

What about when it's down? An Application for the Enhancement of the SAS® Middle Tier User Experience

Christopher Blake, Royal Bank of Scotland

ABSTRACT

The SAS® Web Application Server goes down and the user is presented with an error message. The error messages in the SAS® 9.4 middle tier are the default ones that are shipped with the underlying VMware vFabric Web Server and are seen by many users as too technical and uninformative.

This paper describes an application called "Errors" that was developed at the Royal Bank of Scotland that has been implemented across its 9.4 estate to provide a much better user experience for when things go wrong. In addition, regardless of communications, users always try to access an application if it is available. This paper goes into detail about a feature of the Errors application that RBS uses to prevent this. This feature is used to control access to the web applications during scheduled outage windows and it provides capability for IP and location-based access as well as others. This paper also documents features and capabilities that RBS would like to introduce to the application.

INTRODUCTION

When something breaks, you will want to fix it, or at the very least you will want some help in understanding what has gone wrong and what you should do. As a platform administrator, you may also want to stop people from compounding an existing issue and prevent users from having an experience that leaves them frustrated and disengaged.

It appears that many of those who write software don't think about the layman when writing their error messages. They are often extremely technical and whilst specific, also unhelpful to the lay-user. Apache themselves describe the Apache web server default error messages as "stark, uninformative, and can be intimidating to site users".

At the Royal Bank of Scotland (RBS), we have looked to augment our SAS® 9.4 estate with an application to allow our platform and application support teams to create meaningful and informative error messages for their user communities. This application is imaginatively named "Errors".

THE PROBLEM

At RBS we have a large SAS estate with an equally diverse range of users and uses. Two of our largest platforms though have significant web use. One platform (~45,000 users) runs custom HTML and JavaScript applications using the SAS® Stored Process web application to harness the power of SAS. The other is a SAS® Visual Analytics environment which at the moment has ~1,600 users but is being rebuilt having been sized for 10,000 users.

The two platforms are very different. They have very different user communities and the applications that run on the two platforms are very different. What the platforms have in common though is the same unhelpful default technical error messages. We wanted to change this. We wanted to be able to show users specific messages for their environment that was, as much as possible, personal to them. For example why the environment was down and when it is likely to be back up. Or, why they aren't seeing the page that they expected too. They should also be able to see the contact details of the relevant support teams or business champions for further information. And none of this should compromise on the page that they are seeing being really pretty and intuitive.

Our primary development focus was on the platform with ~45,000 users. Not only is this the biggest SAS application in the bank but it is also considered a business critical platform. It is used by branch staff and is visible by customers on a daily basis.

OUR SOLUTION

As mentioned in the introduction, what we came up with to solve this problem was Errors. Errors is a PHP based web application that can be deployed to the SAS® Web Server to enhance the default functionality and the error handling. It makes use of Apache's `ErrorDocument` directive to surface custom error documents when specific HTTP error codes occur. Errors includes these error documents and allows platform administrators to customize them on-the-fly and without the need for down time. The application can be deployed in an environment configured for high availability to a web server cluster in a shared nothing or shared configuration architecture.

THE HOME PAGE

The starting point of the application was designed to give the platform administrators an overview of the current configuration. It shows information for each HTTP error code where it has checked that there is some text to display if the error is called and that web server configuration contains an appropriate `ErrorDocument` directive.

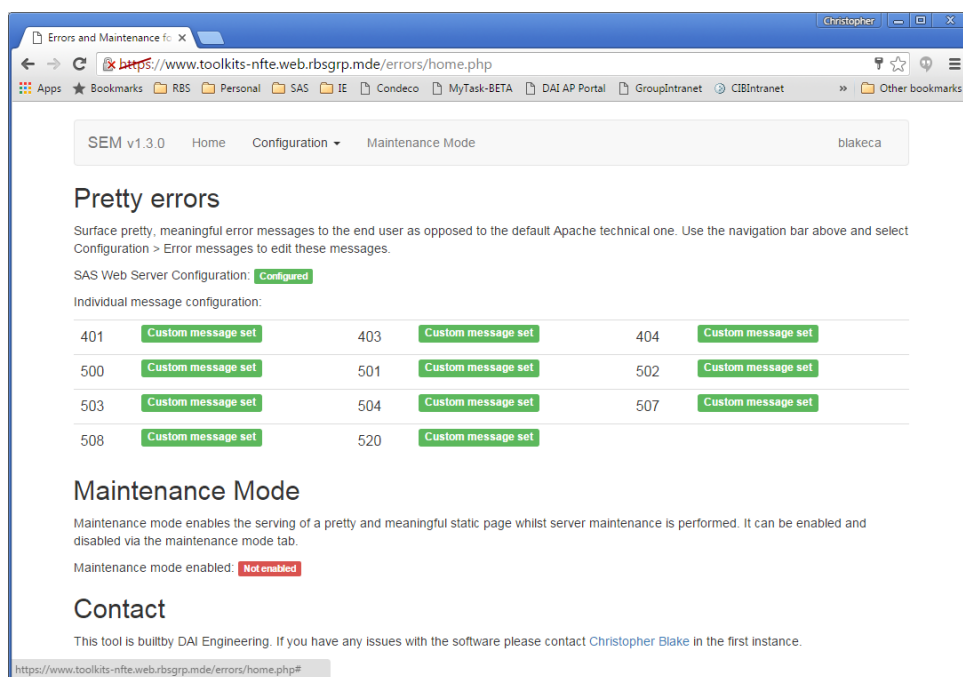


Figure 1. The home page of the Errors application

THE ERROR MESSAGES

This is the part of the application where a platform administrator is able to make on-the-fly changes to the text that is displayed on the screen when a particular error message is triggered. Errors supports HTML tags in the error message text which can be used to enhance the messages being displayed. The Errors application is a Bootstrap-based application which means that any HTML that works within the Bootstrap framework is supported within the error messages. In addition, after saving an error message, the user is able to view that what error would look like to the user without having to trigger the error itself.

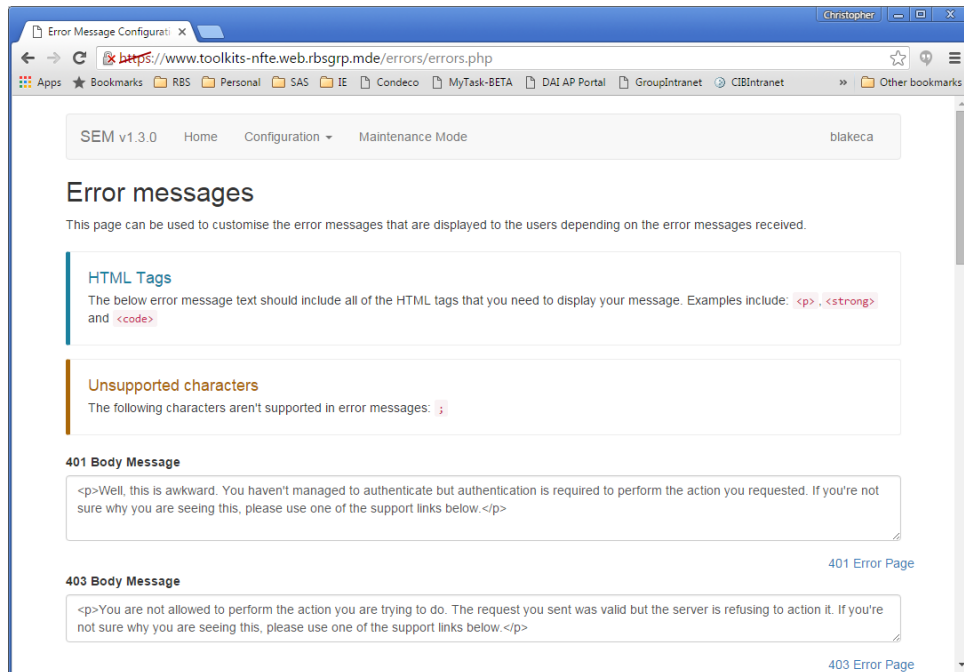


Figure 2. The error message configuration page of the Errors application

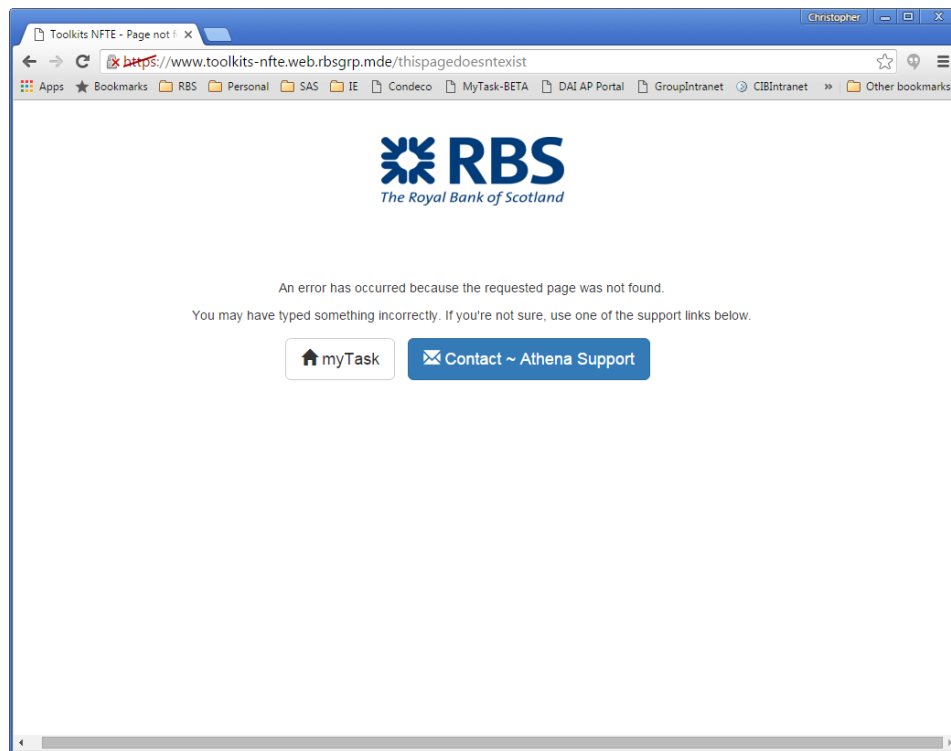


Figure 3. An example of the 404 error message page that is displayed when the requested resource doesn't exist

THE SUPPORT LINKS

On each of the pages where an error is displayed, such as figure 3 above, links are provided to allow a user to contact a support team or access some other type of help resources. As with the error messages, all of these links can take advantage of all of the available features of the Bootstrap framework; by default, three buttons are configured taking the users to different resources; a web application, and two email links formatted as buttons.

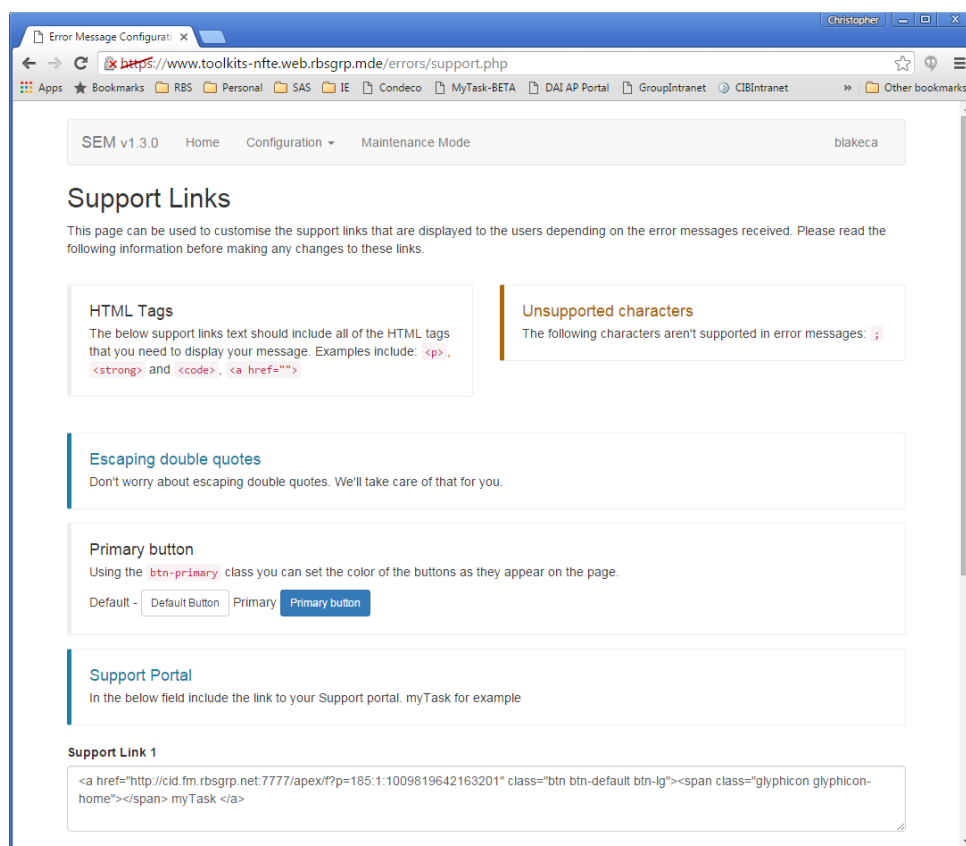


Figure 4. The support links configuration screen

GENERAL APPLICATION SETTINGS

This part of the application is used to configure global settings within the application. This includes things such as the Active Directory domain that you want this instance of the Errors application to be secured against (we don't want to just let anyone login and change stuff), the short and long name of the application or platform which is used in the title bar of the browser when a user gets an error (as seen in Figure 3), and the nodes of the web server cluster if the middle tier is clustered.

SEM v1.3.0 Home Configuration Maintenance Mode blakeca

General settings

This page can be used to customise the general settings .

AD Domain

M01EUROPA

MTR Number

This is only required for systems deployed in MTE

Application Name

Toolkits NFTE

Application Long Name

This is an optional value that can be left blank

Toolkits NFTE

Support Team Name

This value is not currently used by the application but is planned for a later release

~ Athena Support

Support Team Mailbox

This value is not currently used by the application but is planned for a later release

FM-019400@rbos.co.uk

Web Server Cluster Nodes

vsttkm01n vsttkm02n vsttkm03n

Application directory

This value is the top level directory. For example: toolkits.

Figure 5. The general settings where global variables are configured

MAINTENANCE MODE

The maintenance mode is a key feature of the application and allows RBS' platform administrators to surface pretty pages to a user while the system is down for a known reason. At RBS, this feature was developed with two key use cases in mind.

The first use case is for during regular system maintenance. This could be a weekly or monthly restart of some services or it could be something more involved such as the installation of a hotfix. The vast majority of the users of the platform where the Errors application is installed are web users (95%+) so they would see the notification. We have also found that for the significant majority of the tasks where we are performing some kind of maintenance on the system we are able to keep the web server running allowing us to surface these messages.

In addition to that, on the platform where we run a number of custom HTML applications with a stored process backend we rebuild the indexes on some of the database tables once a week, out-of-hours. This process can take some time and while the process is taking place we don't want the users to access the application. Rather than turning the web server off, which would block access to all other applications hosted on that platform, we are able to, for a particular URI, disable the traffic and show the users the maintenance page instead. While the user cannot do the work they wanted, they get a far better experience. They are told why they are seeing the message with an expectation of when the system will be available and perhaps some further instructions if they have any questions.

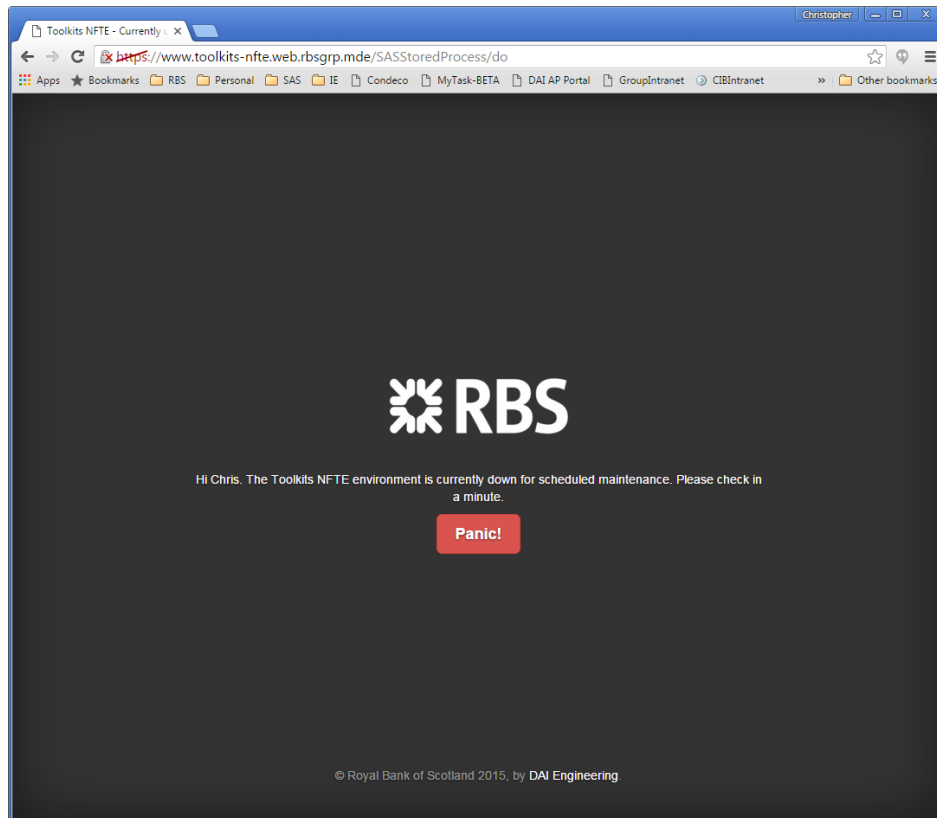


Figure 6. The screen a user is presented with when they try to hit a web application when the system is in maintenance mode

The Errors application allows the whole middle tier to be put into maintenance mode (enabled by pushing the “system panic” button) or on a URI by URI basis where the URI is a folder/application that exists on the SAS® webserver.

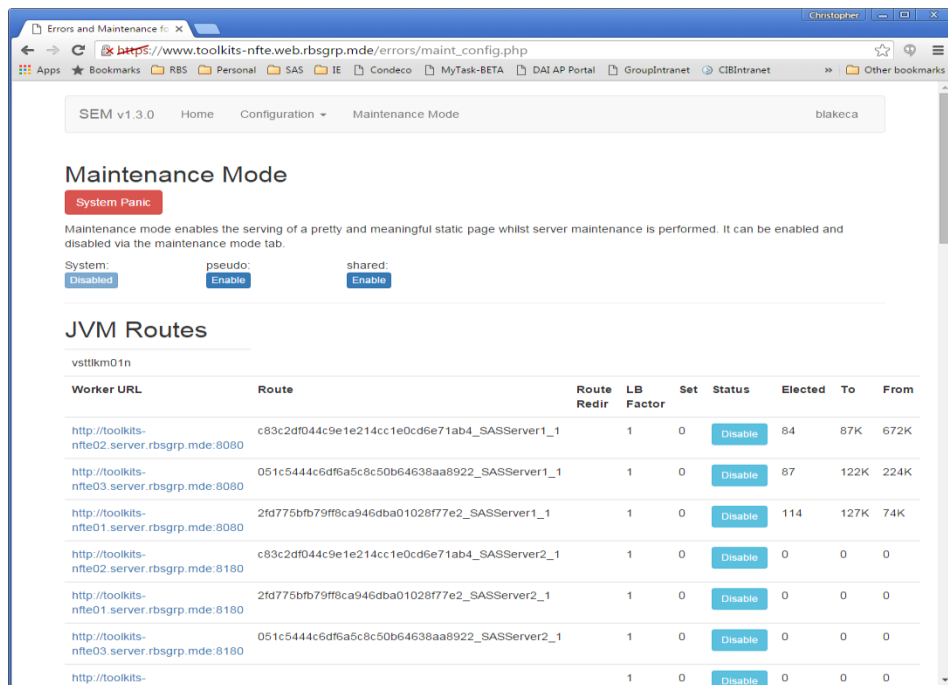


Figure 7. The maintenance mode screen when maintenance mode is not enabled and traffic flows as expected

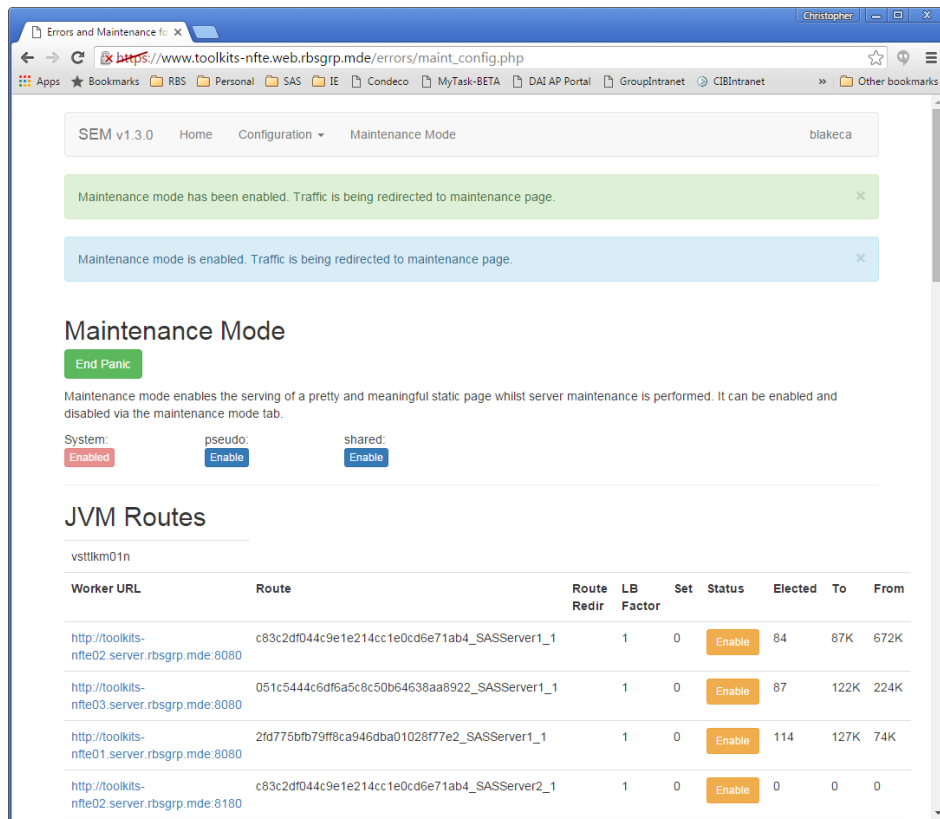


Figure 8. The maintenance mode screen when the whole system is in maintenance mode

In the same way as with the error messages, it is possible to customize the text and links on the maintenance holding page. As with the error message pages, it is possible to take advantage of the full Bootstrap framework's functionality in addition to having the ability to preview any changes that you have made without actually enabling the maintenance mode.

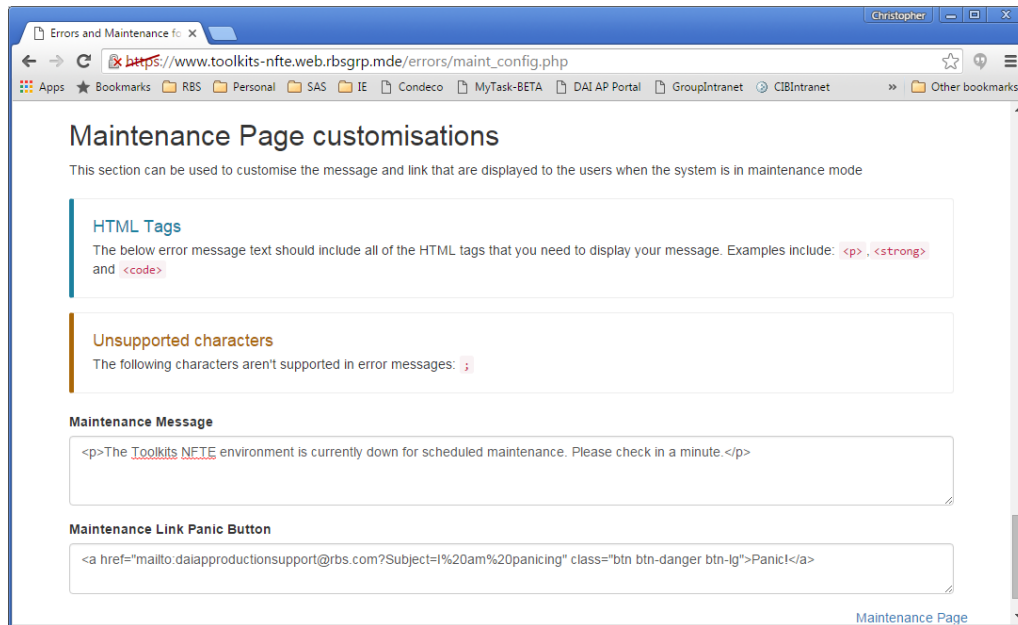


Figure 9. The ability to customize the maintenance page message

In addition to being able to surface pretty, on-the-fly customizable holding pages, the maintenance configuration section of the application also has two other key pieces of functionality.

The first, Alternative routes, allows a platform administrator to redirect a request for a particular page or URL to be redirected to a different one which may or may not be local to same environment (hosted on the same web server). At RBS we have used this functionality for application migrations. When an application is migrated from one platform to another there can easily be teething problems. The alternative routes functionality of the Errors application has allowed our platform administrators the ability to quickly redirect users seamlessly back to the original platform when these issues occur. This makes for a significantly improved user experience and fewer incidents being raised with the support teams.

The second, Special people, was developed to support business checkout of a modified or updated application. It is used to allow a subset of people access to an application based on their IP address. Inevitably when you have a large user community and you are relying on things like email to communicate downtime there will be individuals or groups of users who are not checking their emails and don't know to not use an application. The special people functionality allows you to have a custom HTML application in maintenance mode but for some people to be able to use the application. Once they have confirmed that all of the functionality is working as expected maintenance mode for that application can be disabled.

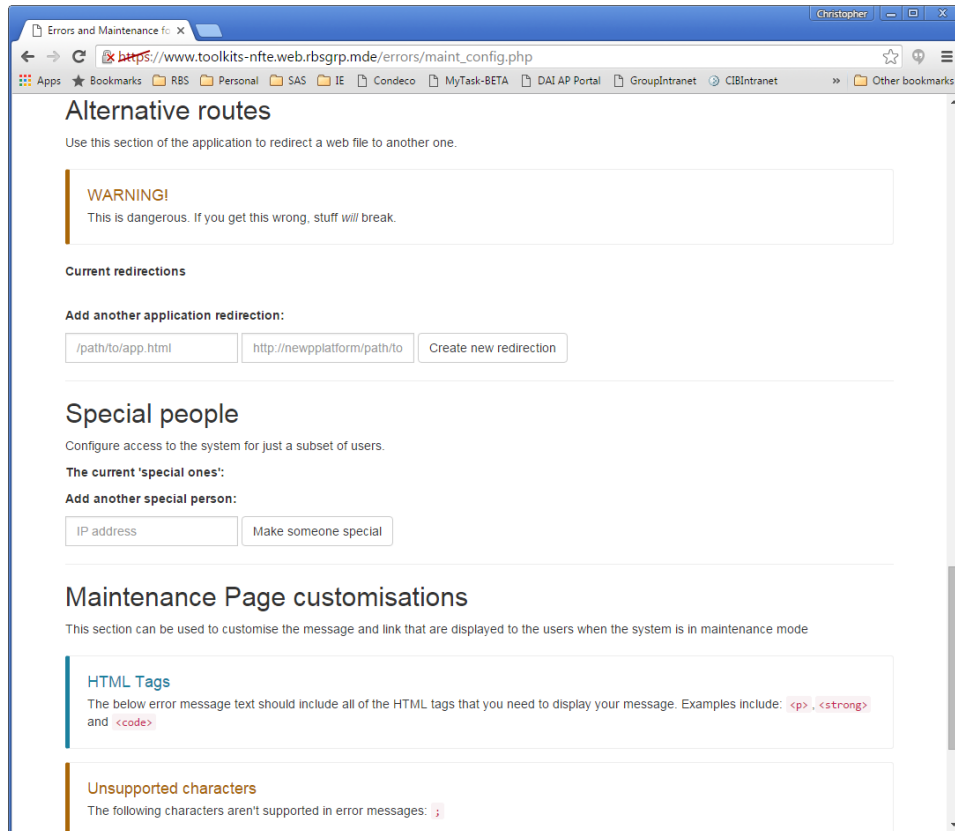


Figure 10. The Alternative routes and Special people configuration pages

FUTURE WORK

There are a number of areas of the Errors application that the RBS team is looking to develop over the coming months.

NON-IP BASED SPECIAL PEOPLE

At RBS we are moving away from physical desktops and more towards virtual machine with only a thin client machine on your desk. Some of these machines are "regular" VMs; however, for the vast majority of head office staff this is a terminal services connection to a Windows server. What this means for the Errors application is that we have found a significant number of users from a very small range of IP addresses.

As you can see from the graph below, this is the total range of IP address that have accessed our production SAS® Visual Analytics platform (12 IP addresses). These 12 addresses either have some very busy people on them as they are generating up to 60,000 web server events an hour or there is something else going on. What this has meant is that we are looking for a new way to identify users to be able identify them in order for them to be able to complete checkout activities. We are looking at location based or userid based cookies but we haven't come up with anything that works for us just yet.

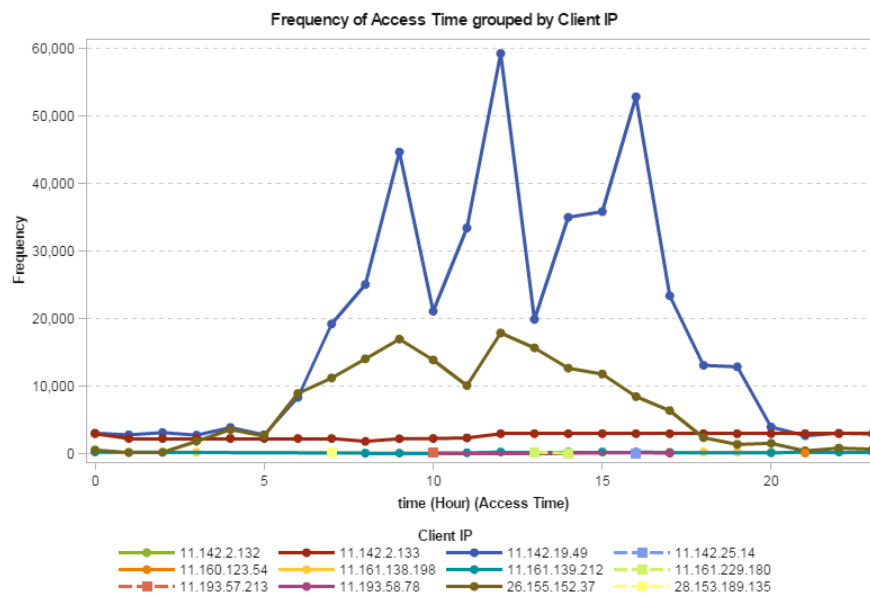


Figure 11. All the IP addresses that access a SAS VA platform over a 3 day period

CAPTURING ALL ERROR MESSAGES

The architecture of the Errors application at the moment relies on the underlying web server throwing an HTTP error code. If no error code is thrown by the web server then the errors application is not invoked. We have found this to be an issue for some errors that are thrown by the SAS web applications. Sometimes, there will be a 500 or other error in the application but the application itself handles this and sends back a HTTP error page. The problem with this error is that generally includes a Java stack trace which is not usually something that our user community wants to see. We haven't found a way to overcome this problem yet.

FULL INTEGRATION WITH THE SAS® WEB APPLICATION SERVER

At the moment the Errors application uses the `.htaccess` file functionality within the Apache web server to enable a lot of the features. SAS uses the Apache module `mod_proxy` to proxy requests through the web server and on to the web application server. This is a problem for Errors because we have found that the rules for `mod_proxy` are executed before the `.htaccess` rules are executed. This has been an issue for Errors because it has meant, thus far, we have been unable to implement maintenance mode for the SAS Web Applications (SAS® Visual Analytics, SAS® Information Delivery Portal, etc.) without disabling the traffic going to the web application server. This has prevented us from being able to use the special people functionality of the Errors application to facilitate limited business checkout whereby only a small, selected subset of the user community can use an application such as SAS® Studio or SAS® Visual Analytics.

From some research that we have done we think that this functionality could be achieved by re-writing SAS' `mod_proxy` rules with `mod_jk` but isn't something that we have tested at the moment. Also, for SAS® 9.4, SAS specifically moved away from using `mod_jk` for facilitating back end web application server connections because of its inability to support encryption.

BETTER SAS® MOBILE BI ERRORS

At the moment, the production version of the Errors application is only deployed with the capability for customizing the error messages that are generated by the SAS® Web Server. We are significantly growing our SAS® Visual Analytics use and our users are keen to start making use of the mobile BI capability. To this end, we have been working on a new version of the Errors application that has the ability to edit the error messages that are shown in the SAS® Mobile BI application from the default to something more applicable to use. An example of this would be the generic "Please contact your SAS

Administrator" message. We have changed this to be the team and mailbox name "Please contact ~DASH for information".

The difference between the mobile errors and the web server error messages that we have so far is that they are less on-the-fly. This is because the configuration file from which these error messages are read are within the JVM configuration which is read into memory when the JVM starts. In order for any updates to the error messages to be picked up the JVM needs to be restarted. I don't think this will be a big problem for us; these errors are not changed regularly once they have been set by the application owners.

CONCLUSION

The default, unhelpful, stark and intimidating error messages that are the default with the SAS® 9.4 Web Server do not promote or encourage a strong or engaging user experience. With the development of Errors, RBS has tried to alleviate some of that pain as we as introduce additional features that can help make the life of a SAS platform administrator looking after a system with a middle tier more bearable.

We have shown that it is possible to create an application that allows a platform administrator to customize the error messages seen by users on-the-fly with no downtime. We have also seen that it is possible to create a maintenance mode, allowing platform administrators to take a SAS Middle Tier offline while being able to maintain an enjoyable user experience. Further, we have shown the additional features that we have added to the Errors application to make managing our large and critical middle tiers easier.

REFERENCES

Apache HTTP Server, Accessed February 20th 2016 <https://httpd.apache.org/docs/2.4/custom-error.html>

ACKNOWLEDGMENTS

There are a significant number of people, too many to list here, who have helped make the Errors application what it is today and have helped with this paper. The author would like to a special mention to the following people for their significant contributions: Dileep Pournami, Sergey Iglov, Ekaitz Goienola, Janet Hussey, Kylene Birnie, and Ross Sansom.

RECOMMENDED READING

- <http://getbootstrap.com/>
- *SAS(R) 9.4 Intelligence Platform: Middle-Tier Administration Guide*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Christopher Blake
2 ½ Devonshire Sq, London, EC2M 4BA, United Kingdom
+44 207 672 9316
christopher.blake@rbs.com || cjb.blake@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.