

Syllabus for the Ranorex® Certified Professional Certificate

Version CP-2.0 – 2017/09/20 – Valid for Ranorex 7.0, 7.1, 7.2

Introduction

We're glad you've decided to become a Ranorex Certified Professional! This syllabus will give you an overview of the topics you will need to cover in your preparation to achieve this goal. It also specifies where you can find the relevant study materials. In addition to studying the theoretical resources on Ranorex, we recommend that you have at least three months of practical experience with the program before taking the exam.

Resources

The Ranorex User Guide is your go-to learning resource for all things Ranorex. Please pay special attention to the introductory chapter, lessons 1 through 9, and the section on RanoreXPath. Our screencasts on the Ranorex website also provide useful information on specific topics. Our various blog posts and the FAQ section on the Ranorex website are helpful resources as well. Finally, hands-on work experience with Ranorex is also an excellent way of preparing for the exam.

Syllabus

General overview

- The basic concepts of quality assurance, understanding important terms such as “automated testing”, “functional testing”, “performance testing”, etc.
- The functions available in Ranorex (record/replay, actions editor, development environment, etc.)
- The system requirements for running Ranorex Studio, e.g. the operating systems it can be run on

Ranorex Studio – The layout

- The views available in Ranorex Studio and their main uses
- The structure of a Ranorex solution, the different types of projects available, and their main uses
- All elements of a Ranorex test suite and their main uses

Lesson 1: Getting started

- The platforms recordings can be made on
- The types of actions typically recorded
- Modifying recorded actions, the types of modifications available

Lesson 2: Ranorex modules

- Ranorex modules and their main uses
- The Ranorex approach to modularity, reusability, and refactoring
- Assigning modules to test cases

Lesson 3: Data-driven testing

- Module variables and their main uses
- Parameters and the types of parameters available in Ranorex Studio
- Reasons for using variables and parameters, ways to bind them, and the scope of variables and parameters available
- Sharing values in a test case or in an entire test suite
- Data sources and the different types of data sources supported
- Reasons for having sensible default values for module variables and modifying variables
- Repository variables and creating, modifying, and binding them
- Reasons for using variables in RanoreXPath expressions and the challenges that can be overcome by using them
- Carrying out data binding (direction of data flow) and binding multiple variables to an external data column
- Data binding as a cause of looping test cases
- Modifying external test data once the data columns are bound
- Differences in data binding when running tests or parts of a test (e.g. running from a test suite vs. playing back a module)

Lesson 4: Ranorex test suite

- Ranorex test suite projects and library projects and their main uses
- The structure of a Ranorex test suite and using it to implement multiple projects of different types
- Understanding and using smart folders
- Ranorex run configurations and their main uses
- The purpose and use of setup regions and teardown regions
- Available error behavior settings and their meanings
- The behavior of setup/teardown regions when they are combined with various error behavior settings
- Ways of running Ranorex tests outside of Ranorex Studio and reasons for doing so
- The most important parameters for running Ranorex tests from the command line

Lesson 4: Ranorex test suite (continued)

- Report levels, reasons for using them, their hierarchical structure, and where their settings are located
- The different ways of replaying in Ranorex Studio and the situations they are suitable for
- Using module groups, carrying out data binding, and assigning groups to test cases

Lesson 5: Ranorex Recorder

- Using Ranorex Recorder and its main functions
- The relevance of the timing behavior (real time vs. standardized time) used for recording and replaying
- Modifying recorded actions
- Causes for automatic splitting of recorded key sequences and ways to remerge them
- Validation actions and calling up the validation dialog during a recording
- Switching between object-based and image-based recording steps during a recording
- The main benefits of linking actions in Ranorex Recorder to assigned repository items
- Turbo mode and its purpose
- Defining a repository item's search timeout
- Creating module variables in Ranorex Recorder
- Reasons for assigning sensible default values to module variables, ways of modifying variables
- The smart actions available in Ranorex Recorder and using them
- The smart actions "Get Value", "Set Value", "Close Application", and "Invoke Action" and their uses

Lesson 6: UI mapping with Ranorex repository

- Purpose and use of the object repository
- Reasons for the strict separation of a repository and test actions
- Reasons for making modifications to repository items, different ways of modifying repository items
- The different elements in a Ranorex repository and their main uses
- The purpose and use of repository cleanup
- The different ways of adding items to the repository
- The difference between "Search Timeout" and "Effective Timeout"

Lesson 7: Code modules

- Reasons for using code modules

Lesson 8: Reporting

- The messages typically logged to the report
- Filtering log messages in a report and the filtering criteria that can be used
- The causes for an exception being thrown during a test run if UI elements are not found and strategies to fix this issue
- The test container states shown in the report and their meanings
- Using report actions in Ranorex Recorder and the content that can be logged

Lesson 9: Ranorex Spy

- Using Ranorex Spy and usage scenarios
- The different ways of tracking available and the situations they are suitable for
- Ranorex snapshots and how they are different from screenshots
- Purpose and use of the the Path Editor

RanoreXPath

- The syntax of RanoreXPath expressions
- The structure of RanoreXPath expressions
- Special RanoreXPath syntax elements (//, ?, *)