

## TECHNICAL BRIEF

# Functional Testing using *Studio Fx*

### Overview

IP Services are characterized by enormous complexity, constant change, and custom building blocks. Each IP Service has its own unique configuration, which makes the testing of them an unprecedented challenge.

### Challenges

Test teams at equipment manufacturers as well as network operators are struggling to meet the intense time requirements necessary to effectively perform functional testing of these services and their underlying products. Current approaches fall into two broad categories – Manual testing or using off-the-shelf protocol test tools.

Manual testing is proving to be slow and expensive and it offers inadequate test coverage. Off-the-shelf protocol test tools provide static test cases, are difficult to customize, and are mostly designed for load or performance testing and not for functional testing. These legacy tools are fundamentally outdated for functional testing of today's dynamic and complex IP Services.

### Solution from Mu Dynamics

Mu Dynamics provides a fresh, adaptive approach designed specifically for the testing of IP Services. As a key product of the award-winning Mu Test Suite, Studio Fx enables testers to gain a more than 10-fold effectiveness in their testing of IP Services. Using Mu Test Suite, operators and vendor quality assurance (QA) teams perform thorough functional testing of their IP Services at a fraction of the time and cost being spent today, while dramatically improving quality.

This document covers the following topics:

1. An overview of Studio Fx and functional testing
2. A description of several customer use cases for functional testing with Studio Fx
3. A walkthrough of creating and executing functional tests using Studio Fx



[www.mudynamics.com](http://www.mudynamics.com)  
[info@mudynamics.com](mailto:info@mudynamics.com)  
866-276-4640 toll-free  
408-329-6330 international  
408-329-6317 fax

## Functional Testing

Functional testing is a critical aspect of testing IP Services. It determines whether the individual features and functionality of the IP Service behaves as specified. In short, functional testing answers the question: “Does the IP Service work correctly?” Much of functional testing done today is either performed manually or using off-the-shelf protocol test tools.

### Challenges with Current Solutions

The problems with a manual approach are as follows:

- Time-consuming to create tests
- Results in poor test coverage
- Expensive

In situations where off-the-shelf test tools are available, they are often limited to only allowing the tester to run the specific canned tests supplied by the test vendor. The problems with using such tools are as follows:

- **Unrealistic emulation**  
Canned test cases are unrealistic and are not representative of the IP Service environment
- **Inflexible**  
Mostly built for packet-blasting and performance testing, such tools are difficult to customize for functional testing

### Benefits of the Mu Solution

- **Functional testing tailored to your IP Service**  
Instead of using static, canned test cases from the test vendor, Studio Fx allows you to rapidly create test cases modeled on your IP Service, starting from real packet captures. These packet captures can be from your lab network, from the production network or from third party sources such as [www.pcapr.net](http://www.pcapr.net). Functional tests created with this approach are inherently highly relevant, tailored to your service, and therefore dramatically improve test coverage.
- **Compressed test cycles**  
By starting with packet captures, Studio Fx compresses the test cycle time and helps your team complete testing faster and with less effort. There is no longer the need to write and maintain custom test scripts.
- **Built-in automation**  
Studio Fx helps increase the level of automation in the test process in multiple ways:
  - Studio Fx automatically re-creates message exchanges and auto-generates thousands of test cases with minimal manual effort
  - The tests generated are designed for reuse across the organization and shared between vendors and operators
  - The test case execution can be automated via popular API-driven test frameworks

“The majority of product launches are notorious for being late.”

“Test coverage is less than 40% for releases.”

Source: Sequent Learning,  
NSP Partners

## Test Creation Approach

The following characteristics make the Mu functional test solution unique:

- **Test Creation via Multiple Content Sources**

Studio Fx creates tests from a rich set of sources, including:

- **Customer content**

This input takes the form of captured real-world IP Service traffic from your lab environment or production network

- **Community content**

Through Mu's online community pcapr.net, you have access to the world's largest set of packet captures (pcaps) providing a rich set of traffic patterns that can be used for your own testing needs

- **Mu-provided content**

The Mu Test Suite comes with a base set of protocol message suites that provide you with a starting point for test creation



- **Stateful Scenario Re-creation**

Studio Fx takes this content and transforms it into a set of semantically meaningful, dynamically generated, stateful test cases. These test cases accurately represent the original communication exchange, and thus enable you to run the most relevant and realistic functional tests. Furthermore, you can easily define the pass/fail criteria for each test scenario and determine whether features are functioning as expected.

## Workflow

The following example explores the workflow using Studio Fx for the DPI functional testing use case.

This DPI example will demonstrate that Studio Fx users can rapidly define custom test content, typically within 30 minutes. With Studio Fx, there is no need to write custom scripts, and the resulting tests are a better fit to the unique, custom and chang-

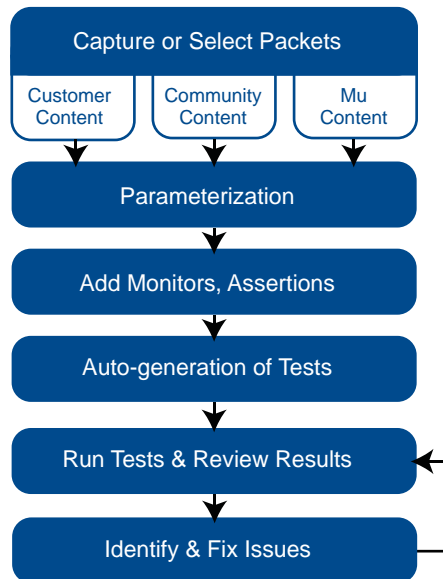
## Use Cases

Leading network operators and equipment makers are leveraging Studio Fx for their functional testing efforts. Below are some examples of how Studio Fx is used in various IP Services domains:

- In the **Mobile LTE/SAE** space, Studio Fx is often used for testing the billing and charging operations between the systems in the evolved packet core. This involves DIAMETER messages with various AVP (Attribute Value Pair) combinations being parameterized during tests. Before Studio Fx, testers struggled to use inflexible off-the-shelf tools and then resorted to manual scripting. With Studio Fx they were able to automate the process and achieve greater test coverage.
- In the **VoIP** space, many test teams use Studio Fx for testing voice signaling and call control operations. Here the tester parameterizes elements within a VoIP call (e.g., Skinny, SIP), and cycles through different extensions to test different call flows and functions. Before Studio Fx, the test team was spending over 50% of their total time doing this manually. With Studio Fx they dramatically reduced this to a small fraction of their time.
- In **Routing**, Studio Fx is instrumental for testing features like BGP transitive attributes. Here, various combinations of BGP attribute update messages are tested to determine if route flaps occur as expected. Previously, it took a team of 3 testers over 3 months to develop and perform this testing manually. With Studio Fx the effort was reduced to just 1 tester working less than 1 week.
- In **Deep Packet Inspection (DPI)**, testers often use Studio Fx for validating features like URL classification, content filtering and lawful interception. Here, various types of HTTP/P2P traffic are sent through the DPI to verify if the rules built into the detection engines work as expected. Today's IP networks are seeing a rapid explosion in types of traffic profiles as well as new threat patterns. Before Studio Fx, testers used canned test cases or manual tests which resulted in poor test coverage. With Studio Fx, testers have access to a rich set of inputs, including input from the IP Service itself, thus enabling the tester to stay on top of the latest traffic profiles and threats.

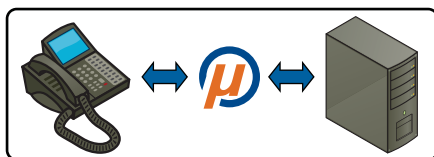
ing product or service attributes. These test cases are based on a model of the IP Service that is following the exact sequence of steps — even involving multiple protocols — as in the real service. The test cases all include their own pass/fail criteria so they run unattended and can be re-run whenever results don't match expectations.

The workflow of using Studio Fx for DPI functional testing includes the steps shown below:



## 1. Capture

You can obtain test content by capturing IP Service traffic by placing the Mu Test Suite in bridge mode in the lab as service elements (e.g. router, gateway, server, and appliance) communicate to each other. Flow selection is used to extract specific flows of interest from a large capture.



It is important for you to start with a set that is focused on the aspect of the IP Service that you need to test, e.g., VoIP phone registration, VoIP phone call setup and teardown, BGP route update, etc. Extraneous packets should be eliminated, and the Flow Selection tool is provided to do this.

Once the test content is ready, it is uploaded to Studio Fx.

## Features at a Glance

**Capture:** You can use Mu Test Suite to capture packets by turning on bridge mode or span mode and placing the Mu in-line on the lab network. Once the packets are captured Studio Fx enables you to select flows of interest by providing a user interface that displays the traffic by host IDs, flows and packets.

**Stateful Scenario Re-Creation:** Studio Fx offers the unique capability to parse packet captures and re-create the exchange in a stateful manner. The transports are re-created and the relationships between fields such as lengths and checksums are automatically computed, making it the most realistic test possible.

**Parameterization:** Parameterization consists of selecting fields and converting them into one of two types – options or variables.

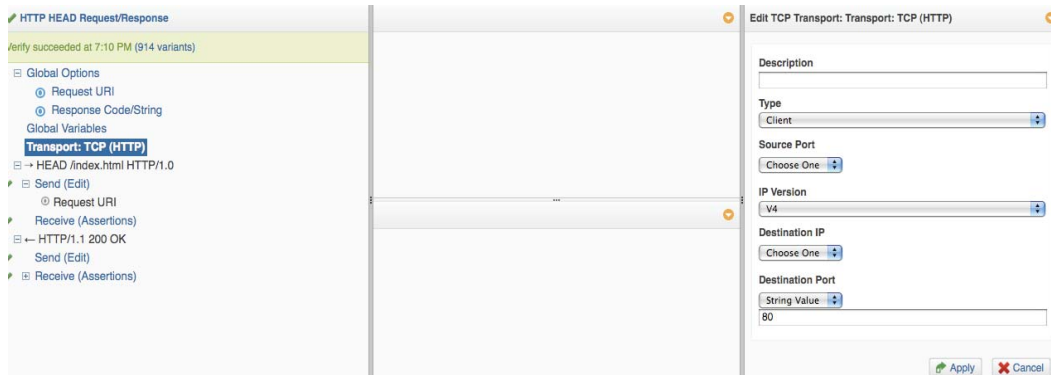
- **Options:** Options or parameters allow you to supply thousands of dynamic data sets through the UI or via a spreadsheet (.csv) file. You can also use Mu Functions to auto-generate thousands of relevant modified values, thus dramatically improving test coverage. Examples of these fields are IP addresses, email addresses, SIP URIs, Call IDs etc.
- **Variables:** In many exchanges there is a need to reuse received values in a later part of the scenario during a subsequent Transmit. Studio Fx allows you to define variables for any received value and re-use it later in the scenario. Examples of this include session ids, cookies and tokens.

**Channels for Monitoring:** You can set up channels over SSH to monitor key test criteria on various devices and servers in the test environment. These channels can be used for test setup and teardown as well as for asserting pass or fail conditions based on custom commands. The channels are inserted as a test step along with other test steps in the scenario.

**Assertions:** Assertions are a way to validate expected results in a test by adding custom pass/fail criteria. You can select a field or a set of fields and add an Assertion that compares the received value in that field to an expected value that you specify. Examples include HTTP “200 OK” responses, BGP Error Code fields, RADIUS response code fields and any other response a peer device might send to a request.

## 2. Stateful Scenario Re-creation

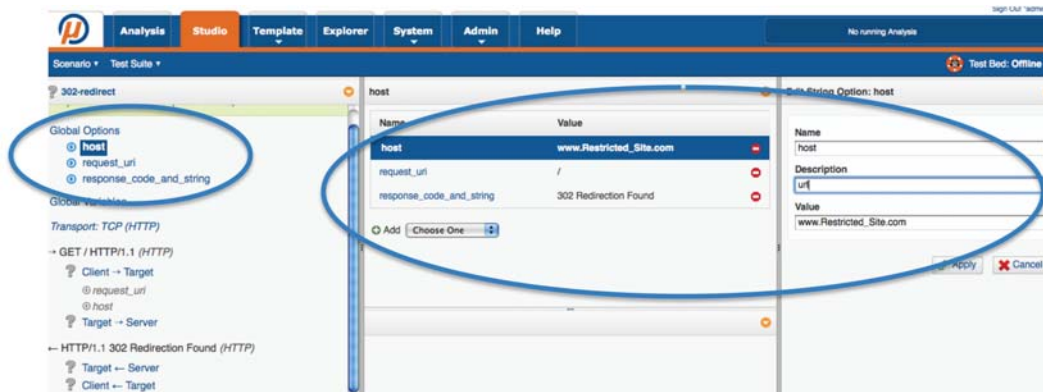
Studio Fx automatically parses the contents, assigns appropriate transports and allows you to edit the payload. In this example, the user has selected an HTTP HEAD request transaction and will run tests for this scenario.



## 3. Parameterization

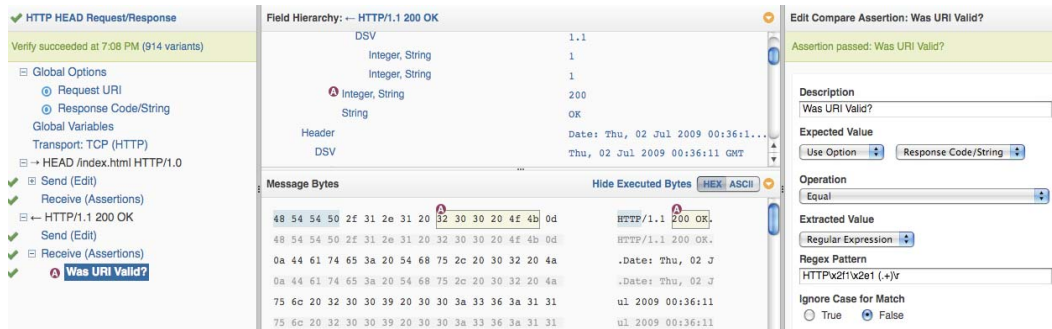
- **Options:** You can optionally parameterize fields and convert them into Options. These options are driven by user-provided data that can be supplied via the UI or by using an external spreadsheet (.csv file). Mu functions are used to auto-generate hundreds or thousands of additional tests based on the one base test-case, thus greatly expanding test coverage.

In this example the HTTP URL field as well as the username and password fields are being parameterized. Depending on the username as well as the URL being requested, the content inspection device will either allow or block the transaction.



- **Variables:** You can optionally convert fields that are defined within incoming messages into reusable Variables. For example, if the target defines a session ID, Studio Fx will be able to refer to the correct session ID later on in the conversation within this test case. Doing so maintains the state and ensures that each test case's new session is created to be functionally indistinguishable from the session in the original pcap.

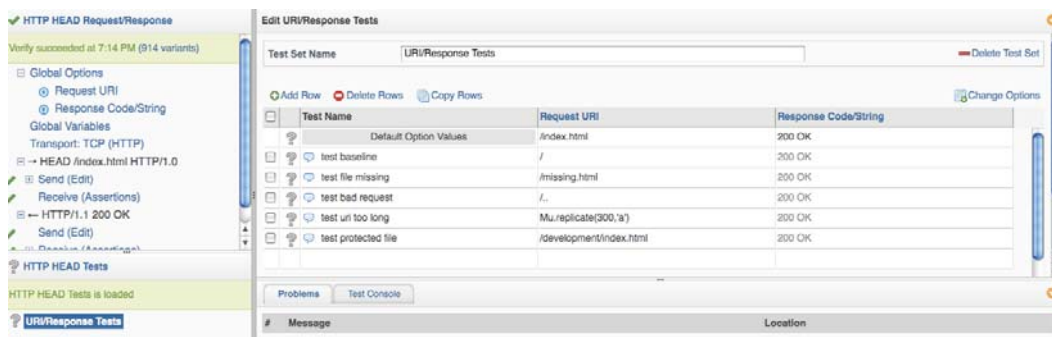
Here, the user creates a test suite that comprises of multiple option value pairs. These parameterized option fields will now be driven by Mu Functions or by customer-supplied data against a target service.



#### 4. Monitors & Assertions

You then can create SSH channels to other hosts on the testbed to kick-start external setup or teardown functions. These SSH channels can also be used to monitor key aspects of the test being run to flag pass/fail conditions.

In addition, you can add assertions that check for specific response fields within any incoming message and then flag a failure if the assertion fails.

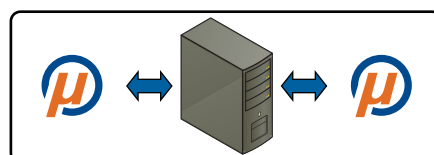


#### 5. Verify and Save

You then verify that the Scenario just created will execute successfully against a live target from the UI. The UI is highly interactive and displays both the outgoing and the incoming messages, enabling a quick way to verify test execution. You can then save the Scenario and the Test Suite.

#### 6. Run/Execute

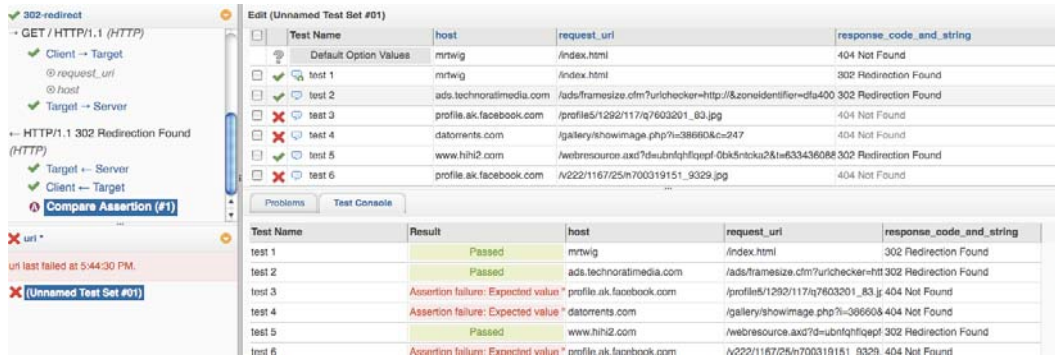
This Scenario template is now run against the live target. Studio Fx can act as a client or it can emulate both the client and the server on both sides of the system under test, thus acting as a testbed-in-a-box to test the system in the middle. The tests can be run from the UI or through a remote automation API. The Scenario template can also be shared across the organization and it becomes a re-usable and configurable test asset helping organizations scale their testing over time. Examples of this are reusing the template for regression testing and other types of testing such as Resilience testing or Security testing.



## 7. Review Results

Now you can review results, and rerun the failed tests in debug mode.

At the end of the test, results are summarized into a detailed html report with pass-fail tags being applied for each test case. This report can then be shared across the test organization.



The screenshot displays the Studio Fx test results interface. On the left, a tree view shows the test structure, including a '302-redirect' test set and its sub-components like 'Client -> Target' and 'Target -> Server'. The main area shows a table of test results for 'Unnamed Test Set #01'. The table has columns for Test Name, Result, host, request\_url, and response\_code\_and\_string. Test 1 and 2 passed, while tests 3, 4, and 6 failed with 'Assertion failure: Expected value' errors. Test 5 passed. The interface also includes a 'Problems' and 'Test Console' section at the bottom.

Test Name	Result	host	request_url	response_code_and_string
Default Option Values		mrtwig	/index.html	404 Not Found
test 1		mrtwig	/index.html	302 Redirection Found
test 2		ads.technoratimedia.com	/ads/framesize.cfm?urlchecker=http://&zoidentifier=dtb400	302 Redirection Found
test 3	Assertion failure: Expected value	profile.ak.facebook.com	/profile/1292/117/q/603201_83.jpg	404 Not Found
test 4	Assertion failure: Expected value	datorents.com	/gallery/showimage.php?i=38660&c=247	404 Not Found
test 5		www.hih2.com	/webresource.axd?id=ubnqhfqpf-0bk5ncka2&l=633436088	302 Redirection Found
test 6	Assertion failure: Expected value	profile.ak.facebook.com	/v/222/1167/25/n/700319151_9329.jpg	404 Not Found

## Summary

Studio Fx radically improves functional testing by providing a new way to rapidly define and automate functional testing tasks that are difficult to do manually, yet which are critical for the health of your unique, ever-changing multi-vendor IP Service environment.

It is critical for operators and vendors to address this dimension of testing in order to be able to eliminate the skyrocketing costs caused by field outages and field reported bugs. A proactive approach to testing IP Services is needed to prevent customer dissatisfaction, churn and field fire drills.

As a key component of the Mu Test Suite, Studio Fx helps vendors and operators save valuable time and money in their IP Service development and deployment, while significantly improving quality.

