

Test automation has picked up a poor reputation in the testing industry. It has been proven that capture/replay is not a good foundation for building robust, maintainable test systems. It has also been found that script-based approaches have their own limitations. Often the best testers on a project do not have the technical background necessary to apply automation. Testers should be writing tests, not automation code.

### **The next generation of test tools provide a simple solution for both automated and manual testing**

The general rule of thumb is that if your application can be tested with any of the leading UI test tools, like Visual Studio 2010 Coded UI, HP QuickTest Professional or WinRunner, Micro Focus SilkTest, TestPartner or QARun etc. then Axe will work. If the underlying tool works with your application then so will Axe. Some customisation and configuration may be required by an automation technician to hook-up any custom functions you may have written e.g. for synchronisation purposes.

### **Can Axe be integrated with test management products like Team Foundation Server or HP Quality Center?**

- Axe integrates with all leading test management tools, including TFS and HP Quality Centre (QC).
- Axe tests are defined in TFS or QC just like any other existing manual or automated test. This means that the full power of QC can be used for linking tests to Requirements and managing the full test creation process e.g. assigning tests to individuals, setting priorities and due dates etc.
- QC just contains the 'header' information for the test such as test id, priority, owner etc. The actual test itself is defined in the Axe spreadsheets. So there is no duplication of data and hence no need for synchronisation.
- Automated test runs can then be defined in the Test Lab and used by Axe to control which tests are built and executed.
- After the tests have run Axe provides the capability to import the results back into QC thus creating a fully joined-up and managed test process.
- The full power of QC's reporting capability can then be utilised to produce reports and analysis of the test runs.
- The integration with Axe is achieved by the open QC API. As such it is easy to further customise the Axe integration with QC to meet any specific requirements.
- Detailed information about Axe integration with QC and other test management tools can be found in the Axe documentation, which is supplied with software implementations.

### **Which Operating Systems (OS) are supported by Axe?**

Any Windows platform that supports .NET 4.0; you are more likely to be restricted by the requirements of your GUI tool than Axe.

### **GUI tools commonly have trouble with multiple IE windows or complex on screen objects, and take some configuration or technical solutions to get around it. Can Axe handle these?**

Yes. If the GUI tool can do it, so can Axe.

### **Commonly we have to input synchronisation code for scenarios like: "Wait until progress bar disappears" or "Wait for pop-up window to appear". It does not seem clear to us how or if this can be done with Axe.**

This is very easily done, especially if you have already written your custom sync functions. There are several ways this can be easily integrated with Axe, e.g. overridden actions, new actions, and custom classes. This is all covered in the Axe Technician Manual and training.

### **Can Axe be used to check and verify special hardware items like ATM Machines, Card handlers etc?**

Only if there is an underlying test tool to do this. If there is an API then Axe Harness may be able to test it via that. See our website for further details.

### **What training and support is required?**

We recommend at least Axe Tester and Technician training session and can provide implementation consultancy if required.

### **How are features such as Parameterisation, Logical flow control and Repetition handled in Axe?**

There are many different ways to parameterise and externalise data with Axe, e.g. databases, XML files and external Excel spreadsheets. We do not encourage use of logic control in tests as this is more of a programming construct, although it is possible to incorporate through the customisation features. With Axe the focus is on simple business process flows, not coding. There are various techniques for repetitive tests - the architecture is very open and flexible.

### **Is the code generated by Axe optimum code or does it requires any fine tuning for better performance results?**

Yes, code generated by Axe are optimised and often run quicker than scripts coded by hand.

### **Can the Axe functionality be batch scheduled to run from a test management tool like HP Quality Center (QC) or TFS Lab Manager, and can results be reported back for test management?**

All Axe functionality can be driven by the command-line and hence can easily be called from scheduling tools. Axe is fully integrated with QC and TFS and allows test runs to be defined from the Test Lab, and Axe results imported.

### **Do we need to upgrade Axe software when new upgrades are made to the supported GUI Tools?**

Sometimes; it depends on how much the GUI Tool changes. This doesn't happen very often. Odin gets the updates out very quickly because they are partners with Microsoft, HP, Micro Focus, IBM and other vendors and therefore have prompt access to all latest product versions. All updates are available as part of the regular standard Axe support package.

### **Object identification and object-mapping; design-time versus run-time; how is the interaction with Axe achieved and what sort of effort is required to keep it maintainable?**

Object maps can be created at design time by putting the object and class information in a simple Excel spreadsheet. There is a trade-off between the ease of use and versatility of Excel versus a GUI tools own object map editor.

### **How is the object identification affected by object irregularities at run-time (changing properties or an extra instance of IE, for example)?**

Generally this is not a problem. If the GUI tool deals with it (and most do) then it is not a problem with Axe either.

### **What is the full command set and how expandable is it?**

There are 5 core testing actions, SET, GET, VAL, LOAD and SAVE, and a number of extensions to these defined in the Testers Guide (available on request). It is very easy for an Axe Technician to override and extend the base action set.

### **How are the expected results defined?**

Primarily this is done in the Excel spreadsheets in the 'data' column. More complex results, e.g. multiple lines, can be referenced in separate files.

### **What are the error recovery capabilities?**

Axe has very powerful built-in error recovery because Axe generates code that has extensive error checking.

### **What CLI parameters are available for running Axe from the command line for integration to other tools (e.g. Schedulers, CM tools etc.)?**

All the main options on the Axe UI can be executed via the command line. Axe supports a fully integrated process, this is how our client Microsoft uses it for example.

### **What back-end testing support and interrogation does Axe provide?**

Axe Harness provides support for database queries, web services and APIs. It is also very easy to integrate interaction with databases and other SOA components with UI tests, and this is a common requirement.

### **Can the tool documentation cover screen capture/compare etc (needed for audit purposes)?**

Yes, this can be done.



tel: +44(0) 118 903 6101 email: info@odintech.com  
fax: +44(0) 118 903 6100 web: www.odintech.com

Atlantic House - Imperial Way  
Reading - RG2 0TD - UK