Application Coverage for Web Applications

In this section:

Introduction

You can monitor and collect coverage data from .NET managed code during manual or automated functional tests performed on a running web application that is deployed on IIS server. You can also send coverage data and test results to DTP. The application coverage information can be displayed in the DTP Coverage Explorer (see the "Coverage Explorer" chapter in the DTP user manual), which provides insights about how well the application is tested, as well as the quality of your tests.

.NET Core Web Applications

dotTEST can collect coverage for .NET Core web applications that are deployed on IIS server. Alternatively, you can use the `coverage.exe` tool shipped with dotTEST; see Application Coverage for Standalone Applications for details.

Prerequisites

The following components are required for collecting coverage:

- Internet Information Services (IIS) version 7.5 or higher
- Coverage Agent Manager (CAM) (contact your Parasoft representative) or SOAtest

Process Overview

dotTEST ships with a component called the coverage agent. The coverage agent is attached to the application under test (AUT) and monitors the code being executed as the AUT runs. When the coverage agent is attached to the AUT, a REST API is exposed that enables you to mark the beginning and end of each test and test session. During test execution, interactions with the coverage agent and AUT are written to a dynamic coverage data file, which contains markers that specify which lines of code were touched.

The steps that follow:

- dotTEST processes the dynamic coverage data and static coverage data, and merges them to create coverage.xml file.
- A coverage.xml file, which contains the coverage information, is sent to DTP.
- When DTP receives the coverage data, it's loaded into a coverage image, which is a special tag that enables you to aggregate coverage data from runs with the same build ID.
- The coverage image enables you to associate coverage information with specific tests.

Test results are also sent to DTP from the tool that executes the tests (i.e., SOAtest, tests executed by dotTEST, manual tests, etc.) in a report.xml file. If the build IDs for the coverage.xml file and the report match, DTP is able to correlate the data and display the coverage information.

If you use a source control system, ensure that your source control settings are properly configured; see Source Control Settings.

Configuring the Application Under Test for Coverage

The following steps are required to prepare the application under test (AUT):

1. The static coverage file must be generated. The static coverage file contains metadata about user classes, methods, and lines; see Generating the Static Coverage File.
2. The coverage agent must be attached to the AUT; see Attaching the Coverage Agent to the AUT.

Generating the Static Coverage File

1. Run the following test configuration on the solution:

```
dottestcli.exe -config "builtin://Collect Static Coverage" -solution SOLUTION_PATH
```

2. The dottestcli console output will indicate where the static coverage data is saved:

```
Saving static coverage information into: 'C:\Users\[USER]\Documents\Parasoft\dotTEST\Coverage\Static\
[FILE].data
```

Customizing Scope of Coverage

By default, coverage is measured for the entire web application. You can customize the scope of coverage by adding the following switches when collecting static coverage to measure specific parts of the application (see Configuring the Test Scope, for usage information):

```
dottestcli.exe -config "builtin://Collect Static Coverage"
-solution "C:\Devel\FooSolution\FooSolution.sln"
-resource "FooSolution/QuxProject"
-include "C:\Devel\FooSolution\src\QuxProject\**\*.cs"
-exclude "C:\Devel\FooSolution\src\QuxProject\**\tests\**\*.cs"
```

The `-resource` switch points to a path inside the solution, while the `-include` and `-exclude` switches should be paths in the file system.

The scope information is stored in a scope configuration file, which can be provided to the IIS manager tool during web server configuration (see Attaching the Coverage Agent to the AUT). The output from the console will indicate the location of the scope configuration file:

```
Saving static coverage scope configuration into: 'C:\Users\[USER]
\Documents\Parasoft\dotTEST\Coverage\Static\scope.instrumentation.txt'
```

It is not possible to use the application coverage scope file for web projects that are compiled on IIS. This is because the target assemblies of IIS compilations are named unpredictably. Scope files can be used safely when the assembly name loaded by IIS can be predetermined before coverage collection starts.

Attaching the Coverage Agent to the AUT

1. Copy the [INSTALLATION_DIR]\integration\IIS directory to the machine were IIS is installed and the web application is deployed.
2. Run a console as an Administrator
3. Invoke the dotTEST IIS Manager tool on this machine to enable runtime coverage collection inside IIS:

```
dottest_iismanager.exe
```

   You may need to configure the dotTEST IIS Manager with additional options, see IIS Manager Options.

   dottest_iismanager initializes the environment for the web server (IIS) and behaves like a service, enabling you to execute tests and collect coverage. The service is ready and waiting for commands as long as the following message is printed to the output:

```
Write 'exit' and hit Enter to close dottest_iismanager
```

   A test session and test can be started even if the tested website or application has not been loaded yet.

4. Ensure that port 8050 (default port for the coverage agent) allows HTTP traffic in firewall settings on this machine. You can change the coverage agent port number if the default port is unavailable.
5. Open the website or application.
6. Go to the following address to check the status of the coverage agent: `http://host:8050/status`
   You should receive the following response:

```
{"session":null,"test":null}
```

Collecting Coverage from Multiple Users

You can collect coverage information for multiple users that are simultaneously accessing the same web application server. This requires launching the dotTEST IIS Manager with the `-multiuser` switch:

```
dottest_iismanager.exe -multiuser
```

See the Coverage Agent Manager (CAM) section of the DTP documentation for details.

Changing IIS Idle Timeouts

By default, IIS application pool processes are recycled after 20 minutes of idle time, which can have negative consequences on a test session. You can prevent this behavior by changing the default value so that people working with the application do not experience unexpected stops and restarts during a test session.

1. Start the Internet Information Services (IIS) Manager.
2. Open the Application Pools node.
3. Choose the pool for your web application.
4. Click **Advanced Settings** in the Actions panel.
5. In Process Model section, change the Idle Time-out (minutes) setting to a value better-suited to your testing practices.

IIS Manager Options

Option	Value	Description
-scope	`[path]`	A path to the scope configuration file. Required if the scope is other than the default; see Customizing Scope of Coverage.
-agentTimeout	`[miliseconds]`	Specifies the timeout for connection with the Coverage Agent. Adjust the timeout to your machine capabilities. The default value is 1500 ms. If you provide 0 or a negative value, the connection attempt will not timeout, which may lead to a considerable slowdown or hang the tool.
-port	`[port number]`	Specifies the port number when you start dottest_iismanager if the default port is unavailable.
-multiuser		Enables collecting coverage information for multiple users; see Collecting Coverage from Multiple Users.

Test Configuration and Execution with SOAtest

You can use SOAtest to run functional tests (refer the Application Coverage chapter of the SOAtest documentation to set up the test configuration), as well as execute manual tests. At the end of the test session, coverage will be saved in `runtime_coverage_[timestamp].data` files in the directory specified in SOAtest. This information will eventually be merged with the static coverage data to create a coverage.xml file and uploaded to DTP.

Uploading Test Results to DTP

If you use dotTEST with CAM

1. Go to **Report Center** in the DTP interface.
2. Click the gear icon and choose **Report Center Settings> Additional Settings> Report Center Administration> Tools> Data Collector Upload Form** (requires admin permissions).
3. Click **Choose File** and browse for the report.xml file you downloaded from CAM.
4. Click the **Upload** button to upload the file to DTP.

If you use SOAtest

For tests executed by SOAtest, the SOAtest XML report will need to be uploaded to DTP. See the "Uploading Rest Results to DTP" section in the Application Coverage topic in the SOAtest documentation for details.

Generating a Dynamic Coverage Data File and Uploading It to DTP

1. Ensure that dotTEST is properly configured, including DTP, scope and authorship settings. See Connecting to DTP, Sending Results and Publishing Source Code to DTP, Configuration.
2. Configure the following settings in the dottestcli.properties file in order to properly merge coverage data:
   - `report.coverage.images` - Specifies a set of tags that are used to create coverage images in DTP. A coverage image is a unique identifier

for aggregating coverage data from runs with the same build ID. DTP supports up to three coverage images per report.
- `session.tag` - Specifies a unique identifier for the test run and is used to distinguish different runs on the same build.
- `build.id` - Specifies a build identifier used to label results. It may be unique for each build, but it may also label several test sessions executed during a specified build.

3. Copy the runtime coverage and static coverage files to the same machine and run `dottestcli` with the following switches:

    - `-runtimeCoverage`: Specifies the path to runtime coverage that you download with CAM (see Coverage Agent Manager (CAM) section of the DTP documentation for details). You can provide a path to an individual .data file with coverage information from one testing session, or a path to a folder that contains many .data files from multiple testing sessions.
    - `-staticCoverage`: Specifies the path to the static coverage file (see Generating the Static Coverage File).

    ```
    dottestcli.exe -runtimeCoverage [path] -report [path] -publish -settings [path] -out [path] -
    staticCoverage [path]
    ```

Stopping Dynamic Coverage Data Collection

You can stop the process of collecting dynamic coverage data in one of the following ways:

1. Write `exit` in the open console when the following message will be printed to the output to stop dottest_iismanager:

    ```
    Write 'exit' and hit Enter to close dottest_iismanager
    ```

2. Send a request to the service by entering the following URL in the browser: `http://host:port/shutdown`

Stop dottest_iismanager only when all test sessions are finished. Application coverage will no longer be collected when the service stops, so it is important that dottest_iismanager runs continously while performing tests to collect coverage.

If any errors occur when dottest_iismanager exits, which prevent the clean-up of the Web Server environment, then execute dottest_iismanager with the `-stop` option to bring back the original Web Server environment and settings:

```
dottest_iismanager.exe -stop
```

Reviewing Coverage in DTP

You can use the Coverage Explorer in DTP to review the application coverage achieved during test execution. See the DTP documentation for details on viewing coverage information.

Known Limitations

- You can download coverage information that was collected in a test session. Coverage data collected when no test session was active cannot be downloaded.
- If multiple users are simultaneously accessing the same web application, the coverage data they collect may be mixed. To ensure that coverage is properly associated with individual users, the multiuser mode must be enabled (see Collecting Coverage from Multiple Users).
- The HTTP or HTTPS protocols are required to enable the multiuser mode, as the user-specific information must be provided within the HTTP header.
- In the multiuser mode, collecting coverage for WCF-based applications requires that they have the ASP.NET compatibility mode enabled.
- In the multiuser mode, the "default" user (the user who has not specified their ID) may collect extra coverage information from other users who are accessing the same web application.
- In the multiuser mode, assigning coverage collected for multithreaded application to individual users is limited. Coverage data for child threads is not assigned to the user who is actually accessing the application, but to the "default" user.
- Coverage data collected web initialization is not assigned to a specific user, but to the "default" user.
- The application coverage scope file cannot be used for WebSite projects as they may get recompiled by IIS server and change the name of the target assembly. Scope files can be safely used for Web Applications.