

# Curriculum:

## Ranorex Certificate

v1.0.1 - 2014/08/14

### Introduction

The purpose of this document is to provide detailed information about the Ranorex Certificate examination and to aid in your preparation. In the following sections, a brief outline of the required knowledge to pass the

exam is given. Unless mentioned explicitly otherwise, background information can be found in the relevant chapter of the Ranorex User Guide.

### Sources

The main source for your preparation will likely be the [Ranorex User Guide](#). Please pay special attention to the [introductory chapter](#), lessons 1 through 9 and the section on the [RanoreXPath](#). Additionally, the screencasts provided on the Ranorex website might provide useful information on [specific topics](#).

Furthermore, helpful information can be gained from various [blog posts](#), the [FAQ section](#) on the Ranorex website and of course from practical work with Ranorex. We recommend that you have at least three months of experience working with Ranorex before attempting the exam.

## What You Should Know

### General Overview

- » The basic concepts of quality assurance, including understanding of important terms like “automated testing” and “functional testing”, “performance testing”, etc.
- » The scope of functionality available within Ranorex (record & replay, actions editor, development environment, etc.)
- » The prerequisites for running Ranorex Studio, e.g. on which operating systems can it be run

### Ranorex Studio – The Layout

- » Which views are available in Ranorex Studio and what they are primarily used for
- » The structure of a Ranorex Solution, the different types of projects available and what they are primarily used for
- » All elements of a Ranorex Test Suite and what they are primarily used for

### Getting Started – Lesson 1

- » On which platforms recordings can be made
- » Which types of actions are typically recorded
- » How recorded actions can be modified and what modifications are possible

### Ranorex Modules – Lesson 2

- » What Ranorex modules are and what they are used for
- » The Ranorex approach to modularity, reusability and refactoring
- » How modules can be assigned to test cases

### Data Driven Testing – Lesson 3

- » What module variables are and what they are primarily used for
- » What parameters are and which types of parameters are available in Ranorex Studio
- » Reasons for using variables and parameters, ways to bind them and the scope of variables and parameters
- » How values can be shared in a test case or in an entire test suite
- » What data sources are and which different types of data sources are supported
- » Reasons for having meaningful default values for module variables and how variables can be modified
- » What repository variables are and how they can be created, modified and bound
- » Reasons for using variables in RanoreXPath expressions, and the challenges that can be overcome by using them
- » How data binding can be carried out (direction of data flow) and how multiple variables can be bound to an external data column
- » How data binding can lead to looping test cases
- » How external test data can be modified once the data columns are bound
- » In which way does running tests or parts of a test bind data (e.g. running from a test suite vs. playing back a module)

### Ranorex Test Suite – Lesson 4

- » What Ranorex Test Suite projects and library projects are and what they are primarily used for
- » The structure of a Ranorex Solution and how it can be used to provide multiple projects of different types

- » What Ranorex “Run Configurations” are and how they can be used
- » What setup regions and teardown regions are for, and how they can be used
- » Which “Error Behavior” settings are provided and what do they mean
- » How setup/teardown regions behave in combination with various “Error Behavior” settings
- » In which ways can Ranorex tests be run outside Ranorex Studio, and why this could be useful
- » What are the most important parameters for running Ranorex tests from the command line
- » What report levels are, what reasons are there to use them, how are they hierarchically structured, where these setting can be made
- » Which different ways are provided for replaying in Ranorex Studio, and in which situations each way makes sense
- » What module groups are useful for, how data binding can be carried out and how groups can be assigned to test cases

### Ranorex Recorder – Lesson 5

- » How Ranorex Recorder can be used and which main functions are available
- » Which timing behavior (real time vs. standardized time) is used for recording and replaying
- » How recorded actions can be modified
- » Why recorded key sequences might be split up automatically and how they can be merged
- » What a validation action is and how the validation dialog can be called up during a recording

- » How someone can switch between object based and image based recording steps during a recording
- » What the main benefits are of linking actions in the Recorder with assigned repository items
- » Why turbo mode is available and what this mode does
- » Definition of a repository item's search timeout
- » How module variables can be created in the Ranorex Recorder
- » Reasons for having meaningful default values for module variables and how variables can be modified
- » Which smart actions are available within the Ranorex Recorder and how they can be used
- » What the smart actions "Get Value", "Set Value", "Close Application" and "Invoke Action" can be used for

### UI Mapping – Ranorex Repository

- » What the object repository is used for and how it can be used
- » Meaningful reasons for the strict separation of a repository and test actions
- » Important reasons for making modifications to repository items Different ways of modifying repository items
- » Which different elements are in a Ranorex Repository, and what they are primarily used for
- » What repository cleanup can be used for, and how it can be used
- » Different ways for adding items to the repository
- » The difference between "Search Timeout" and "Effective Timeout"

### Code Modules – Lesson 7

- » What reasons there might be for using code modules

### Reporting – Lesson 8

- » Which messages are typically logged to the report
- » How log messages can be filtered in a report, and filtering criteria that can be used
- » Why an exception is thrown during a test run if UI elements are not found, and strategies to fix this issue
- » Which test case states are presented in the report and what they mean
- » How report actions can be used in the Ranorex Recorder and which content can be logged

### Ranorex Spy – Lesson 9

- » Scenarios for using the Ranorex Spy and how it can be used
- » Which different ways of tracking are available and in which situations one might be better than another
- » What Ranorex Snapshots are and how they are different from screenshots
- » How and why the Path Editor can be used

### RanoreXPath

- » The syntax of RanoreXPath expressions
- » How RanoreXPath expressions are structured
- » Special RanoreXPath syntax elements (`//`, `?`, `*`)