

TECHNICAL BRIEF

Interoperability 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 interoperability testing of these services and their underlying products. The current approach is mostly manual testing, which results in delayed releases, constant firefighting after the release of the product and skyrocketing costs to both vendors and operators alike.

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 interoperability testing of IP Services. Using the Mu Test Suite, operators and vendor quality assurance (QA) teams perform thorough interoperability 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 interoperability testing
2. A description of several customer use cases for interoperability testing with Studio Fx
3. A walkthrough of creating and executing interoperability tests using Studio Fx



www.mudynamics.com
info@mudynamics.com
866-276-4640 toll-free
408-329-6330 international
408-329-6317 fax

Interoperability Testing

Interoperability testing is a critical aspect of testing IP Services. Test teams have to deal with a list of other devices and software that are required to interact with their product on a production network. In addition, for each of these systems there usually are a number of older versions that also need to be test against to ensure compatibility. Much of interoperability testing done today is manual.

Challenges with Current Solutions

The problems with a manual approach are as follows:

- Time-consuming to create testbed and tests
- Limited test coverage
- Expensive

Benefits of the Mu Solution

The interoperability test solution from Mu Dynamics provides the following unique capabilities:

- **Impersonation of 3rd party devices/systems**
Using Studio Fx, vendors and operator test teams can impersonate any 3rd party or internal system. Starting from captured traffic, testers can rapidly and accurately recreate the message exchange.
- **Field issue replication**
Field issues are often the cause for delaying new product releases and for damaging the reputation of services with customers. Studio Fx helps improve the process of identifying field issues and testing fixes by bringing in captured traffic and quickly recreating the original scenario. Operators send captured traffic that represents problems to the vendor for accurate re-creation. Reproducing an issue takes up the most significant portion of the time spent in developing a fix and this can be significantly reduced by using Studio Fx.
- **Avoid expensive lab build-outs**
Vendors and operators can avoid expensive multi-vendor lab build-outs by using Studio Fx to replicate multiple instances of 3rd party systems. This significantly saves capital and operational expenditure as the test team needs only to purchase one instance of the 3rd party system.

Test Creation Approach

The following characteristics make the Mu interoperability 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.

- **Field driven content**
This input takes the form of captured real-world IP service traffic from the customer's production network environment.



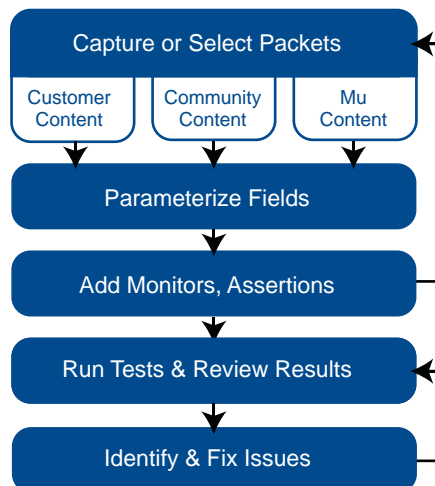
- **Stateful Scenario Re-creation**
Studio Fx takes this content and transforms it into a set of semantically meaningful, dynamically generated, stateful test cases. These test cases accurately represent the original communication exchange, and thus enable you to run the most relevant and realistic interoperability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

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

Example Workflow

This DPI example will demonstrate that Studio Fx users can rapidly re-create client traffic, typically within 30 minutes. With Studio Fx, there is no need to manually setup clients and client applications, and the resulting tests are highly accurate and scalable. 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 interoperability testing includes the steps shown in the diagram on the right:



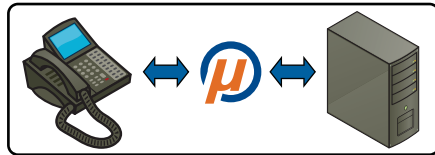
Use Cases

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

- In the **Mobile** space, Studio Fx is used for **replicating field issues**. Starting with pcaps, the testers can statefully replicate specific messages that their gateway device was unable to process on the production network. *Before Studio Fx, test teams struggled to use inflexible off-the-shelf tools and ultimately resorted to manual scripting to recreate specific patterns seen on their network. With Studio Fx they were able to automate the process and accurately recreate flows to test.*
- In the **VoIP** space, test teams use Studio Fx for testing soft-switches and session border controllers. Here the tester use Studio Fx to **emulate various IP elements** such as SIP phones, media gateways, IAD and IMS network elements. Since the soft-switch needs to interoperate with a large number of 3rd party systems, Studio is an integral part of their testing. *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 addition the organization sees significant cost savings, as they scale this testing to other labs and developers without having to purchase additional 3rd party systems.*
- In **Security Gateways** and **Deep Packet Inspection (DPI)** devices, testers use Studio Fx for **emulating various types of clients** by having Studio Fx replay the client messages statefully. Features like proxy authentication, web caching, traffic classification and access control can be tested without having to maintain and configure actual clients for every test run. This is especially useful during regression runs when large numbers of clients can be tested in an automated fashion. *Before Studio Fx, testers had to manually configure and exercise various clients in order to test the Gateway device. This proved to be extremely time-consuming and resulted in testers having to limit their testing due to time to market pressures. With Studio Fx, testers can quickly and accurately emulate clients, thus enabling the tester to stay on top of the growing matrix of clients and applications.*

1. Capture or Select

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. Studio Fx can also be used to replicate a field issue by selecting the packet trace associated with the field issue.



It is important for you to start with a set that is limited to the exchange that you need to test. Extraneous packets should be eliminated, and the Flow Selection tool is provided to do this. It is also important to limit the exchange to 15 to 20 individual messages that are targeted to test one transaction – e.g., VoIP phone registration, VoIP phone call setup and teardown, BGP route update, etc.

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

2. Stateful Scenario Re-creation

Studio Fx assumes the role of the client and statefully re-creates the scenario. It 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

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.

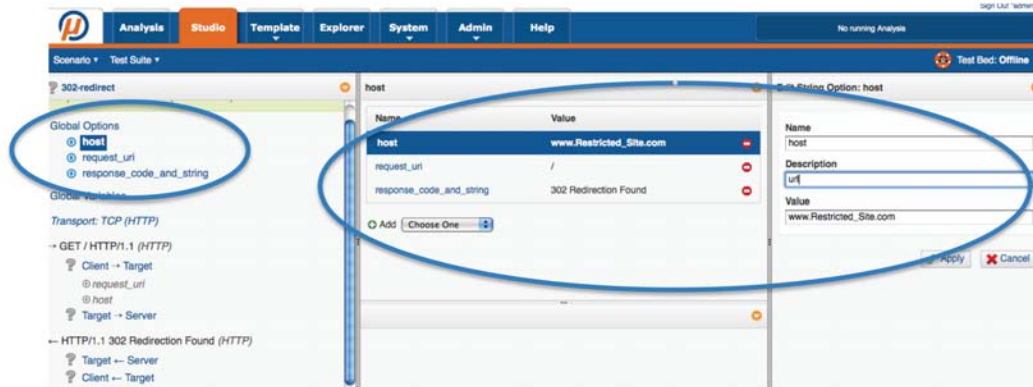
- **Options:** Options or parameters allow you to supply thousands of dynamic data sets through the UI or via a spreadsheet (.csv) file. You can also use Mu Functions to auto-generate thousands of relevant modified values, thus dramatically improving test coverage. Examples of these fields are IP addresses, 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.

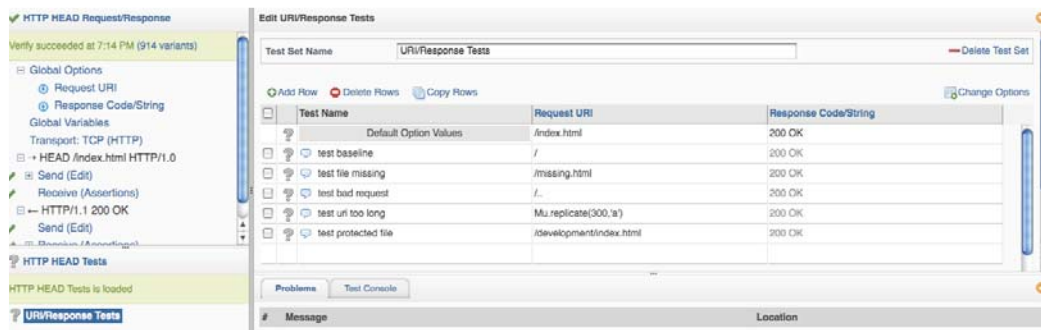
Functions are used to expand the number of test cases available, starting from the one base test case.

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 part of incoming messages and make them reusable Variables throughout the Mu Test Suite. For example, if the target defines a session ID, Studio Fx will be able to refer to the correct session ID later on. Doing so maintains the state and ensures that the 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.

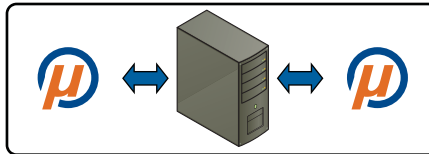
The screenshot shows a test execution interface. On the left, a tree view shows the test steps: 'Verify succeeded at 7:00 PM (914 variants)', 'Global Options', 'Request URI', 'Response Code/String', 'Global Variables', 'Transport: TCP (HTTP)', 'HEAD /index.html HTTP/1.0', 'Send (Edit)', 'Receive (Assertions)', 'HTTP/1.1 200 OK', 'Send (Edit)', 'Receive (Assertions)', and 'Was URI Valid?'. The central pane shows the 'Field Hierarchy' for 'HTTP/1.1 200 OK' with fields like 'DSV', 'Integer, String', and 'Header'. Below this is a 'Message Bytes' section showing hex and ASCII representations of the message. The right pane is titled 'Edit Compare Assertion: Was URI Valid?' and shows the assertion configuration, including a description, expected value, operation (Equal), and a regular expression pattern: 'HTTPx2f1w2n1 {+}r'.

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. At this point, you can go back and capture or select pcaps from other scenarios or systems that are of interest. In this manner you can build a library of scenarios representing various 3rd party or in-house clients, versions and field reported scenarios.

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.

The screenshot shows a test results review interface. On the left, a tree view shows the test steps: '302-redirect', 'GET / HTTP/1.1 (HTTP)', 'Client -> Target', 'Request URI', 'Host', 'Target -> Server', 'HTTP/1.1 302 Redirection Found (HTTP)', 'Target -> Server', and 'Client -> Target'. The main area is titled 'Edit (Unnamed Test Set #01)' and contains a table of test results.

Test Name	host	request_uri	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://&zoneid=400	302 Redirection Found
test 3	profile.ak.facebook.com	/profile/51292/117/q/7603201_83.jpg	404 Not Found
test 4	datorrents.com	/gallery/showimage.php?i=38660&c=247	404 Not Found
test 5	www.hih2.com	/webresource.axd?d=ubn1qhqepf-Obk5ntcka2&t=633436088	302 Redirection Found
test 6	profile.ak.facebook.com	/v222/1167/25/n/700319151_9329.jpg	404 Not Found

Below the table, there is a 'Problems' section with a 'Test Console' tab. The console shows a list of test results with columns for Test Name, Result, host, request_uri, and response_code_and_string.

Test Name	Result	host	request_uri	response_code_and_string
test 1	Passed	mrtwig	/index.html	302 Redirection Found
test 2	Passed	ads.technoratimedia.com	/ads/frameSize.cfm?urlchecker=htt	302 Redirection Found
test 3	Assertion failure: Expected value	profile.ak.facebook.com	/profile/51292/117/q/7603201_83.j	404 Not Found
test 4	Assertion failure: Expected value	datorrents.com	/gallery/showimage.php?i=38660&	404 Not Found
test 5	Passed	www.hih2.com	/webresource.axd?d=ubn1qhqepf	302 Redirection Found
test 6	Assertion failure: Expected value	profile.ak.facebook.com	/v222/1167/25/n/700319151_9329	404 Not Found

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.

Summary

Studio Fx provides a new way for you to rapidly address your interoperability testing needs that are critical for the health of your unique service in an ever-changing multi-vendor IP environment. It helps you generate these tests significantly faster than the existing manual methods. It also helps cut the costs of acquiring additional 3rd party systems.

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 product 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.