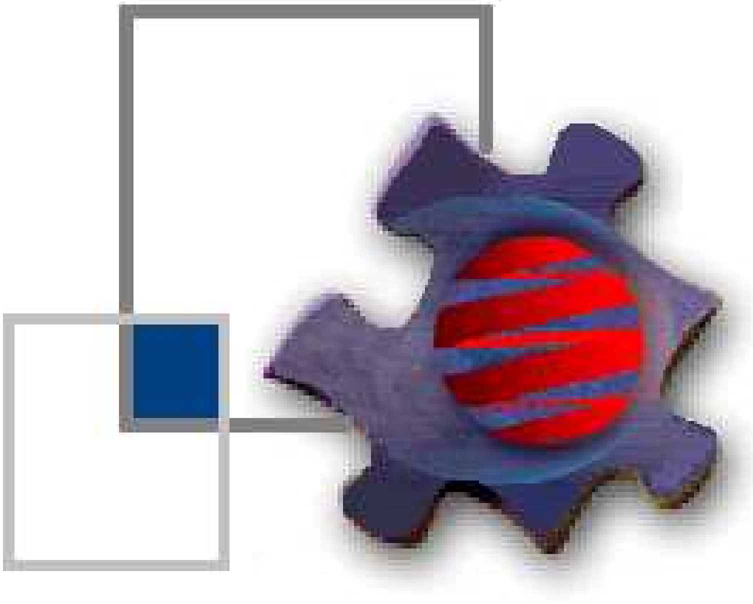


Testing websites automatically with *JStudio SiteWalker* *Professional 2.5*

Contents:

All information of this document relates to the Professional Edition only. Please find further product and purchase information at www.jstudio.de

Enjoy working in your projects with JStudio SiteWalker Professional !

Frank Jarsch
Jarsch Software Studio
Kattowitzer Strasse 26
51065 Köln (Cologne)

1. Automated Tests – a disillusion story

In medium and large projects automating tests for web based client functionality by using script based test tools and programmed test algorithm using test libraries reaches limits that can't be handled. Very often high potential and expensive test tools requires high skilled testing stuff and a complex runtime environment – which raises efforts immense. Often maintainance and working on changes of testing scripts and routines caused by application changes leads to inefficiency and often gets loosing control. In the result, in many projects tests will be automated only selectively anymore or done completely manually.

By definition, testing is a repetitive activity. The very nature of application software development dictates that no matter which methods are employed to carry out testing (manual or automated), they remain repetitious throughout the development lifecycle. Automation of testing processes allows to complete the tedious, repetitive work while human personnel perform other tasks.

How it looks in reality

Huge phases for manually tests by business departments blocks valuable business and IT resources for long time periods. Repeating those tests after changes or bug fixes raises the whole testing scenario again – and raises also all efforts too. Testing selectively often do not cover all critical parts and might lead to surprises at application processes that had not been tested.

Manually tests will be done by testers eye – and are limited to the testers business and IT skill and it's concentration after hundreds of tests has been done. All test scenarios often bases on test case definitions which often had not been updated after some periods, so tests became imprecise. Exact repetition and a secured quality result is nearly impossible.

After software aproval tests by the customer in most cases the application will be enhanced and maintained further on. Often new releases either will not tested again completely or starts all testing activities with huge efforts again.

In practise today quality and testing stuff is not part of the development team or join the team at a later project phase. So often the whole knowledgement of business or technical processes needed to specify and create tests must be communicated and transferred before. In complex projects communication increase considerable efforts.

All these negative items leads to disillusionment in projects practise when using automated tests.

Increased technical and business complexness increases the corresponding requirements for quality management and testing issues also. If there are errors they were not found during testing period the damage for software vendors might be to much to count on fortune only concerning quality of tests.

JStudio SiteWalker Professional provides all capabilities to make automated tests more efficient, more comfortable, and more easier to assist project stuff in software testing – to contribute in reducing costs and efforts significant.

2. How JStudio SiteWalker supports your projects

Avoid searching errors – let them find

JStudio SiteWalker Professional offers an extensive and intuitive solution for automating websites and webapplication to automate functional tests, regression tests with flexible validation. Web based processes can be completely automated and all content can be validated automatically to avoid searching errors – just let them find.

Creating tests without beeing expert

Beyond unit tests, JStudio SiteWalker increases the concepts of developer tests in the area of web based GUI tests and enables developers and business staff to reach high level of tests coverage for client side Functional Tests and Regression Tests. Without expert knowledgement of test automization JStudio SiteWalker enables unexperienced testers to get first productive test results within minutes. By using browser automation - navigations, automatically filled form inputs - used with static values or imported test data from Excel or databases JStudio SiteWalker fits to many requirements of automated business processes for websites or web applications. For each element to be tested, test cases can be defined as assertions by using the Quick Validator - without programming.

Data diven tests & module driven tests

Automated workflows can be separated in modules and executed depending on the results of failures of other workflows. Many test cases, based on single workflows, can be executed repetitive by using Task Iteration to execute data driven tests using different types of datasources, i.e. Microsoft Excel or databases. If specialized, more complex validation is required, e.g. comparing or validating displayed content values, this can be done by using JavaScript and defining assertions or launching different test cases using the specific Insert Declares.
All results and customized test case descriptions for the executed automated tests are saved into defined reports in HTML format and can be displayed and evaluated after accomplish your tests directly in the browser. They shows an overview page to get into any details of test case that are happened and logged during test procedure.

Reduced maintainance

Tests can be changed when websites has been changed just be changing elements attribute or to re-assign HTML elements by mouse click. So JStudio SiteWalker reduces efforts of maintainance of automated tests to enables the whole potential during the full software life cycle.

By using browser automation, automated form inputs - using static values or retrieved values from data sources - and by defining individual script operations and validations as quick dialog definition or JavaScript routines, JStudio SiteWalker fits to a large number of test scenarios for automated tests in the area of web based applications.

Test automation is not for free, but it pays off.

3. Automated tests as authentic user scenarios

Test cases defined for testing web application on client side can be created to replay any automated test scenario, because all actions will be done like a human user would do these actions manually using the browser. By using JavaScript Insert Declares, fail conditions or assertions can be defined to validate values or structure of loaded web documents. Test results can be traced to report files that can be displayed and validated with the browser. Because flexible automation processes can be defined by using all this features, JStudio SiteWalker reduces high efforts when creating and defining test cases. By using the capture & replay concept, creating test cases can be done in an easy and fast way to leave more time for higher quality test case definitions and quality security.

Simultanous ./. authentical workflow scenarios

For testing web applications automatically known unit tests libraries are available that offers a lot of functionalities to simulate web actions of a browser. In default, no browsers are part of the environment to execute automated tests, but user defined routines to simulate those browser behaviour by programmed tests. Also some other capture & replay automation products are available that allows to capture actions of the scenario by recording them with a browser. In default, actions are saved as propriatary script languages, which are executed to play all actions.
Executing programs or scripts just simulates the actions elements, because not the action element itself will be activated to proceed the navigation action. E.g. a hyperlink address was saved or defined on recording or developement time, but its action will be done later by navigating the address via http. The hyperlink element itself is not needed anymore to proceed the test, so the test would have been proceeded correctly even if the element is not placed within the browser document anymore.

Element based testing

In opposite to this behaviour JStudio SiteWalker just notes identifier attributes during recording to re-identify any element at runtime. So each HTML element that was activated and applied to a Navigation Task must be included in the web document at runtime also. To find the element is required to activate the element itself in the web document for starting the characteristic action, e.g. a hyperlink will be clicked in real. Also this behaviour triggers all events that are normally binded to this action. This concept allows automated navigation in a high authentic way that is needed for more efficient test cases and enhanced test authenticity.

4. What kind of tests can be done with JStudio SiteWalker ?

JStudio SiteWalker enables you to create Functional tests, Integration tests, Regression tests as Black Box tests.

- **Functional** - functional testing is the most basic type of testing. It examines whether the application works in the way the designers intended. Functional testing forms the foundation for most of the other types of software testing.

- **Integration** - Integration testing evaluates whether an application communicates and works with other elements in the computing environment. Integration testing ensures that complex system components can share information and coordinate to deliver the desired results.

- **Regression** - Regression testing is used to ensure that new or updated versions of an application work as planned. Regression tests examine new versions of the application to ensure that they do not produce negative effects on other parts of a system.

Client side tests

Only the result view on client side - the web document itself - will be tested without testing the further logical operations needed to create a page and the displayed results within. In general, each automated website navigation is a functional test. Activating hyperlinks, filling input elements and loading the specific result pages indicates that the application works correctly and the integrity of the navigation and GUI processes or scripts is valid. The retrieved result page also includes content values that can be interpreted - depending on individual application purposes and business logic - as results that have to be tested. This document content can be validated by logical operations and fail definitions using fast and flexible validation rule dialog or JavaScript routines.
JStudio SiteWalker supports automated tests of websites and their content validation in very efficient way because all specific web navigations can be done without any programming. Individual test case definitions by JavaScript is provided only for those cases you need validation of documents content in more specific way - what can be done easily without any specific runtime environment.

Using the Capture & Replay concept and simple validation rules, JStudio SiteWalker reduces IT knowledgement needed to enable business stuff specifying most test cases directly. Additional for developers the integration of JavaScript allows access onto all elements within the browser document context and unrestricted opportunities for enhanced validations and logical operations.

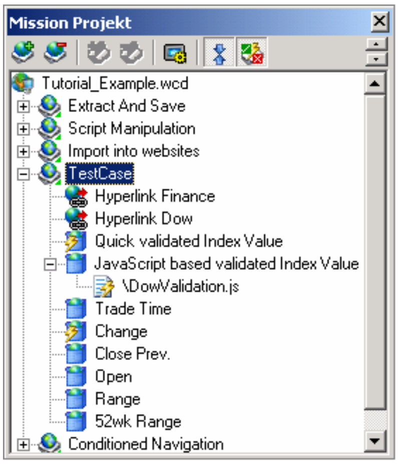
Dare for changes

Automation allows the testing organization to perform consistent and repeatable tests. For complex web projects, particulary projects using high level of descriptive components (XML, XSLT) or when working in iteration phases, creating automated tests should be started at an early stage of the application to validate the correctness at any development stage just by executing the test cases. Even, when applications need to be deployed across different hardware or software platforms, integration tests can be created and repeated on target platforms to ensure that new platforms operate consistently. Because no specific environment is needed - all tests will be done within JStudio SiteWalker - test efforts are minimized. Testing on every stage finds all changes and errors and helps you to get more transparency of the real state of your project and gives certainty to dare changes on running systems.

5. Creating automated tests

5.1. The Mission Project – the Testsuite

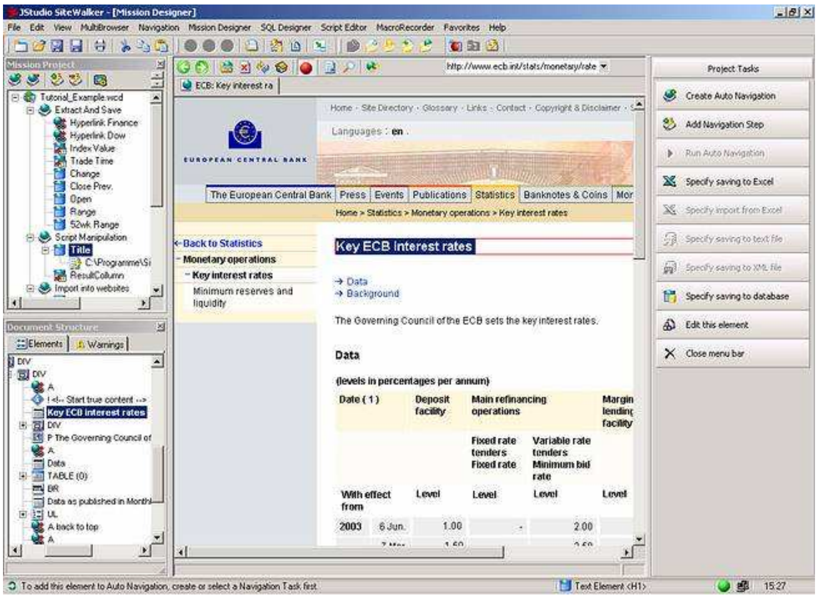
In the Mission Project Pane all elements and sources will be displayed already added to the Mission Project – the Testsuite and all contained Testcases.
All central funktionalities needed to manage and operate the current project elements can be reached via context menu or the standard menu in the JStudio SiteWalker menu bar. The nodes in the project tree view can be edited and changed at any time.



Picture: Mission Project as Testsuite in the Mission Project Pane

5.2. The view *Mission Designer*

The view Mission Designer is the central perspective of JStudio SiteWalker.
Using this view and the integrated browser processes will be recorded – just by surfing and navigating the web application - click yourself through all pages by following hyperlinks etc. Clicked elements of the current page will be - depending on your settings – automatically or after confirmation - added to the Mission Project. With a few mouse clicks you can configure your Mission Project, which can be 'played' on every time as Auto Navigation.



The Governing Council of the ECB sets the key interest rates.

Data

(levels in percentages per annum)

Date (1)		Deposit facility	Main refinancing operations		Margin lending facility
			Fixed rate tenders Fixed rate	Variable rate tenders Minimum bid rate	
With effect from		Level	Level	Level	Level
2003	6 Jun.	1.00	-	2.00	

To add this element to Auto Navigation, create or select a Navigation Task first

Picture: View *Mission Designer* for recording automated web navigations

In JStudio SiteWalker all automated tests are – browser based - automated web navigations. Recording and configuring automated Navigation Tasks is the base for all more specific test scenarios. Therefore created Navigation Tasks represents an complete and - depending on the process to test - fixed workflow process. By creating a Navigation Task with intention for testing web applications the Navigation Tasks can be executed as Test Case. Attending to this more specific purpose all further configuration tasks that can be done leads into the test case definition. E.g. all description settings that can be done should have been used for description of the test case. All outputs or inputs required defined in the Test Case now are used with the intention to enter test data or extract test results. In general all test case steps and traced output can be saved to a test report, a specific file output to complete the service for testing web applications.

5.3. Creating Navigation Tasks as Test Case

5.3.1. Navigation Task = web based process

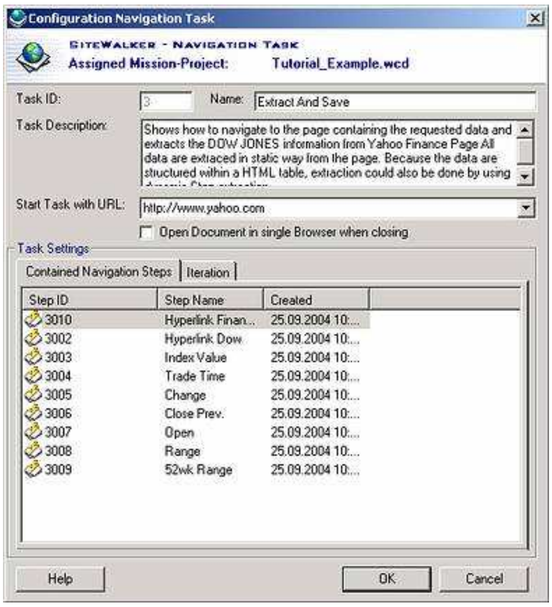
In general, a Navigation Task has to be created to automate a test case. A Navigation Task is the starting point to begin a surf session which should be executed automatically at a later time. Like normal surfing, a Task starts on entering an internet address to load a document into the browser. If the requested document is loaded and parsed, every user activity on the loaded document will be added to the Navigation Task as 'Navigation Step', so Navigation Tasks contains child Navigation Steps as sequence of user actions – that represents the web based workflow process.

5.3.2. Navigation Task = Test Case

For generating meaningful Test Reports while executing Test Cases, please describe the Test Case definition in the description field of the configuration dialog of the base Navigation Task. Describe the workflow to be tested and the expected results of navigation steps or values that are expected to be displayed within the result page. These information are important to be available within the test reports.

Record all single user actions by applying Navigation Steps to the current Task. Also use description fields to enter information what they are used or validated for.

If the automated navigation web process has been completed, please test it that it runs and validates correct and simulates the recorded workflow process as required to focus on the critical issues to be tested.



Assigned Mission-Project: Tutorial_Example.wcd

Task ID: 3

Name: Extract And Save

Task Description: Shows how to navigate to the page containing the requested data and extracts the DOW JONES information from Yahoo Finance Page All data are extraced in static way from the page. Because the data are structured within a HTML table, extraction could also be done by using

Start Task with URL: http://www.yahoo.com

☐ Open Document in single Browser when closing

Task Settings

Contained Navigation Steps | Iteration

Step ID	Step Name	Created
3010	Hyperlink Finan...	25.09.2004 10:...
3002	Hyperlink Dow	25.09.2004 10:...
3003	Index Value	25.09.2004 10:...
3004	Trade Time	25.09.2004 10:...
3005	Change	25.09.2004 10:...
3006	Close Prev.	25.09.2004 10:...
3007	Open	25.09.2004 10:...
3008	Range	25.09.2004 10:...
3009	52wk Range	25.09.2004 10:...

Picture: Test Case definition with Task Dialog and sequence of steps

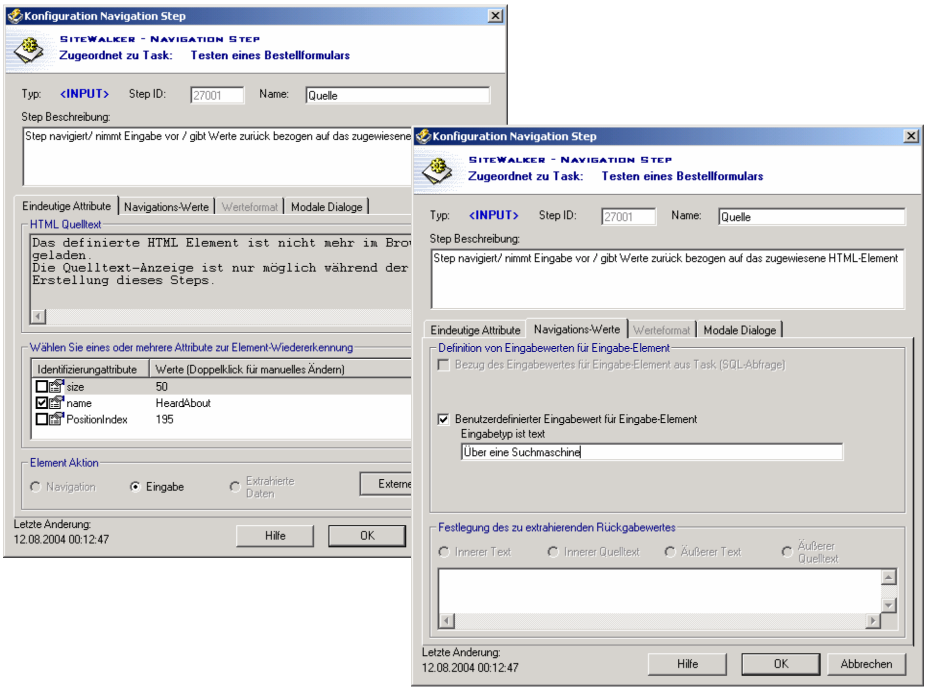
5.4. Recording user actions as Navigation Steps

5.4.1. Navigation Step = User action

Navigation Steps represents any actions within web documents in a browser tab. When a document is loaded many activities can be done within. You can make use of most HTML elements. Useable elements will be displayed in red highlighted frame. Every click on a hyperlink, image, button or posting forms processes an action that must be repeatable and processed later at Auto Navigation runtime. To do so each activated navigation element must be applied and assigned to a

Navigation Task as Navigation Step to trace all action in the right order. Textboxes, Selectboxes, Checkboxes or option boxes will be added, when activated or edited. At least there are content or format elements including any kind of text - that should be extracted and saved. When a text element is activated it will be added in the same way to the Mission Project.

Recording with JStudio SiteWalker means clicking the HTML element within the browser (or picking it from documents tree view) to apply it as Navigation Step to the project. So Steps can be changed, inserted or added at any time to create the web navigation process in iterative way, because no script is generated sequentially by doing "recording".



Konfiguration Navigation Step

SITEWALKER - NAVIGATION STEP

Zugeordnet zu Task: Testen eines Bestellformulars

Typ: <INPUT> Step ID: 27001 Name: Quelle

Step Beschreibung:

Step navigiert/ nimmt Eingabe vor / gibt Werte zurück bezogen auf das zugewiesene

HTML Quelltext

```
Das definierte HTML Element ist nicht mehr im Bro
geladen.
Die Quelltext-Anzeige ist nur möglich während der
Erstellung dieses Steps.
```

Wählen Sie eines oder mehrere Attribute zur Element-Wiedererkennung

Identifizierungattribute	Werte (Doppelklick für manuelles Ändern)
□ size	50
☑ name	HeardAbout
□ PositionIndex	195

Element Aktion

○ Navigation ● Eingabe ○ Extrahierte Daten

Letzte Anderung:
12.08.2004 00:12:47

Konfiguration Navigation Step

SITEWALKER - NAVIGATION STEP

Zugeordnet zu Task: Testen eines Bestellformulars

Typ: <INPUT> Step ID: 27001 Name: Quelle

Step Beschreibung:

Step navigiert/ nimmt Eingabe vor / gibt Werte zurück bezogen auf das zugewiesene HTML-Element

Definition von Eingabewerten für Eingabe-Element

□ Bezug des Eingabewertes für Eingabe-Element aus Task (SQL-Abfrage)

☑ Benutzerdefinierter Eingabewert für Eingabe-Element
Eingabetyp ist text
Über eine Suchmaschine

Festlegung des zu extrahierenden Rückgabewertes

○ Innerer Text ○ Innerer Quelltext ○ Äußerer Text ○ Äußerer Quelltext

Letzte Anderung:
12.08.2004 00:12:47

Picture: Configuration Settings for Navigation Steps

5.4.2. Enhanced actions for Navigation Steps

Depending from type of Step element many enhanced actions can be specified for step elements to configure behaviour and functionality:

- Definition of HTML element re-identifier attributes and using of wildcards and patterns to get high flexibility on variable web document content
- Definition of static input data or data from external data (Microsoft Excel or databases) sources for input controls
- Particular in web applications, modal message or input boxes are used to display a message, to confirm an action by the user or to enter a user defined value. But for automated navigations modal dialogs stops any processes on your computer until the user has reacted to it. So in JStudio SiteWalker those

dialogs can be deactivated to prevent the automation process to be held. This can be done in the same way as dialogs normally would do. The action or value the user manually would do or would enter can be defined at design time and will be automatically done or entered during runtime without displaying the dialog.

- Using loops for flexible working with table item elements
- Enabling / disabling of event element handlers to get high flexibility in elements behaviour to be tested
- Specification of validation rules by dialog based Quick Validation for each HTML element within a loaded web document
- Execution of JavaScript statements and complete script files to specify complex validation routines programmatically

Note: Websites content from different domains (displayed in different frames) only can be handled in reduced functionality mode, because of „Crossframe Security" that does not allow to access to elements from different frames.

5.5. Testing the web navigation

The automated web navigation should be executed to test the recorded functional process. All Navigation Steps searches for their corresponding HTML elements within the browser document: navigation elements will be executed, input elements will be edited; text elements will be extracted. Each action output is traced and displayed within the Output Console of perspective *Auto Navigation Center*.

For executing navigations at design time, processes that might changes data consistency can be disabled during auto navigation, e.g. suppressing saving to Excel or databases, running scripts or submitting auto filled forms.

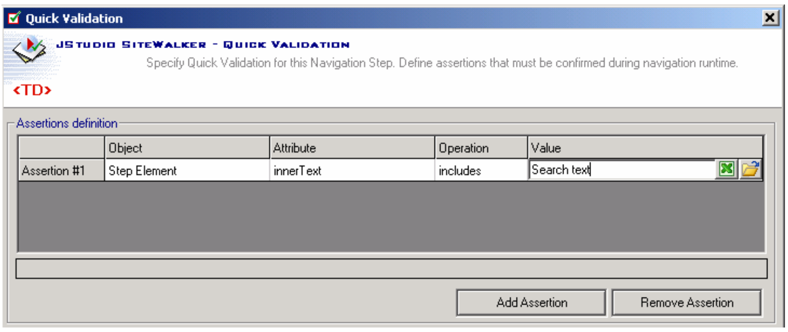
If the process runs without errors, further quality checks – using validation – can be specified.

5.6. Validations, test conditions or assertions

When testing software functional tests are just the basic tests. Additional to this, very often it is required to execute logical operations, validations of data value by comparing test data values and default values. Therefore assertions must be defined to ensure that a specific state or data value will be checked depending on the assertion conditions. If the assertion fails - the specified condition is false - then the whole test fails. If not, the test was successful.

The Quick Validation

The Quick Validator provides fast and flexible validation of websites that can be defined in a comfortable and easy way - without any programming. By defining validation rules as assertions, the user specifies conditions and expressions for HTML elements, that has to be confirmed at auto navigation runtime, whether they are correct or not. Values for comparison can be entered or read from Excel or text files.



Specify Quick Validation for this Navigation Step. Define assertions that must be confirmed during navigation runtime.

<TD>

Assertions definition

	Object	Attribute	Operation	Value
Assertion #1	Step Element	innerText	includes	Search text

Picture: Specify validation rules (“assertions”) with Quick Validation

Actions depending on validation results

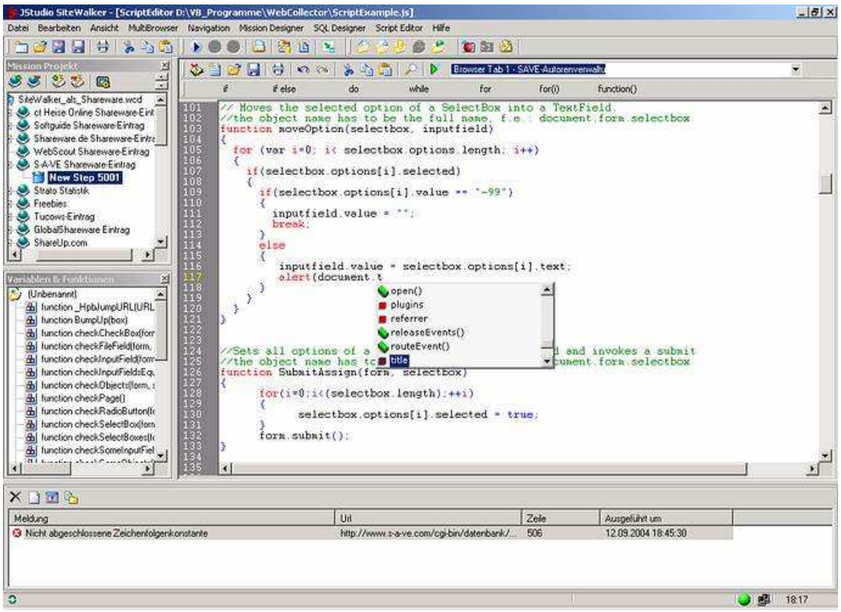
For each validation navigation, actions can be specified that will be performed when the current Step has been validated. These actions offers a powerful way to cancel current Tasks or Steps in case of failures and goto different Tasks or Steps. This enables defining Tasks or Steps as test modules that will be executed depending on validation of websites content to simulate human process decisions. Also allows defining loops to proceed workflows as long as validation will fail.

Creating process moduls

Validation and specified actions can be used to evaluate content of the web documents to decide further process if dependencies between state of the document and process exists. So, e.g. if a customer has been already entered and is shown in the screen it might be necessary to go to the next performing task – and skip the current task. These kind of process branching provides a very high level of authentic process automation.

Enhanced manipulation and validation by JavaScript

JStudio SiteWalker provides using the JavaScript - the script language that reduces efforts, because most web developer already know - so unrestricted access to all elements of a loaded page enables you to define any test conditions on your own without any restrictions you might have using proprietary libraries. To execute scripts it is just necessary to create the script and to assign the JavaScript file to the current Navigation Step. By using JavaScript in this way, Inline Declares to define fail definitions can be used.


JStudio SiteWalker - [ScriptEditor D:\VB_Programme\WebCollector\ScriptExample.js]
Datei Bearbeiten Ansicht MultiBrowser Navigation Mission Designer SQL Designer Script Editor Hilfe
Mission Projekt
Variablen & Funktionen
Browser Tab 1 - SAVE-Autorenverwaltu
if
if else
do
while
for
for(i)
function()
open()
plugins
referrer
releaseEvents()
routeEvent()
title
18:17

```
// Moves the selected option of a SelectBox into a TextField.
//the object name has to be the full name, f.e.: document.form.selectbox
function moveOption(selectbox, inputfield)
{
  for (var i=0; i< selectbox.options.length; i++)
  {
    if(selectbox.options[i].selected)
    {
      if(selectbox.options[i].value == "-99")
      {
        inputfield.value = "";
        break;
      }
      else
      {
       inputfield.value = selectbox.options[i].text;
       alert(document.t
      }
    }
  }
}


//Sets all options of a                      d and invokes a submit
//the object name has tc                     cument.form.selectbox
function SubmitAssign(form, selectbox)
{
      for(i=0;i<(selectbox.length);++i)
      {
            selectbox.options[i].selected = true;
      }
      form.submit();
}
```

Meldung	Url	Zeile	Ausgeführt um
Nicht abgeschlossene Zeichenfolgenkonstante	http://www.s-a-ve.com/cgi-bin/datenbank/...	506	12.09.2004 18:45:30

Picture: Editing JavaScript files with the *DHTML Script Editor*

JStudio SiteWalker integrates a *DHTML Script Editor*, where JavaScript files can be created and edited in order to execute functions for validation or manipulation of extracted data during a running Auto Navigation. Edit JavaScript files and specify for which element they should be executed to get more flexibility. Powerful validation expressions or fail definitions for automated website tests can be defined. Also branched navigation processes can be started or stopped depending from validation results.

5.7. Analog recording with Macro Recorder

The Macro Recorder allows to record all user events, raised by user actions, to replay it at a later time. So it is possible to repeat a whole surf session of all user actions, e.g. mouse moves, enter data, clicks etc., exactly in the way the user did before.

Normally in JStudio SiteWalker all elements of websites are activated in object sensitive mode. This means that every activated (clicked) HTML element in the browser will be referred by JStudio SiteWalker and added to the Mission Project as Navigation Step Node. But websites can also include non sensitive elements - non HTML - e.g. applets, ActiveX elements, Flash animations. These kind of elements are included in the website as single object tag, but might support specific complex functionalities inside and display a real GUI, like separated applications. This GUI supports their own text fields, buttons etc. It is not possible to access these specific controls within those embedded applications/ modules via HTML object reference so they can not used by JStudio SiteWalker for Auto Navigations.

An additional problem can be the using of message boxes occurred by the browser. Instead of using HTML based messages, many web developer uses script inside to raise a standard - non HTML based - Windows message box by the browser functionality.

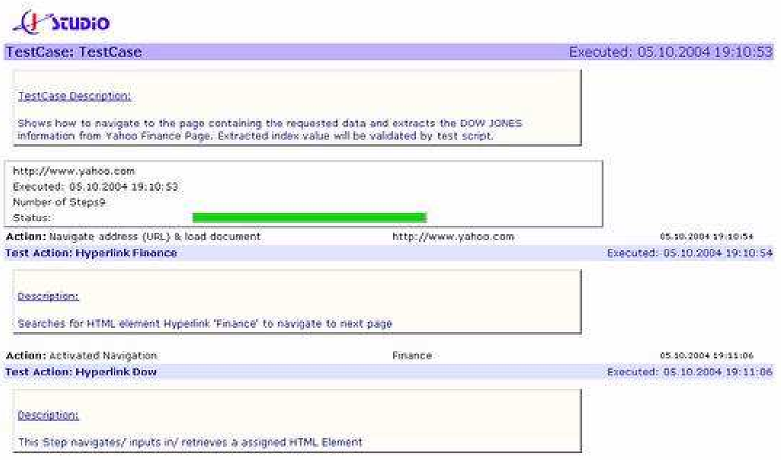
However, using the Macro Recorder supports you to handle those embedded elements. The Macro Recorder records all events and mouse moves raised by the user based on coordinates. All user activities will be recorded and saved as macro file. By starting the macro all actions will be replayed exactly in the recorded way. Because macro files can be added to the Mission Project as specific ‚Macro Step' the macro can be executed during Auto Navigation runtime. If an Auto Navigation is running and should handle a website containing animated intro, the played macro can move the mouse to the displayed graphical link and trigger the wanted action.

All standard browser functionalities available in browser toolbar can also be recorded (e.g. entering urls, back etc.)!

5.8. Generate Test Reports

Reporting

By using browser automation - navigations, automatically filled form inputs – used with static values or imported values from datasources - and the avaibility of customized script operations executed in websites - JStudio SiteWalker fits to many requirements of automated test cases for websites or web applications. Running - manually or scheduled - automated workflows of your sites JStudio SiteWalker simulates a human user to test or re-test specified workflows. All results for the executed automated tests can be saved into definied report file in HTML format and can be displayed and evaluated after finishing your tests directly in the browser.



TestCase: TestCase Executed: 05.10.2004 19:10:53

TestCase Description:

Shows how to navigate to the page containing the requested data and extracts the DOW JONES information from Yahoo Finance Page. Extracted index value will be validated by test script.

http://www.yahoo.com
Executed: 05.10.2004 19:10:53
Number of Steps9
Status:

Action: Navigate address (URL) & load document http://www.yahoo.com 05.10.2004 19:10:54

Test Action: Hyperlink Finance Executed: 05.10.2004 19:10:54

Description:

Searches for HTML element Hyperlink 'Finance' to navigate to next page

Action: Activated Navigation Finance 05.10.2004 19:11:06

Test Action: Hyperlink Dow Executed: 05.10.2004 19:11:06

Description:

This Step navigates/ inputs in/ retrieves a assigned HTML Element

Picture: Detailed Test Case report

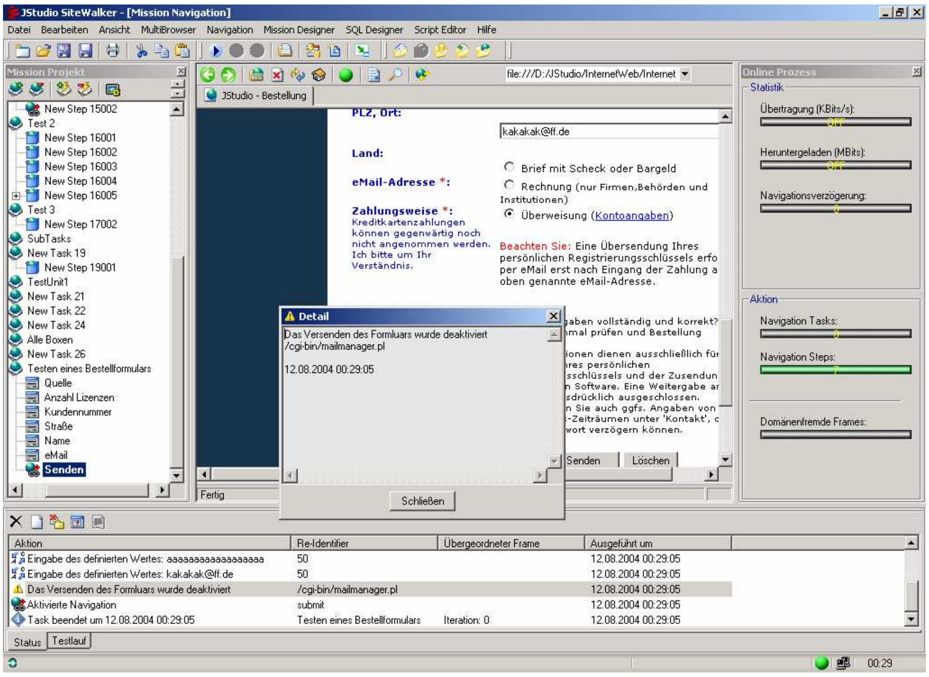
Logging

All output information of Auto Navigation operations will be logged and displayed in the Navigation Center Output Console and also inserted into the specified Test Report. Execution and result reporting of Tasks can be enabled and disabled for each single Task via Mission Configuration dialog. This allows you to separate Tasks to switch with Tasks and different processes.

6. Execute automated tests

6.1. The View *Auto Navigation Center*

The view *Auto Navigation Center* allows running and controlling automated web navigations. Therefore all actions performed by a web navigation will be logged and lists all results of those actions. Use this view to make sufficient tests of your specified Auto Navigation and to trap any errors to ensure valid web navigations executing automatically without any further control or manually tasks. The browser performes all specified actions and allows watching all browser actions directly.



Das Versenden des Formluars wurde deaktiviert
/cgi-bin/mailmanager.pl

12.08.2004 00:29:05

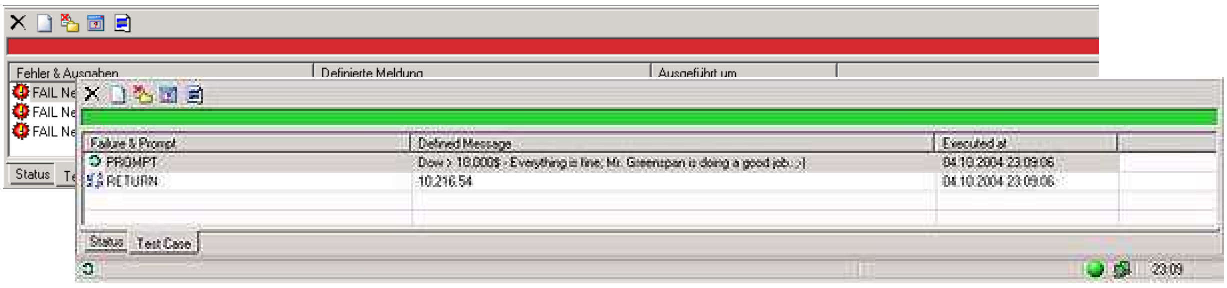
Aktion	Re-Identifier	Übergeordneter Frame	Ausgeführt um
Eingabe des definierten Wertes: aaaaaaaaaaaaaaaaa	50		12.08.2004 00:29:05
Eingabe des definierten Wertes: kakakak@ff.de	50		12.08.2004 00:29:05
⚠ Das Versenden des Formluars wurde deaktiviert	/cgi-bin/mailmanager.pl		12.08.2004 00:29:05
Aktivierte Navigation	submit		12.08.2004 00:29:05
Task beendet um 12.08.2004 00:29:05	Testen eines Bestellformulars	Iteration: 0	12.08.2004 00:29:05

Picture: Auto Navigation Center

6.2. Trace Test Cases

Additional to traced standard output messages, the register Test Case allows to display the test state more comfortabely. Therefore red or green progress bar displays the current state of the tests. Red means, that a defined test definition (Inline Declare 'Fail' command) has failed so the test case also fails. Green means, no failure definitions has executed and the test has succeed. To make a Test Case fail interpretation of script has to be run through only one fail definition. After test failure all other test steps - independing from the state - will be diplayed as failure in red.

A Test Case always is a automated Navigation that consists of further workflow actions. So the Mission Procect can be executed as Test Case. Then all activated Navigation Tasks are actions that can fail. If one action fails, the test has failed. If one single Navigation Task is executed as Test Case all its Navigation Steps are actions that can fail. The Test Case will fail, if one single Step fails.



Fehler & Ausgaben	Definierte Meldung	Ausgeführt um
FAIL Ne		
FAIL Ne		
FAIL Ne		

Failure & Prompt	Defined Message	Executed at
PROMPT	Dow > 10.000$ - Everything is fine; Mr. Greenspan is doing a good job. ;-)	04.10.2004 23:09:06
RETURN	10.216,54	04.10.2004 23:09:06

Picture: Tracing Test Cases results on Output Console

6.3. Remote Messages

JStudio SiteWalker allows sending state messages to remote computer via eMail and remote Pop Up to watch & control JStudio SiteWalker processes and Auto Navigations from another computer than JStudio SiteWalker is running. Specified messages will be sent on manually or time scheduled started Mission web navigations.

You can specify, whether a remote message should be sent and also the recipients of the message

by eMail

Activate this option, if State Messages should be send via eMail during Auto Navigation runtime. Just specify the receipient mail address to do so.
Note: To send eMails you also have to specify a valid SMTP server & port in Options tab '*Connection*'

By Remote Popup

Activate this option to use Microsoft Network Messages to pop up remote messages on specified computer. The receipient computer must be logged in the network. If problems occur please ask your system administrator because net messaging (netsend.exe) could be disabled by the network configuration settings. Just enter receipients user name or computer name (also IP) in the text field.

Specify when messages should be sent (Events).

- *On Auto Navigation end*
  Sends State message at the end of an Auto Navigation.

- *On Auto Navigation start*
  Sends State message before starting an Auto Navigation.

- *On any problems/ errors*
  Sends State message when errors occures during Auto Navigation. Note: Depending of the Mission Project size and the number of errors this could be a high number of messages.

- *On State change Online ./. Offline*
  Sends State message on changing offline/ online state. This option could be fine to watch online periods when using a modem connection to the internet.

6.4. Time Scheduler

Navigation Missions – the Testsuite - can be executed completely automatically using scheduler. To do this, the starting time must be specified and the timer must be activated. The activated timer checks the current system time and starts the Auto Navigation if the time as specified for start is reached.

You can specify, whether JStudio SiteWalker should run in background when AutoNavigation scheduler is activated.

6.5. Launching JStudio SiteWalker from command line

JStudio SiteWalker can also be started from any command line tools. This enables using JStudio SiteWalker launched by a batch program for automated processes for testing web applications or automated built processes (e.g. Ant). So you can run all Auto Navigations (as Test Cases) created with JStudio SiteWalker to make website testing & validation part of a complete build process.

Launching JStudio SiteWalker from command line is very simple - only 2 parameters can be defined:

Syntax: **sitewalker.exe -*Filename* [-*close*] [-*save*] [-*Profile=profilename*]**

File name: Absolute Path/ File name of the Mission Project to be executed automatically

Close: Optional. Define, whether JStudio SiteWalker has to be closed after runnig the Mission Project.

Save: Optional. Define, whether JStudio SiteWalker has to save the Mission Project before closing. Use this to save Excel references changed by iteration, if specified.

Profile: Optional. Define, whether and which parameter profile has to be used to resolve specified parameter during navigation runtime.

Further settings, needed to configure the project, e.g. what tasks are enabled /disabled, definition of test reports etc. must be configured in the Mission Project settings before.

Note: deactivate modal dialog Startup Manager before starting from command line to prevent displaying and stopping the launch. You also should ensure that all modal dialogs may raise during workflow process should be automated by JStudio SiteWalker Step configuration.

When entering the command JStudio SiteWalker launches and is shown on the screen to run the defined Mission Project.

JStudio SiteWalker returns a result before closing. So if you need to get the results of automated tests, specify option -close to quit the application and retrieve return values that can be evaluated by the launching script or batch file.

Return values:

0 - The specified Mission Project has been executed without any errors There are no navigation errors; no fail definition has occured.

1 - There are navigation errors during running the project. User defined fail definitions has occured and make the test case failed.

3 - There are navigation errors during running the project and User defined fail definitions has occured and make the test case failed

>3 An application error occured, the mission project could not be finished successfully

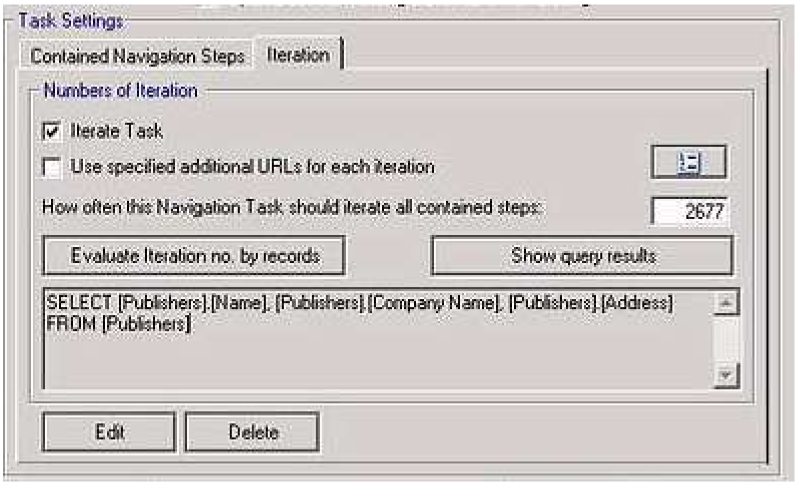
For more detailed information about the results that can be evaluated automatically a XML output for test reports is also available.

7. External data sources & repeating processes

7.1. General

Task Iteration can be executed as AutoNavigation by using Excel Sheets as data source or alternatively using ODBC data source during runtime. If Task Iteration is defined by using Excel for each iteration cycle data values will be read from the next line of the specified Excel Sheet. All you have to do is to specify how often this should be done. By using data values from ODBC you have to specify a SQL Query that retrieves one ore more result records. This SQL Query has to be assigned to the Navigation Task. For all the Steps beeing destinated e.g. to use retrieved data from your data source to import them into HTML input controls on the specified page only specify using a resultset for the parent Task.

This functionality gives you the opportunity to automate complete data imports from existing datasources into websites or test websites by using test data. This kind of import through a web applications front end allows you to import data by using the full functionality already implemented in the front ends, especially for initial loads in new systems. It automates the workflow process like it would be done by a human user. The same imports by programmed loaders processing directly to the underlaying database often requires the implementation of the same logic validations or triggered processes that already exists in the web application and are used than doing the input of data manually by human user.



☑ Iterate Task

☐ Use specified additional URLs for each iteration

How often this Navigation Task should iterate all contained steps: 2677

```
SELECT [Publishers].[Name], [Publishers].[Company Name], [Publishers].[Address]
FROM [Publishers]
```

Picture: Task Iteration defintion using SQL based data source

7.2. Using data from Microsoft Excel

Using this opportunity allows to import data into websites input control executing identical processes with using variable data at different times. So Task Iteration enables executing the same navigation workflow with different input data or data to be extracted.
The number of iterations must be specified freely. For entering data into website forms it is assumed that for each iteration data are available in your Excel data sheet that you may have specified for number of iterations.

For each input control you want to input data from a Excel file an Excel cell reference must be defined where to start retrieving values. Therefore just click into the cell within a collumn containing the first value for an iteration. For each iteration cycle the next line value will be retrieved automatically. Note: To do this, for each input element a collumn of data in Excel must exist.

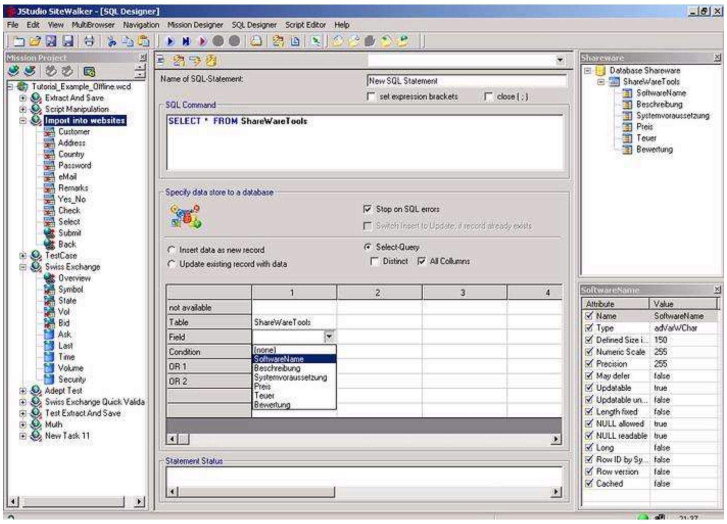
Activate the checkbox '*Increment Line*' when specifying the cell reference on dialog of Excel reference definition. This ensures that after processing every Excel access the next cell reference will be calculated (increment 1) to ensure that the next value from the next line will be used on next iteration cycle. Note: If this checkbox is not activated the steps refers always to the same cell (on saving: the content of the cell will be overwritten; on import: for each cycle always the same value will be imported to a control)

Activate the Task Iteration (checkbox in dialog *Configuration Navigation Task*, tab Iteration') and enter the number of iteration cycles. This should be the number of lines of your Excel file.

7.3. Using data from an ODBC database

Its the same procedure handling iterations by Excel, but the data must retrieved from databases. This needs to define a Select statement to retrieve a resultset, those data are used to enter into website forms. So - in opposite to excel approach - the number of iterations is given by the number of retrieved records within the resultset.
Approach to import data from ODBC data source (databases):

Create a SQL Select query with *SQLDesigner* which retrieves the data from your datasources (also validate the results by executing the statement in SQLDesigner. Assign this valid statement to the Navigation Task (Select statement can only be assigned to Tasks) representing your automated workflow process.



Name of SQL-Statement: New SQL Statement

□ set expression brackets □ close (;)

SQL Command

```
SELECT * FROM ShareWareTools
```

Specify data store to a database

☑ Stop on SQL errors
□ Switch Insert to Update, if record already exists

○ Insert data as new record
○ Update existing record with data
● Select-Query
□ Distinct ☑ All Columns

	1	2	3	4
not available				
Table	ShareWareTools			
Field	(none) SoftwareName Beschreibung Systemvoraussetzung Preis Teuer Bewertung			
Condition				
OR 1				
OR 2				

SoftwareName

Attribute	Value
☑ Name	SoftwareName
☑ Type	adVarWChar
☑ Defined Size i...	150
☑ Numeric Scale	255
☑ Precision	255
☑ May defer	false
☑ Updatable	true
☑ Updatable un...	false
☑ Length fixed	false
☑ NULL allowed	true
☑ NULL readable	true
☑ Long	false
☑ Row ID by Sy...	false
☑ Row version	false
☑ Cached	false

Picture: SQL statement definition using view *SQL-Designer*

Activate the Task Iteration (checkbox in dialog *Configuration Navigation Task*, tab Iteration') and enter the number of iteration cycles). The number of iteration depends on the number of records the resultset includes when the statement was executed during runtime) You can evaluate this number of records by pressing the button '*Evaluate iteration no by records*'.

For each Navigation Step assigned to input data values from the retrieved recordset please specify import from SQL database in the configuration dialog of the Step, tab '*Input values*'. Therefore activate the checkbox '*interative external database input value*' and select one of the retrieved recordset collumns from the select list below.

8. Import Navigation Tasks

To work in projects with many testers on different computers it is necessary to merge separately created automation results into one project file – the central Testsuite. By using Navigation Task import any Tasks from one mission project file can be imported into the project that is currently loaded in JStudio SiteWalker.
To import Navigation Tasks into other projects, JStudio SiteWalker provides using relative paths for your external source file references, like Excel or scripts which might used by imported test cases. Those resources must be available when executing test cases on other computers.

9. Additional Notes

The introduction of automated testing into the business environment involves far more than buying and installing an automated testing tool. In fact, effective automation is predicated on the idea that a manual testing process already exists since there is not a technology in existence today that performs automatic testing. So it is recommended that testing organizations begin their testing projects with a structured approach.
Within the testing environment, a quality assurance process is defined as a set of related steps designed to ensure or verify that a software application meets the requirements of the business user.

Before attempting to automate tests with JStudio SiteWalker, a solid grasp of basic testing processes is needed, as is an understanding of what automated testing can accomplish, and an idea of which tests are good candidates for automation. In fact, not all tests should, or can be, automated. When considering which tests to automate, focus should be placed on those manual test activities which take the longest time to set up including those manual tests that require the highest number of repetitive tasks and which are run the most frequently.

Test should be specified and described clear transparency to enable other testers understanding and change them on application changes and to make all information available in reports known by the tester who created the test scenario. Especially information of the needed environment, permissions, process conditions, needed data, the testers name and date etc. should be entered into the description fields of Tasks and Steps. If Steps or Tasks are configured to do some special process, e.g. many validation and Goto statements to branch processes, then please describe them. The time will come when this information is required to preserve quality standards of created tests or otherwise tests quality will get lost piece by piece.

Not the testing tool ensures the quality of your development results. It's a structured quality process and the tester, the developer, the business specialist who perform it. And JStudio SiteWalker just enables them to do the job.