A Boolean Society Publications Series
The George Boole Foundation Ltd
72 New Bond Street, London W1S 1RR

GBF-DAI-PES-003

ECMAScript World

An introduction to the Power ECMAScripting series.

This document sets out the objectives, some perspectives on this approach, its significance and where we intend to end up.

Objectives

The objective of this series is to provide programmers and people interested in learning to produce server-end scripts in ECMAScript with a step-by-step explanation of how achieve this, what software is required and how this is setup with respect to hardware located on the client and server sides and, in some cases, at intermediate locations.

Some perspectives on this approach

It is worth emphasising that we are dealing here with state of the art proven scripting techniques which have been in operation and proven for over 10 years in demanding commercial industrial and service functions as well as supporting governance operations. We are not dealing here with "concepts" or "ideas" that might work; we are dealing with proven state of the art.

In addition what will be discussed is not a new front designed to compete with existing systems and operating systems but rather something that is entirely complementary to existing software products. So that in terms of client platforms it is cross-compatible and at the server end use is made of conventional Microsoft™ Server Operating Systems. The approach to be described is one where other implementations such as Adobe Flash, the Swish line of products, multi-media, various MS programming environments, VoIP and barcoding techniques, run in a seamless fashion. Virtual Client Technology client platforms so far have been designed to run in any MS Windows OS and to control the OS and any Windows compatible software running on the client computer.

The basic configuration of what will be described is an authoring system which outputs script in ready-to-run-server-side format. The script is an advanced ECMAScript core implementation containing extensions in the form of convenient functions, many of which support decision analysis, which do not exist in core ECMAScript. The script can accept any ECMAScript as part of the content except for some quite minor details that will be identified at the appropriate stage in this series. The script runs at the server-side under a dedicated server software running on a MS Windows OS server. All the user requires to access the server-based applications is a www standards-compliant browser; the client OS is not a constraint.

We will not refer here to the slow and more conventional client-side implementations of JavaScript and ECMAScript because there are around 100 excellent forums and blogs accessible on the www as well as technology news media who keep this aspect of ECMAScript well up to date. Also, during this introductory phase we will not go into the current continuing progress of the ECMAScript core standards being advanced under the auspices of the ECMA, although we do take this progress seriously. These matters are a matter for future discussions.

Telephone: +44 (0) 751 298 778
http://www.boolean.org.uk
info@boolean.org.uk

Significance

The significance of this system is that it represents the most powerful implementation of ECMAScript-based web-based operations because three components of the system configuration can be optimised for processing speed as well as complexity of data processing. The components are:

- server-side implementation
- extended functionality in the server-side environment
- client platforms

The combined optimization takes ECMAScript implementations up to speeds of native Windows implementations at the client end and, in some cases, can exceed the client side speed. It is pointless to elaborate this particular point in a vacuum. It is better to provide specific examples and bench test results which we will provide together with online demonstrations during the course of this series with online demonstrations.

In this context, this issue of "speed" needs to be assessed from the standpoint of systems engineering economics in that, as designs and implementations become more complex, we are looking for improvements in the power-to-monetary-outlay ratio where power is a measure of combined process complexity, precision and speed of execution and response times on the client-side.

The name of this series "Power ECMAScripting" should be recognised as being justified in this context.

Where we intend to end up

Those who follow this series should be capable of scripting and setting up server-side operational decision analysis systems and will know how to optimise the final configuration so as to achieve the lowest cost implementation without compromising power or security. We will be setting up a range of online decision analysis system demos and aspects of these systems will feature in this series or as notes released in association with each demo.

McNeill, H.W., 12th March, 2010

Upated: 13th March, 2010 [typos & clarity]

A4-format