

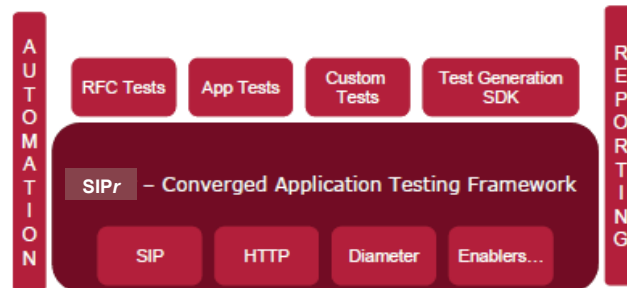


Application Testing AGNITY SIPr

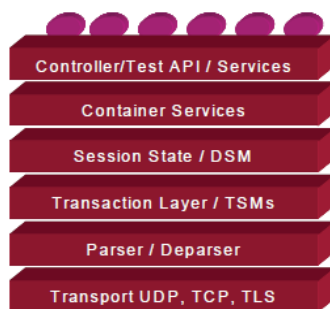
With the emergence of IMS, Web 2.0 and SOA, a new breed of converged applications and services are mushrooming in the market. The quality assurance of these applications by means of testing is posing a great challenge for manufacturers and service providers. In addressing this gap AGNITY has taken a unique approach and created an Application Testing Framework called SIPr. SIPr provides an application server style programming model and offers a range of constructs using Ruby, a very powerful yet simple and user-friendly programming language. SIPr draws its power from the flexibility and configurability supported at each layer of its protocol engine, thus facilitating creation of any complex testcase for these converged applications.

Key Components

The figure at right shows the high level components of the SIPr platform. At the center are the SIPr protocol engine and the application framework. Around it are several modules that provide functionality like tooling to create, manage and deploy tests and controllers, pre-packaged tests and applications, an API for controller and test generation and some helper scripts around automation and reporting.



Controllers and Tests



SIPr Architecture

SIPr is a layered framework and at first glance looks very similar to any SIP stack. Where it is different and actually shines is the level of flexibility and extensibility it offers to the users. That's the reason why it is ideally suited for application testing. The figure below shows the internal layers of SIPr architecture. On top of this layered architecture runs the "Controllers and Tests." Controller is a piece of code that the user would write and deploy on the SIPr platform. It contains the business logic to decipher the incoming protocol messages and send out its own messages. A Test on the other hand is again a deployable piece of code that contains a controller in it and besides doing all the messaging it also validates the call flow and messages.

Key Features

1. SIPr is pluggable at each layer, starting from transport layer one can plug-in a transport filter chain. These filters can be applied on incoming or outgoing messages and these filters can be defined both at the global SIPr level affecting all messages and also at the individual test-case level. These filters can be used for simulating transport level features such as simulating packet drop, content modification, etc.
2. SIPr has extensive in-built parsers for most SIP headers; users can not only define their own parsers for new/custom headers but also override the existing parsers to address custom extensions or even create malformed headers for testing.
3. SIPr's transaction handling is FSM (Finite State Machine) based and SIPr provides simple method for controlling transactional behavior by providing a mechanism of call back handlers. Users can completely control the transaction state machine and also modify it using these callbacks. Besides this all transaction timers can be configured at global, or controller or session or even down to individual message level, allowing users to create various timer sensitive test cases.
4. Actually almost everything is exposed as configuration at all levels with appropriate defaults. Everything in the protocol engine can be changed at SIPr level, Controller level, Session level, Message level.
5. The Session layer offers a combination of rich API set and configurations for controllers and tests to create/manage messages and sessions (dialogs).
6. The container services are responsible for controller deployment/selection, message routing, timers, logging, session recording, flow validations, test lifecycle, thread management, configuration management, etc.



Functionality to be Tested with SIPr

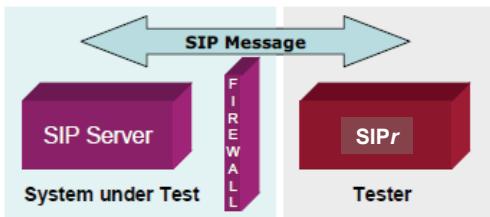
SIPr is an application development framework. It is a platform which exposes an API for writing converged applications. These applications, so called Controllers, are written and deployed on the SIPr framework that manages the protocol sessions and messaging.

Although conceptually a controller is similar to a Servlet in Java, SIPr offers a much richer and powerful API than a Servlet. Through its openness and flexibility almost any converged application – consisting of SIP, HTTP, Diameter and RTP – can be tested with SIPr. The typical models include:

- SIP UA, SIP B2BUA
- Proxy Server, SIP Registrar
- SIP Subscribe/Notify, SIP MWI
- Privacy mechanisms for SIP
- Service Route Header, etc.
- HTTP requests and responses
- RTP Media testing with multiple codecs

Prebuilt Tests and Test Management

SIPr supports augmentation of prepackaged, plug-and-play testcases. This rich collection of ready-to-use testcases and device profiles draws on the wealth of expertise and know-how of the users and developers of SIPr.



The comprehensive prebuilt testcases let the user get started with any testing exercise in a development lab and scale up to QA and IOT lab. The testcases can be configured to run in an automated environment. The accompanying reporting infrastructure provides an easy-to-use mechanism for storing and displaying the summary of the test results.

Standards Supported

A number of RFCs, such as RFC 3261 (SIP), 3262, 3265 (SIP Specific Event), 2327 (SDP), 2617 (Digest), 2833 (DTMF), 1889 (RTP), are natively supported by SIPr.

Since SIPr is an application development framework, complying with a new RFC that does not change any basic protocol behavior is simple work. Even for stack-affecting changes, hooks are provided at each layer to modify the default behavior of the stack by writing extensions.

Operating Environment

Operating System supported: Windows
Linux Language supported: Ruby

Conclusion

SIPr 1.0 is rolled out with the goal of helping the community of users optimize their efforts and time in testing their most complex converged applications.

The accompanying user documentation and training from the experts at AGNITY helps jumpstart the whole test automation process. In addition, a number of value-add enhancements, including full support for Diameter, are planned in an upcoming product release that will further enhance the richness of this testing process.



About AGNITY, Inc.

AGNITY is a global technology and consulting services firm dedicated to helping businesses, network equipment manufacturers and service providers design, develop and deploy application solutions. AGNITY ACONYX™ is the industry's leading Service-Oriented Communications Architecture (SOCA) enabling communication service components to be rapidly and cost-effectively deployed within the applications infrastructure. With more than 125 employees in Silicon Valley, Plano, Toronto and Noida, AGNITY is one of the only firms of its kind providing global support exclusively on application design, solution development and application migration and deployment. AGNITY has forged strong relationships with equipment providers and service delivery platform (SDP) vendors, serving more than 25 customers, including five Tier 1 service providers. For more information, visit www.agnity.com.



AGNITY, Inc. Headquarters

42840 Christy Street Suite 102
Fremont CA 94538 USA
Tel: 510-270-2669
Fax: 510-353-1302
Email: info@agnity.com

www.agnity.com