

Wizards for Creating Projects, .tst Files, and Tests

For help selecting and completing the available project, .tst file, and test creation wizards, refer to the descriptions and links in the following tables:

- [General Wizard Options](#)
- [REST and SOA Wizard Options](#)
- [Web Wizard Options](#)

General Wizard Options

Wizard	Description	Reference
Import Test (.tst file) Project	Imports a project that was previously exported through the SOAtest export feature.	Importing a Test Suite Exporting a Test Suite
Empty Test (.tst file) Project	Creates an empty project or .tst "skeleton" to which you can add the desired tests.	Creating an Empty Project
Reference Test (.tst) File Project	Uses an existing project/.tst file as a building block for constructing another—reusing your existing test assets.	Using Test Suite References
SOAtest Java Project	Creates a new Eclipse Java project that has access to SOAtest's Extensibility API. You can later configure SOAtest scripts and Extension tools to invoke classes from the new Java project.	Using Eclipse Java Projects in SOAtest
Project from Existing SOAtest or WebKing Test Suites	Creates a new SOAtest project based on pre-existing SOAtest or WebKing test suites (including test suites from earlier generations of SOAtest or WebKing).	Setting Up Projects for Existing Tests
SOA Policy Configuration File	Creates a SOA policy file that addresses concerns such as interoperability and compliance to industry standards, as well as maintainability and best practices.	SOA Policy Enforcement: Overview
Data Model Definition File	Creates a data model definition file that tells SOAtest how the organization's fixed length messages are structured. SOAtest will use this information to convert these messages to and from XML.	Fixed Length Client and Fixed Length Call Back

REST and SOA Wizard Options

Many of the wizards below also provide an option to create "policy enforcers" for WSDLs and SOAP messages.

Wizard	Description	Reference
AmberPoint Management System	Creates: <ul style="list-style-type: none">• Functional tests for the messages captured in a specified plain text traffic log/trace file.• Regression controls that validate whether each request continues to have the expected response (the response captured in the file) when the messages are replayed.	Creating Tests From AmberPoint Management System
BEA Aqualogic repository	Creates: <ul style="list-style-type: none">• The types of tests specified in the ALER repository (for example, functional tests, WSDL tests).	Creating Tests From Oracle Enterprise or BEA Aqualogic Repositories
BPEL	Creates: <ul style="list-style-type: none">• BPEL semantics tests.• WSDL tests.• BPEL process tests.• BPEL partner tests.	Creating Tests From BPEL Files
Java Message Service (JMS)	Configures SOAtest to: <ul style="list-style-type: none">• Monitor transactions that pass through the a JMS system.• Generate functional test cases that replay and check the monitored messages.	Creating Tests From JMS System Transactions

RAML	Creates: <ul style="list-style-type: none"> Functional tests for the end-points found in a RAML description. 	Creating Tests From a RAML Definition
Record from traffic	Records HTTP, JMS, or MQ traffic. You can later create the following from the generated traffic file: <ul style="list-style-type: none"> Functional tests for the messages captured in the specified traffic file. Regression controls that validate whether each request continues to have the expected response (the response captured in the file) when the messages are replayed. 	Creating Tests From Recorded HTTP, JMS or MQ Traffic
Sonic Enterprise Service Bus	Configures SOAtest to: <ul style="list-style-type: none"> Monitor transactions that pass through a Sonic ESB system. Generate functional test cases that replay and check the monitored messages. 	Creating Tests From Sonic ESB Transactions
OpenAPI/Swagger	Creates: <ul style="list-style-type: none"> Functional tests for the end-points found in a OpenAPI/Swagger description. 	Creating Tests From a OpenAPI /Swagger Definition
TIBCO Enterprise Messaging Service	Configures SOAtest to: <ul style="list-style-type: none"> Monitor transactions that pass through a TIBCO EMS system. Generate functional test cases that replay and check the monitored messages. 	Creating Tests From TIBCO EMS Transactions
Traffic	Creates: <ul style="list-style-type: none"> Functional tests for the HTTP, JMS, or MQ messages captured in the specified traffic file. Regression controls that validate whether each request continues to have the expected response (the response captured in the file) when the messages are replayed. 	Creating Tests From Recorded HTTP, JMS or MQ Traffic
UDDI	Creates: <ul style="list-style-type: none"> Functional tests for each service operation retrieved from the query. WSDL tests to ensure that the WSDL conforms to the schema and passes XML validation tests. 	Creating Tests From a UDDI
WADL	Creates: <ul style="list-style-type: none"> Functional tests for each operation defined in the WADL. 	Creating Tests From a WADL
WSIL	Creates: <ul style="list-style-type: none"> Functional tests for each object associated with the WSDLs within the specified WSIL. WSDL tests to ensure that the WSDL conforms to the schema and passes XML validation tests. 	Creating Tests From a WSIL
WSDL	Creates: <ul style="list-style-type: none"> Functional tests for each operation defined in the WSDL. Comprehensive WSDL tests to ensure that the WSDL conforms to the schema and passes XML validation tests. 	Creating Tests From a WSDL
XML Schema	Creates: <ul style="list-style-type: none"> Functional tests (SOAP Client or Messaging Client) for elements in an XML schema. 	Creating Tests From XML Schema

Web Wizard Options

Wizard	Description	Reference
Record web scenario	Creates functional tests that represent the actions taken as you exercise the application from a browser.	Browser Recording and Playback

Scan web application	Configures a test that crawls the specified Web application (e.g., to prepare for static analysis).	Configuring SOAtest to Scan a Web UI
----------------------	---	--