



# Etaliq @ RTAC09

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)



# Agenda

- 1 Why Automation?
- 2 Automation Today
- 3 Etaлиq – Easy Test Automation
- 4 ETA System Architecture
- 5 ETA Target Environment
- 6 ETA vs Tcl
- 7 Live ETA Demo
  - Task #1: Execution
  - Task #2: Execution Summarization
- 8 Case Studies
- 9 Conclusion

innovation in automation

# Why Automation?

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)



## Why Automation?

**Everyone needs to...**

**Automation must...**

innovation in automation



## Why Automation?

**Everyone needs to...**

- Save time

**Automation must...**

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality

### Automation must...

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality
- Find *bugs* sooner

### Automation must...

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality
- Find *bugs* sooner

### Automation must...

- Reduce effort

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality
- Find *bugs* sooner

### Automation must...

- Reduce effort
- Be *reliable*

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality
- Find *bugs* sooner

### Automation must...

- Reduce effort
- Be *reliable*
- Be *fast*

innovation in automation



## Why Automation?

### Everyone needs to...

- Save time
- Increase quality
- Find *bugs* sooner

### Automation must...

- Reduce effort
- Be *reliable*
- Be *fast*
- Be *easy*

innovation in automation

# Automation Today

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)

## Automation Environment

### Manual DevTest Engineer

- Test strategies/plans
- On-demand scripting

### Automation Engineer

- Translate test case to code
- Execute and review
- First-line problem determination

### Manual DevTest

- Subject Matter Expert (SME)
- Defines how to test
- Defines what to test
- Creates *personal...*
  - automation as required
  - Tcl regular expressions

innovation in automation

## Automation Environment

### Manual DevTest Engineer

- Test strategies/plans
- On-demand scripting

### Automation Engineer

- Translate test case to code
- Execute and review
- First-line problem determination

### Automation Engineer

- Not SME
- Translates *verbage* test plan to running code
- Creates log output
- Reviews log output
- Determines if...
  - *environmental* error or
  - *real* problem

innovation in automation



# Automation Environment Infrastructure

## Many files:

- libraries, scripts, job files,
- node definitions, map files,
- log files, environment settings

## Many systems:

- code libraries, device control,
- TGEN libraries, analysis,
- logging, reporting, revision control,
- reservation, scheduling, ...

## Expertise required

- Unix, Tcl, regular expressions,
- basic DUT behavior,
- many file formats,
- TGEN functions,
- libraries,
- log interpretation

innovation in automation



# **Etaliq – Easy Test Automation**

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)



## ETA – Primary ROI Methods

### Features:

- Test plan & case scripting
  - 25 instructions, coding time
- Resource usage
  - Syntax check
  - Simulated nodes
  - Hierarchy: Groups that FAIL don't run tests
- Log review & results reporting
  - Fully integrated indexed logs
  - Customizable summarized results reporting

innovation in automation



### Files:

- node table,
- execution table,
- test plan

### Single system:

- Execution Engine, TGEN types,
- log reports, summarized reporting,
- File Manager,
- Scheduler

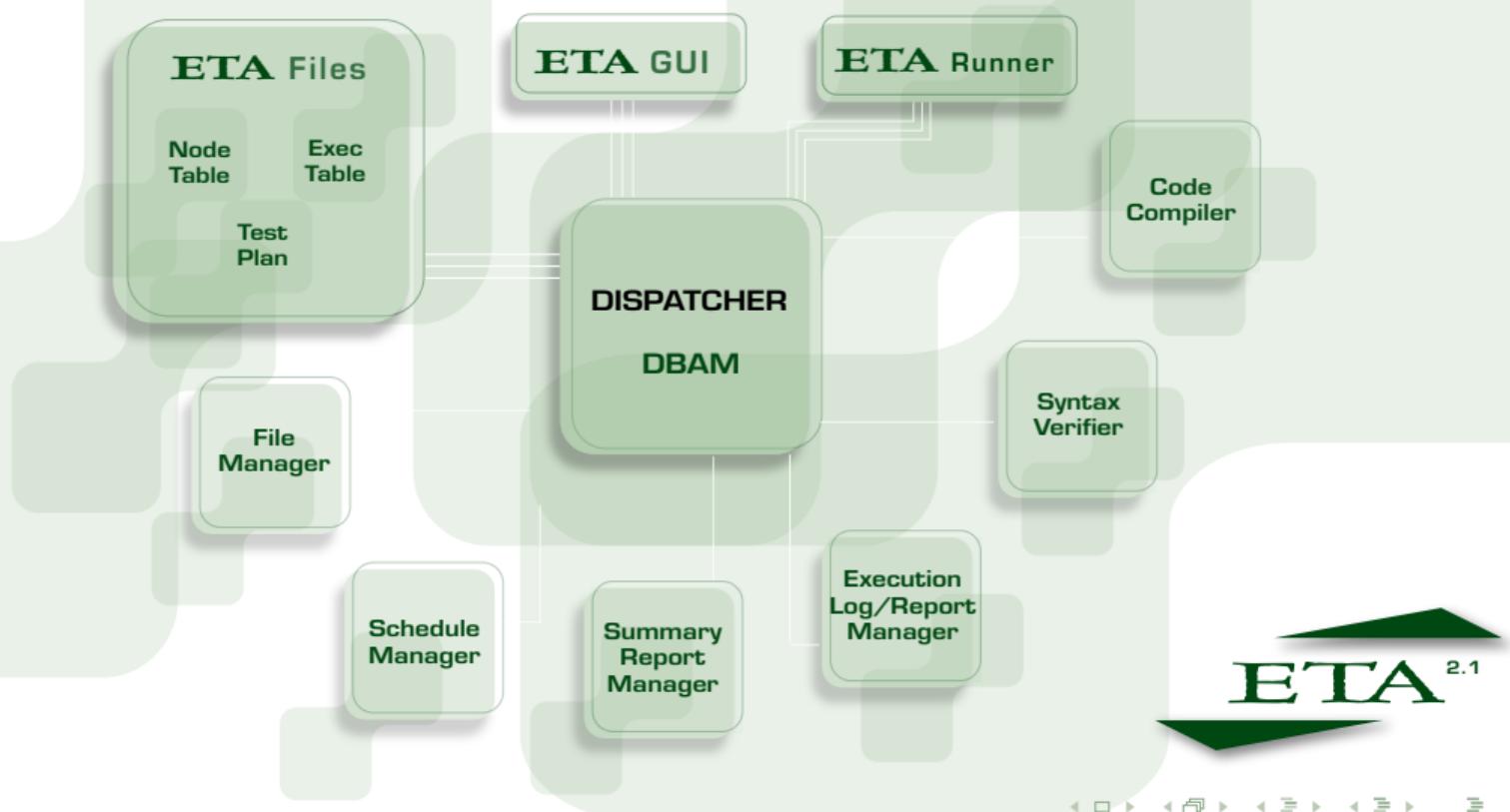
innovation in automation

## ETA – Automation Environment Infrastructure

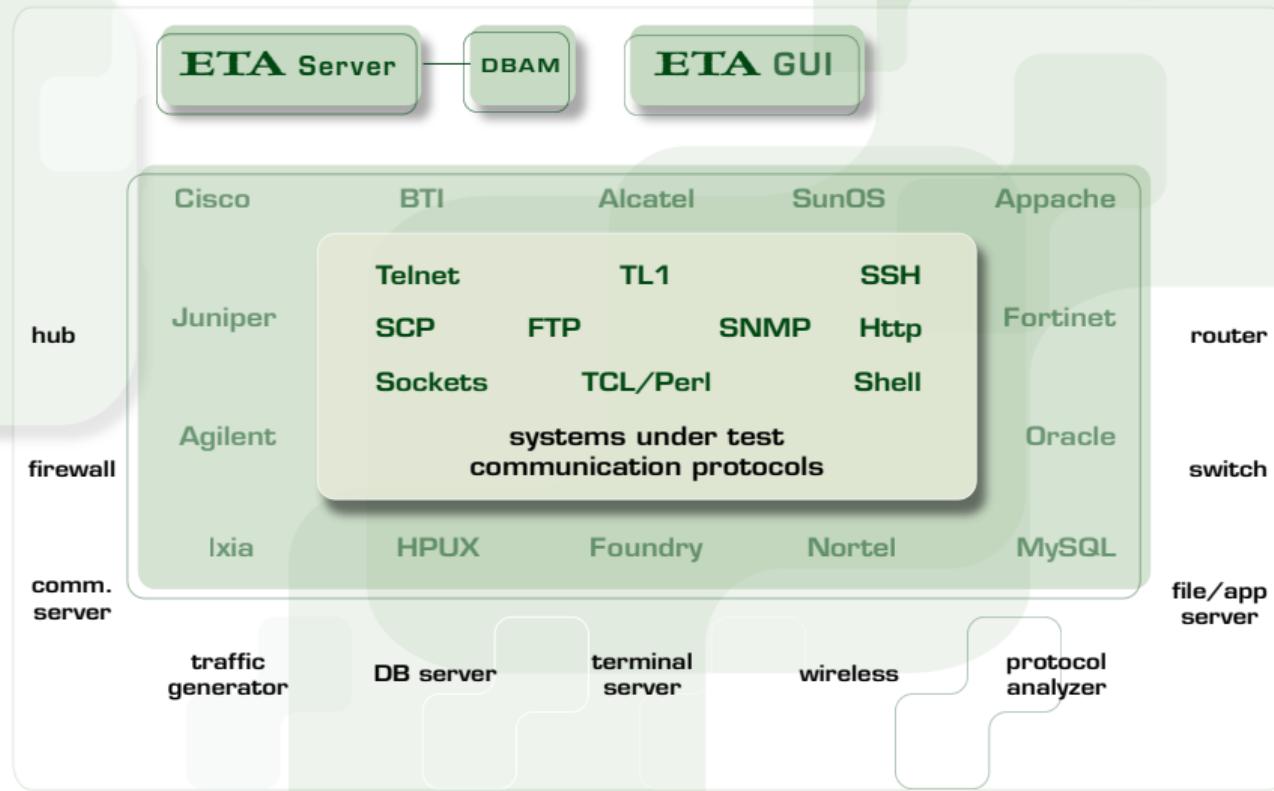
### Expertise required

- basic DUT behavior,
- minimal file formats,
- TGEN functions

# system architecture



# target environment



# **ETA vs Tcl**

innovation in automation

innovation in automation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)

## Send, Receive, and Verify!

- The ETA test plan template includes all related documentation.
- The **SEND** verb allows sending commands to nodes and handles all communications.
- The **RESULTLIST** verb defines expected results, no need for complicated parsing!
- **RESULTLIST** variables are associated with **SEND** instances and automate verification.
- Values are easily extracted from node output simply by referencing the **SEND** verb, no need for complicated parsing!
- The **EXPR** sub-verb provides powerful access to Boolean expressions and arithmetics, including seamless IP expressions.

# ETA

Test Case Identifier: **etaVsTcl\_1**

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp
```

```
# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))
```

```
# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (1/3)

```
set RouterNode1 csMdNode1
set intf $::nodes_array($RouterNode1,INTF1)

if { ![$node_connect $RouterNode1] } {
    error "Cannot continue: Connection to node $RouterNode1 failed!"
}

set cmd "show int $intf"
if { ![$node_send_receive $RouterNode1 $cmd\r "# *$" dispIntf] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
}

if { ![$regexp "line protocol is (\[^ \]+)" $dispIntf - valLineProto] } {
    log_failure "line protocol value not found in '$cmd' output!"
    set ::test_result Fail
    set valLineProto "unknown"
} elseif { $valLineProto != "up" } {
    log_failure "line protocol is '$valLineProto' but expected to be 'up'!"
    set ::test_result Fail
} else {
    log_info "line protocol is '$valLineProto', as expected."
}
...
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (1/3)

```
set RouterNode1 csMdNode1
set intf $::nodes_array($RouterNode1,INTF1)

if { ![$node_connect $RouterNode1] } {
    error "Cannot continue: Connection to node $RouterNode1 failed!"
}

set cmd "show int $intf"
if { ![$node_send_receive $RouterNode1 $cmd\r "# *$ dispIntf] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
}

if { ![$regexp "line protocol is (\[^ \]+)" $dispIntf - valLineProto] } {
    log_failure "line protocol value not found in '$cmd' output!"
    set ::test_result Fail
    set valLineProto "unknown"
} elseif { $valLineProto != "up" } {
    log_failure "line protocol is '$valLineProto' but expected to be 'up'!"
    set ::test_result Fail
} else {
    log_info "line protocol is '$valLineProto', as expected."
}
...
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (1/3)

```
set RouterNode1 csMdNode1
set intf $::nodes_array($RouterNode1,INTF1)

if { ![$node_connect $RouterNode1] } {
    error "Cannot continue: Connection to node $RouterNode1 failed!"
}

set cmd "show int $intf"
if { ![$node_send_receive $RouterNode1 $cmd\r "# *$" dispIntf] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
}

if { ![$regexp "line protocol is (\[^ \]+)" $dispIntf - valLineProto] } {
    log_failure "line protocol value not found in '$cmd' output!"
    set ::test_result Fail
    set valLineProto "unknown"
} elseif { $valLineProto != "up" } {
    log_failure "line protocol is '$valLineProto' but expected to be 'up'!"
    set ::test_result Fail
} else {
    log_info "line protocol is '$valLineProto', as expected."
}
...
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (2/3)

```
...
if { ![regexp "([0-9]+) input errors" $dispIntf - valInputErrors] } {
    log_failure "input errors value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputErrors "unknown"
} elseif { $valInputErrors != 0 } {
    log_failure "input errors value is $valInputErrors but expected to be 0!"
    set ::test_result Fail
} else {
    log_info "input errors value is $valInputErrors, as expected."
}

if { ![regexp "5 minute input rate\[^\\n\]*\\
  ([0-9]+) packets/sec" $dispIntf - valInputRate] } {
    log_failure "5 minute input rate value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputRate "unknown"
} elseif { $valInputRate <= 0 } {
    log_failure "5 minute input rate value is $valInputRate but expected\
      to greater than 0!"
    set ::test_result Fail
} else {
    log_info "5 minute input rate value is $valInputRate, as expected."
}
...
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (2/3)

```
...
if { ! [regexp "([0-9]+) input errors" $dispIntf - valInputErrors] } {
    log_failure "input errors value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputErrors "unknown"
} elseif { $valInputErrors != 0 } {
    log_failure "input errors value is $valInputErrors but expected to be 0!"
    set ::test_result Fail
} else {
    log_info "input errors value is $valInputErrors, as expected."
}

if { ! [regexp "5 minute input rate[^\\n]*\\
  ([0-9]+) packets/sec" $dispIntf - valInputRate] } {
    log_failure "5 minute input rate value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputRate "unknown"
} elseif { $valInputRate <= 0 } {
    log_failure "5 minute input rate value is $valInputRate but expected\\
      to greater than 0!"
    set ::test_result Fail
} else {
    log_info "5 minute input rate value is $valInputRate, as expected."
}
...
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (3/3)

```
...
if { ! [regexp "Internet address is\
  (\[0-9\]+\.\[0-9\]+\.\[0-9\]+\.\[0-9\]+)" $dispIntf - valSrcIP] } {
  log_failure "Internet address value not found in '$cmd' output!"
  set ::test_result Fail
  set valSrcIP "unknown"
} else {
  set bytes [split $valSrcIP .]
  set bytes [lreplace $bytes 3 3 [expr { [lindex $bytes 3] ^ 3 }]]
  set valDestIP [join $bytes .]

  set cmd "ping $valDestIP"
  if { ! [node_send_receive $RouterNode1 $cmd\r "# *$" dispPing] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
  }
  if { ! [regexp "Success rate is 100 percent" $dispPing] } {
    log_failure "Ping failed!"
    set ::test_result Fail
  } else {
    log_info "Ping succeeded."
  }
}

log_info "test result: ::test_result"
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))
```

```
# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (3/3)

```
...
if { ! [regexp "Internet address is\
  (\[0-9\]+\.\[0-9\]+\.\[0-9\]+\.\[0-9\]+)" $dispIntf - valSrcIP] } {
  log_failure "Internet address value not found in '$cmd' output!"
  set ::test_result Fail
  set valSrcIP "unknown"
} else {
  set bytes [split $valSrcIP .]
  set bytes [lreplace $bytes 3 3 [expr { [lindex $bytes 3] ^ 3 }]]
  set valDestIP [join $bytes .]

  set cmd "ping $valDestIP"
  if { ! [node_send_receive $RouterNode1 $cmd\r "# *$" dispPing] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
  }
  if { ! [regexp "Success rate is 100 percent" $dispPing] } {
    log_failure "Ping failed!"
    set ::test_result Fail
  } else {
    log_info "Ping succeeded."
  }
}

log_info "test result: ::test_result"
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (3/3)

```
...
if { ! [regexp "Internet address is\
  (\[0-9\]+\.\[0-9\]+\.\[0-9\]+\.\[0-9\]+)" $dispIntf - valSrcIP] } {
  log_failure "Internet address value not found in '$cmd' output!"
  set ::test_result Fail
  set valSrcIP "unknown"
} else {
  set bytes [split $valSrcIP .]
  set bytes [lreplace $bytes 3 3 [expr { [lindex $bytes 3] ^ 3 }]]
  set valDestIP [join $bytes .]

  set cmd "ping $valDestIP"
  if { ! [node_send_receive $RouterNode1 $cmd\r "# *$" dispPing] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
  }
  if { ! [regexp "Success rate is 100 percent" $dispPing] } {
    log_failure "Ping failed!"
    set ::test_result Fail
  } else {
    log_info "Ping succeeded."
  }
}

log_info "test result: ::test_result"
```

# ETA

Test Case Identifier: `etaVsTcl_1`

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

```
RESULTLIST(CheckIntfUp)
  "line protocol = up"
  "input errors = 0"
  "AFTER 5 minute input rate  ATTR packets/sec > 0"
RESULTLIST(CheckPingSuccess)
  "Success rate = 100 percent"
```

Setup:

Steps:

```
# Show and check the local interface and statistics
SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"
# (p2p, ^.3 does .1 <-> .2)
SET SrcIPnet dispIntf(Internet address)
SET DestIP  EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface
SEND RouterNode1 "ping DestIP" CheckPingSuccess
```

Cleanup:

# Tcl (3/3)

```
...
if { ! [regexp "Internet address is\
  (\[0-9\]+\.\[0-9\]+\.\[0-9\]+\.\[0-9\]+)" $dispIntf - valSrcIP] } {
  log_failure "Internet address value not found in '$cmd' output!"
  set ::test_result Fail
  set valSrcIP "unknown"
} else {
  set bytes [split $valSrcIP .]
  set bytes [lreplace $bytes 3 3 [expr { [lindex $bytes 3] ^ 3 }]]
  set valDestIP [join $bytes .]

  set cmd "ping $valDestIP"
  if { ! [node_send_receive $RouterNode1 $cmd\r "# *$" dispPing] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterNode1!"
  }
  if { ! [regexp "Success rate is 100 percent" $dispPing] } {
    log_failure "Ping failed!"
    set ::test_result Fail
  } else {
    log_info "Ping succeeded."
  }
}

log_info "test result: ::test_result"
```

# ETA

Test Case Identifier: **etaVsTcl\_1**

Title: ETA vs Tcl Comparison

Keywords: demo send receive verify

Objectives:

Compare the ease and power of ETA to traditional Tcl code.

Common:

**RESULTLIST(CheckIntfUp)**

"line protocol = up"

"input errors = 0"

"AFTER 5 minute input rate ATTR packets/sec > 0"

**RESULTLIST(CheckPingSuccess)**

"Success rate = 100 percent"

Setup:

Steps:

# Show and check the local interface and statistics

SEND RouterNode1 "show int Serial0" dispIntf CheckIntfUp

# Derive remote IP based on local IP found in "dispIntf"

# (p2p, ^.3 does .1 <-> .2)

SET SrcIPnet dispIntf(Internet address)

SET DestIP EXPR(ipHost(SrcIPnet ^ 0.0.0.3))

# Ping the remote interface

SEND RouterNode1 "ping DestIP" CheckPingSuccess

Cleanup:

## 6 verb commands!

# Tcl

```
set RouterModel csM6Model
set intf $::nodes_array($RouterModel,INTF1)
if { [ $::node_connect $RouterModel ] } {
    error "Cannot continue: Connection to node $RouterModel failed!"
}
set cmd "show int $intf"
if { [ $::node_send_receive $RouterModel $cmd "# $intf dispIntf" ] } {
    error "Cannot continue: Failed to send '$cmd' to node $RouterModel!"
}

if { [ regexp "Line protocol is (\|^.)" $dispIntf - valLineProto ] } {
    log_failure "Line protocol value not found in '$cmd' output!"
    set ::test_result Fail
    set valLineProto "Unknown"
} elseif { $valLineProto == "up" } {
    log_failure "Line protocol is '$valLineProto' but expected to be 'up'!"
    set ::test_result Fail
} else {
    log_info "Line protocol is '$valLineProto', as expected."
}

if { [ regexp "(\^S+\^S+) input errors" $dispIntf - valInputErrors ] } {
    log_failure "Input errors value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputErrors "Unknown"
} elseif { $valInputErrors == "0" } {
    log_failure "Input errors value is $valInputErrors but expected to be 0!"
    set ::test_result Fail
} else {
    log_info "Input errors value is $valInputErrors, as expected."
}

if { [ regexp "5 minute input rate(\^S\^S) (\^S+\^S+) packets/sec" $dispIntf - valInputRate ] } {
    log_failure "5 minute input rate value not found in '$cmd' output!"
    set ::test_result Fail
    set valInputRate "Unknown"
} elseif { $valInputRate > 0 } {
    log_failure "5 minute input rate value is $valInputRate but expected to be greater than 0!"
    set ::test_result Fail
} else {
    log_info "5 minute input rate value is $valInputRate, as expected."
}

if { [ regexp "Internet address is (\^S+\^S+)\.(\^S+\^S)\.(\^S+\^S)\.(\^S+\^S+)" $dispIntf - valSrcIP ] } {
    log_failure "Internet address value not found in '$cmd' output!"
    set ::test_result Fail
    set valSrcIP "Unknown"
} else {
    set bytes [split $valSrcIP 3]
    set bytesIP [lrange $bytes 0 2]
    set valDestIP [lrange $bytes 3]
    set valDestIP [join $bytesIP]
    set valDestIP [expr { [lindex $bytesIP 0] * 256 + [lindex $bytesIP 1] }]

    set cmd "ping $valDestIP"
    if { [ $::node_send_receive $RouterModel $cmd "# $intf dispPing" ] } {
        error "Cannot continue: Failed to send '$cmd' to node $RouterModel!"
    }
    if { [ regexp "Success rate = 100 percent" $dispPing ] } {
        log_failure "Ping failed!"
        set ::test_result Fail
    } else {
        log_info "Ping succeeded."
    }
}
log_info "test result: ::test_result"
```

## 60 lines!

# **Send, Receive, and Verify!**

## **Products**

Routers   Switches   Hubs   Gateways   *Firewalls*   Servers  
Workstations   *Databases*   Analyzers

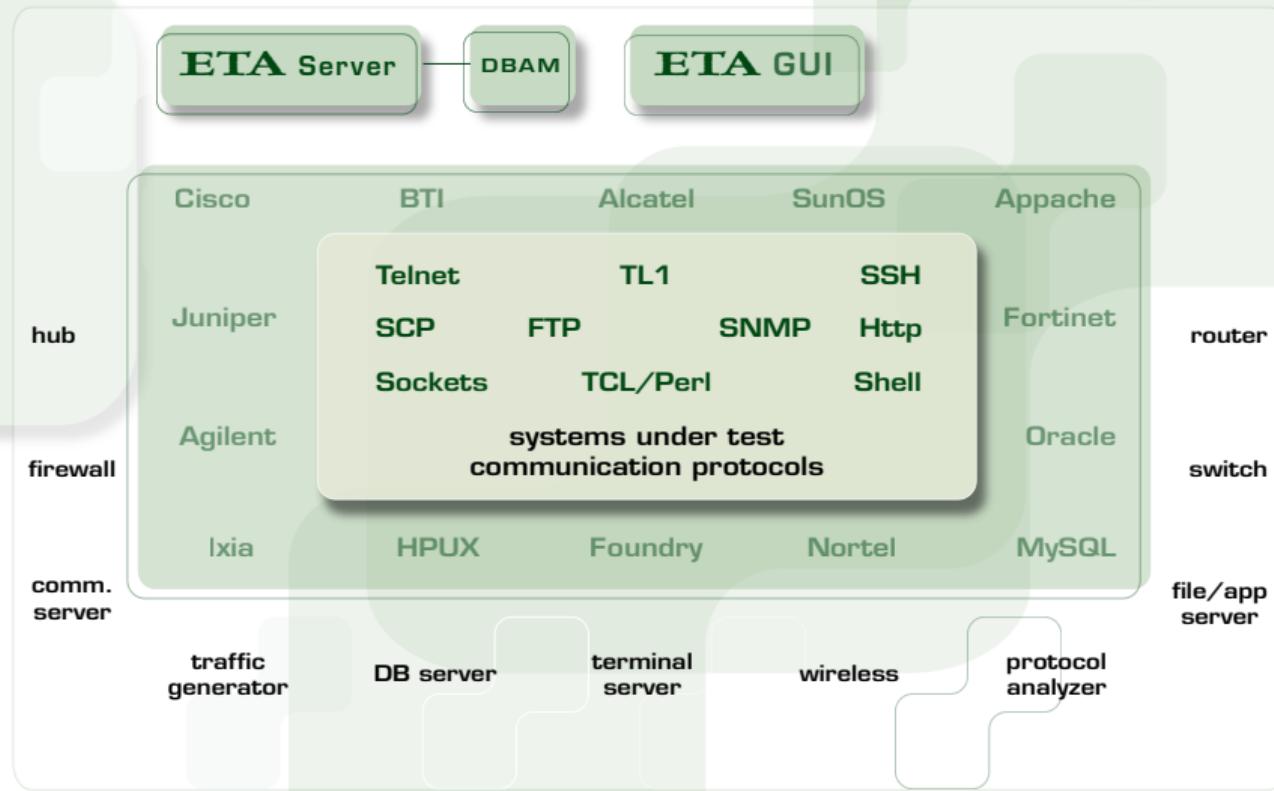
## **UUT Protocols**

Telnet   TL1   SSH   SFTP   FTP   SNMP   XML   Sockets   SQL  
Tcl   Perl   Shell   Files   HTML   HTTP   HTTPS

## **Vendors**

Cisco   Juniper   Nortel   Alcatel   Lucent   F5   IBM   Ciena   MySQL  
*Apache*   Sun   Linux   HP   Agilent   Ixia   Spirent   Microsoft

# target environment





## Task #1: Execution

Learn to:

- schedule an execution
- review an execution's reports
- navigate using relative position jumping
- update the test case code

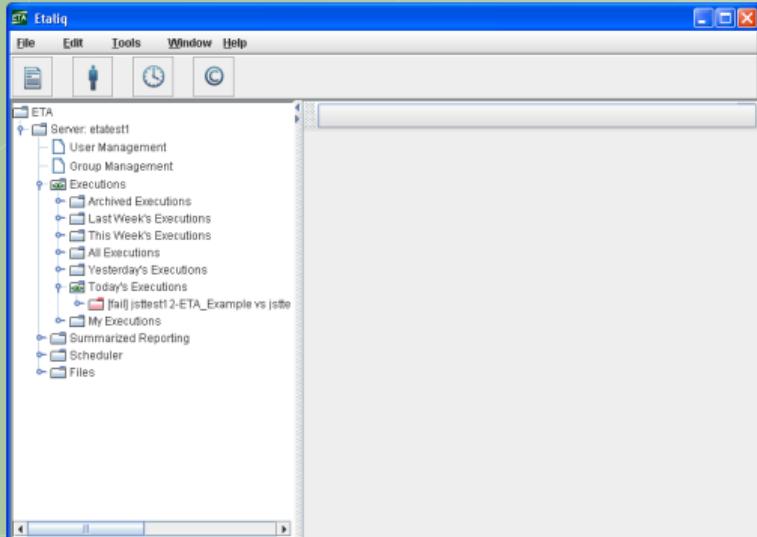
Register at <http://www.etaliq.com> to access the **Live ETA Demo**

innovation in automation



## Task #1: Execution

List of executions



- Expand the *Executions* tree item to display execution filters

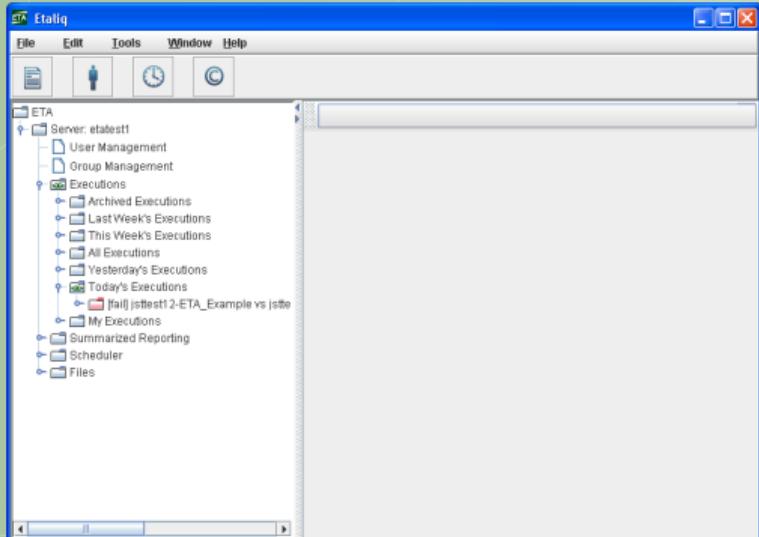
innovation in automation

[www.etaliq.com](http://www.etaliq.com)



## Task #1: Execution

List of executions



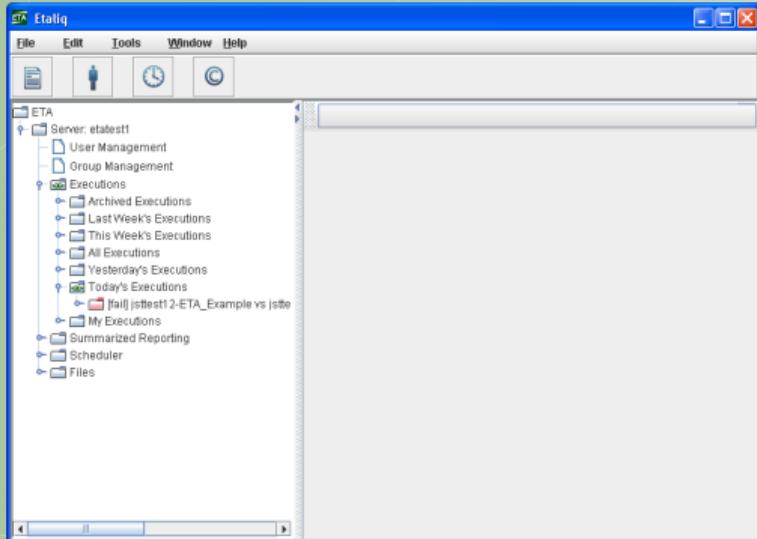
- Expand the *Executions* tree item to display execution filters
- Expand the *Today's Executions* execution filter

innovation in automation



# Task #1: Execution

List of executions



- Expand the *Executions* tree item to display execution filters
- Expand the *Today's Executions* execution filter
- Any executions just scheduled or executed today will appear under this tree item
- Several default execution filters are available
- Any number of custom execution filters can be created using the *Filter Management* tool

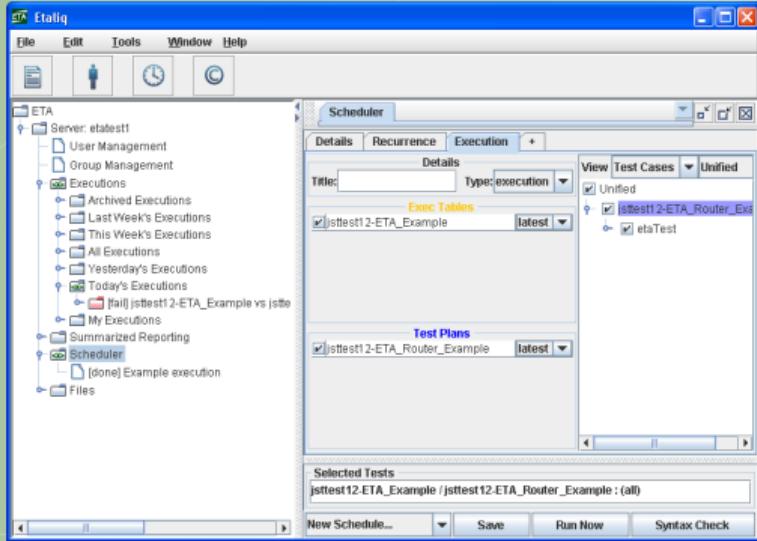
innovation in automation

[www.etaliq.com](http://www.etaliq.com)



# Task #1: Execution

Starting the Schedule



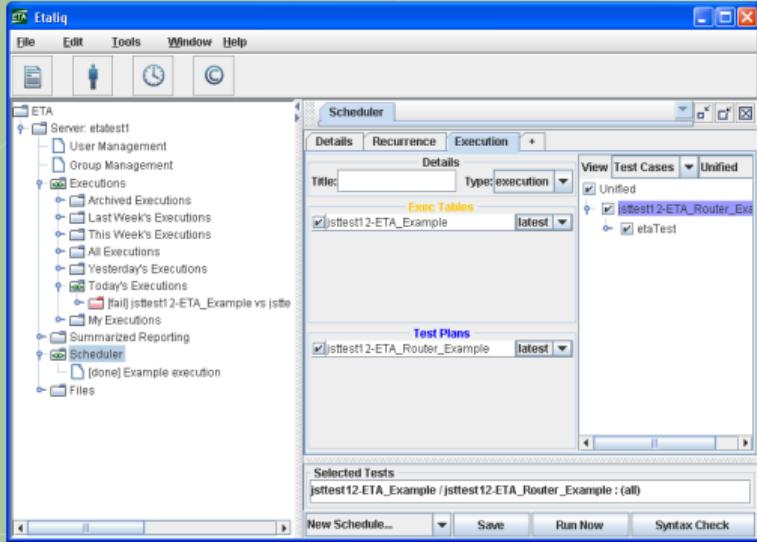
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane

innovation in automation



# Task #1: Execution

Starting the Schedule



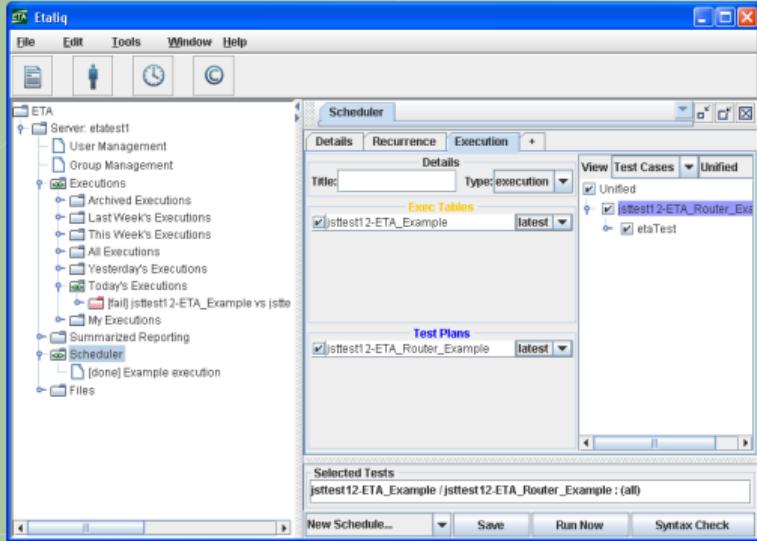
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane
- Expand the *Scheduler* tree item to display existing schedules

innovation in automation



# Task #1: Execution

Starting the Schedule



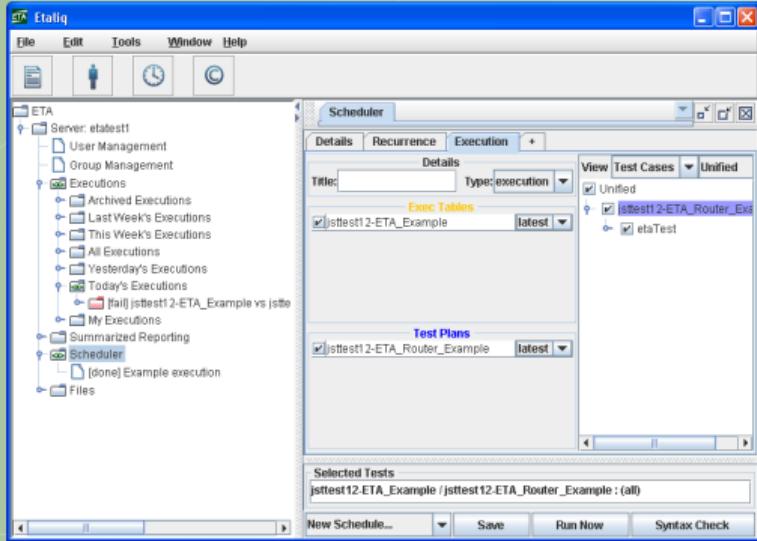
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane
- Expand the *Scheduler* tree item to display existing schedules
- Select the *Scheduler's Execution* tab

innovation in automation



# Task #1: Execution

Starting the Schedule



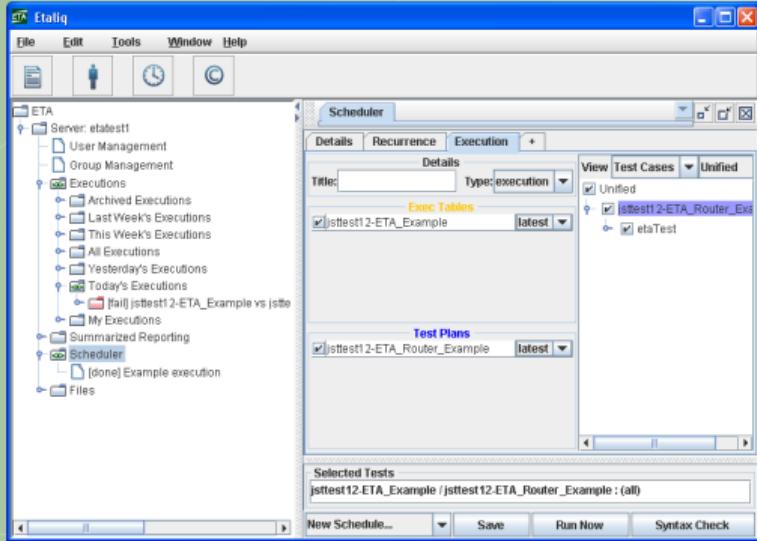
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane
- Expand the *Scheduler* tree item to display existing schedules
- Select the *Scheduler's Execution* tab
- Select the execution table named **user-ETA\_Example**

innovation in automation



# Task #1: Execution

Starting the Schedule



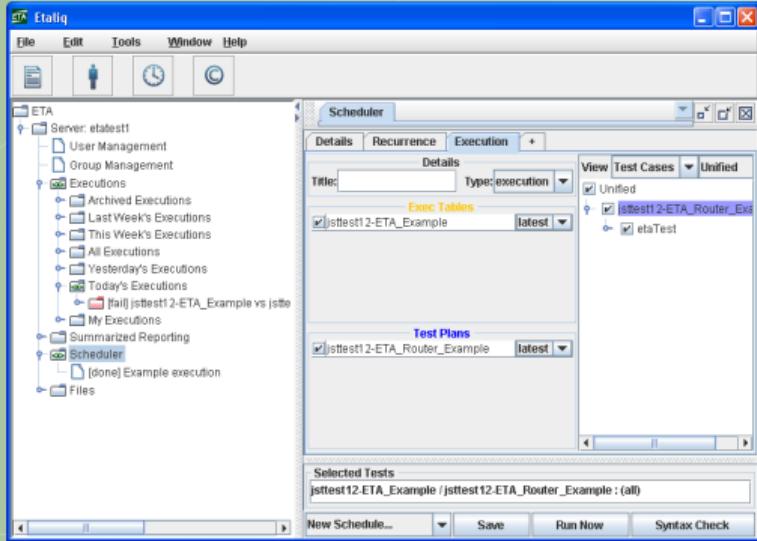
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane
- Expand the *Scheduler* tree item to display existing schedules
- Select the *Scheduler's Execution* tab
- Select the execution table named **user-ETA\_Example**
- Select the test plan named **user-ETA\_Router\_Example**

innovation in automation



# Task #1: Execution

## Starting the Schedule



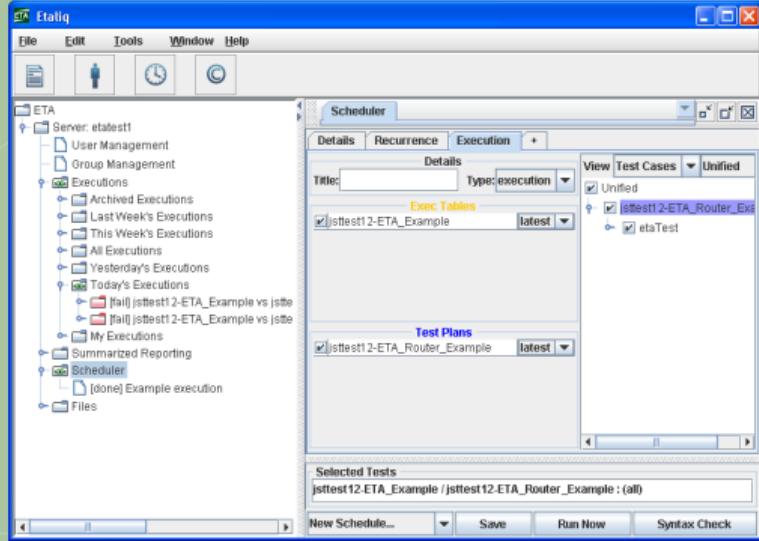
- Select the *Scheduler* link from the *Left Tree* to activate the *Scheduler* tool in the right pane
- Expand the *Scheduler* tree item to display existing schedules
- Select the *Scheduler's Execution* tab
- Select the execution table named **user-ETA\_Example**
- Select the test plan named **user-ETA\_Router\_Example**
- Click the **Run Now** button

innovation in automation



# Task #1: Execution

Live Summary Report



- The schedule just started appears under *Today's Executions* in the *Left Tree*

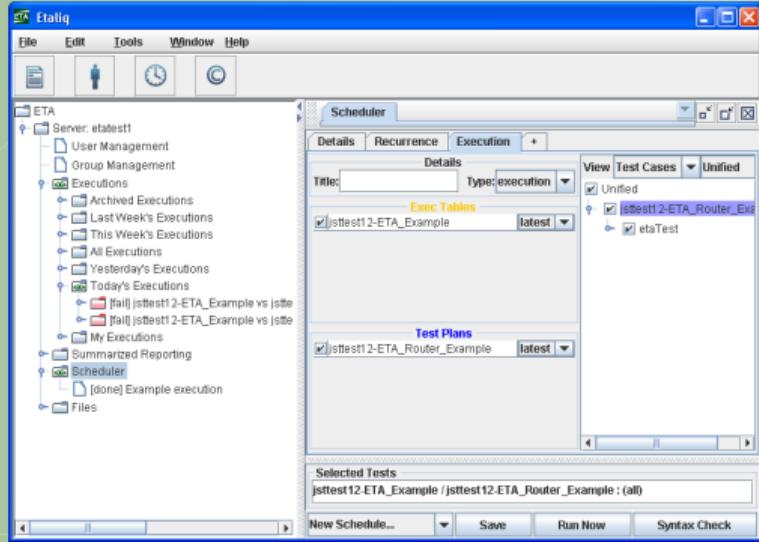
innovation in automation

[www.etaliq.com](http://www.etaliq.com)



# Task #1: Execution

Live Summary Report



- The schedule just started appears under *Today's Executions* in the *Left Tree*
- Expand the **schedule** to reveal any verification and test plan executions

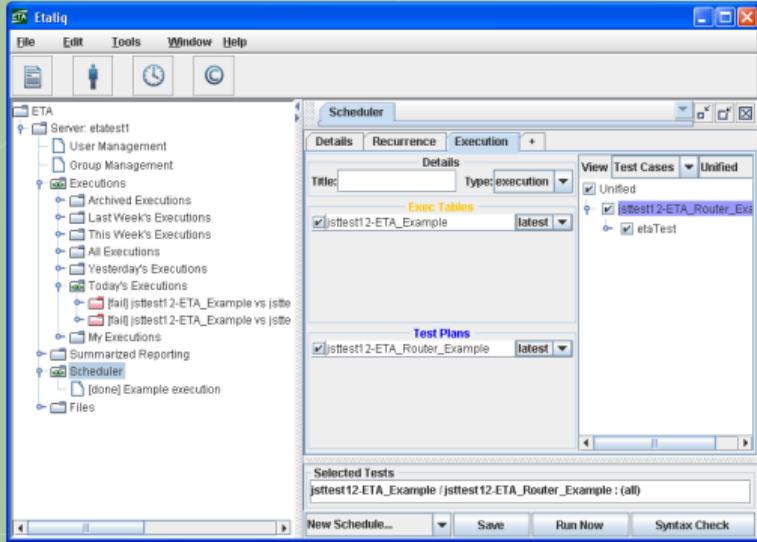
innovation in automation

[www.etaliq.com](http://www.etaliq.com)



# Task #1: Execution

Live Summary Report



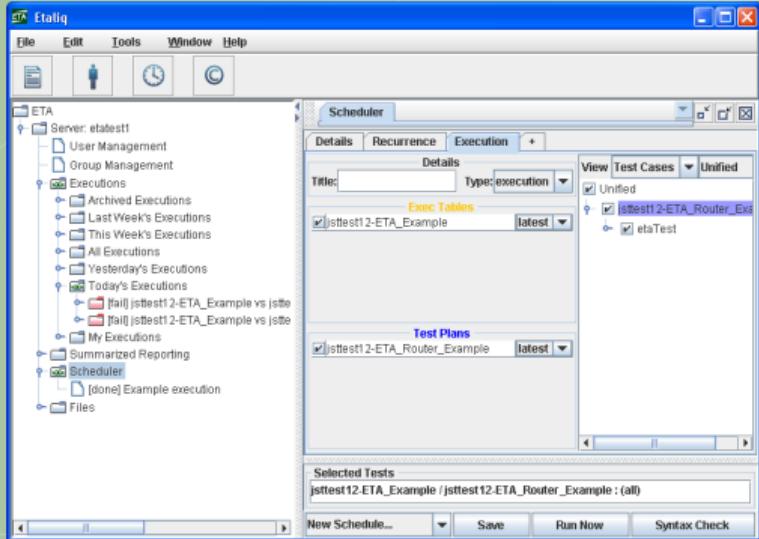
- The schedule just started appears under *Today's Executions* in the *Left Tree*
- Expand the schedule to reveal any verification and test plan executions
- Select the **execution** to see the live reports in the right pane

innovation in automation



# Task #1: Execution

Live Summary Report



- The schedule just started appears under *Today's Executions* in the *Left Tree*
- Expand the schedule to reveal any verification and test plan executions
- Select the execution to see the live reports in the right pane
- Click the *Right arrow* on the *Resize Bar* in the middle of the screen to hide the *Left Tree* and leave more room to the *View Panel*

innovation in automation



# Task #1: Execution

Live Summary Report

TYPE	VERDICT	SMVERSL	TEST PLAN	TEST CASE	TITLE
execTable	PASS		jsttest12-BexecTable		
group	CHILDFAIL		jsttest12-BetaTest		ETA tool test gr
group	CHILDFAIL		jsttest12-BetaTest_routertest		ETA tool test
group	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_1		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_2		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_3		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_4		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification_1		ETA tool test, p
group	CHILDFAIL		jsttest12-BetaTest_routertest_showInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_2		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_3		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_4		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_5		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_showIPRouteVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showIPRouteVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showIPTrafficVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showIPTrafficVerification_1		ETA tool test, p
group	CHILDFAIL		jsttest12-BetaTest_routertest_showVersionVerification		ETA tool test
test	FAIL		jsttest12-BetaTest_routertest_showVersionVerification_1		ETA tool test, p

- The default selected tab is the *Summary Report*
- This tab updates live to reflect the verdict of each executed group and test

innovation in automation



# Task #1: Execution

Live Summary Report

TYPE	VERDICT	SMVERSL	TEST PLAN	TEST CASE	TITLE
execTable	PASS		jsttest12-BexecTable		
group	CHILDFAIL		jsttest12-BetaTest		ETA tool test gr
group	CHILDFAIL		jsttest12-BetaTest_routertest		ETA tool test
group	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_1		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_2		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_3		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_4		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification_1		ETA tool test, p
group	CHILDFAIL		jsttest12-BetaTest_routertest_showInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_2		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_3		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_4		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_5		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_showIPRouteVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showIPRouteVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showIPTrafficVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showIPTrafficVerification_1		ETA tool test, p
group	CHILDFAIL		jsttest12-BetaTest_routertest_showVersionVerification		ETA tool test
test	FAIL		jsttest12-BetaTest_routertest_showVersionVerification_1		ETA tool test, p

- The default selected tab is the *Summary Report*
- This tab updates live to reflect the verdict of each executed group and test
- Select this test from the table:  
**ETA\_RouterTest\_showInterfaceVerification\_4**

innovation in automation



# Task #1: Execution

Live Summary Report

The screenshot shows the ETALIQ software interface with the title bar "Etaiq" and the window title "Scheduler jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:15:38)". The main area displays a table with columns: TYPE, VERDICT, SWVERSL, TEST PLAN, TEST CASE, and TITLE. The table lists various test cases and their outcomes. The bottom navigation bar includes tabs for Attributes, Syntax Report, Detailed Report, Summary Report, Console.RouterNode1, and jsttest12-ETA\_Router\_Example.jst.

TYPE	VERDICT	SWVERSL	TEST PLAN	TEST CASE	TITLE
execTable	PASS		jsttest12-BexecTable		
group	CHILDFAIL		jsttest12-BetaTest		ETA tool test gr
group	CHILDFAIL		jsttest12-BetaTest_routertest		ETA tool test
group	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_1		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_2		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_3		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_basicSendReceiveVerify_4		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_configInterfaceVerification_1		ETA tool test, p
group	CHILDFAIL		jsttest12-BetaTest_routertest_showInterfaceVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_2		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_3		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showInterfaceVerification_4		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_showInterfaceVerification_5		ETA tool test, p
group	PASS		jsttest12-BetaTest_routertest_shouIPRouteVerification		ETA tool test
test	PASS		jsttest12-BetaTest_routertest_shouIPRouteVerification_1		ETA tool test, p
test	PASS		jsttest12-BetaTest_routertest_shouIPTrafficVerification		ETA tool test
group	CHILDFAIL		jsttest12-BetaTest_routertest_shouIPTrafficVerification_1		ETA tool test, p
test	FAIL		jsttest12-BetaTest_routertest_showVersionVerification		ETA tool test
			jsttest12-BetaTest_routertest_showVersionVerification_1		ETA tool test, p

- The default selected tab is the *Summary Report*
- This tab updates live to reflect the verdict of each executed group and test
- Select this test from the table:  
ETA\_RouterTest\_showInterfaceVerification\_4
- Select the *Detailed Report* tab to jump directly to the details of its execution

innovation in automation



# Task #1: Execution

Live Detailed Report

The screenshot shows the Etaliq application window with the title "Scheduler jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:15:38)". The main pane displays a hierarchical tree of test cases and steps. The tree starts with "etaTest\_routertest\_showInterfaceVerification\_1" under "etaTest\_routertest\_showInterfaceVerification". This node has five children, each ending in \_1 through \_5. Each of these five nodes has a "CLEANUP" child node. The entire tree is contained within a "COMMON" section, which itself has "SETUP" and "CLEANUP" sections. The status of each node is indicated by icons: PASS (green), FAIL (red), and CHILDFAIL (yellow). The bottom of the window shows tabs for "Attributes", "Syntax Report", "Detailed Report", "Summary Report", "Console.RouterNode1", and "jsttest12-ETA\_Example.execTable".

- The detailed report is a hierarchical representation of every test group, test case, section and step

innovation in automation



# Task #1: Execution

Live Detailed Report

The screenshot shows the Etaliq application window titled "Scheduler jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:15:38)". The main pane displays a hierarchical tree of test cases and steps. The tree starts with "etaTest\_routertest\_showInterfaceVerification\_1" and branches into "etaTest\_routertest\_showInterfaceVerification\_2", "etaTest\_routertest\_showInterfaceVerification\_3", "etaTest\_routertest\_showInterfaceVerification\_4", and "etaTest\_routertest\_showInterfaceVerification\_5". Each of these further branches into "COMMON", "SETUP", and "CLEANUP" sections. The "etaTest\_routertest\_showIPRouteVerification" section also follows this structure. The "etaTest\_routertest\_showIPTrafficVerification" and "etaTest\_routertest\_showVersionVerification" sections follow a similar pattern. The "etaTest\_routertest\_showVersionVerification\_1" step is expanded to show its own "COMMON", "SETUP", and "CLEANUP" sections. The bottom of the window shows tabs for "Attributes", "Syntax Report", "Detailed Report", "Summary Report", "Console.RouterNode1", and "jsttest12-ETA\_Example.execTable (".

- The detailed report is a hierarchical representation of every test group, test case, section and step
- Every step executed is logged along with its verdict, details of execution and any node results

innovation in automation



# Task #1: Execution

Finding Failures

**Scheduler jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:15:38)**

VERDICT	COMMAND
PASS	etaTest_routertest_showInterfaceVerification_1
PASS	etaTest_routertest_showInterfaceVerification_2
<b>FAIL</b>	etaTest_routertest_showInterfaceVerification_3
<b>FAIL</b>	etaTest_routertest_showInterfaceVerification_4
PASS	↳ COMMON
PASS	↳ SETUP
PASS	↳ STEPS
PASS	↳ SEND RouterModel "clear counters"
PASS	↳ SLEEP 3
PASS	↳ SEND RouterModel "sh int Serial0" interfaceIncreasingCounters RouterModel(INTERFACE)
PASS	↳ WAIT 30 RouterModel "ping 192.1.1." ZERO_PERCENT WaitInterval=10
PASS	↳ SEND RouterModel "sh int Serial0" interfaceIncreasingCounters RouterModel(INTERFACE)
PASS	↳ SEND RouterModel "sh int Serial0" interfaceNoChangeCounters RouterModel(INTERFACE)
PASS	↳ CONFIG RouterModel SHUTDOWNINTERFACE
PASS	↳ SEND RouterModel "sh int Serial0" interfaceNoChangeCounters RouterModel(INTERFACE)
PASS	↳ Deferred verification
PASS	↳ CLEANUP
PASS	↳ etaTest_routertest_showInterfaceVerification_5
PASS	↳ CLEANUP
PASS	↳ etaTest_routertest_showIPRouteVerification
PASS	↳ COMMON
PASS	↳ SETUP
PASS	↳ etaTest_routertest_showIPRouteVerification_1

**Attributes** **Syntax Report** **Detailed Report** **Summary Report** **Console.RouterNode1** **jsttest12-ETA\_Example.execTable (**

- Focus by selecting the currently highlighted line

innovation in automation



# Task #1: Execution

Finding Failures

The screenshot shows the Etalq application window with the title "Scheduler jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:15:38)". The main area displays a table with two columns: "VERDICT" and "COMMAND". The "VERDICT" column contains several "FAIL" entries, with the fourth one highlighted in blue. The "COMMAND" column lists various test steps, including "etaTest\_routertest\_showInterfaceVerification\_1" through "etaTest\_routertest\_showInterfaceVerification\_5", "COMMON", "SETUP", "STEPS", and "etaTest\_routertest\_showIPRouteVerification". Below the table, tabs for "Attributes", "Syntax Report", "Detailed Report", "Summary Report", "Console.RouterNode1", and "jsttest12-ETA\_Example.execTable" are visible.

VERDICT	COMMAND
PASS	etaTest_routertest_showInterfaceVerification_1
PASS	etaTest_routertest_showInterfaceVerification_2
FAIL	etaTest_routertest_showInterfaceVerification_3
FAIL	etaTest_routertest_showInterfaceVerification_4
PASS	COMMON
PASS	SETUP
PASS	STEPS
PASS	SEND RouterModel "clear counters"
PASS	SLEEP 3
PASS	SEND RouterModel "sh int Serial0" interfaceIncreasingCounters RouterModel(INTF)
PASS	WAIT 30 RouterModel "ping 192.1.1.1" ZERO_PERCENT WaitInterval=10
PASS	SEND RouterModel "sh int Serial0" interfaceIncreasingCounters RouterModel(INTF)
PASS	SEND RouterModel "sh int Serial0" interfaceNoChangeCounters RouterModel(INTF)
PASS	CONFIG RouterModel SHUTINTERFACE
PASS	SEND RouterModel "sh int Serial0" interfaceNoChangeCounters RouterModel(INTF)
PASS	Deferred verification
PASS	CLEANUP
PASS	etaTest_routertest_showInterfaceVerification_5
PASS	CLEANUP
PASS	etaTest_routertest_showIPRouteVerification
PASS	COMMON
PASS	SETUP
PASS	etaTest_routertest_showIPRouteVerification_1

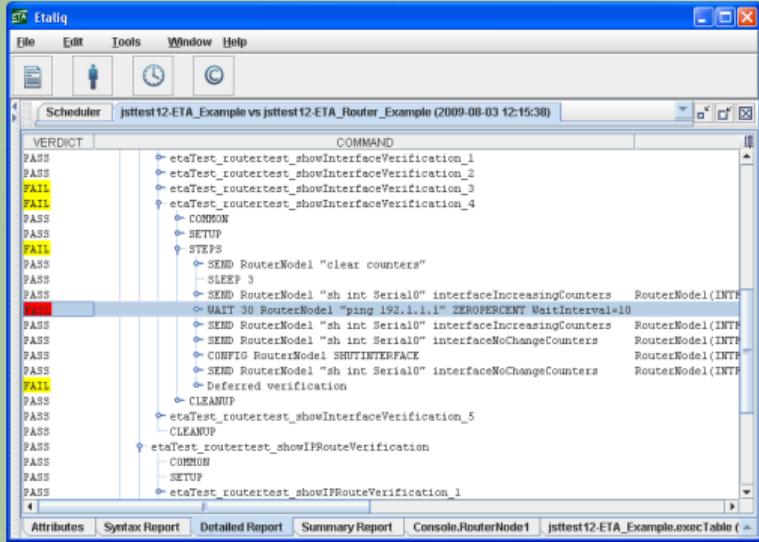
- Focus by selecting the currently highlighted line
- Press **F4** to find the first failure

innovation in automation



# Task #1: Execution

Finding Failures



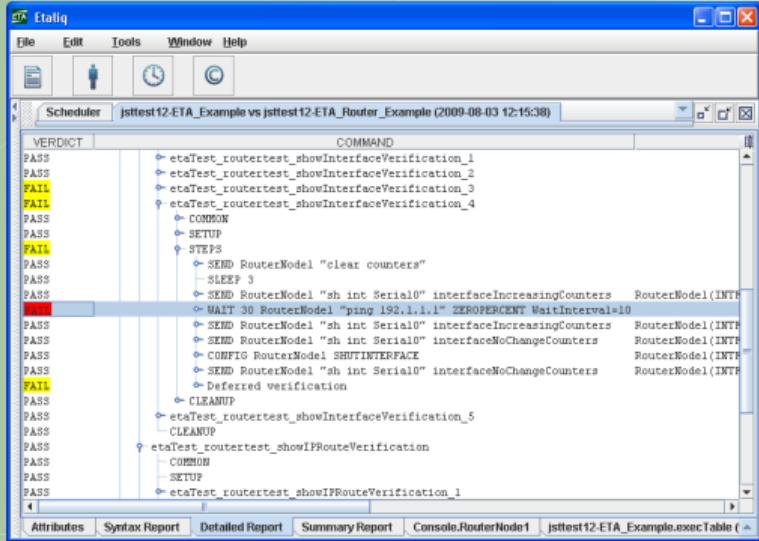
- Focus by selecting the currently highlighted line
- Press F4 to find the first failure
- Press F4 again to find the next failure

innovation in automation



# Task #1: Execution

Finding Failures



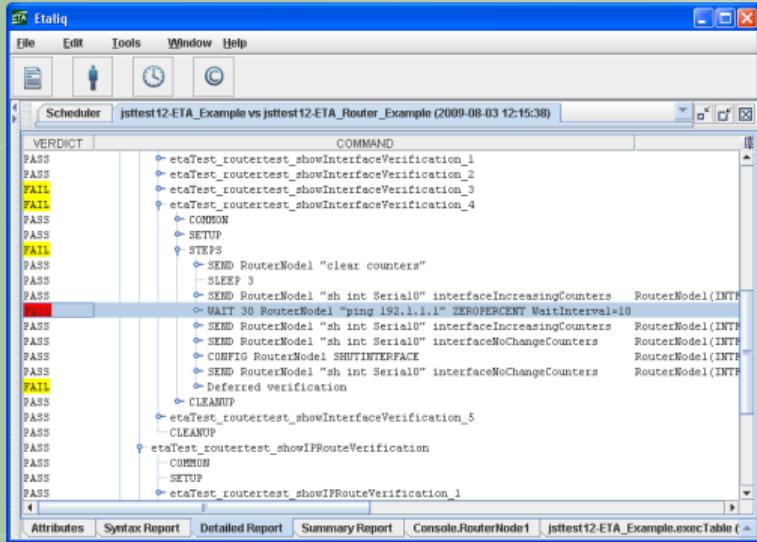
- Focus by selecting the currently highlighted line
- Press F4 to find the first failure
- Press F4 again to find the next failure
- Focus by selecting the currently highlighted line

innovation in automation



# Task #1: Execution

Finding Failures



- Focus by selecting the currently highlighted line
- Press F4 to find the first failure
- Press F4 again to find the next failure
- Focus by selecting the currently highlighted line
- Select the *user-ETA\_Router\_Example.testPlan* tab to jump directly to the test case code  
(access hidden tabs using the *Tab Selector* arrow on the right side of the tab list)

innovation in automation



# Task #1: Execution

Updating Test Code

```

Etalix
File Edit Tools Window Help
Scheduler jstest12-ETA_Example vs jstest12-ETA_Router_Example (2009-08-03 12:15:38)
Test Case Identifier: etaTest_koutertest_showInterfaceVerification_5
Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jstest12-ETA_Router_Example.test

MANUFACTURER("intellectualquality")
"packets output a"
"packets output =="
"packets output EQUAL"

Setup:
1. CONFIG RouterModel IP

Steps:
1. SEND RouterModel "clear counters"
2. SLEEP 3
3. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
4. WAIT 30 RouterModel "ping 192.1.1.10" ZERO_PERCENT WaitInterval=10
5. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
6. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters
7. CONFIG RouterModel SHUTINTERFACE
8. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters

Cleanup:
1. CONFIG RouterModel NOSHUTINTERFACE
2. SLEEP 5 s

Test Case Identifier: etaTest_koutertest_showInterfaceVerification_5
Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jstest12-ETA_Router_Example.test

```

- Select the **File → Unlock** menu item

innovation in automation



# Task #1: Execution

Updating Test Code

```

Etalix
File Edit Tools Window Help
Scheduler jsttest12-ETA_Example vs jsttest12-ETA_Router_Example (2009-08-03 12:15:38)
  "packets output"
  "packets output =="
  "packets output EQUAL"

Setup:
1. CONFIG RouterModel IP

Steps:
1. SEND RouterModel "clear counters"
2. SLEEP 3
3. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
4. WAIT 30 RouterModel "ping 192.1.1.10" ZERO_PERCENT WaitInterval=10
5. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
6. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters
7. CONFIG RouterModel SHUTINTERFACE
8. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters

Cleanup:
1. CONFIG RouterModel NOSHUTINTERFACE
2. SLEEP 5 s

Test Case Identifier: etaTest_koutertest_showInterfaceVerification_5
Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jsttest12-ETA_Router_Example.test

```

- Select the *File* → *Unlock* menu item
- Update the **test case code**  
(For example, change the ping destination address from 192.168.1.1 to 192.168.1.10)

innovation in automation



# Task #1: Execution

Updating Test Code

```

Etaliq
File Edit Tools Window Help
Scheduler jstest12-ETA_Example vs jstest12-ETA_Router_Example (2009-08-03 12:15:38)
"packets output"
"packets output =="
"packets output EQUAL"

Setup:
1. CONFIG RouterModel IP

Steps:
1. SEND RouterModel "clear counters"
2. SLEEP 3
3. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
4. WAIT 30 RouterModel "ping 192.1.1.10" ZERO_PERCENT WaitInterval=10
5. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
6. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters
7. CONFIG RouterModel SHUTINTERFACE
8. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters

Cleanup:
1. CONFIG RouterModel NOSHUTINTERFACE
2. SLEEP 5 s

Test Case Identifier: etaTest_koutertest_showInterfaceVerification_5
Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jstest12-ETA_Router_Example.test

```

- Select the *File* → *Unlock* menu item
- Update the test case code  
(For example, change the ping destination address from 192.168.1.1 to 192.168.1.10)
- Select *File* → *Save* to save the changes  
(A “Success” dialog will pop-up to confirm the operation)

innovation in automation



# Task #1: Execution

Updating Test Code

```

Scheduler jsctest12-ETA_Example vs jsctest12-ETA_Router_Example (2009-08-03 12:15:38)

Setup:
  packets output "a"
  packets output ""
  packets output EQUAL

Steps:
  1. CONFIG RouterModel IP

  1. SEND RouterModel "clear counters"
  2. SLEEP 3
  3. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
  4. WAIT 30 RouterModel "ping 192.1.1.10" ZERO_PERCENT WaitInterval=10
  5. SEND RouterModel "sh int Serial0" interfaceIncreasingCounters
  6. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters
  7. CONFIG RouterModel SHUTINTERFACE
  8. SEND RouterModel "sh int Serial0" interfaceNoChangeCounters

Cleanup:
  1. CONFIG RouterModel NOSHUTINTERFACE
  2. SLEEP 5 s

Test Case Identifier: etaTest_koutertest_showInterfaceVerification_5
Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jsctest12-ETA_Router_Example.test

```

- Select the *File* → *Unlock* menu item
- Update the test case code  
(For example, change the ping destination address from 192.168.1.1 to 192.168.1.10)
- Select *File* → *Save* to save the changes  
(A “Success” dialog will pop-up to confirm the operation)
- Your test is ready to execute again.

**It can't get any easier than this!**

innovation in automation



## Task #2: Execution Summarization

Learn to:

- create a summarized execution report
- drill-down to access finer details
- access details of execution reports

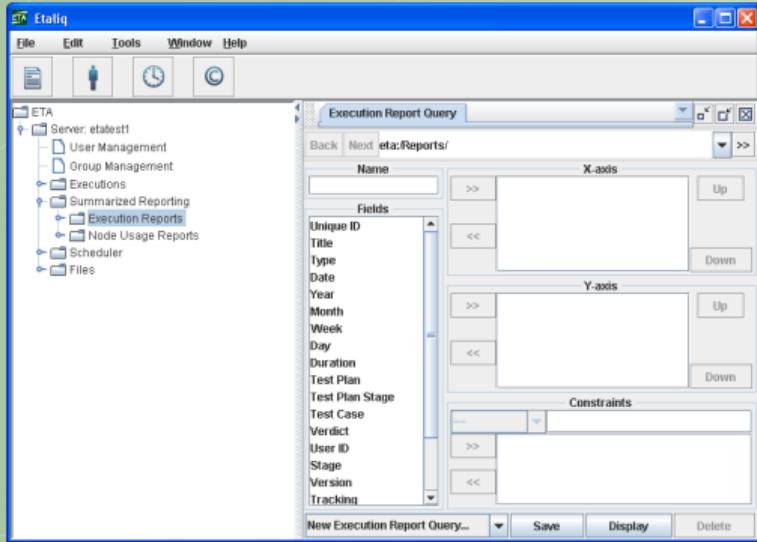
Register at <http://www.etaliq.com> to access the *Live ETA Demo*

innovation in automation



# Task #2: Execution Summarization

Summarized Reporting



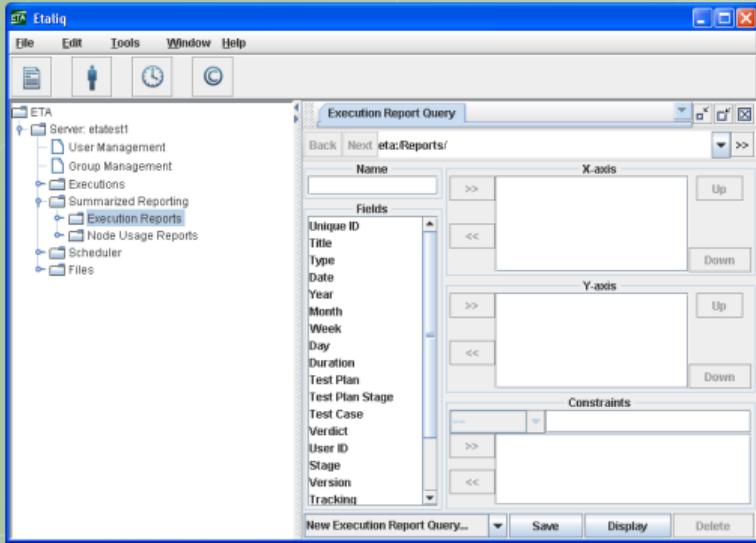
- Expand the *Summarized Reporting* tree item to display a list of report types

innovation in automation



# Task #2: Execution Summarization

Summarized Reporting



- Expand the *Summarized Reporting* tree item to display a list of report types
- *Execution Reports* rolls up the results of multiple executions to provide summarized *Execution Reports*

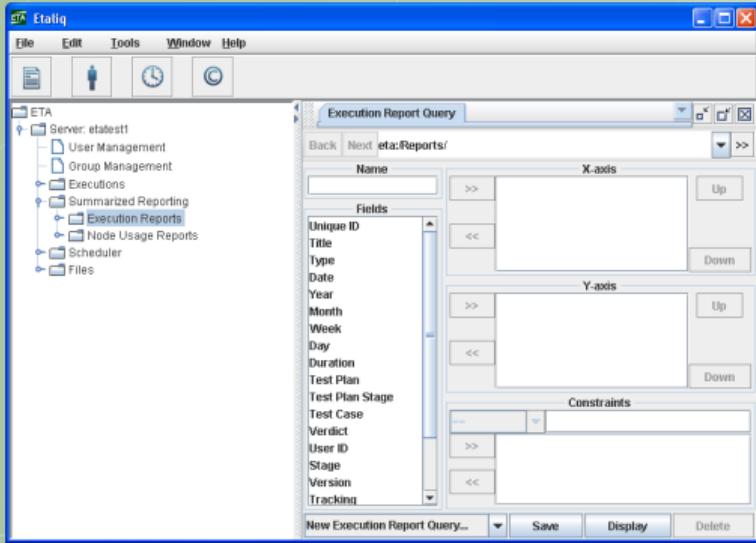
innovation in automation

[www.etaliq.com](http://www.etaliq.com)



# Task #2: Execution Summarization

Summarized Reporting



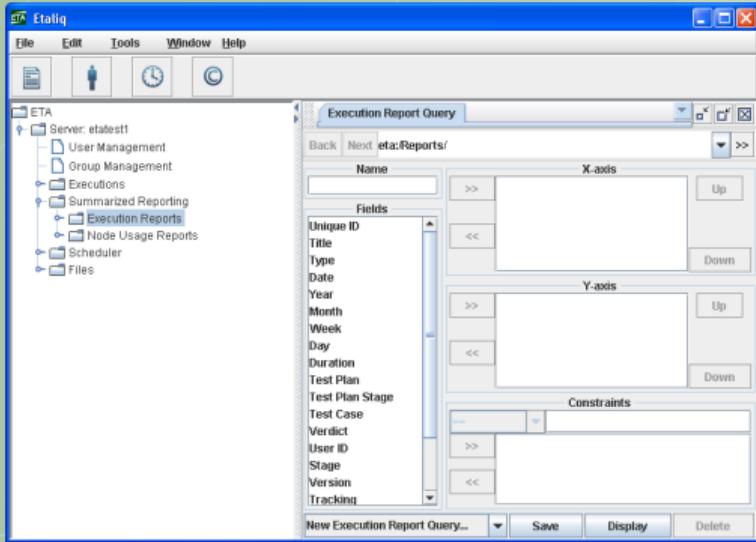
- Expand the *Summarized Reporting* tree item to display a list of report types
- *Execution Reports* rolls up the results of multiple executions to provide summarized *Execution Reports*
- *Node Usage Reports* tracks the usage of nodes by executions and manual lockouts to provide summarized *Node Usage Reports*

innovation in automation



# Task #2: Execution Summarization

Summarized Reporting



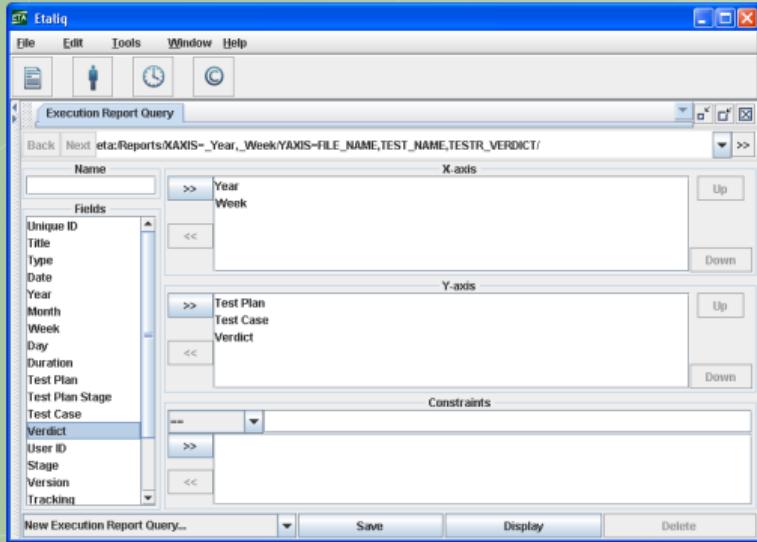
- Expand the *Summarized Reporting* tree item to display a list of report types
- *Execution Reports* rolls up the results of multiple executions to provide summarized *Execution Reports*
- *Node Usage Reports* tracks the usage of nodes by executions and manual lockouts to provide summarized *Node Usage Reports*
- Select the *Execution Reports* link to activate the *Execution Report Query*

innovation in automation



## Task #2: Execution Summarization

### Execution Report Query



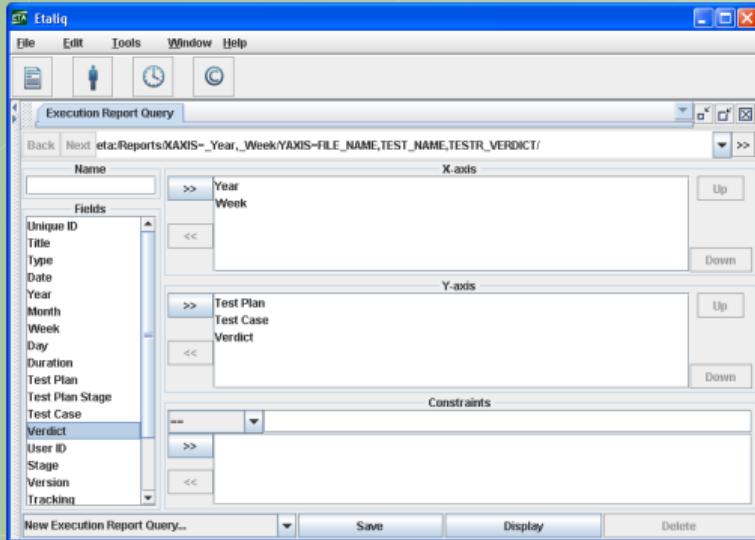
- From this screen, many combinations of fields can be selected to appear on the X- or Y-axis of the report

innovation in automation



# Task #2: Execution Summarization

## Execution Report Query



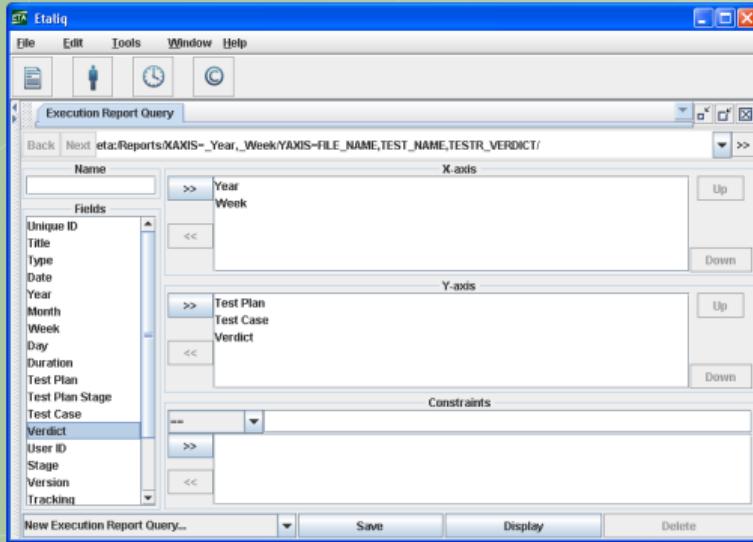
- From this screen, many combinations of fields can be selected to appear on the X- or Y-axis of the report
- For each of **Year** and **Week**, select the field and press the top "**>>**" button to add it to the X-axis

innovation in automation



# Task #2: Execution Summarization

## Execution Report Query



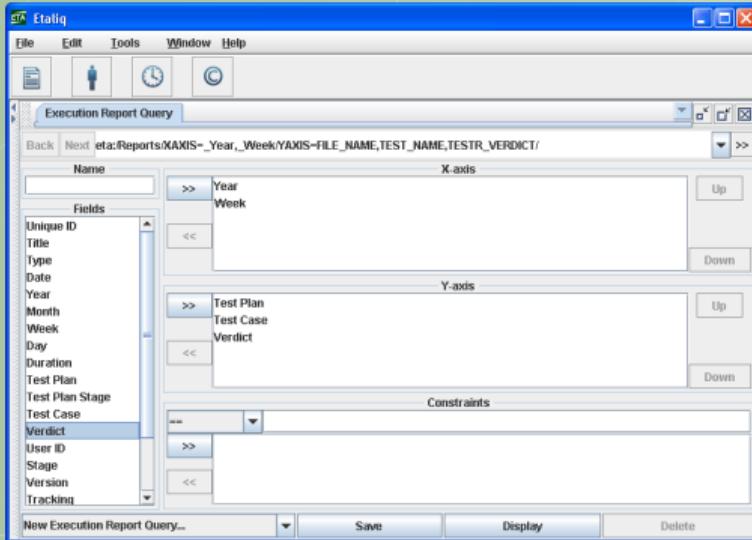
- From this screen, many combinations of fields can be selected to appear on the X- or Y-axis of the report
- For each of Year and Week, select the field and press the top “>>” button to add it to the X-axis
- For each of *Test Plan*, *Test Case* and *Verdict*, select the field and press the middle “>>” button to add it to the Y-axis

innovation in automation



# Task #2: Execution Summarization

## Execution Report Query



- From this screen, many combinations of fields can be selected to appear on the X- or Y-axis of the report
- For each of Year and Week, select the field and press the top “>>” button to add it to the X-axis
- For each of Test Plan, Test Case and Verdict, select the field and press the middle “>>” button to add it to the Y-axis
- Press the *Display* button to create the report

innovation in automation



# Task #2: Execution Summarization

## Execution Report

Etaliq

File Edit Tools Window Help

Execution Report Query Execution Report

Back Next eta:Reports/XAXIS=\_Year,\_Week/YAXIS=FILE\_NAME,TEST\_NAME,TESTR\_VERDICT/

Execution Report

			2009		TOTAL
			31	32	
jtest12-ETA_Router_Example	etaTest	CHILDFAIL	1	2	6
	etaTest_routertest	CHILDFAIL	1	2	6
	etaTest_routertest_basicSendReceiveVerify	PASS	1	2	6
	etaTest_routertest_basicSendReceiveVerify_1	PASS	1	2	6
	etaTest_routertest_basicSendReceiveVerify_2	PASS	1	2	6
	etaTest_routertest_basicSendReceiveVerify_3	PASS	1	2	6
	etaTest_routertest_basicSendReceiveVerify_4	PASS	1	2	6
	etaTest_routertest_configInterfaceVerification	CHILDFAIL	1	0	1
		PASS	0	2	2
	etaTest_routertest_configInterfaceVerification_1	FAIL	1	0	1
		PASS	0	2	2
	etaTest_routertest_showInterfaceVerification	CHILDFAIL	1	2	6
	etaTest_routertest_showInterfaceVerification_1	PASS	1	2	6

- Following any link will “drill-down” to the next level of report

innovation in automation



# Task #2: Execution Summarization

## Execution Report

The screenshot shows the Etaliq software interface with a title bar "Etaliq" and a menu bar "File Edit Tools Window Help". Below the menu is a toolbar with icons for file operations. The main window has tabs "Execution Report Query" and "Execution Report". The "Execution Report" tab is active, displaying a report titled "Execution Report" with a table. The table has columns "FILE\_NAME=jettest12-ETA\_Router\_Example" and "TEST\_NAME=etaTest". The table contains the following data:

	2009	TOTAL
CHILDFAIL	31 32	
etaTest	1 5	6
TOTAL	1 5	6

Below the table, there is a section titled "Constraints" containing the following text:

```
FILE_NAME=jettest12-ETA_Router_Example  
&& TEST_NAME=etaTest
```

- Following any link will “drill-down” to the next level of report
- When a **header link** is followed, the created report will only include that **row or column**

innovation in automation



## Task #2: Execution Summarization

### Execution Report

The screenshot shows the Etaliq software interface with the following details:

- Execution Report Table:**

	2009	TOTAL
31	32	
<a href="#">jsttest12-ETA_Router_Example</a>	<a href="#">etaTest</a>	<a href="#">CHILDFAIL</a>
TOTAL	1	6
	1	6
- Constraints:**

```
FILE_NAME=jsttest12-ETA_Router_Example
&& TEST_NAME=etaTest
```

- Following any link will “drill-down” to the next level of report
- When a header link is followed, the created report will only include that row or column
- When a **cell link** is followed, an **Executions table** will be displayed of all the executions that matched that cell

innovation in automation



## Task #2: Execution Summarization

### Execution Table

Details	Title	Schedule	Test Plan	Exec Table	TEST_NAME	User ID	Start	Length	Status
Details	jstest12-ETAjstest12-ETAlphaTest	jstest12	2009-08-03 11:41		complete	C			
Details	jstest12-ETAjstest12-ETAlphaTest	jstest12	2009-08-03 11:43		complete	C			
Details	jstest12-ETAjstest12-ETAlphaTest	jstest12	2009-08-03 11:21		complete	C			
Details	jstest12-ETAjstest12-ETAlphaTest	jstest12	2009-08-03 11:20		complete	C			
Details	jstest12-ETAjstest12-ETAlphaTest	jstest12	2009-08-03 11:21		complete	C			

- Within the Executions table, a *Details* button appears in the first column of every individual execution.

innovation in automation



## Task #2: Execution Summarization

### Execution Table

ETA EtaLiq

File Edit Tools Window Help

Execution Report Executions jsttest12-ETA\_Example vs jsttest12-ETA\_Router\_Example (2009-08-03 12:08:02)

TYPE VERDICT SWVERSL TEST PLAN TEST CASE TITLE EXEC START EXEC END EXEC DUR... EXEC USER

TYPE	VERDICT	SWVERSL	TEST PLAN	TEST CASE	TITLE	EXEC START	EXEC END	EXEC DUR...	EXEC USER
execTable	PASS		jsttest12-ExecTable		2009-08-03 2009-08-03 1				root
group	CHILDFAIL		jsttest12-BetaTest	ETA tool	tz2009-08-03 2009-08-03 135				root
group	CHILDFAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 135				root
group	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 22				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 1				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 10				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 8				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 1				root
group	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 9				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 9				root
group	CHILDFAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 79				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 7				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 7				root
test	FAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 9				root
test	FAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 47				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 7				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 1				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 0				root
test	PASS		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 3				root
group	CHILDFAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 2				root
test	FAIL		jsttest12-BetaTest	_routeTA tool	tz2009-08-03 2009-08-03 2				root

Attributes Syntax Report Detailed Report Summary Report Console.RouterNode1 jsttest12-ETA\_Example.execTable (

- Within the Executions table, a *Details* button appears in the first column of every individual execution.
- Click one to open all the **report files** associated with that **specific execution**.

All information is just a click away!

innovation in automation

# Case Studies

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)

## Case Study #1

### Test Automation: In-house Isn't the Answer

#### Who

- Telecom hardware start-up
- 50 engineers

#### Challenge

- Rapid black-box device testing
- Long term regression tests
- Broad coverage

#### Solution

- Old Linux server
- ETA Client and Server install
- A Student Engineer

#### Result

- Useful tests immediately
- 140 tests in less three months
- Protocol errors corrected
- Impressive customer feedback

Download the full text at <http://www.etaliq.com>

innovation in automation

## Case Study #2

### Test Automation: Do More Testing in Less Time

#### Who

- Carrier class network equipment provider
- Over 10,000 engineers

#### Solution

- ETA Client and Server Install
- Develop a flexible table-based configuration suite

#### Challenge

- Configure DUT in preparation for manual testing
- Currently takes each engineer 2–6 weeks

#### Result

- 6 person months meet 99% of requirements
- Provided 80 additional test cases to verify stability
- Became a licenced Etaliq ETA customer
- ROI well beyond expectations

Download the full text at <http://www.etaliq.com>

innovation in automation

## Case Study #3

### Test Automation: Do More Testing with Less Hardware

#### Who

- Carrier class network equipment provider
- Over 10,000 engineers

#### Challenge

- Determine current testbed configuration, and adapt test cases to suit
- Give summarized reports of result by combination

#### Solution

- ETA Client and Server Install
- 4 person months development time
- Developed a configuration reader
- Created 130 adaptable tests creating 2,200 unique results

#### Result

- Up to 2,500 results per day, per testbed
- ROI well beyond expectations

Download the full text at <http://www.etaliq.com>

innovation in automation



## Case Study #4

### Test Automation: Solve Problems Quickly

#### Who

- Carrier class network equipment provider
- Over 10,000 engineers

#### Challenge

- Place the DUT under heavy stress
- Allow other tests in parallel

#### Solution

- ETA Client and Server Install
- 6 person days development time

#### Result

- Solution delivered
- Fast, reliable, customizable
- ROI well beyond expectations

Download the full text at <http://www.etaliq.com>

innovation in automation

# Conclusion

innovation in automation

innovation

innovation in automation

innovation in automation

[www.etaliq.com](http://www.etaliq.com)



## Innovation in Automation

**Etaliq** revolutionizes and accelerates automated testing by providing an integrated framework where test plans and scripts are merged into a single document using **ETA**'s patent-pending command language.

innovation in automation



## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?

innovation in automation



## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?

innovation in automation



## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?
- Have your Subject Matter Experts (SMEs) write the automation?

innovation in automation

## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?
- Have your Subject Matter Experts (SMEs) write the automation?
- Create and verify your automation prior to feature availability?

innovation in automation



## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?
- Have your Subject Matter Experts (SMEs) write the automation?
- Create and verify your automation prior to feature availability?
- Get summary pass/fail reporting real time by day/week/month?

innovation in automation

## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?
- Have your Subject Matter Experts (SMEs) write the automation?
- Create and verify your automation prior to feature availability?
- Get summary pass/fail reporting real time by day/week/month?
- Create table-driven setups?

innovation in automation

## Manual & Automation Related Questions

What if you could

- Substantially reduce automation complexity?
- Use automation to assist in manual testing?
- Have your Subject Matter Experts (SMEs) write the automation?
- Create and verify your automation prior to feature availability?
- Get summary pass/fail reporting real time by day/week/month?
- Create table-driven setups?
- Create your automation in 1/10<sup>th</sup> the time?

innovation in automation



## Conclusion

If the answer is YES.

innovation in automation



## Conclusion

If the answer is YES.

We can help.

innovation in automation



## Conclusion

If the answer is YES.

innovation in automation



## Conclusion

If the answer is YES.

# ETALIQ

innovation in automation intelligent quality