Performance Tuning of jBASE Web Builder

INTRODUCTION

As jBASE Web Builder is written using jBASE technology, it is only usually necessary to tune web builder for extremely high server loads.

This document assumes web builder development knowledge and knowledge of the target operating system.

There are three main ways to improve the performance of a web builder server.

Firstly, tuning the configuration of the web builder application server and the operating system so that they are running at their peak performance.

Secondly, tuning the application so that the pages it delivers are reduced in size.

Thirdly, optimising the web builder application so that the number of subroutine calls are reduced.

It should be stressed that performance tuning and optimisation is not necessarily an exact science and experimentation will be required.

INFRASTRUCTURE TUNING

There are a number of changes which may be made within the environment to improve performance.

Environment variables are used in jBASE to define the locations of data files, subroutines and executables. These environment variables usually hold multiple operating system paths separated by commas.

At run time, jBASE searches through these paths, one by one, until it comes across the object it needs. By making changes to the order that these paths appear in can mean that jBASE finds the objects it needs more rapidly.

The JEDIFILEPATH environment variable stores the paths that jBASE searches for data files. It makes sense that the first path stored in the

environment variable is the optimum place to store files which are frequently accessed.

Similarly, the JBCOBJECTLIST environment variable is used to discover subroutines which have been compiled, so it makes sense that the first path entry should be where the most commonly used subroutines should be catalogued to.

Lastly, the PATH environment variable is used to track down executables, so a frequently used executable should be in a path which appears towards the start of the PATH.

There is a caveat, and that is that web builder itself uses many data files, subroutines and executables. Experimenting with these settings may well result in a degradation of performance as jWB files and programs might be relegated further down the list!

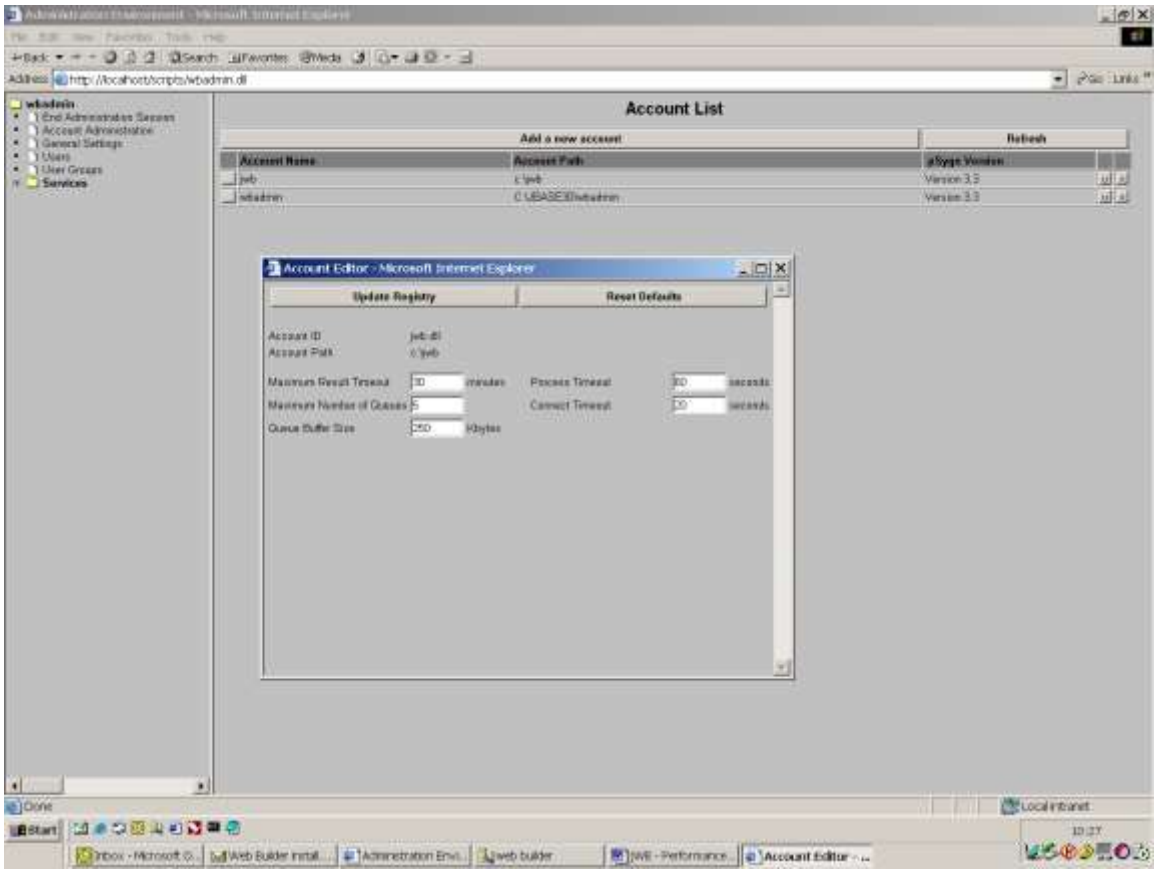
Another factor that will affect the performance is the sizing of the files in use by the application. It is a good idea to periodically check that both the data files used by the application, and the data files used by jWB are in tip-top shape by running the jBASE JRF utility. The data files used by jWB are stored in the account directory and all have a prefix of sys.

The user connects to the web builder application by way of some kind of connector which then connects to one of a number of queues.

There is an optimum number of these queues which will vary according to the load being placed on the server, and the characteristics of the server itself.

The number of queues belonging to an account is set when the account is created and may be modified by using the web builder administration account.

Fire up the administration account (wbadmin.dll), and select account administration. To set the number of queues, select the account and change the Maximum number of queues property.



REDUCING PAGE SIZES

In general, the smaller the pages that make up a web application, the better the overall performance of the application.

Avoid lots of pages. When requesting a page, the browser will need to make a separate request for each image on the page.

Avoid unnecessary frameset use. Each frame will result in a separate request to the server, and if the contents of each of the frames is a web builder page, this will increase the load on web builder accordingly.

Tables are fairly expensive for the browser to render, particularly nested tables.

Cascading stylesheets can be used so that the appearance of your application is defined only once rather than embedded style elements for each component of your web application.

OPTIMISING THE APPLICATION

Optimising the application itself tends to be an exercise in reducing the amount of code that web builder has to execute.

There is a tool in jBASE which allows profiling of BASIC code. This can give you valuable information about where your application is spending most of it's time.

DIRT tags are an incredibly useful feature within web builder which allow macro replacement. They can be overused however, as each DIRT tag encountered will result in a subroutine call in web builder.

For example, if a page uses several extract DIRT tags to pull data out of the same file, that file will be opened, read and closed again every time that DIRT tag is encountered. It would be much more efficient to read in all the values in a jBASE pre-page routine and load the values that way.

Another technique is to call a DIRT tag once, and store the return value in a hidden HTML object. Then whenever the page needs to refer to that value, javascript may be used to extract the value.

Every development class will also call a web builder subroutine, so if more than one component can be combined, the number of subroutine calls will be reduced.

For example, if a page contains two or more adjacent HTML objects, they could be combined into one HTML object.

Nested object references will also result in an expensive page generation process. It is far better to only have one level of object reference if possible. This will also improve the maintainability of the pages.

Document Created 11th January 2002
Author: Martin Bailey