White Paper

Scapa Technologies

Scapa® Test and Performance Platform

- Stress Testing
- Load Testing
- Capacity Test
- Soak Testing
- Scalability Testing
- Performance Testing
- Benchmarking
- Reliability Testing
- Performance Tuning
- Performance Optimization
- Bottleneck Identification
- Maximizing User Densities
- Migration Testing
- Performance Comparison
- Server Consolidation Testing
- Diagnostic Testing
- Right Sizing Systems
- Service Availability

Performance Testing for BMC Remedy®

IT Service Management Suite

bmcsoftware

bmcsoftware TAP Validated

bmcsoftware TECHNOLOGY PARTNER

CONTENTS

Executive Summary	3
Introduction	4
ITSM Testing	5
Analysis of System Performance	5
The Edge	5
The Envelope	5
The Signature	6
The Impact of Change	6
Performance Analysis of Remedy Architectures	6
Defining Synthetic Load	6
Baseline Service Performance Test	7
Sanity Check	7
Synthetic Load Calibration	8
Driving Synthetic Load	8
BMC Remedy ITSM and Scapa TPP	9
Integration Points	9
Remedy API Complexities	10
ITSM Integration	12
Other Scapa TPP Features	13

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

EXECUTIVE SUMMARY

BMC Remedy Action Request System
IT Service Management Suite
http://www.bmc.com

Market coverage in more than 124 countries; Major offices located in Houston and Austin, TX; San Jose, CA; Boston, MA; Amsterdam; Singapore; Tel Aviv, Israel; and Pune, India; Approximately 6,000 employees worldwide; Member of S&P 500

In addition to its unique level of integration with BMC Remedy Action Request Systems at the C and Java AR Server APIs, Scapa Test and Performance Platform (TPP) also provides integration with BMC Remedy IT Service Management Suite (ITSM) at the HTTP layer. This is provided in a single, unified tool under a single licence.

Scapa TPP significantly reduces the complexity and cost of performance testing, and thus reduces the risk of changing and growing a Remedy environment to reflect the needs of a dynamic business.

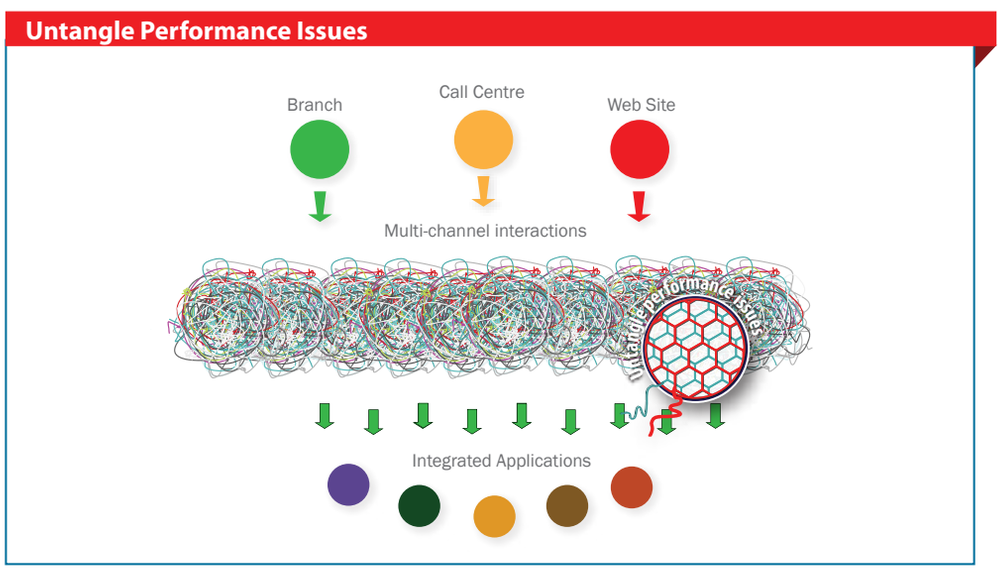
NOTE: Scapa TPP supports all versions of BMC Software's Remedy ITSM

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

INTRODUCTION

Introduction

Remedy is commonly used within organizations to manage customer interactions through one or more channels, such as call centres or the Internet, through one of the ITSM applications. Its versatility is underlined both by its ability to be customized and its large domain coverage. Remedy applications typically represent the layer interfacing with customers and play a key role in delivering and supporting business services driven by a large user base.



Therefore, these applications can be defined as “mission-critical” and are required to be highly reliable, scalable and flexible. The applications may have to be adapted quickly to changes in the size of the user base, the type of usage and the required workflows supporting a diverse set of usage scenarios. This set of requirements demands a unified testing methodology that supports testing at all stages in the lifecycle of an application and allows the testing to be adapted as quickly as the applications themselves.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126| Tel. UK : +44 (0)131 208 0652

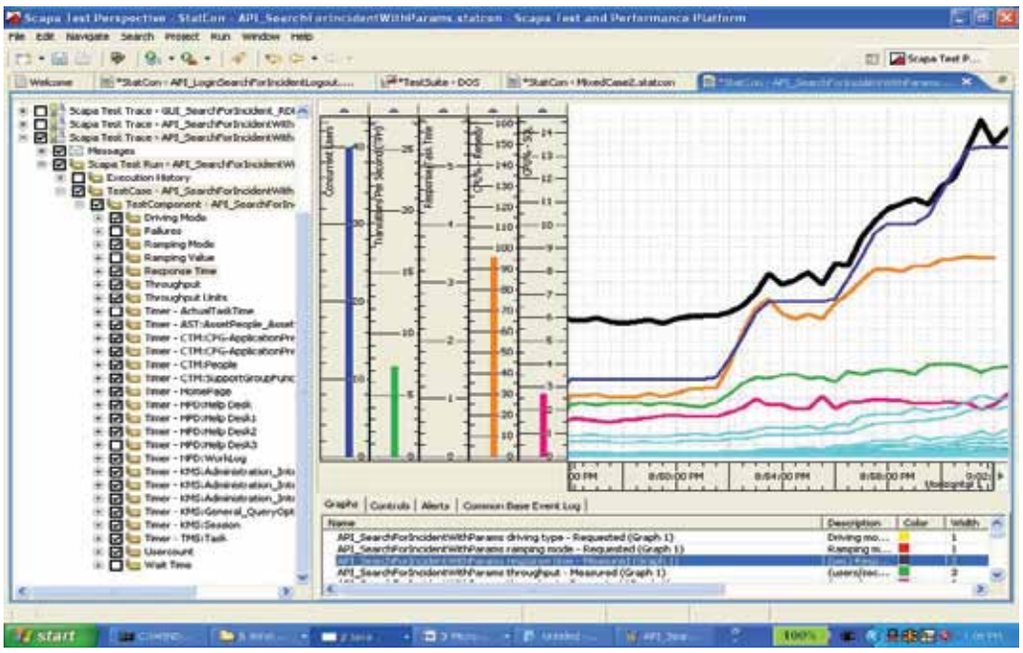
BMC Remedy ITSM Testing

BMC Remedy ITSM Testing
Analysis of Systems Performance

The role of performance testing and analysis is, as the system is changed or scaled up, to ensure it is sufficiently responsive before customers are exposed to it. In addition, the same technology can be used to ensure that a live system is performing well under the load applied by real users.

The Edge

All systems “break” under sufficient load. Typical behavior is shown in the following diagram (see Scapa TPP screen shot) for a system driven with a controlled throughput, a constant think time and a user count which is allowed to vary. In this example, the system response time (black line) increases beyond acceptable and sustainable levels when throughput (orange line) is driven above capacity.



The Envelope

For most systems, as shown above, there is a sharp edge beyond which the performance of the system is not acceptable. The number of users or transactions per second that this corresponds to will vary, however, for different user transactions. If the variability in the transaction mix is represented as a multidimensional workload space, there is part of that space (we refer to this as an “envelope”) for which system performance is acceptable, and outside of which performance becomes unacceptable.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126| Tel. UK : +44 (0)131 208 0652

The Signature

Systems “break” for a reason (or reasons), typically because one or other system resource saturates or serializes. It is this that determines the location of the edges of the “envelope” of acceptable workloads in the space of possible workloads. Associated with the edge of the envelope is a “signature” of associated system metrics. In the diagram, the red line is CPU, the light blue line is the application server queue and the dark blue line is the user count (the overall number of users resident in the stack). The signature of the edge, in the above example, is that the application server queue length is around 5, and CPU is around 90%.

There is usually a small set number of transactions that show a set of differentiated “signatures”. That is, they cause the architecture to fail for a set of identifiably different reasons. These signatures can be used to map out the edge of the envelope.

The Impact of Change

In a changing environment, the objective is to ensure that the current workload of the system is always within the envelope. Changes to the user activity may cause it to spill out of the edges of the envelope. Improvements to the system may then allow the envelope to expand to compensate. However, poorly implemented changes to the system may actually reduce the size of the envelope.

Performance Analysis of Remedy Architectures

The basic requirement is to map out the edge of the envelope and determine whether production usage is within the envelope and, where appropriate, make amendments to the system to expand the envelope. This can be done by testing a production or test system, and/or monitoring the production system.

Defining Synthetic Load

One of the initial performance testing activities is defining a suitable load that would need to be recreated in a test to simulate user activity. Reliability of the test results would depend on the quality of the test load definition. A number of different approaches are possible:

Benchmark Kits.

Normally restricted to a single ITSM application and dependent on a specific version, configuration and data. While this approach may reduce overall time needed for testing, test results are often of little value as they do not reflect the real environment or real user data, ignoring one of the most important factors of the overall system health and performance. Using existing scalability figures for different architectures would provide a similar level of information.

Replaying a single hour/day load.

This approach will try to record and then replay a live system load. These are usually one-off tests, as they depend on data at one specific point in time. These tests could be quite complex to complete and are not suitable for performance problem analysis or system optimization as there is no clear understanding of the user activity that is being recreated to produce results.

Performance Testing based on User Acceptance Testing (UAT) scripts.

While providing good coverage of the application functionality, this approach is often time consuming without providing any proportional benefits. This type of testing does not take into account any specifics of the system architecture and system usage.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

Application Specific load.
The first stage of the process is to understand the main characteristics of the systems being deployed, as most performance problems arise from the complex relationship between applications, their usage patterns and the infrastructure on which they are deployed. The selection of the transactions to be modelled would, therefore, be based not only on the frequency and importance of individual actions but also the underlying system architecture and data being used. One suggested approach is to start with a small, simple set of transactions and increase the complexity and accuracy of the model with every test iteration. While providing initial results very early, it also simplifies the process of problem analysis and resolution.

The key is doing testing sensibly. There is no point in generating a lot of information that cannot be acted upon, or information that comes too late to impact on decision making processes. Additionally, the cost of the testing process needs to be in proportion with the its value to the business.

Baseline Service Performance Test

The objective of a Baseline Service Performance Test is to find out how the system behaves at a very low level of activity. If performance problems are seen at this level and at this stage, there is time to rectify issues with the application or the infrastructure.

Typically this test is performed as early as possible on the first available environment, and may be repeated at various times during the project and would be based on a few Diagnostic Transactions (e.g. log-in, logout and, perhaps, a read-only transaction like application console load).

If the End-to-End Service metrics are out of the ballpark or acceptable estimates, the Operational Metrics can be used to analyze root cause. Some linear extrapolations can be made on the Operational Metrics, for example, Memory and CPU usage, to give early warning of Capacity problems.

Sanity Check

This may be viewed as an initial testing activity to undertake some preliminary tests to check that the system is relatively sound whilst there is still time for it to be fixed. The focus is normally on gross features of individual ITSM applications, and as a rule of thumb, the following five very simple transactions (to be tested in the following order) will show up most of the fundamental problems:

Login: a transaction which logs a user into the system, authenticates them and then logs them out.

Read: a simple transaction that may be loading one of the ITSM consoles and retrieves a set of results and details, or similar activity that involves accessing persistent data in the system.

Create/Update: a transaction that creates a new entry and/or changes an existing one.

Search: one or more searches providing a set of specific criteria.

Batch: Repeating above test while triggering some of the background batch process that updates the database.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

The philosophy here is not to determine whether the system will perform according to some specific business performance metric, but whether the performance of the environment is in the right ballpark to give confidence. For example, if we expect to create 1000 new incident requests per hour at the peak load and can create only 50 when using a simple workload model, there is a problem that would need to be addressed.

It is often also useful to build tests around transaction rates, (i.e. updates per second etc.) rather than user count.

At this point it is also worth looking at interference amongst transactions in individual applications and, in the case of multiple applications, amongst the applications.

Synthetic Load Calibration

A good starting point may be a set of Sanity Check transactions adjusted to take into account business requirements or levels of activity observed or expected. These are often expressed as the maximum number of users that are accessing the system, the frequency of login/logout and the number of new incidents, changes or similar. This level of information is usually easily accessible but further refinement of a workload might be more difficult. On existing systems, the SQL level activity may provide good metric for work model calibration. The system utilization level of production systems can be used for additional validation. For example, CPU utilization on the production system with live users should be similar to CPU utilization during the test run on equivalent load and on comparable systems. Additional test transactions may need to be added to the workload at this stage.

Driving Synthetic Load

At the core of the problem is the requirement to drive a synthetic load into the Remedy server. The synthetic load is created by virtual (i.e. simulated) users, each of which is performing transactions which, as far as Remedy is concerned, are identical to the activities of real users.

It is necessary to control how many users there are, what they are doing and how often they are doing it. Actions are typically collected through tracing the activity at the HTTP protocol level, and “replaying” that trace by using a test tool to exercise the interface in an equivalent manner. However, it is important to make sure that the actions specified to the interface are actually valid - i.e. they make sense in terms of the state of the workflow and data in the Remedy system at the point in time at which they are replayed And it is typically necessary to introduce variability into the workload to avoid spurious effects from unnatural caching and/or contention in the database or application layers.

During the execution of the synthetic load, it is common to monitor systems performance metrics at a range of points in the Remedy architecture, to identify the signatures of failure.

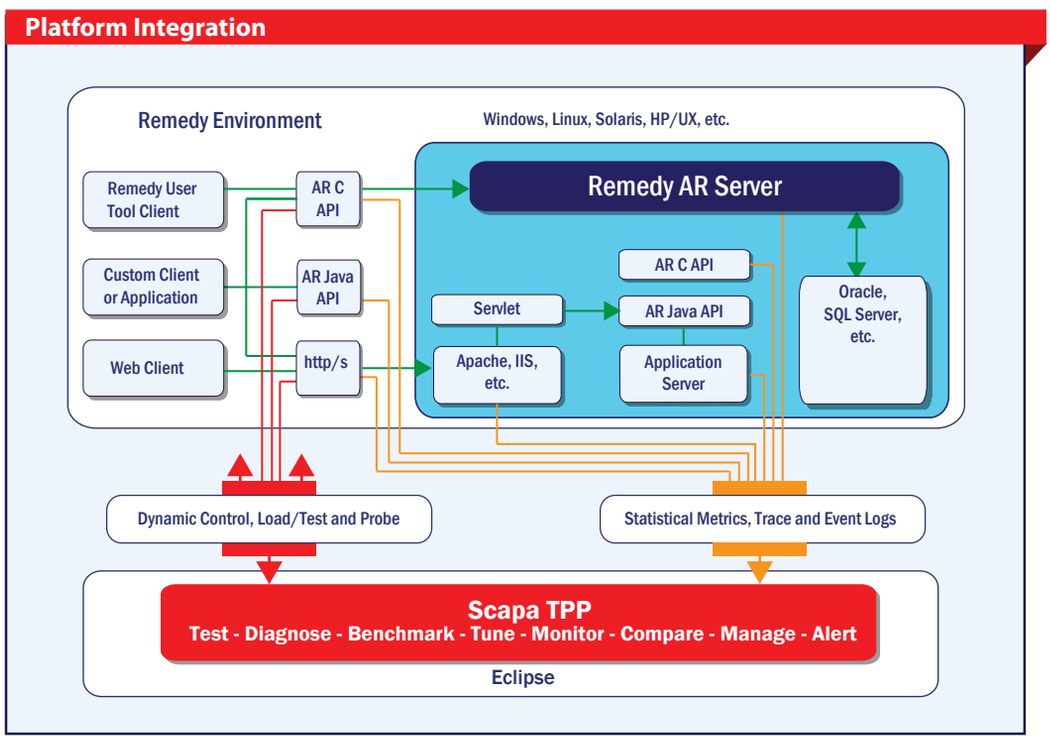
In later-lifecycle activities such as monitoring a production system, signatures usually continue to be measured, but a much smaller number of virtual users are typically simulated, often distributed across multiple geographical locations.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126| Tel. UK : +44 (0)131 208 0652

BMC Remedy ITSM and Scapa TPP

Integration Points

Scapa TPP has a huge number of interfaces, presented through a unified application interface, for driving load and collecting relevant data at different layers of the system. Many of the key touch-points are illustrated in the diagram below.



Scapa TPP can drive different Remedy API/Protocols such as AR C API, AR Java API or HTTP.

In addition, HTTP protocol support speed and ITSM testing success will depend on the level of support for ITSM specific BackChannel servlet calls. Scapa TPP can perform capture/replay and a certain amount of processing at this level, allowing the creation of valid tests and the introduction of data variability. Client – Server interaction is presented in a user readable format.

Whilst this is going on, Scapa TPP can collect system performance data and logs from Windows and most Unix systems, as well as Oracle, SQL server and most other popular relational databases, the Remedy server itself and the Web server.

Note: There are a number of interfaces not shown on the diagram which can usefully be exercised in certain architectures:

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

Scapa TPP can perform capture/replay at the user interface of the Remedy client (pressing buttons etc.) for the Remedy User Tool or a web browser. This is used to retrieve real end user metrics including client-side processing and page rendering.

Scapa TPP can drive the database layer directly, which can be useful in isolating detailed database configuration problems, or to simulate third-party applications that access Remedy data directly from the underlying database.

If the workflow triggers external processes (e.g. Java or HTTP), it is useful to drive them using Scapa TPP independently of the Remedy system.

It is important to remember that Scapa TPP is fully integrated and licensed as a single product. All of the above functionality is available at any time, singly or in combination, allowing Scapa TPP to function in Remedy architectures of any complexity and to help migrate any Remedy architecture to any other.

Remedy API Complexities

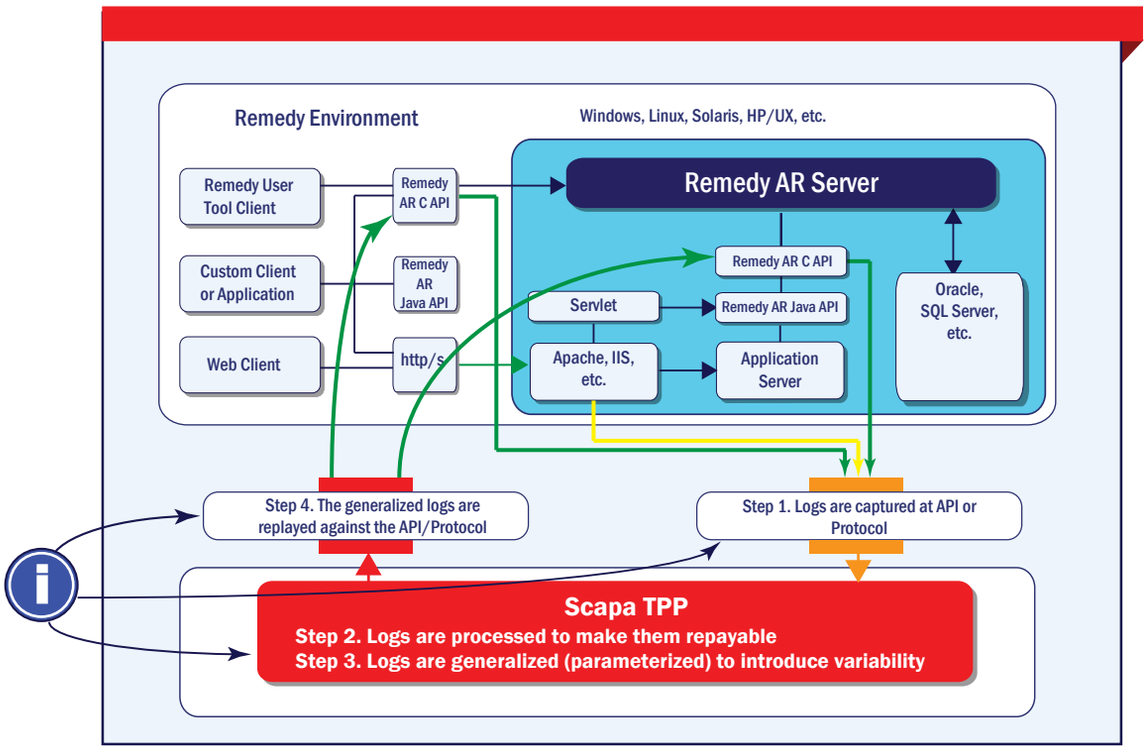
Driving load into ITSM is quite a complex problem. One example is the creation of a problem ticket for a customer contacting a telco call centre for assistance with his broken mobile phone battery. Replicating an appropriate process will involve the creation of a new ticket and there will be various references to information about the customer and the call centre agent which propagate throughout the corresponding sequence of API calls.

All these dependencies and equivalences have to be picked up to replicate the sequence with identical data values without errors when executed by a population of virtual users at test time. Certain pieces of information such as references to a newly created ticket will have to be propagated dynamically as they can not be known before runtime.

Furthermore, the introduction of variability into these test scripts will have to detect and expose dependencies in the workflow, i.e. if the customer is varied, the workflow may subsequently depend on the type of contract each customer has with the telco, and if the mobile phone model is varied, there may be a requirement subsequently to refer to the type of battery that actually fits the phone.

There are four basic processes which are illustrated for HTTP (yellow) and AR C API (green) in the following diagram:

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652



These four processes are:

1) Capture – the process of identifying a sequence of API calls that corresponds to a typical application workflow,

2) Parameterization/equivalence processing – the identification of data items and equivalence between them within the trace which are variable and/or require to be propagated through the trace,

3) Generalization – building on the parameterization, the introduction of variability at selected points in the trace such that large populations of virtual users will execute similar but not identical iterations of the captured trace,

4) Replay – a synthetic load is created by exercising the API according to the parameterized, generalized trace.

Typical workflows can easily require the understanding of hundreds of dependencies which need to be propagated correctly to obtain executable tests and simulate realistic load levels. This is necessary to arrive at a consistent replication of the originally captured workflows and to introduce variability without breaking dependency links.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126| Tel. UK : +44 (0)131 208 0652

ITSM Integration

At the HTTP layer, capture is performed from a standard browser via an instrumented proxy. The proxy supports additional proxies, secure sockets and various forms of complex authentication, and records a trace of the HTTP interaction which, subject to additional processing including interpretation of BackChannel service calls, is replayed against the server. The HTTP replay facility again supports a broad range of encryption and authentication models and/or proxies and is extremely lightweight, so that very significant levels of usage can be simulated from a small client machine.

The Remedy web client speaks to the Remedy server by issuing commands embedded in the HTTP requests. These commands correspond quite closely to the underlying Remedy API calls, with API arguments serialized into and out of the HTTP stream.

Scapa TPP exposes the serialized HTTP/Remedy API references in an XML-based capture/replay trace that forms a Test Component allowing easy parametarization of data values and handling of input/output dependencies. This representation of the Test Component is edited through the Scapa TPP GUI which has a number of tools to support parameterization, equivalence processing and generalization at the HTTP layer (no programming is required). At test time, the XML representation is compiled into a highly performant script format.

Parameterization/equivalence processing

Parameterization/equivalence processing is handled through available Scapa TPP functionality which allows introducing parameters for all occurrences of certain literal values at the click of a button. The necessary script changes are handled automatically and the resulting parameter instances are editable through an easy-to-use GUI.

Generalization

Generalization can be introduced by retrieving alternative field values from forms on a Remedy server. Scapa TPP provides an import facility which allows the user to do selective imports of variants for the created parameters. Through the provided functionality, multiple parameters can be easily populated with variant values from a Remedy Server.

Scapa TPP exposes the Remedy Back Channel API layer into its user interface through an abstraction layer known as a “pseudo-API” which isolates the user from the complexities of the underlying HTTP call and data encoding.

Scapa TPP’s support for the Back Channel API is not ITMS version specific, thereby providing both backward and forward compatibility and removing the need for a ITSM version specific plugin or patch. This helps avoid unnecessary delays if the ITSM version or patch level is changed during the project.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

Other Scapa TPP Features

Scapa TPP is a well-established application performance testing, diagnosis and monitoring tool. It is licensed as a single product with a common user interface and a common data management layer which interlocks all of its functionality. In addition to dedicated Remedy connectivity it has a number of key features:

Dynamic control of load - The load on the servers is controlled and visualized through a single user interface via one or more sliders. The user interface is like a graphic equalizer, which can crank up the load and see where the system breaks, whilst watching performance data inside Scapa TPP.

Control over business throughput as well as user count - If it is important to know the number of transactions per second the system can support rather than the number of users, Scapa TPP allows this to be controlled directly.

Predefined tests and scheduled tests execution - If *dynamic* control is not required, tests can be pre-defined to run in certain ways, for example for regression testing. They can also be scheduled for out of hours execution.

Distributed test execution - Tests can be run from multiple client machines, to measure end-user performance in the various geographies over which the applications are being deployed.

Test result analysis - All the data collected during a test is available for analysis with Scapa TPP's statistical console.

Reporting - Scapa TPP integrates with the Eclipse BIRT framework to provide a full- feature reporting capability with PDF, CSV and Web output, and a broad range of summarization and aggregation facilities, including percentile reporting and other statistical functions. There is also a facility to collect test results to a standard SQL database.

Monitoring - Scapa TPP test execution and system data collection can be scheduled on a continuous basis to provide an ongoing monitoring capability for Remedy applications.

Integration with Enterprise Monitoring tools - In addition to its stand-alone monitoring capabilities, Scapa TPP integrates well into an enterprise reporting tool. Scapa TPP can be used to control load, whilst exporting performance data and raising alerts in the monitoring tool, or the monitoring tool can schedule the execution of load from Scapa TPP.

GUI driven testing - For real end user experience measurements and functional validation.

For more information: quotes/case studies/references please contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126 | Tel. UK : +44 (0)131 208 0652

Other Testing Solutions Available from Scapa Technologies

- Citrix® XenApp™
- Citrix XenDesktop®
- Microsoft Terminal Services
- Microsoft Remote Desktop Services
- VMware® View™
- Other Solutions (Enquire for details)

Part: RWP/PT/1013

For more information, contact Scapa : www.scapatech.com | Contact@scapatech.com
Tel. USA : +1 415 287 4126| Tel. UK : +44 (0)131 208 0652