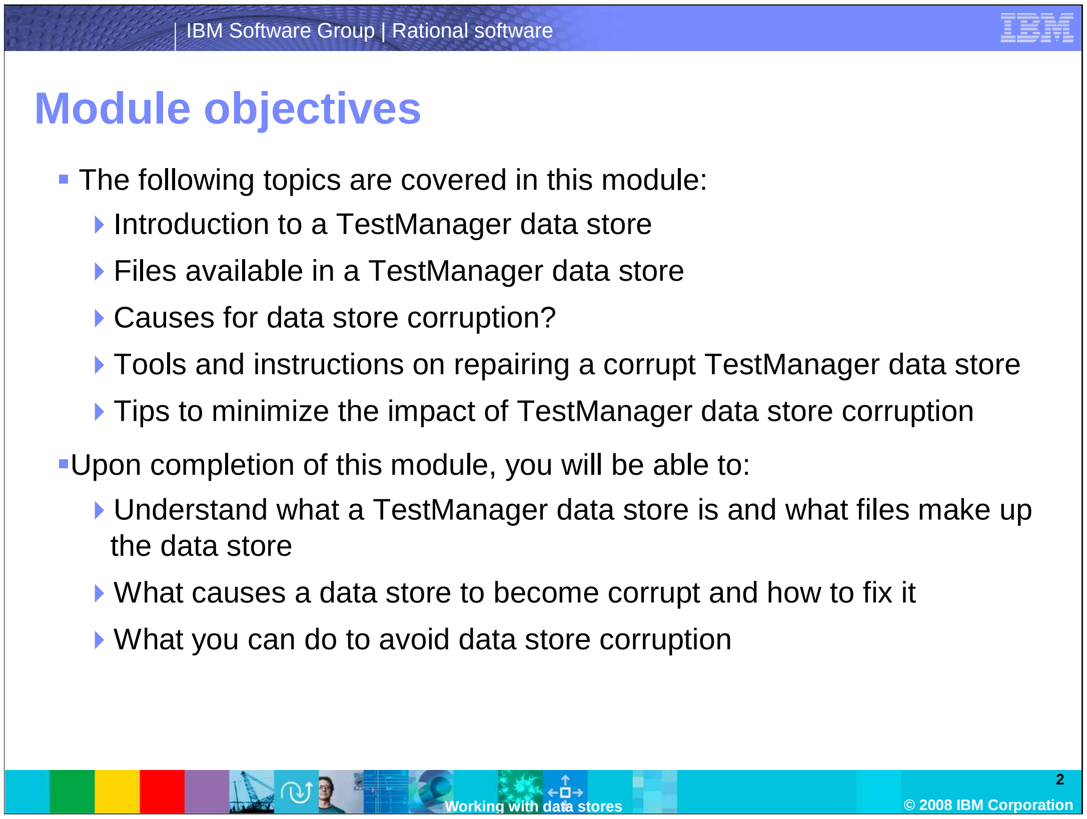
IBM Rational TestManager 7.0

Working with data stores

Updated July 21, 2008

This module will cover working with data stores for IBM Rational TestManager version 7 and above

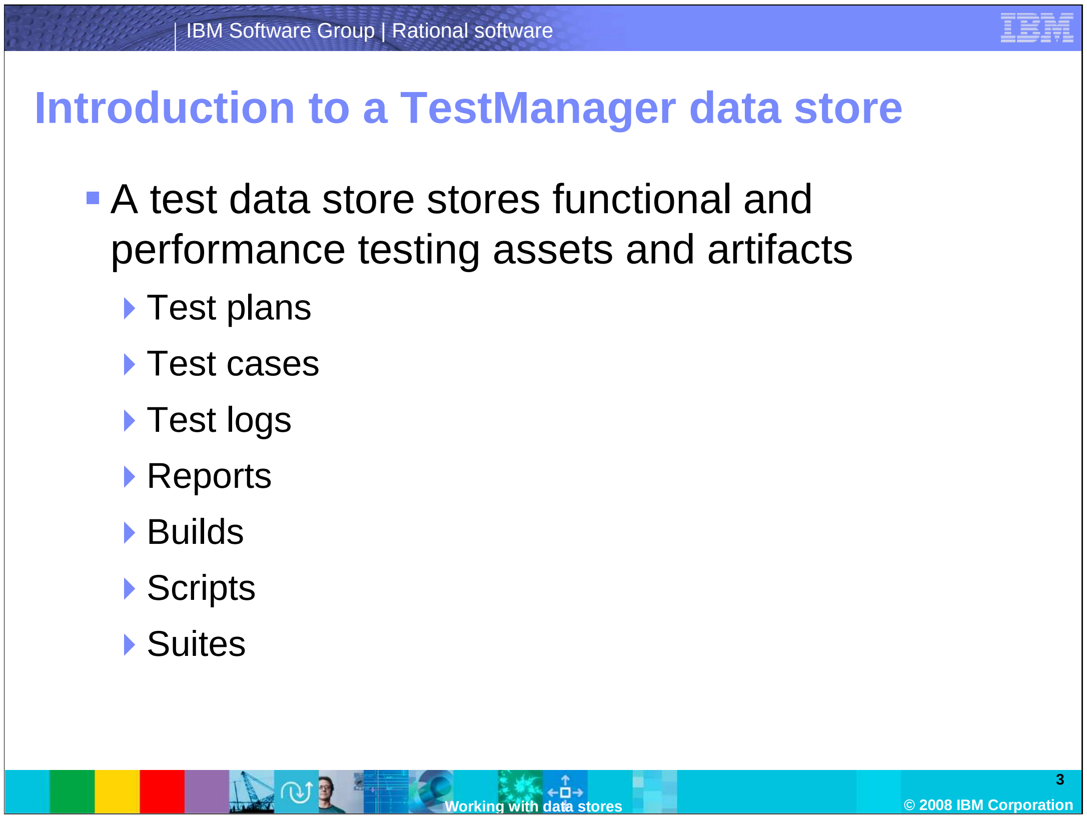


Module objectives

- The following topics are covered in this module:
  - Introduction to a TestManager data store
  - Files available in a TestManager data store
  - Causes for data store corruption?
  - Tools and instructions on repairing a corrupt TestManager data store
  - Tips to minimize the impact of TestManager data store corruption
- Upon completion of this module, you will be able to:
  - Understand what a TestManager data store is and what files make up the data store
  - What causes a data store to become corrupt and how to fix it
  - What you can do to avoid data store corruption

The following topics are covered in this module: Introduction to a TestManager data store, files available in a data store, causes for data store corruption, tools used to fix a data store, repairing a corrupt data store, and tips to minimize the impact of data store corruption.

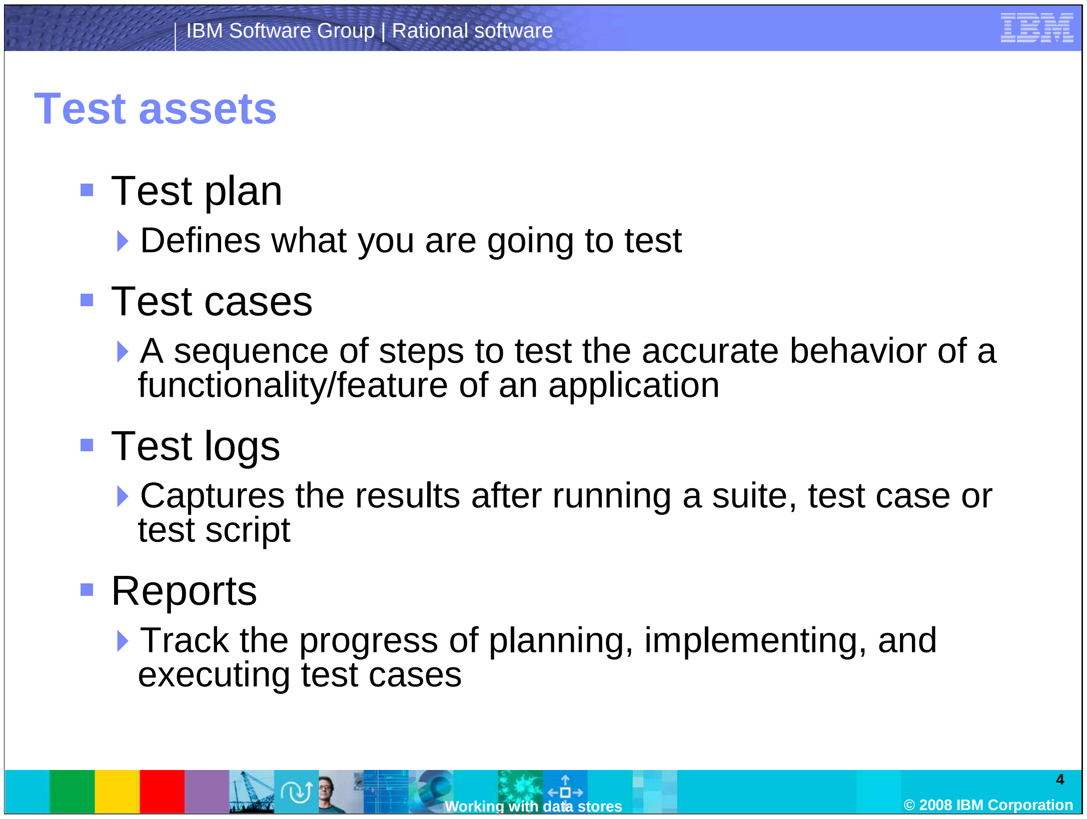
Upon completion of this module, you will be able to: understand what a TestManager data store is and what files make up the data store, what causes a data store to become corrupt and how to fix it, and what you can do to avoid data store corruption.



Introduction to a TestManager data store

- A test data store stores functional and performance testing assets and artifacts
  - Test plans
  - Test cases
  - Test logs
  - Reports
  - Builds
  - Scripts
  - Suites

A Test data store stores application testing assets, including: test plans, test cases, test logs, reports, builds, scripts and suites. The following slides will cover each asset in more detail.



Test assets

- Test plan
  - Defines what you are going to test
- Test cases
  - A sequence of steps to test the accurate behavior of a functionality/feature of an application
- Test logs
  - Captures the results after running a suite, test case or test script
- Reports
  - Track the progress of planning, implementing, and executing test cases

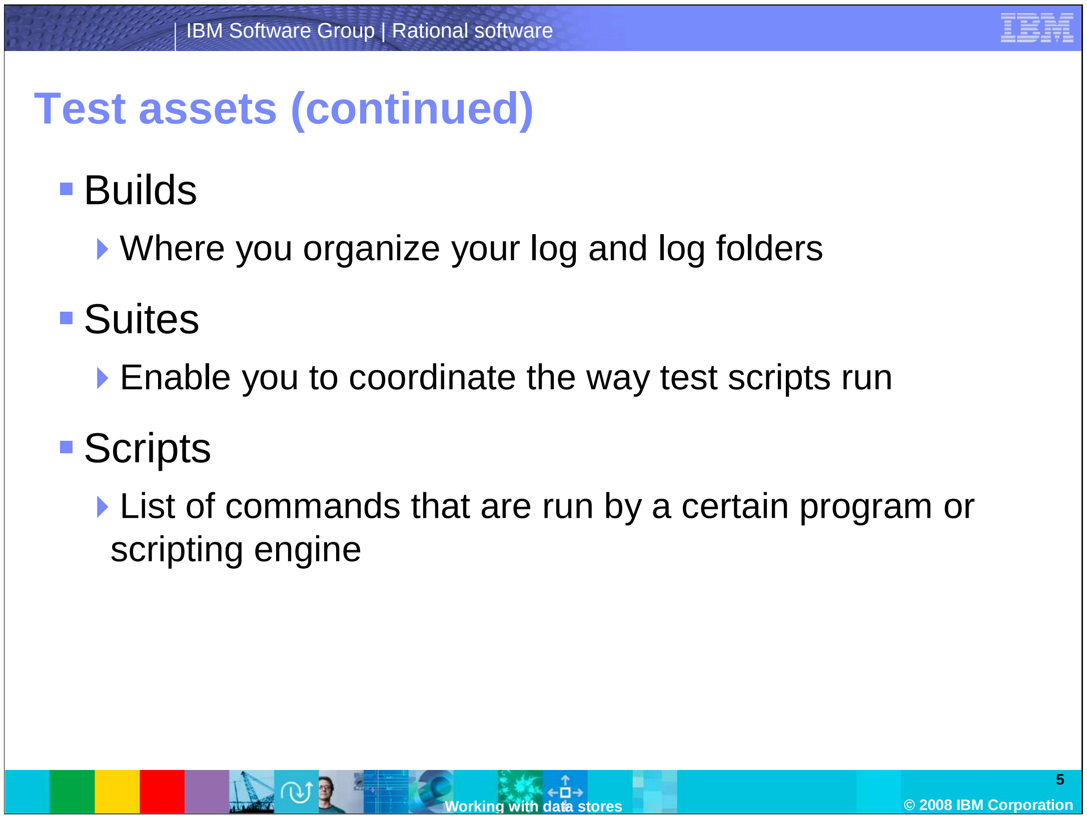
Test Assets are all of the resources that make up the testing project.

A **test plan** defines what you are going to test. Since test planning happens over time, you continually add things to the test plan.

A **test case** is a list of the conditions or issues of what the tester wants to test in a piece of software.   It is essentially a sequence of steps to test the accurate behavior of a functionality/feature of an application**.**

Test Logs is a repository to store passed and failed test cases. These logs capture the results after running a suite, test case, or test script.

Reports track the progress of planning, implementing, and executing test cases. They provide the results of how testing efforts are analyzed and communicated.



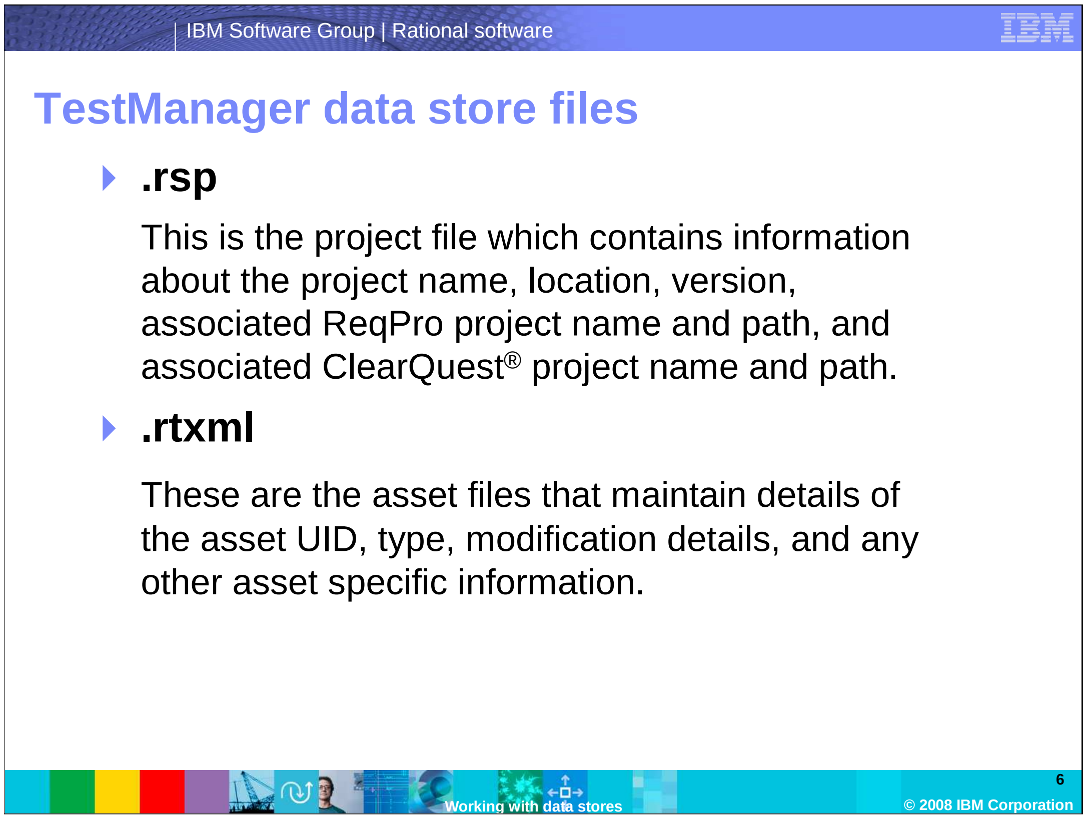
Test assets (continued)

- Builds
  - Where you organize your log and log folders
- Suites
  - Enable you to coordinate the way test scripts run
- Scripts
  - List of commands that are run by a certain program or scripting engine

Builds is a container for the logs and log folders.

Suites enable you to coordinate the way test scripts run. In functional testing, you can use suites to run test scripts concurrently on the computers that are available so that your tests can run more quickly. In performance testing, suites add workload to the server.

Scripts are a list of commands that are run by a certain program or scripting engine. In TestManager, when you run a test script it will create a temporary suite and actually run the suite.



TestManager data store files

- **.rsp**

  This is the project file which contains information about the project name, location, version, associated ReqPro project name and path, and associated ClearQuest® project name and path.

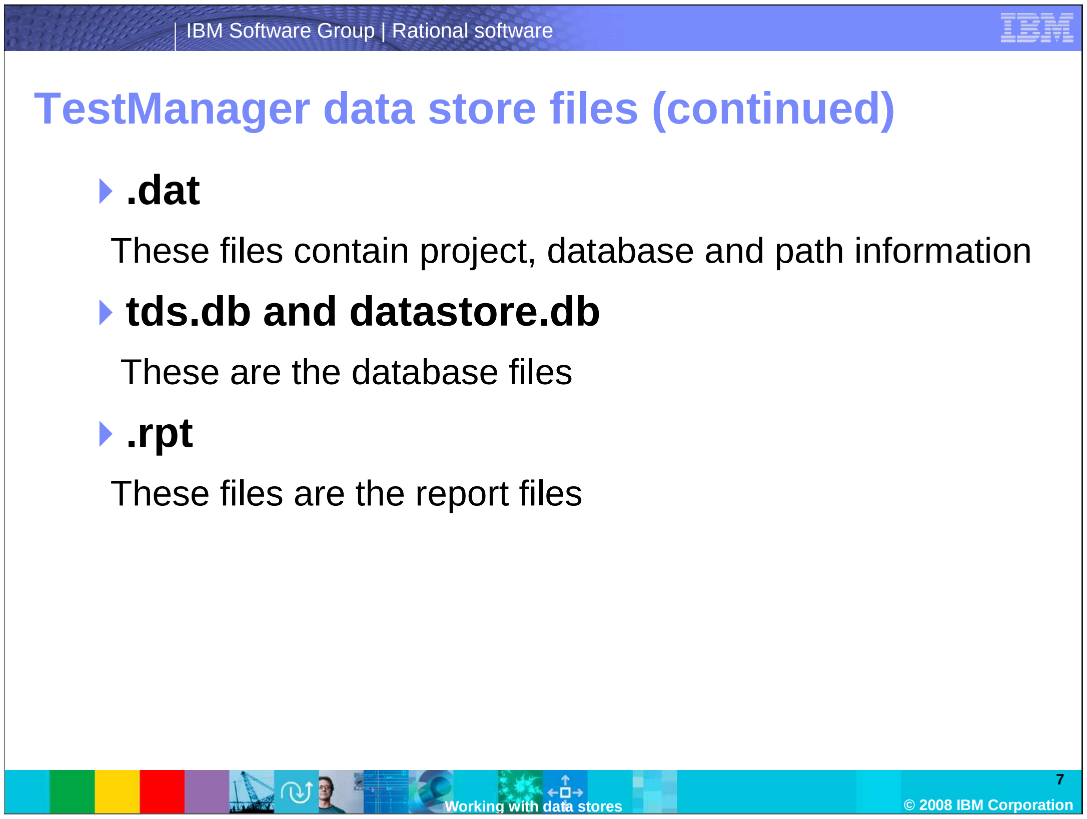
- **.rtxml**

  These are the asset files that maintain details of the asset UID, type, modification details, and any other asset specific information.

There are many important files in a TestManager data store. The following slides cover the files included in a TestManager data store, and explain the function of each file.

The main project file is the **.rsp** file. This is the project file which contains information about the project name, location, version, associated ReqPro project name and path, and the associated ClearQuest project name and path.

.rtxml files are the asset files that maintain details of the asset UID, type, modification details, and any other asset specific information. Every element in a TestManager project has an associated .rtxml file, and, wherever applicable, an associated folder that contains the child elements. For instance, a *test plan* has an .rtxml file and a folder. This folder contains the *test case* folder, and the associated .rtxml files.



TestManager data store files (continued)

- **.dat**

  These files contain project, database and path information
- **tds.db and datastore.db**

  These are the database files
- **.rpt**

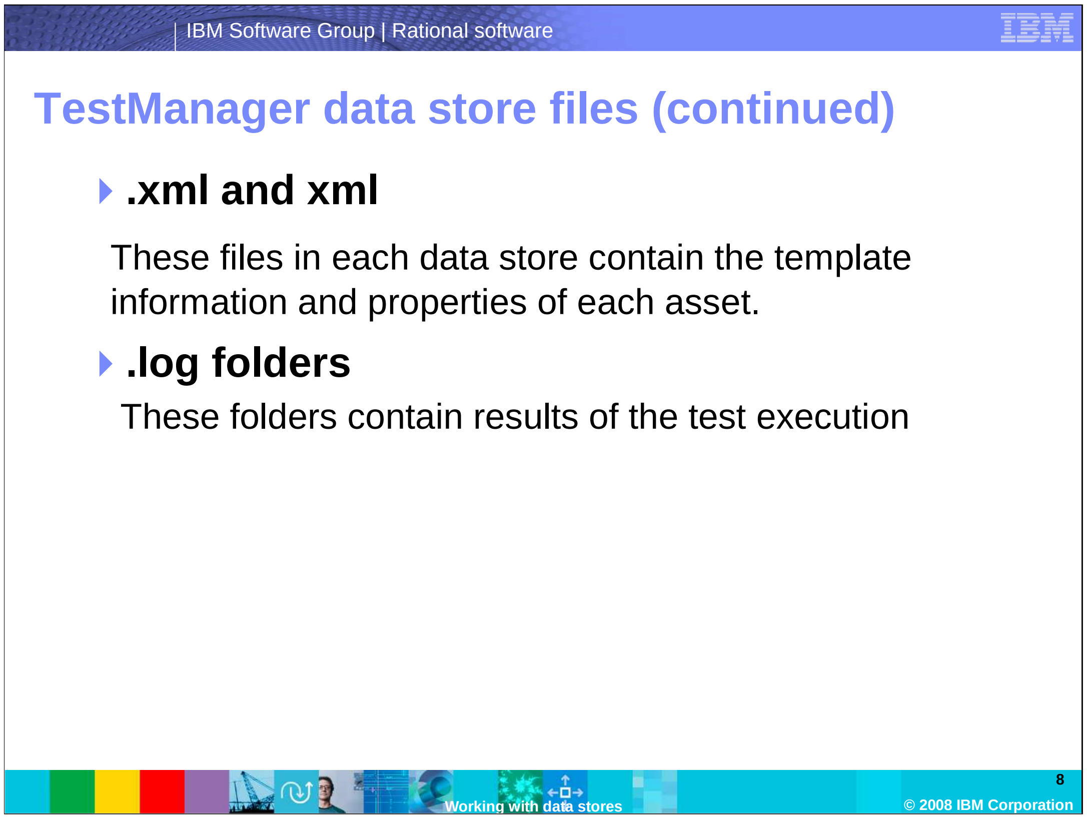
  These files are the report files

There is a **project.dat** file and a **datastore.dat** file in each of the three data store folders: TestUsersAndGroups, TestDatastore, and DefaultTestScriptDatastore.

Project.dat contains the project name and path information. The **Datastore.dat** file contains the database type, name and authentication details.

The Tds.db and datastore.db files are the databases that store the maps of project assets. These are used when the project uses SQLAnywhere as the backend. If you choose MS Access, then tds.mdb and datastore.mdb are used.

The .rpt files are the Crystal Report templates that are run from the Reporting/Analysis tab



TestManager data store files (continued)

- **.xml and xml**

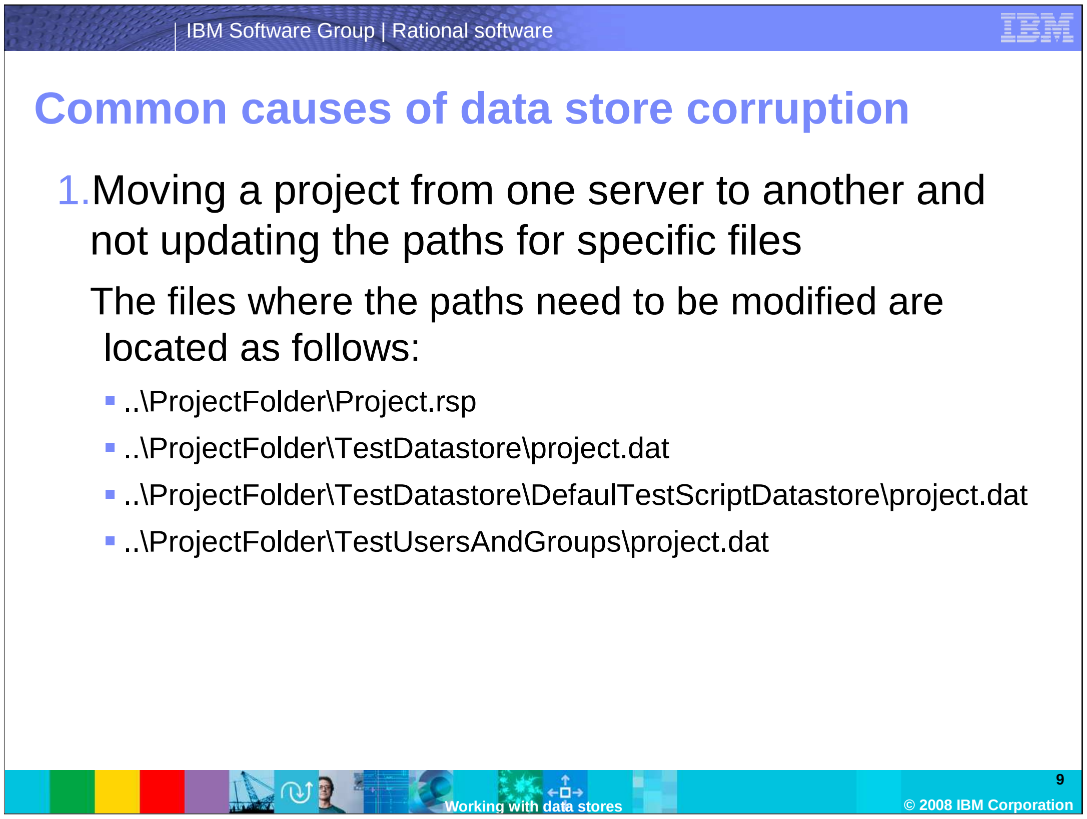
  These files in each data store contain the template information and properties of each asset.

- **.log folders**

  These folders contain results of the test execution

.xml and **xml** files contain template information and properties for each of the test asset files.

.log folders are all under the **TMS_Builds** folders and contain a variety of execution statistics and information displayed in the TestManager log viewer.



Common causes of data store corruption

1. Moving a project from one server to another and not updating the paths for specific files

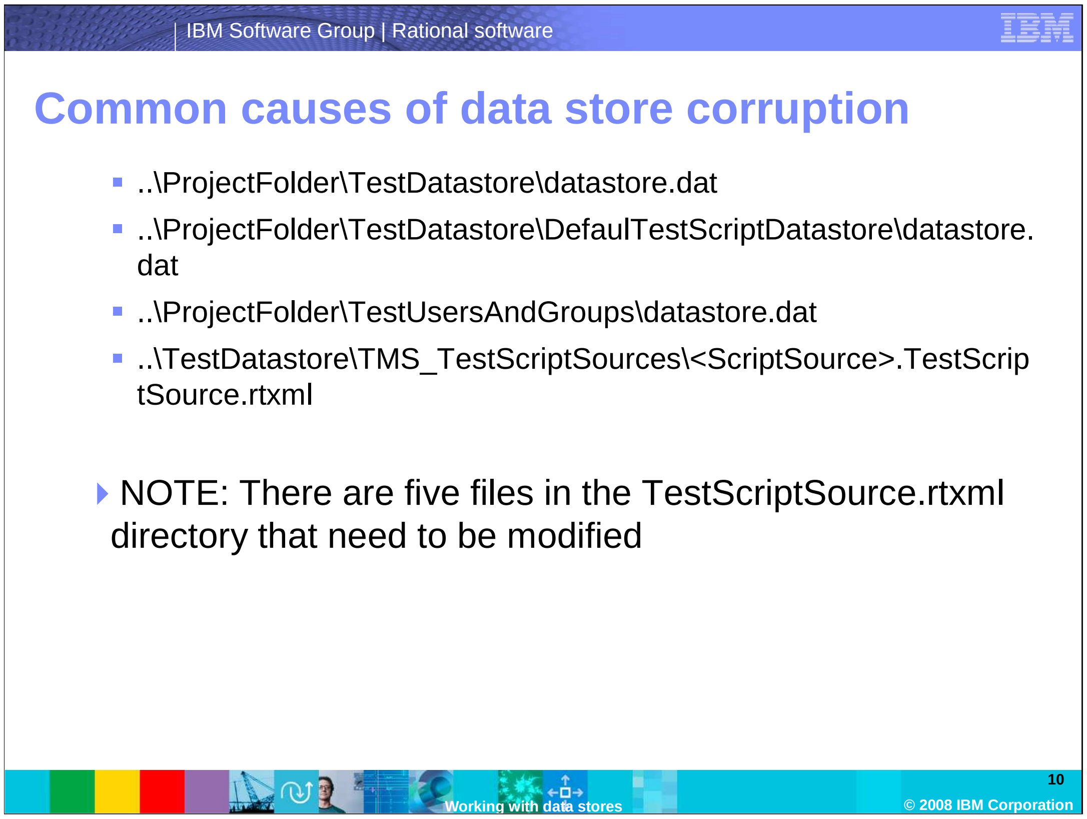
   The files where the paths need to be modified are located as follows:

   - ..\ProjectFolder\Project.rsp
   - ..\ProjectFolder\TestDatastore\project.dat
   - ..\ProjectFolder\TestDatastore\DefaulTestScriptDatastore\project.dat
   - ..\ProjectFolder\TestUsersAndGroups\project.dat

Data Store corruption is when a particular part of the project files become unusable, or, if they are missing.

A common cause of data store corruption occurs when moving a project from one server to another without updating the paths for specific files. In this case, there are 12 paths that need to be modified for the project move to be successful without any corruption.

First, locate the project.rsp and the three project.dat files as shown here.

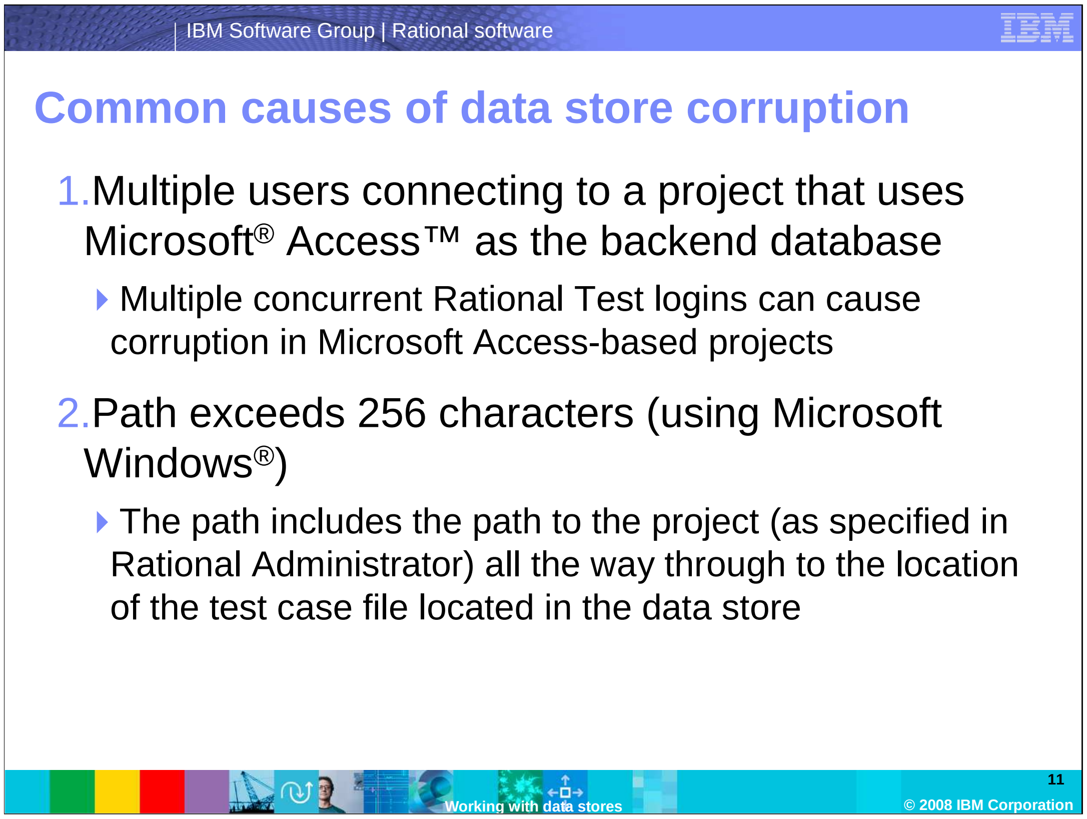


Common causes of data store corruption

- ..\ProjectFolder\TestDatastore\datastore.dat
- ..\ProjectFolder\TestDatastore\DefaulTestScriptDatastore\datastore.dat
- ..\ProjectFolder\TestUsersAndGroups\datastore.dat
- ..\TestDatastore\TMS_TestScriptSources\<ScriptSource>.TestScriptSource.rtxml

▸ NOTE: There are five files in the TestScriptSource.rtxml directory that need to be modified

Finally, there are three datastore.dat files and five test script source files that also need to be modified when moving a project to a different location – for a total of 12 files.


IBM Software Group | Rational software

Common causes of data store corruption

1. Multiple users connecting to a project that uses Microsoft® Access™ as the backend database
   - Multiple concurrent Rational Test logins can cause corruption in Microsoft Access-based projects
2. Path exceeds 256 characters (using Microsoft Windows®)
   - The path includes the path to the project (as specified in Rational Administrator) all the way through to the location of the test case file located in the data store

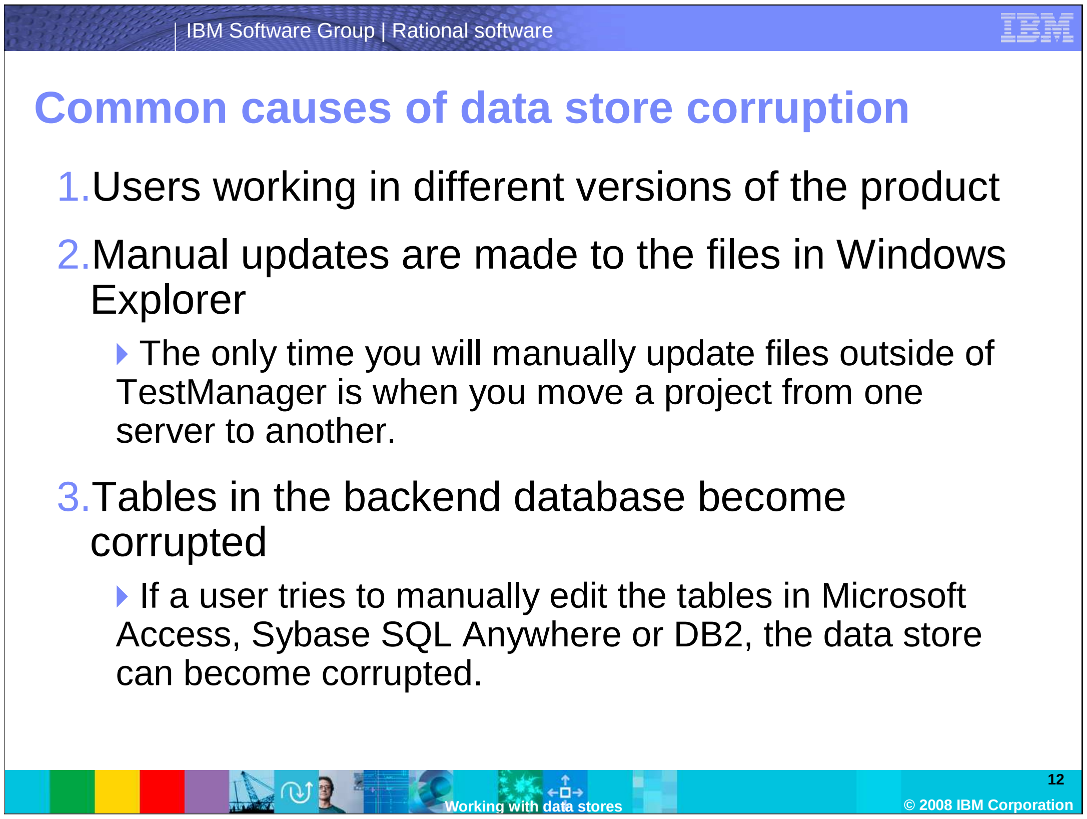
Working with data stores

11

© 2008 IBM Corporation

Another common cause of data store corruption occurs when multiple users connect to a project that uses Microsoft Access as the backend database. In this case, multiple concurrent Rational test logins can cause corruption in MS Access-based projects. Microsoft Access should *only* be used as a backend database if there is only one user working with that data store. Users of TestManager version 7 or later should use DB2®, while users of 2003.06.00 through 2003.06.14 should use Sybase SQL Anywhere. Users of 2003.06.15 and 16 can use either Sybase SQL Anywhere or DB2.

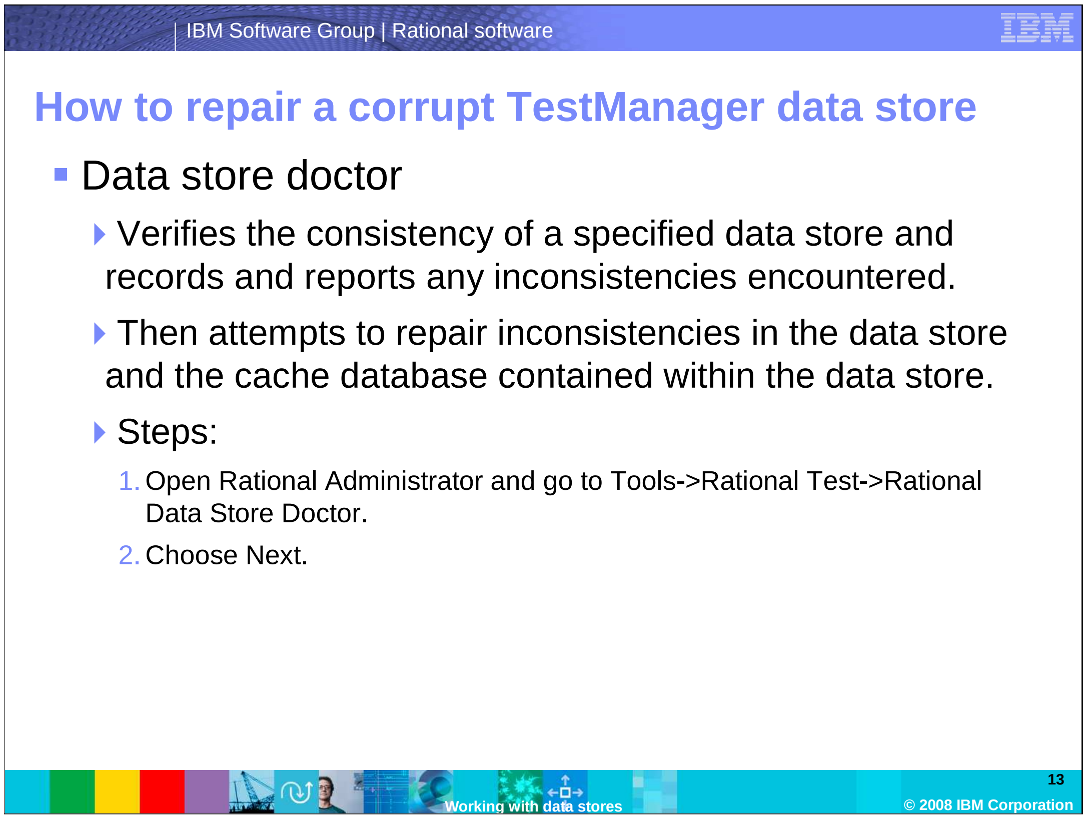
Furthermore, another common cause occurs when the path exceeds 256 characters on Microsoft Windows. The complete path must be less than 256 characters for any operations to be performed on the test case. This means the path includes the entire path to the project, as specified in Rational Administrator, all the way through to the location of the test case file located in the data store. You may experience one of the following errors if the path is too long: “General Data Store error” or "Failure to insert asset. Insertion results in invalid path.”



Common causes of data store corruption

1. Users working in different versions of the product
2. Manual updates are made to the files in Windows Explorer
   - The only time you will manually update files outside of TestManager is when you move a project from one server to another.
3. Tables in the backend database become corrupted
   - If a user tries to manually edit the tables in Microsoft Access, Sybase SQL Anywhere or DB2, the data store can become corrupted.

Users working in different versions of the product may also cause data store corruption. To prevent this, ensure all users on a project use the same version of TestManager. With regards to manual updates, do not manually edit files outside of the TestManager GUI. You should make all of your edits through user interface in the product. By changing the test data store manually through the file system, a user can introduce irreparable damage to the test project. The only time you will manually update files outside of TestManager is when you move a project from one server to another. Last, tables can become corrupt if you try to manually edit the tables in Microsoft Access, Sybase SQL Anywhere, or DB2. Be sure to do your work in TestManager to prevent table corruption.


IBM Software Group | Rational software

How to repair a corrupt TestManager data store

- Data store doctor
  - Verifies the consistency of a specified data store and records and reports any inconsistencies encountered.
  - Then attempts to repair inconsistencies in the data store and the cache database contained within the data store.
  - Steps:
    1. Open Rational Administrator and go to Tools->Rational Test->Rational Data Store Doctor.
    2. Choose Next.

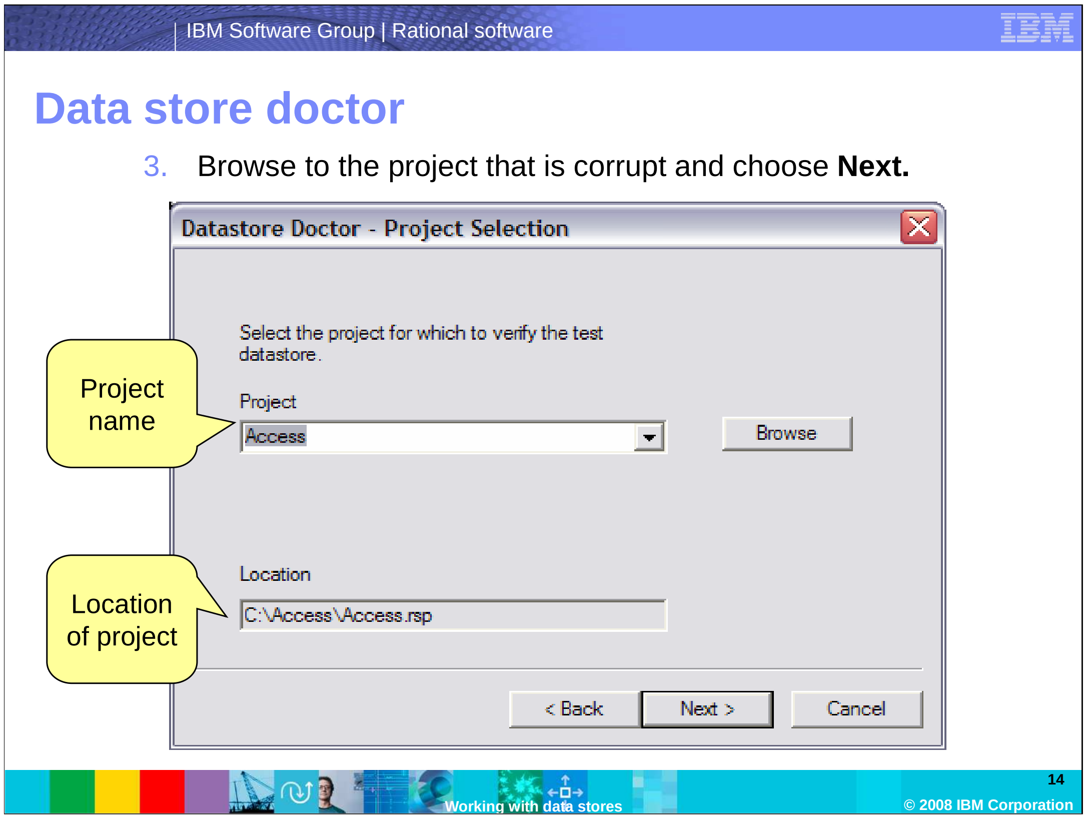
Working with data stores

13

© 2008 IBM Corporation

Now that you know about some common causes of data store corruption, the following few slides will cover ways to repair a corrupt TestManager data store. The Data Store Doctor is a utility for TestManager that will first verify the consistency of a specified data store, and then record and report any inconsistencies encountered. Next, the Data Store Doctor will attempt to repair inconsistencies in the data store and the cache database contained within the data store.

To launch Data Store Doctor, open Rational Administrator and go to Tools->Rational Test->Rational Data Store Doctor. Then choose **Next.**


IBM Software Group | Rational software

Data store doctor

3. Browse to the project that is corrupt and choose **Next.**

Datastore Doctor - Project Selection

Select the project for which to verify the test datastore.

Project: Access

Location: C:\Access\Access.rsp

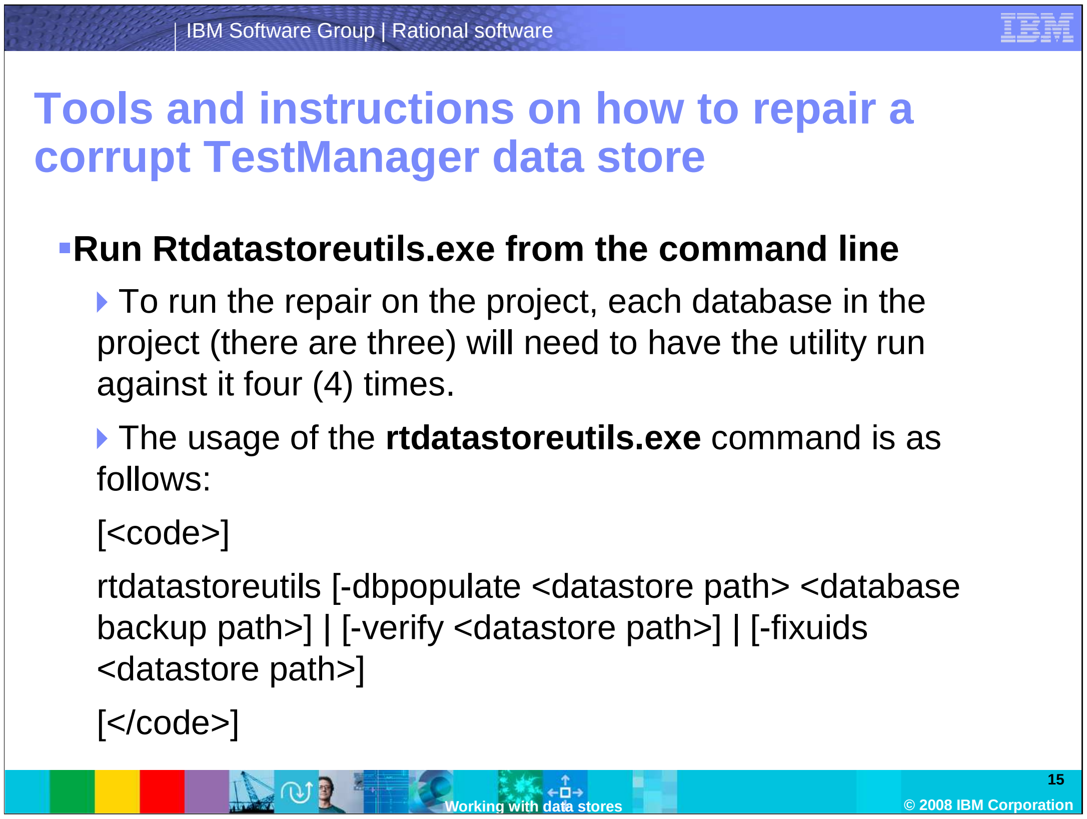
14

© 2008 IBM Corporation

The screen capture shown here appears during step 3. First, browse to the project that is corrupt, and then choose **Next**.

The repair will now run. Once complete, TestManager will return one of the following results:

1) “Verification Complete. No errors were detected”, or,
2) “Verification Complete. Errors were detected. Please view the log for more details”



IBM Software Group | Rational software

Tools and instructions on how to repair a corrupt TestManager data store

- **Run Rtdatastoreutils.exe from the command line**
  - To run the repair on the project, each database in the project (there are three) will need to have the utility run against it four (4) times.
  - The usage of the **rtdatastoreutils.exe** command is as follows:

  [<code>]

  rtdatastoreutils [-dbpopulate <datastore path> <database backup path>] | [-verify <datastore path>] | [-fixuids <datastore path>]

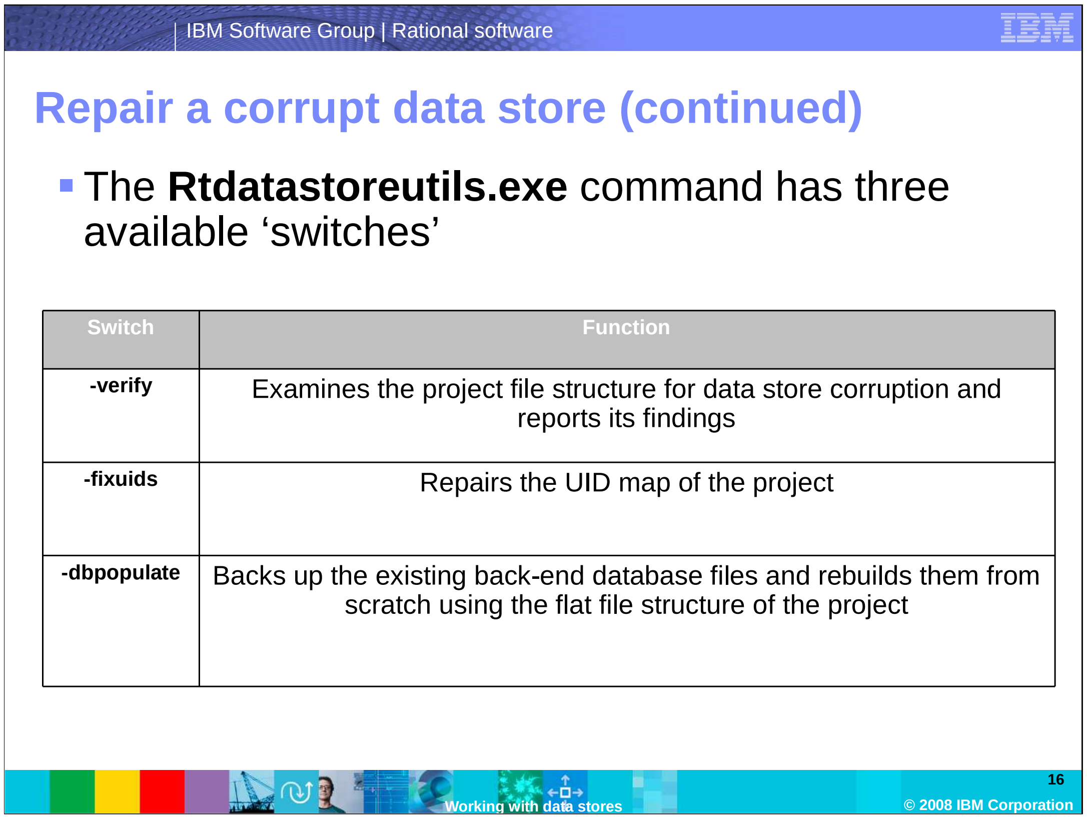
  [</code>]

Working with data stores

15

© 2008 IBM Corporation

If you prefer to attempt to repair a data store by means other than the Data Store Doctor, you can run rtdatastoreutils from the command line. To run the repair on the project, each database in the project will need to have the utility run against it four times. Note that there are three databases in a project. Run the command as shown here.



Repair a corrupt data store (continued)

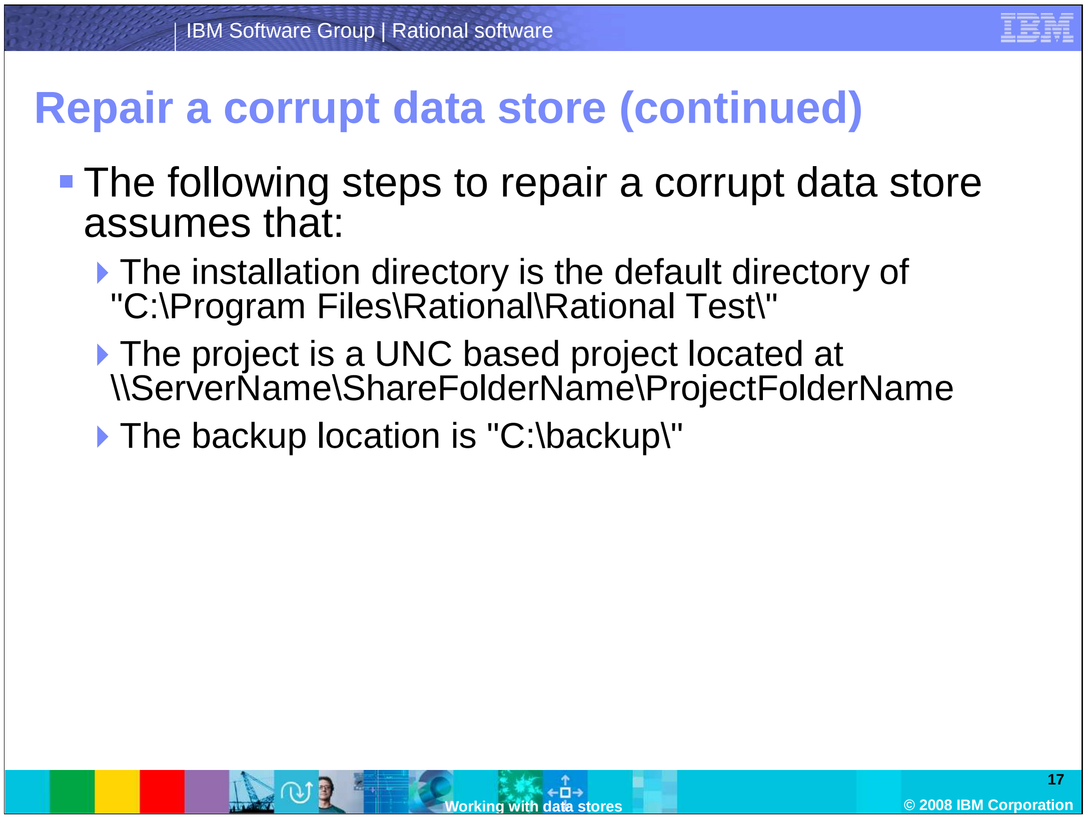
- The **Rtdatastoreutils.exe** command has three available 'switches'

Switch	Function
-verify	Examines the project file structure for data store corruption and reports its findings
-fixuids	Repairs the UID map of the project
-dbpopulate	Backs up the existing back-end database files and rebuilds them from scratch using the flat file structure of the project

Rtdatastoreutils.exe uses one of three switches to repair the data store as shown in the table here: **–verify**, -**fixuids**, or –**dbpopulate**. The –verify switch examines the project file structure for data store corruption and reports its findings. The –fixuids switch repairs the UID map of the project. The –dbpopulate switch backs up the existing back-end database files and rebuilds them from scratch, using the flat file structure of the project.

Note that before running any of the commands, verify that all users are disconnected from the project. If using **–dbpopulate,** it will disconnect all users when it runs.

If the data store being repaired is using Access or SQL Anywhere for the backend database, these commands will run faster if the original file tds.db (with SQL) or tds.mdb (with MS Access) is renamed or moved before starting the commands.

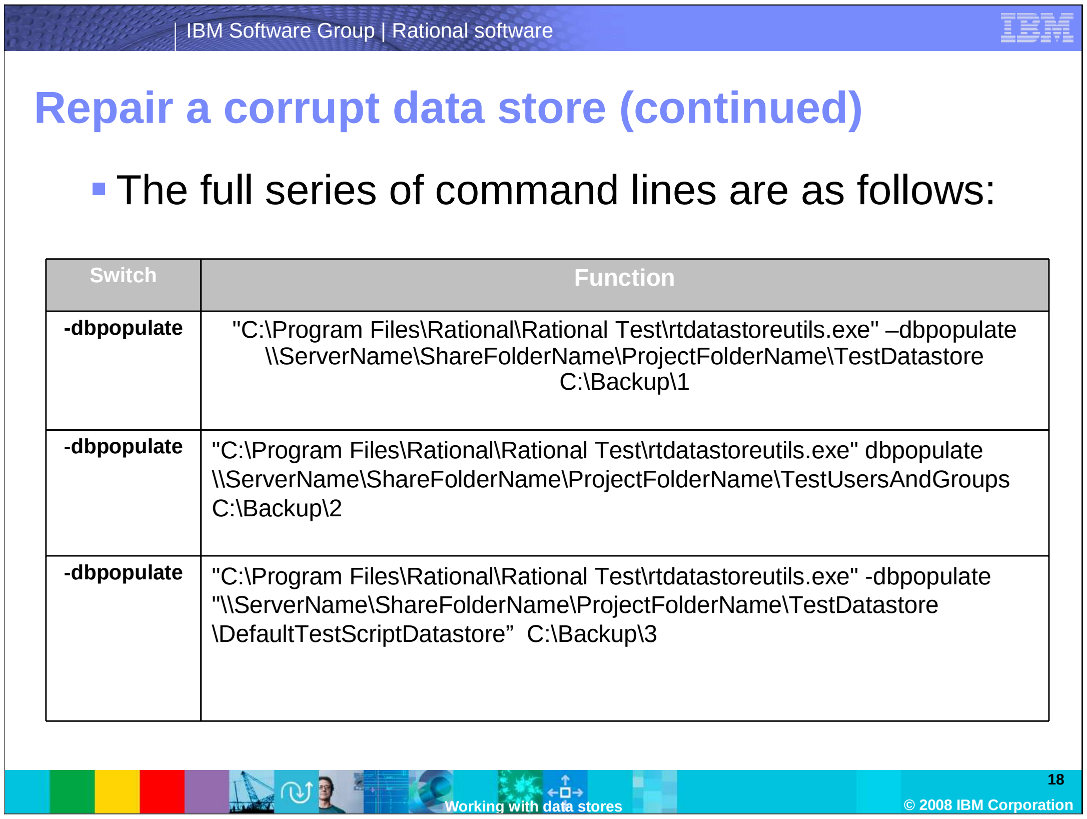


Repair a corrupt data store (continued)

- The following steps to repair a corrupt data store assumes that:
  - The installation directory is the default directory of "C:\Program Files\Rational\Rational Test\"
  - The project is a UNC based project located at \\ServerName\ShareFolderName\ProjectFolderName
  - The backup location is "C:\backup\"

The following steps to repair a corrupt data store assume these three notions: that the installation directory is the default of "C:\Program Files\Rational\Rational Test\“, that the project is a UNC based project located at \\ServerName\ShareFolderName\ProjectFolderName, and that the backup location is "C:\backup\“.

If your setup is different, redirect your commands to reflect the appropriate locations of your default directory, project location, and your backup directory.



Repair a corrupt data store (continued)

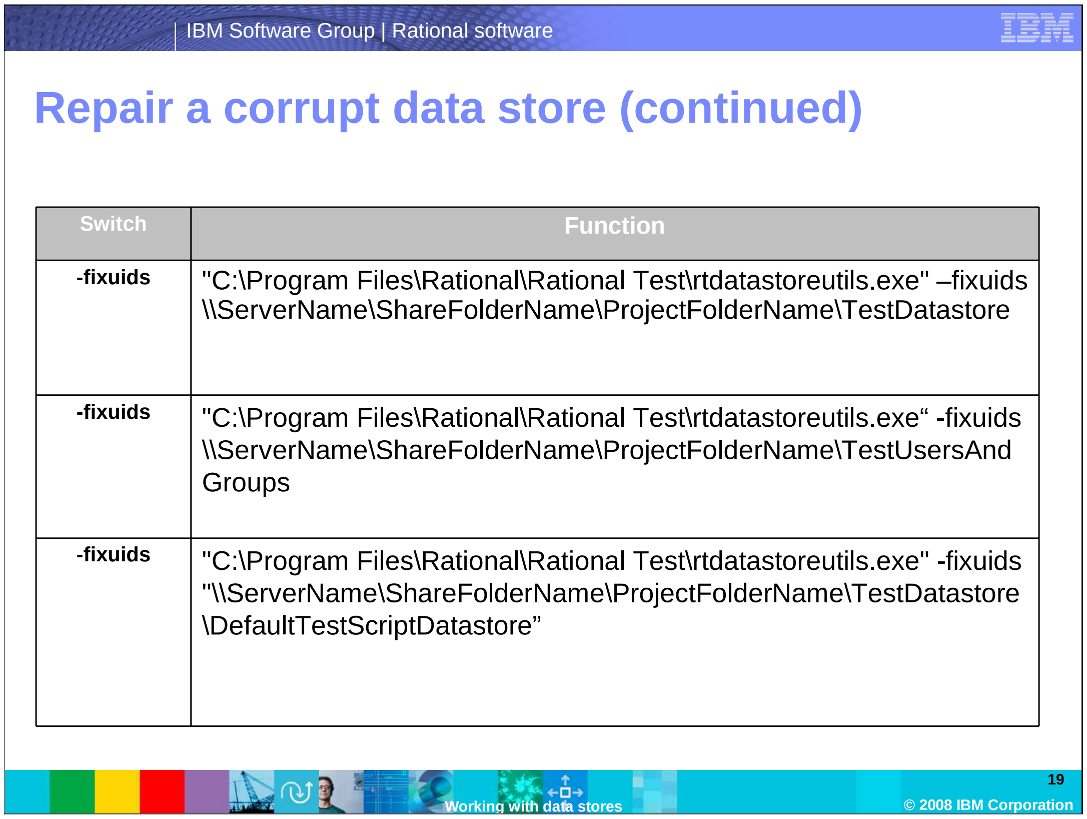
- The full series of command lines are as follows:

Switch	Function
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" –dbpopulate \\ServerName\ShareFolderName\ProjectFolderName\TestDatastore C:\Backup\1
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" dbpopulate \\ServerName\ShareFolderName\ProjectFolderName\TestUsersAndGroups C:\Backup\2
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" -dbpopulate "\\ServerName\ShareFolderName\ProjectFolderName\TestDatastore \DefaultTestScriptDatastore" C:\Backup\3

The first step is to use the –**dbpopulate** switch three times, as shown here. The three different directories are shown in the table. Note: You must run all three commands.

Make sure the chosen backup location ('C:\backup\' in the example) is EMPTY before running. If it is not, the -**dbpopulate** commands will end without running.

Since the commands have the full path to **rtdatastoreutils.exe**, you should include quotations to handle the blank spaces in the default path.


IBM Software Group | Rational software

Repair a corrupt data store (continued)

Switch	Function
-fixuids	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" –fixuids \\ServerName\ShareFolderName\ProjectFolderName\TestDatastore
-fixuids	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe“ -fixuids \\ServerName\ShareFolderName\ProjectFolderName\TestUsersAndGroups
-fixuids	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" -fixuids "\\ServerName\ShareFolderName\ProjectFolderName\TestDatastore\DefaultTestScriptDatastore”

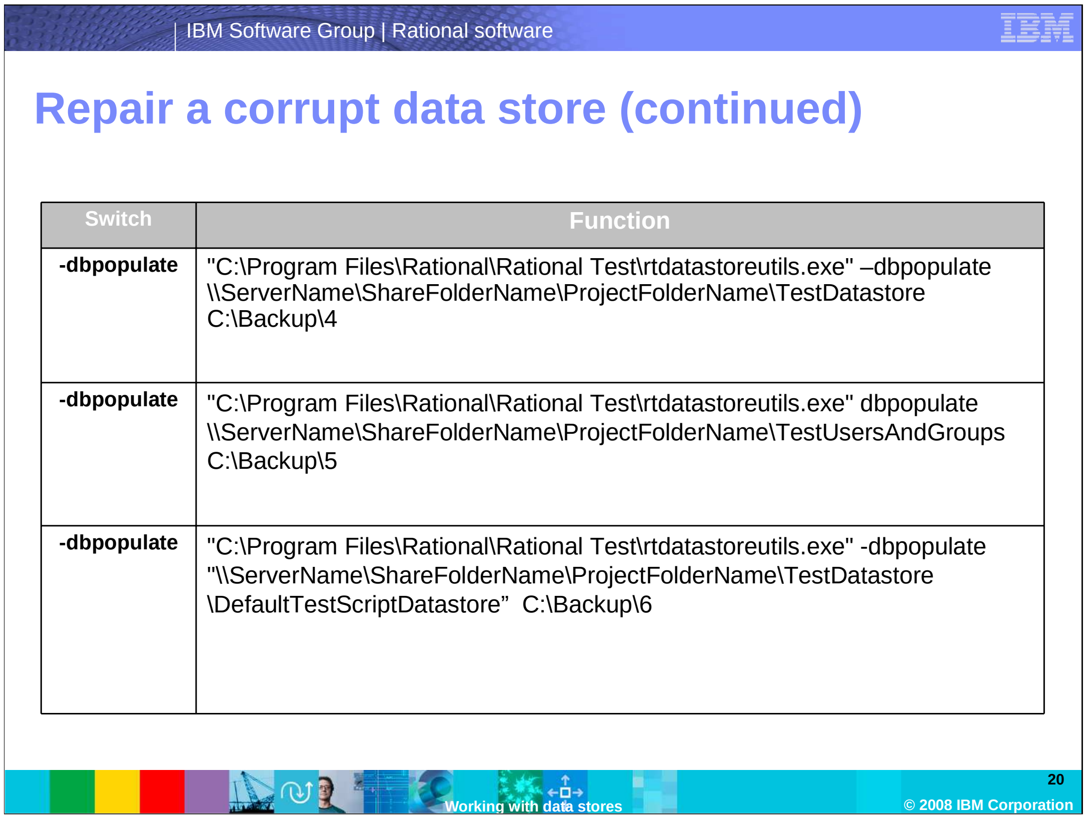
Working with data stores

19

© 2008 IBM Corporation

Like the previous step, you must run these three **–fixuids** commands. The three different directories are shown in the table.

Since the commands listed have the full path to **rtdatastoreutils.exe**, you should include quotations to handle the blank spaces in the default path.

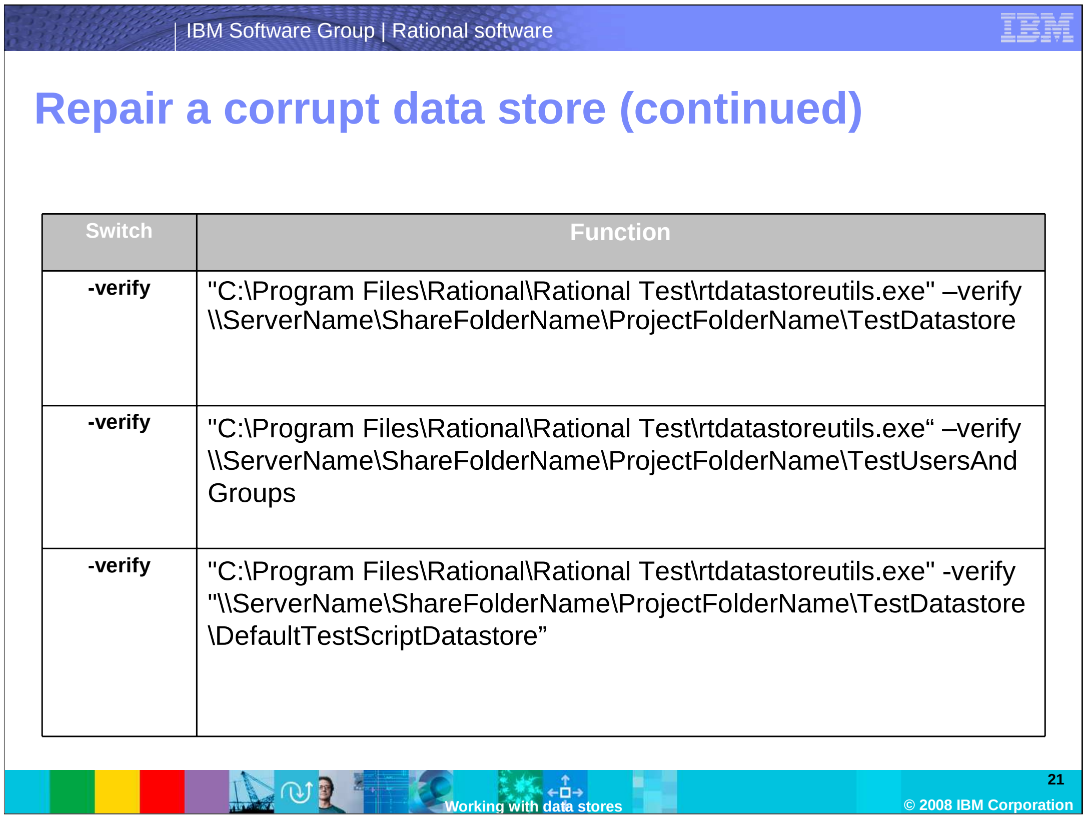


Repair a corrupt data store (continued)

Switch	Function
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" –dbpopulate \\ServerName\ShareFolderName\ProjectFolderName\TestDatastore C:\Backup\4
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" dbpopulate \\ServerName\ShareFolderName\ProjectFolderName\TestUsersAndGroups C:\Backup\5
-dbpopulate	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" -dbpopulate "\\ServerName\ShareFolderName\ProjectFolderName\TestDatastore \DefaultTestScriptDatastore" C:\Backup\6

Next, you will again run the –**dbpopulate** switch three times, as shown here. The three different directories are shown in the table.

Since the commands have the full path to **rtdatastoreutils.exe**, you should include quotations to handle the blank spaces in the default path.

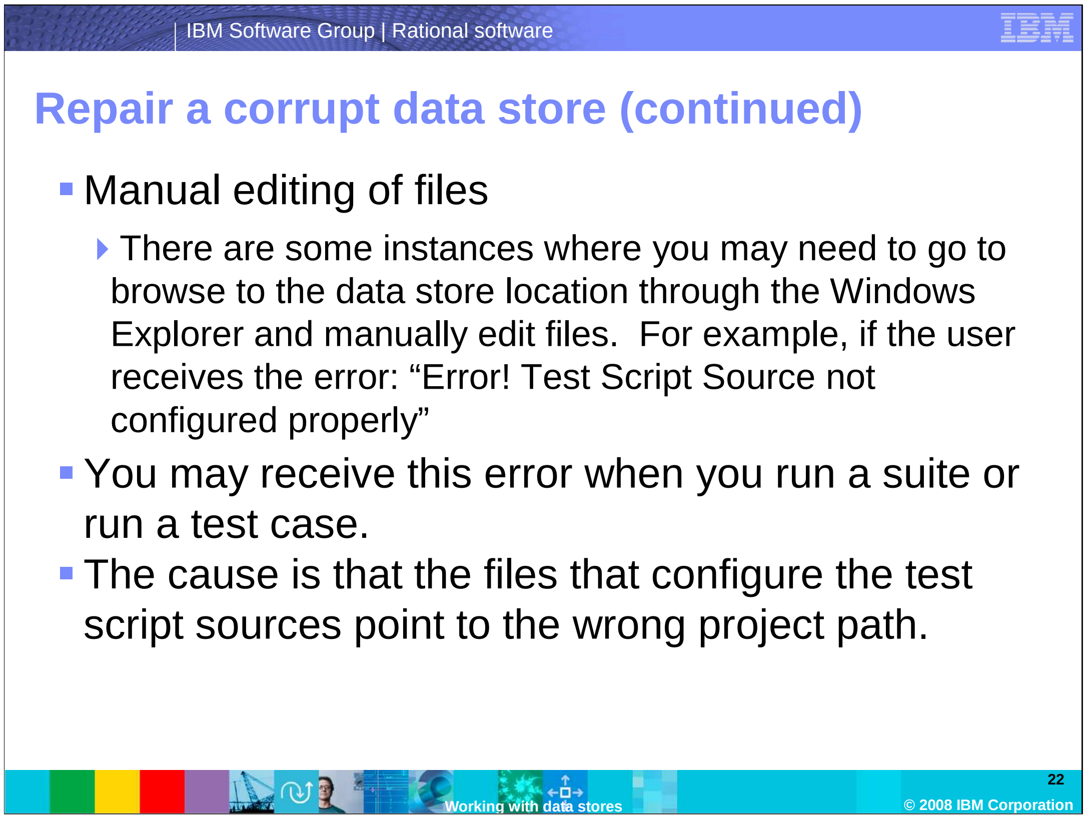


Repair a corrupt data store (continued)

Switch	Function
-verify	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" –verify \\ServerName\ShareFolderName\ProjectFolderName\TestDatastore
-verify	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe“ –verify \\ServerName\ShareFolderName\ProjectFolderName\TestUsersAnd Groups
-verify	"C:\Program Files\Rational\Rational Test\rtdatastoreutils.exe" -verify "\\ServerName\ShareFolderName\ProjectFolderName\TestDatastore \DefaultTestScriptDatastore”

The final step is to use the **–verify** switch three times as shown here. The three different directories are shown in the table. Note: You must run all three commands.

Since the commands have the full path to **rtdatastoreutils.exe**, you should include quotations to handle the blank spaces in the default path.



IBM Software Group | Rational software

Repair a corrupt data store (continued)

- Manual editing of files
  - There are some instances where you may need to go to browse to the data store location through the Windows Explorer and manually edit files. For example, if the user receives the error: “Error! Test Script Source not configured properly”
- You may receive this error when you run a suite or run a test case.
- The cause is that the files that configure the test script sources point to the wrong project path.

Working with data stores

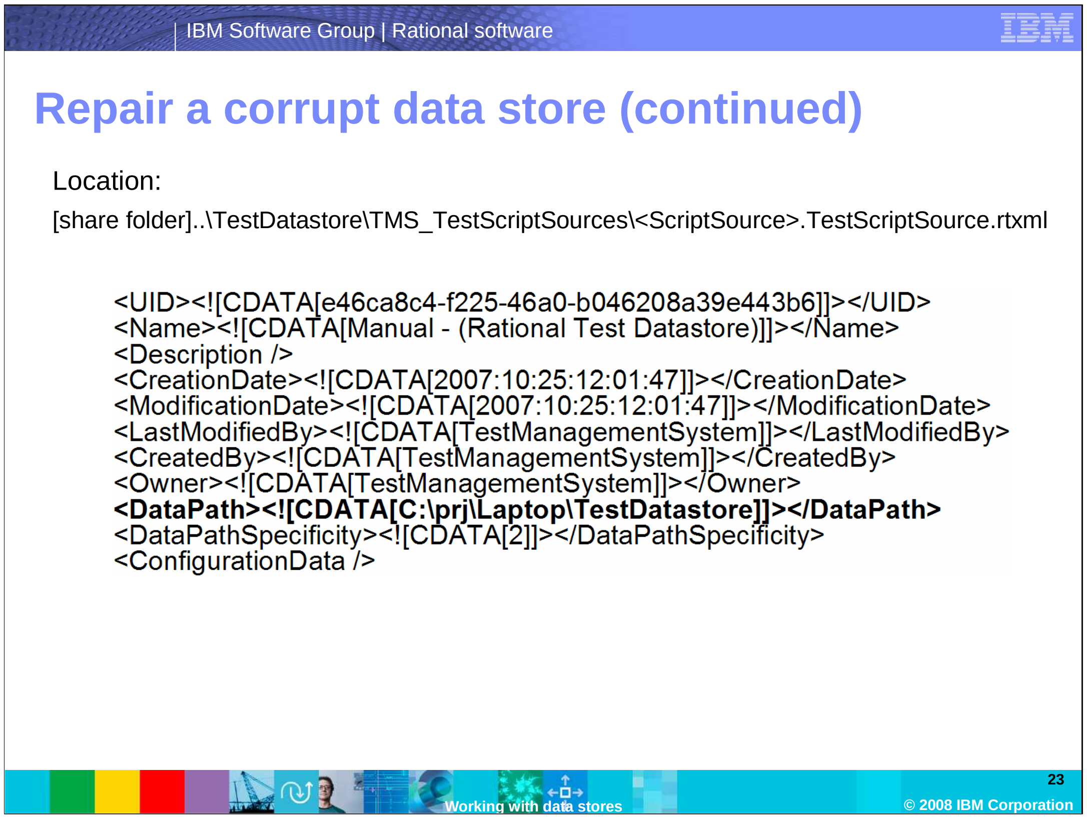
22

© 2008 IBM Corporation

If Data Store Doctor or rtdatastoreutils.exe did not resolve the corruption, you may need to manually edit the files. There are some instances where you may need browse to the data store location through windows explorer and manually edit files.

In this example, the path to the script file is incorrect. You may see **“Error! Test Script Source not configured properly”**

You may receive this error when you run a suite or run a test case. The cause is that the files that configure the test script sources point to the wrong project path.


IBM Software Group | Rational software

Repair a corrupt data store (continued)

Location:

[share folder]..\TestDatastore\TMS_TestScriptSources\<ScriptSource>.TestScriptSource.rtxml

```
<UID><![CDATA[e46ca8c4-f225-46a0-b046208a39e443b6]]></UID>
<Name><![CDATA[Manual - (Rational Test Datastore)]]></Name>
<Description />
<CreationDate><![CDATA[2007:10:25:12:01:47]]></CreationDate>
<ModificationDate><![CDATA[2007:10:25:12:01:47]]></ModificationDate>
<LastModifiedBy><![CDATA[TestManagementSystem]]></LastModifiedBy>
<CreatedBy><![CDATA[TestManagementSystem]]></CreatedBy>
<Owner><![CDATA[TestManagementSystem]]></Owner>
<DataPath><![CDATA[C:\prj\Laptop\TestDatastore]]></DataPath>
<DataPathSpecificity><![CDATA[2]]></DataPathSpecificity>
<ConfigurationData />
```

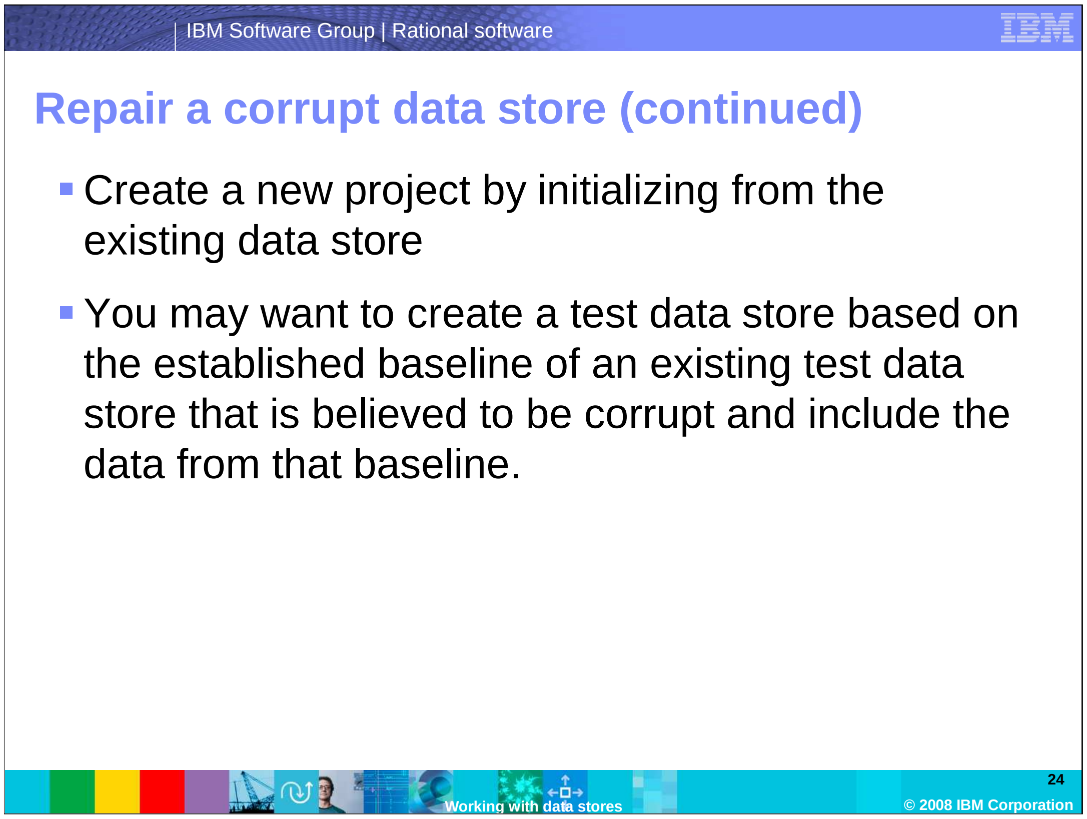
Working with data stores

23

© 2008 IBM Corporation

This slide describes the location of the files that need to be edited. The testscriptsource.rtxml file is located in the share folder at the location shown here. This code example contains the data path that you need to edit.

The highlighted data path line in this example needs to reflect the current path where the test data store is located. Note: there are different rtxml files including: GUI, Manual, Java, Command line, and Visual Basic, that should all be edited.



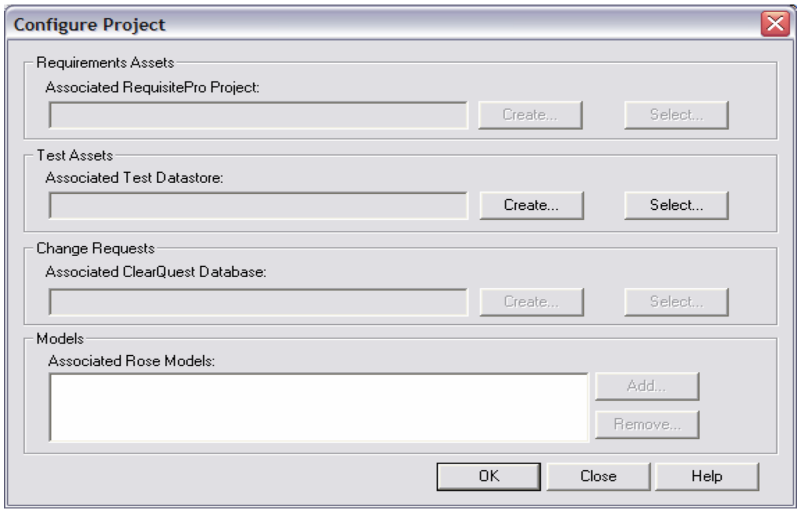
Repair a corrupt data store (continued)

- Create a new project by initializing from the existing data store
- You may want to create a test data store based on the established baseline of an existing test data store that is believed to be corrupt and include the data from that baseline.

Additionally, you can create a new project by initializing from the existing data store. You may want to create a test data store based on the established baseline of an existing test data store that is believed to be corrupt and include the data from that baseline.

Repair a corrupt data store (continued)

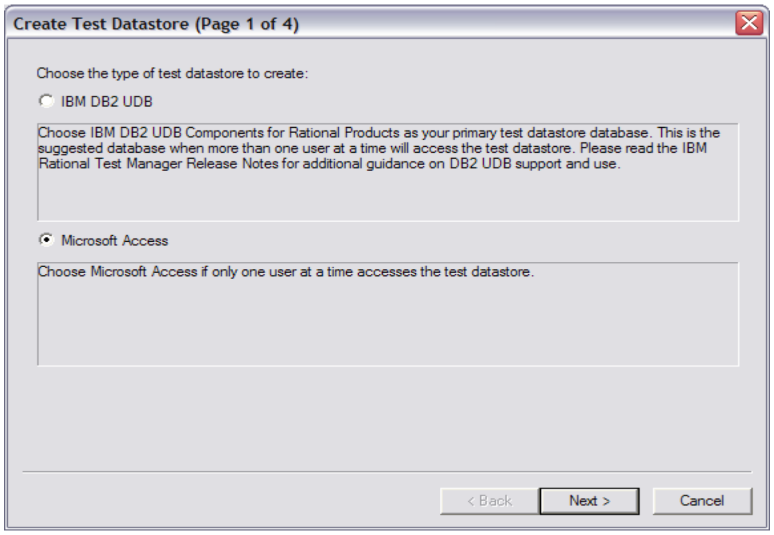
1. On the Configure Project dialog box, click Create in the Test Assets area. The Create Test Data Store wizard starts.



Continuing from the last slide, create a new project by initializing from the existing data store. On the Configure Project dialog box, click Create in the Test Assets area. The Create Test Data Store wizard will appear.

Repair a corrupt data store (continued)

2. On the first page of the wizard, select the vendor database you want to work with. Click Next.



Create Test Datastore (Page 1 of 4)

Choose the type of test datastore to create:

IBM DB2 UDB

Choose IBM DB2 UDB Components for Rational Products as your primary test datastore database. This is the suggested database when more than one user at a time will access the test datastore. Please read the IBM Rational Test Manager Release Notes for additional guidance on DB2 UDB support and use.

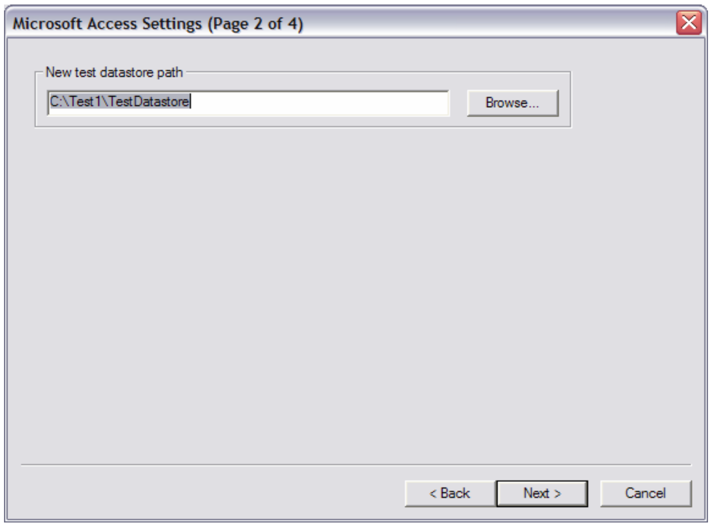
[✓ Microsoft Access]

Choose Microsoft Access if only one user at a time accesses the test datastore.

Second, choose the vendor database you want to associate with this project. Select the appropriate vendor, then click Next.

Repair a corrupt data store (continued)

3. Complete the second page of the wizard with the UNC pathname of the test data store you want to create (for example, \\computerA\project-directory\tests) and other information specific to the vendor database you chose in Step 2. Click Next.



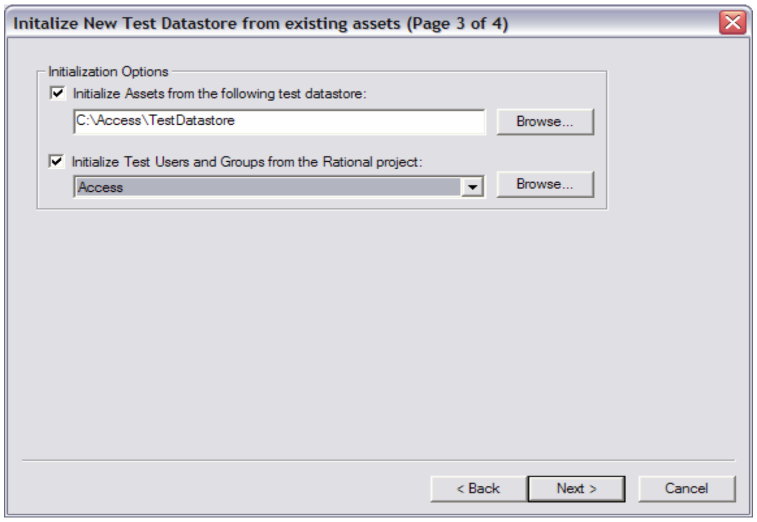
Microsoft Access Settings (Page 2 of 4)

New test datastore path: C:\Test1\TestDatastore

Third, choose the new data store path for the new data store you are creating. Complete the second page of the wizard with the UNC pathname of the test data store you want to create (for example, \\computerA\project-directory\tests) and information specific to the vendor database you chose in Step 2. Once complete, click Next.

Repair a corrupt data store (continued)

4. Choose Initialization options


Initalize New Test Datastore from existing assets (Page 3 of 4)

Initialization Options

☑ Initialize Assets from the following test datastore:

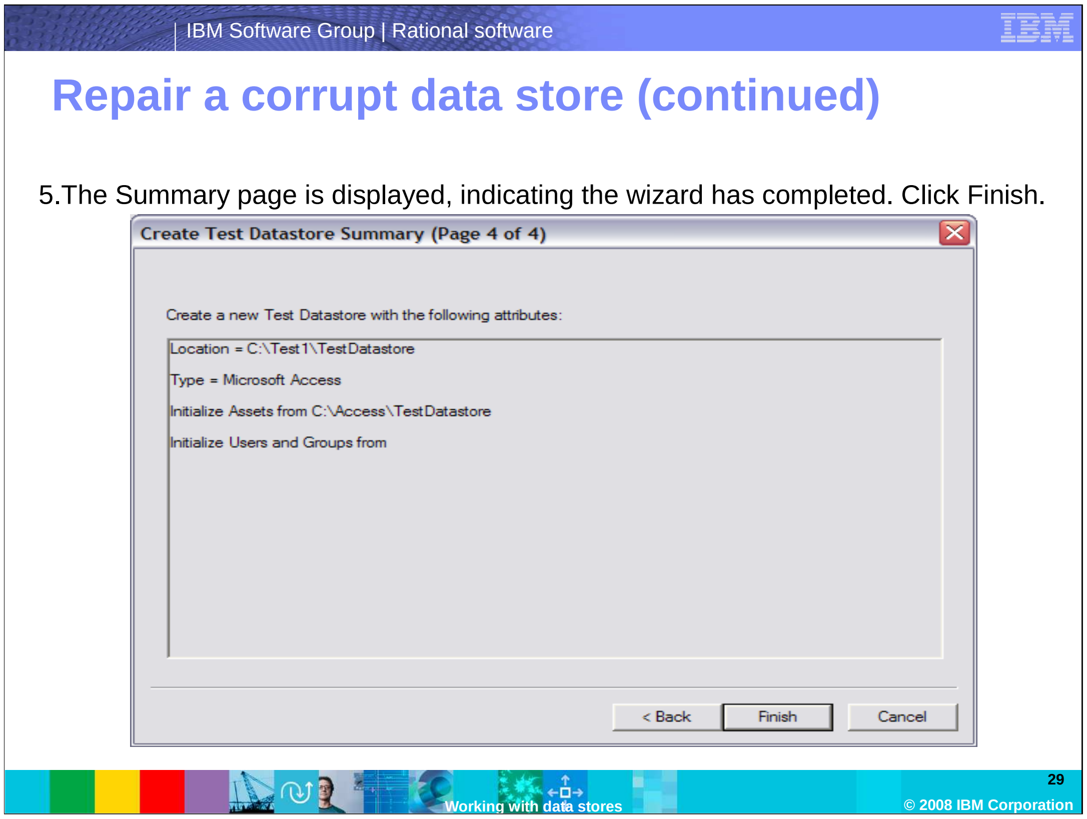
C:\Access\TestDatastore

☑ Initialize Test Users and Groups from the Rational project:

Access

Next, choose the Initialization Options. Check off Initialize Assets from the test data store and browse to the TestDatastore directory that is corrupt. Afterwards, click Next.

Note: You can also choose to initialize from the Test Users and groups for that data store so that you do not have to create new Test Users and Test groups.


IBM Software Group | Rational software

Repair a corrupt data store (continued)

5.The Summary page is displayed, indicating the wizard has completed. Click Finish.

Create Test Datastore Summary (Page 4 of 4)

Create a new Test Datastore with the following attributes:

```
Location = C:\Test1\TestDatastore
Type = Microsoft Access
Initialize Assets from C:\Access\TestDatastore
Initialize Users and Groups from
```

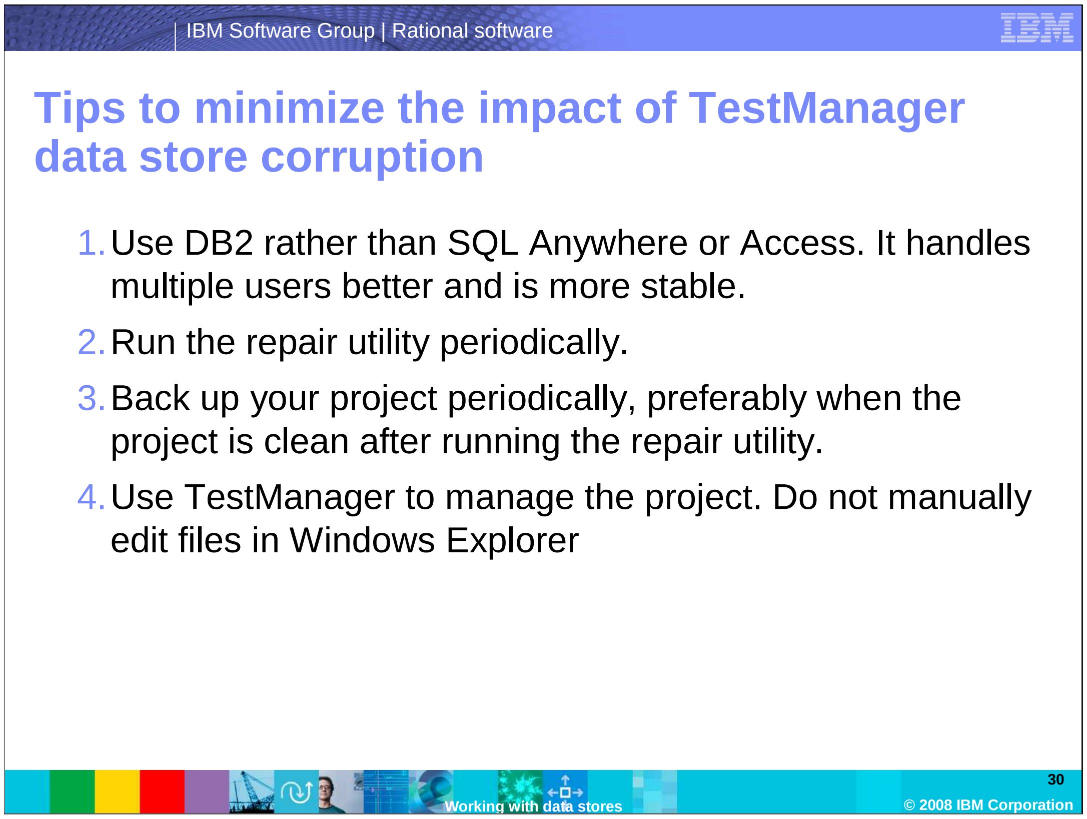
Working with data stores

29

© 2008 IBM Corporation

Finally, the summary page will appear. This describes the location where the TestDatastore is being created, the type of database, where the TestDatastore location is being initialized from, and the Users and Groups the user is initializing from (if any).

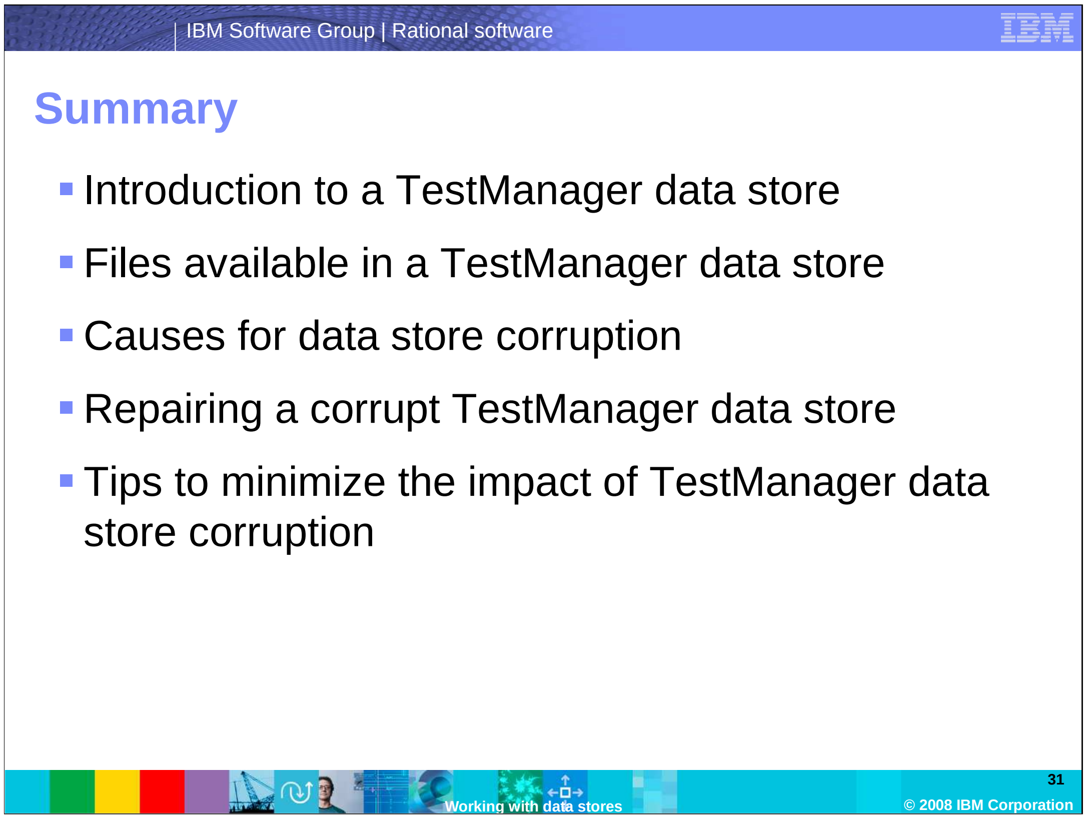
The summary page is displayed indicating the wizard has completed its task.   Click Finish and the new TestDatastore is created.



Tips to minimize the impact of TestManager data store corruption

1. Use DB2 rather than SQL Anywhere or Access. It handles multiple users better and is more stable.
2. Run the repair utility periodically.
3. Back up your project periodically, preferably when the project is clean after running the repair utility.
4. Use TestManager to manage the project. Do not manually edit files in Windows Explorer

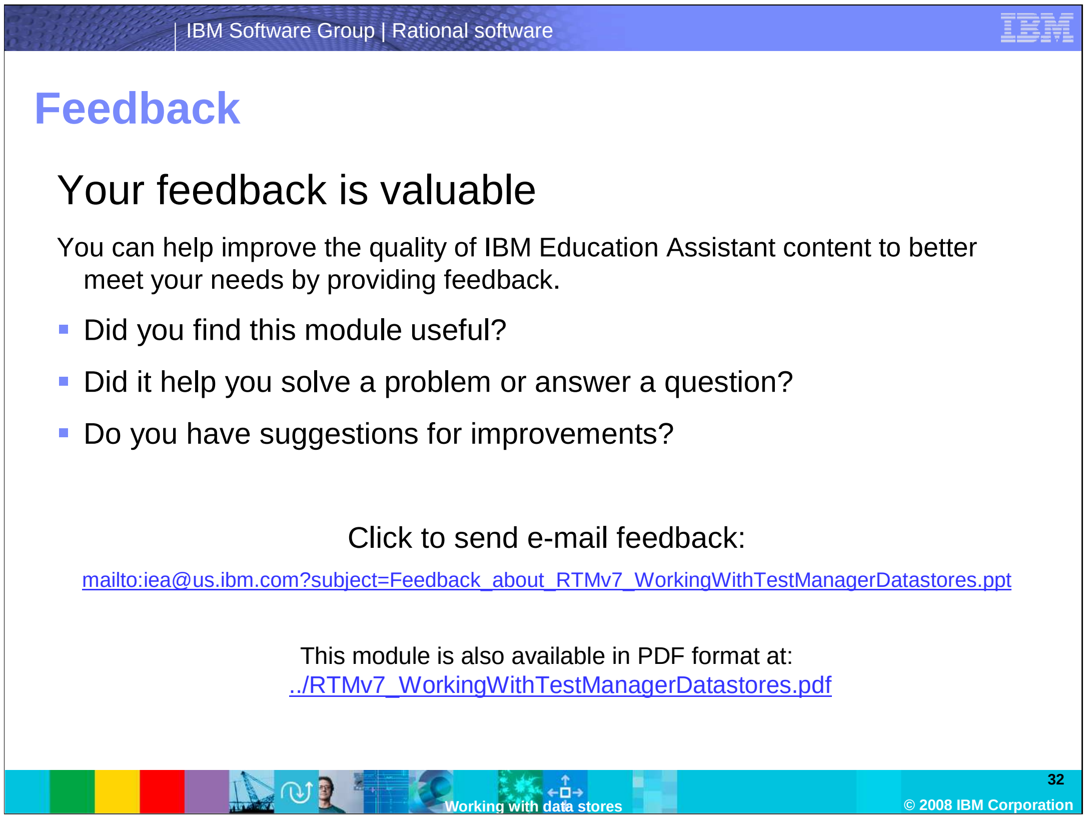
On a final note, here are some tips on how to minimize Data Store corruption:
1) Use DB2 rather than SQL Anywhere or Access. It handles multiple users better and is more stable. 2) Run the repair utility periodically. 3) Back up your project periodically, preferably when the project is clean after running the repair utility. And 4) Use TestManager to manage the project. Do not manually edit files in Windows Explorer.



Summary

- Introduction to a TestManager data store
- Files available in a TestManager data store
- Causes for data store corruption
- Repairing a corrupt TestManager data store
- Tips to minimize the impact of TestManager data store corruption

In summary, this module covered an introduction to a TestManager data store, the various files available in a TestManager data store, provided common causes for data store corruption, showed the steps for repairing a Corrupt TestManager data store, and provided tips to minimize the impact of TestManager data store corruption.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_RTMv7_WorkingWithTestManagerDatastores.ppt

This module is also available in PDF format at:
../RTMv7_WorkingWithTestManagerDatastores.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers