

SBS Network Operating Center Alert Web Console

By

Debra Sweet Barrett

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

March 2005

SBS Network Operating Center Alert Web Console

By

Debra Sweet Barrett

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2005 by Debra Sweet Barrett

The information in this document is proprietary and may not be reproduced or distributed
in whole or in part without the permission of the owner

Debra Sweet Barrett

Date

Professor Tamrisra Sanyal, Faculty Advisor

Date

James F. Sullivan, Department Head

Date

Acknowledgements

I would like to thank Gene O'Shaughnessy, former manager of the SBS Remote Network Services Tools Team for giving me the opportunity to participate in his Siemens Six Sigma Project and for knowing this project would fulfill my senior design requirements. I would like to thank my husband Blaine and daughter Ella for allowing me to take time to work on this project and finally I would like to thank Professor Tamrisra Sanyal for his patience and assistance on this project.

Table of Contents

Section	Page
Acknowledgements	i
Table of Contents	ii
List of Figures	iii
Abstract	iv
1. Statement of Problem	1
2. Description of Solution	1
2.1.1. User Profile	4
2.1.2. Design Protocols	5
3. Deliverables	6
4. Design and Development	8
4.1.1. Timeline	8
4.1.2. Budget	9
4.1.3. Hardware and Software Specifications	10
5. Proof of Design	
5.1.1. Home Web Interface	11
5.1.2. User Settings	12
5.1.3. Reports	13
5.1.4. My Incidents Options	14
5.1.5. SQL Database	18
6. Testing Procedures	21
7. Conclusions and Recommends	21
7.1.1. Conclusions	21
7.1.2. Recommendations	22
Appendix A.	24
A1. Home Page Code Snippet	24
A2. Login	31
A3. Setting Refresh Rate	33
References	34

List of Figures

Figure Number	Page
Figure Number 1. Monitoring with Pagers	3
Figure Number 2. Network Console Main Screen	11
Figure Number 3. Options Pop Up	12
Figure Number 4. Network Console Warning Sound Settings	12
Figure Number 5. Network Console Refresh Rate	13
Figure Number 6. Network Console Report Menu Options	13
Figure Number 7. Web Forms	14
Figure Number 8. Application Overview of hardware and software	17
Figure Number 9. SQL Table Relationships	19
Figure Number10. SQL Tables	21

Abstract

SBS Alert Web Console is a web based alarm console for addressing server and router health events. The target audience for this application is the technical support staff of SBS. The management of SBS is using this application as a sales tool for potential customers. It is a product not available to the general market of server monitoring. The Web interface was created using ASP.net and using SQL Server for event logging. The Web site has simple navigation and provides reporting for management.

Network Operation Center Alert Web Console

1. Statement of the Problem

Siemens Business Services, Inc, Network Services division provides server and network monitoring for several fortune 500/1000 organizations. “Siemens Business Services and the SieQuence® solution to manage over 30,000 servers and network devices along with over 1,000,000 desktops in the U.S. alone. A centralized call center handles over 5,000,000 help desk calls annually. Siemens Business Services has the best-in-class people, process and technology to meet your specific business requirements.” (5) A variety of monitoring tools are currently in place within customer environments for network and server monitoring. Each tool has a main console interface for monitoring events, but the technical support engineers are not customer dedicated; they are required to monitor multiple customer environments. Monitoring several consoles is not the best use of their time and previous alarms are sent to pagers and there was no way to determine if an alarm was received or processed. Figure #1 provides a visual example of how alarms are being processed.

2. Description of the Solution

The Network Operations Alert Web Console was developed to allow one console view for monitoring, to ensure accountability that alarms were being addressed and by what technical support engineer. All tools use simple network management protocol (SNMP) for event forwarding. When an event is detected on the monitored device an SNMP trap is sent to the application specific console. These traps will instead be

forwarded to the Network Operations Alert Web Console. This will provide network support engineers one place to receive, acknowledge and address alarms. The console has a Web interface, with an SQL database for alarms to be logged. Logging alarms will provide data that can be used for trending and reporting purposes to SBS customer base.

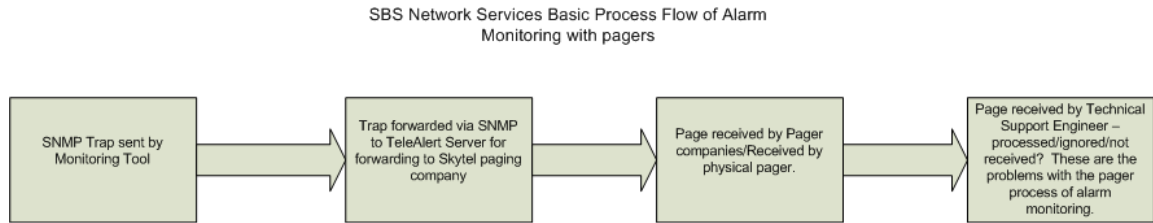


Figure 1. Alarm Flow with Pagers

2.1 User Profile

The Network Operations Alert Web console will be used by three groups within SBS. The three groups include technical support engineers, shift managers and customer account service delivery team.

- 1) Technical Support Engineer: Engineers are expert/proficient computer users in various computer industry hardware and software. These engineers have computer industry technical certifications. Certifications may include COMPTia Network Plus, Microsoft Certified Professionals, Microsoft Certified Systems Engineers and Cisco Certified Network Administrator. Engineers will use the Network Services Alert Web Console to monitor and acknowledge posted events from the various monitoring tools.
- 2) Network Operation Shift Managers: Mid-level computer user experience. Shift managers will periodically view the console to ensure events are being acknowledged in a timely fashion. A Report feature is available for shift managers to view current engineers on-line.
- 3) Service Delivery Team: Mid-level computer user experience. SDT will have view rights within the Network Services Alert Web Console. Team members will be able to view current events for their account and a report of periodical data for a specific customer.

2.2 Design Protocol

The Network Operating Center Alert Web Console application consists of a Web interface developed using ASP.net (3), programming using VB.net for alarm routing and Microsoft SQL 2000(3) database to track, store and maintain a history of the alarm events.

The Home screen is the main interface where alarms are posted after they are received in the database. The default.apx refreshes and checked for new alarms in the database every 60 seconds. Users have to log into the Network console using their Network Services Shared Infrastructure WAN ID.

3. Deliverables

- 1) Web based application that will post valid network and server event traps.
- 2) Engineer accountability will be possible by logging WAN ID when an alarm is addressed.
- 3) Web interface for processing alarms will consist of a simple design for ease of navigation/use.
 - Web interface will be developed using ASP.Net.
 - Alarm format will be same as currently appears on the pagers.
 - Set refresh rate every 60 seconds as default – Pager default rate is also 60 seconds.
- 4) Back end database to store alarms.
 - Database will be used for event tracking, performance trending on a customer specific node and reporting.
 - Database will be developed using Microsoft SQL Server 2000 (current business standard for SBS).
 - Reports to include – Current online users, top 10 alarms per day. Other reports will be developed by the SBS Server Delivery team.
- 5) User manual will be created for new hire technical support engineers on how to do the following:
 - Access the web site (the URL)
 - Sign on using the SI Domain accounts
 - Read events on screen
 - Screen options/enable customer filters

- Acknowledge and close alarms

4. Design and Development

4.1. Timeline

This project was completed in February 2004. The time lines below outline documentation required of the project for Senior Design requirements.

Task	Date Required	Completed Yes/No
Document SQL Tables of Outage Database	November 1, 2004	Yes
Document SQL Views of Outage Database for Design Freeze	December 1, 2004	Yes
Visual Application Overview	November 1, 2004	Yes
Document ASP and VB Code for Senior Design 3	January 2005	Yes
Create user manual for Technical Support Engineers	March 2005	Yes

4.1.2 Budget

Budget - Hardware/Software

Siemens Business Services, Inc. is a Microsoft Solutions provider and already has ample application licenses in place for software required for this application. This application will be placed on existing SBS hardware. There will be no hardware/software costs associated with this project implementation. Sample hardware and software costs have been estimated below:

Database Server		
Software		
Microsoft SQL Server 2000 Enterprise Edition (25-CAL)	\$11,099	(3)
Microsoft Server 2000 Advance Server (25-CAL)	\$3,999	(4)
Microsoft Visual Studio.NET 2003 Enterprise Developer Edition	\$1,799	(4)
Hardware		
Dell PowerEdge 1600SC , Dual Xeon Processors 2.4Ghz, 512K Cache, 4-18GB SCSI Harddrives, 1GB SD Ram	<u>\$2,917</u>	(1)
	\$19,814	
Web Server		
Software		
Microsoft Server 2000 Advance Server (25-CAL)	\$3,999	(3)
Hardware		
Dell PowerEdge 1600SC , Dual Xeon Processors 2.4Ghz, 512K Cache, 4-18GB SCSI Harddrives, 1GB SD Ram	<u>\$2,917</u>	(1)
	<u>\$6,916</u>	
Hardware/Software Total for Project-->		\$26,730

(These estimates are based on full Retail List price - actual costs would be slightly lower due to vendor discounts/relationships.)

4.1.3 Hardware/Software Specifications

The table below represents the current hardware and software configurations of the systems running the Alert Web Console.

Web Server

Server name: USRNSNOTES IP address: 155.45.25.124
NOS: Windows 2000 Advanced Server/SP2
IPC Rack #86, Make/Model: Dell Power Edge 6400
Serial Number 5XFX901
Processor: Quad Processor/ Intel Pentium III Xeon processor

Database Server

Server Name: RNSNMCMGT1 IP Address: 155.45.25.101
NOS: Windows 2000 Advanced Server/SP2
IPC Rack # 85, Make/Model: Dell Power Edge 6450
Serial Number: CG6X801
Processor: Quad Processor/ Intel Pentium III Xeon processor

5. Proof of Design

5.1.1 Web Interface

The Web form for NOC Alert Web Console consists of a HOME screen where all alarms are received. See Figure #2.



Count	Case	Customer Node	Resource	Last Message	Last Occurred	Actions
22	5261472-1101	OWC	ARDS6100	Logical Disk D: Full Disk D: Used % is 100.00: >TH = 95 Free space MB is 2.56: <TH = 100 1498:NT_LogicalDiskSpace	11/2/2004 8:37:56 AM	
42	ignore	MET	AS-INDIPRT01	BACKUP Status is Major Backup Exec Loader Req Attention	11/2/2004 8:32:28 AM	
1		MET	AS-STFYBKUP02	BACKUP Status is Minor Backup Exec Job Failed	11/2/2004 8:31:35 AM	
17	ignore	MET	as-siscbkup14	BACKUP Status is Minor Backup Exec Job Failed	11/2/2004 8:29:31 AM	
2	3151098	MET	Boston-AS_BOSTMEM05	AVAILABILITY ASR Node has ASR Error	11/2/2004 8:28:34 AM	
1		MET	AS-SISCBKUP47	BACKUP Status is Minor Backup Exec Job Failed	11/2/2004 8:28:00 AM	
1		MET	Scranton-AS-SISCBKUP14	BACKUP Met-BETTraps 10.10.60.168 Met-CPQ.SVR.CS is BE_JobFailed , \$VB(1)	11/2/2004 8:27:07 AM	
2		MET	Scranton-AS-SISCBKUP47	BACKUP Met-BETTraps 10.10.30.153 Met-CPQ.SVR.CS is BE_JobFailed , \$VB(1)	11/2/2004 8:25:19 AM	
48	ignore	MET	AS-SISCBKUP49	BACKUP Status is Major Backup Exec Loader Req Attention	11/2/2004 8:22:39 AM	
82	ignore	MET	as-siscbkup41	BACKUP Status is Major Backup Exec Loader Req Attention	11/2/2004 8:20:21 AM	
81	ignore	MET	AS-BLOMBKUP01	BACKUP Status is Major Backup Exec Loader Req Attention	11/2/2004 8:19:54 AM	
74	FPA	MET	AS-RISCM0SP04	HARDWARE Status is Critical Fit To Power Redundancy Lost	11/2/2004 8:18:54 AM	
1		MET	Rensselaer-AS_RISCSOL34	CPU Met-MX_SusCPUUtil-New 10.9.164.83 Met-CPQ.SVR.DB is CPU_Error - Utilization is HIGH.	11/2/2004 8:17:40 AM	
39	5261582-1101	OWC	NJOS6100	AVAILABILITY is DeviceDown 100.179.15.3 Owens-Icmp has become unavailable please investigate.	11/2/2004 8:16:43 AM	
2		MET	AS-SISCBKUP38	BACKUP Status is Minor Backup Exec Job Failed	11/2/2004 8:16:40 AM	
2		MET	Scranton-AS-SISCBKUP38	BACKUP Met-BETTraps 10.10.67.75 Met-CPQ.SVR.CS is BE_JobFailed , \$VB(1)	11/2/2004 8:16:16 AM	
2		OWC	WHOS6628	OS Down Service - SysmonLog SysmonLog down, will not be restarted. 1498:NT_ServiceDown	11/2/2004 8:15:59 AM	

Figure 2. Network Console Home Page

5.1.2 User OPTIONS menu: This feature allows the user to turn off visual and audio notifications and also reset the refresh rate. Audio and Visual alarm notification is on by default and the console refresh rate is set to 60 seconds.

The four elements defined are:

Warning Popup - allows the popup to be set on or off

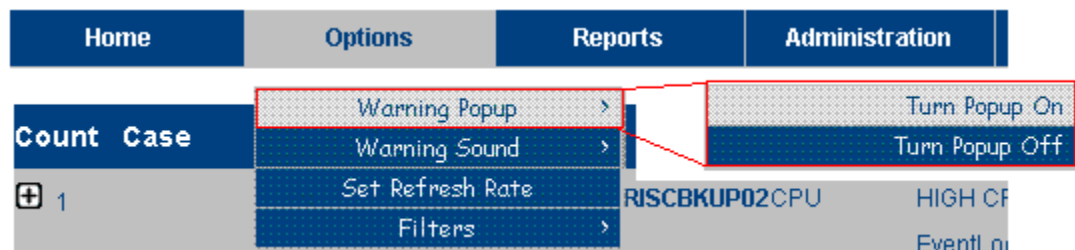


Figure 3. Options-Pop Up

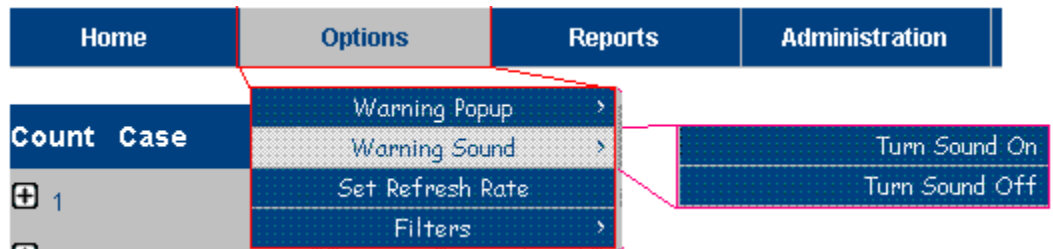


Figure 4. Warning Sound

Allows the user to set the Web Page refresh rate. The default setting = 120 (0.12 second)

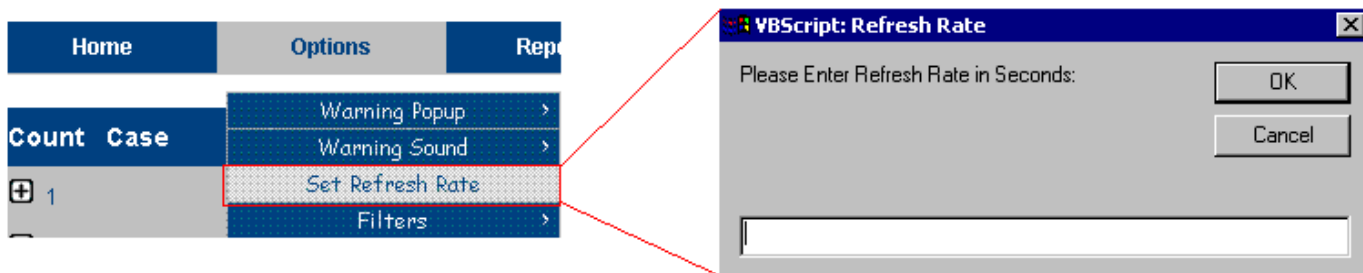


Figure 5. Set Refresh Rate

5.1.3 REPORTS menu: This option will be used by shift managers and the service delivery team to view current users online. Future reports will be added such as daily alarms for a specific customer, outages for device. See Figure #6.

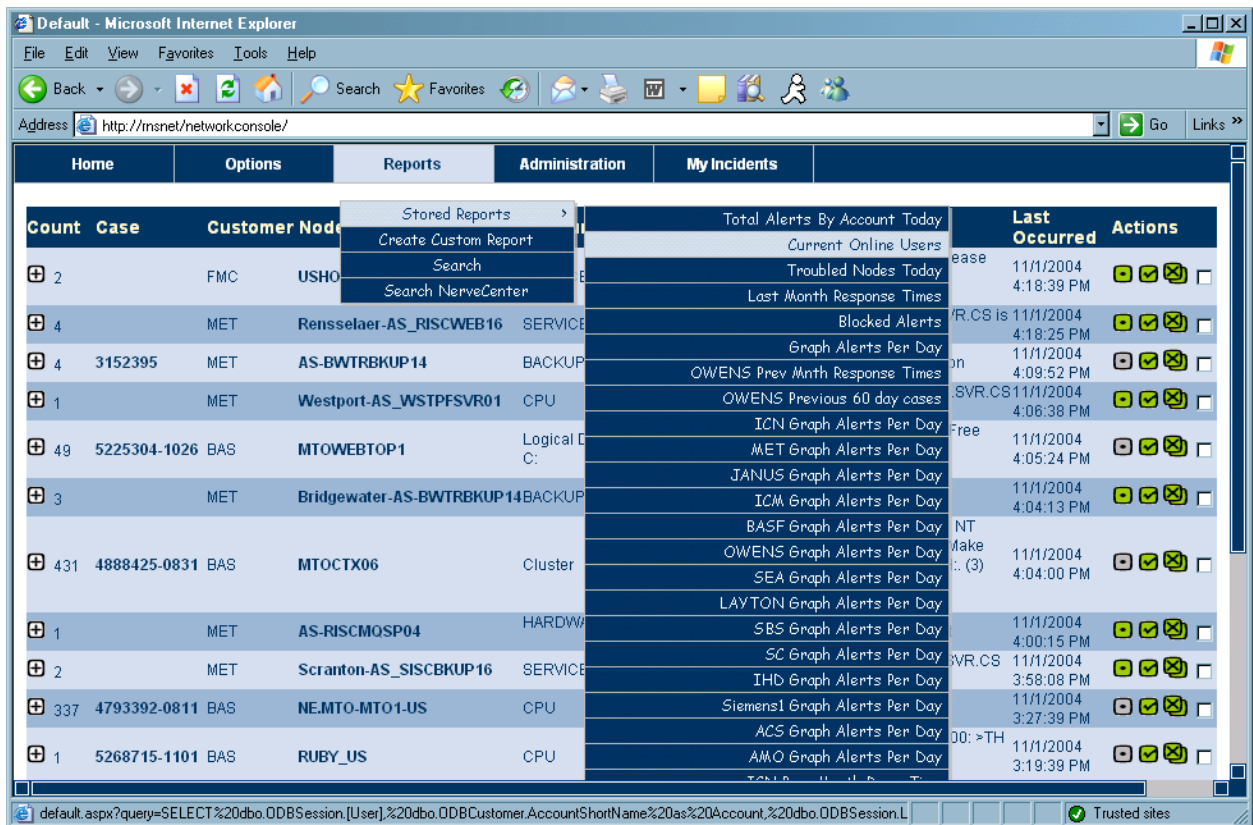


Figure 6. Reports Menu

5.1.4 MY INCIDENTS menu option: After an alarm is accepted by a technical support engineer, the SQL Database record is updated with the Engineers WAN ID and time alarm was acknowledged for accountability.

The web interface will include a header under which alarm information is displayed.

The web form (WebForm1) for Network Console is default.aspx.

Header

Count	Case	Customer Node	Resource	Last Message	Last Occurred	Actions
4		JAN	EXCONP001	Mail	4/29/2004 7:35:04 PM	
16		LAY	TLC-MAIL 1	Mail	4/29/2004 7:34:13 PM	
1	4315350-0429	ICN	Chelmsford-cmf-sw1	AVAILABILITY is DeviceDown 165.218.86.137 ICN-ENTRSYS.SWT has become unavailable, please investigate.	4/29/2004 7:28:01 PM	
2		MET	Rensselaer-AS_RISCBKUP02	CPU	4/29/2004 7:08:51 PM	
1		MET	Scranton-AS_SISCNTSDB01	CPU	4/29/2004 7:04:37 PM	
1		MET	Boston-AT_BOSTSHRAPP01	CPU	4/29/2004 6:30:57 PM	

Body

Footer

Records: 6

CREATE NEW CASES ACKNOWLEDGE SELECTED ALERTS CLOSE SELECTED INCIDENTS

Figure 7. Web Form

Menu Header

- Count – this information indicates how many times an alarm has triggered for a single device. This is a cumulative count for any event that is listed in ‘open’ status in the Outage Database.
- Case – Displays the case number received from Clarify Guardian Tracking Application.

- Customer – Displays the customer short name for the specific customer the event was triggered for.
- Node – Displays the node name for the device in alarm.
- Resource – Alarms are categorized by Hardware, OS (operating system), CPU/Memory, Router, Switch, and Service to assist technical support engineers in determining what skills are required to address the alarm.
- Message – This text field will display a brief description of the event from the various monitoring tools. (i.e. service AUTOUPDATE failed – would indicate to a technical support engineer that an automatic application services has stopped and needs to be restarted)
- Last Occurred – this is time/date stamp of the time the trap was received and posted to the database.

Users will have three actions for processing the alarms.

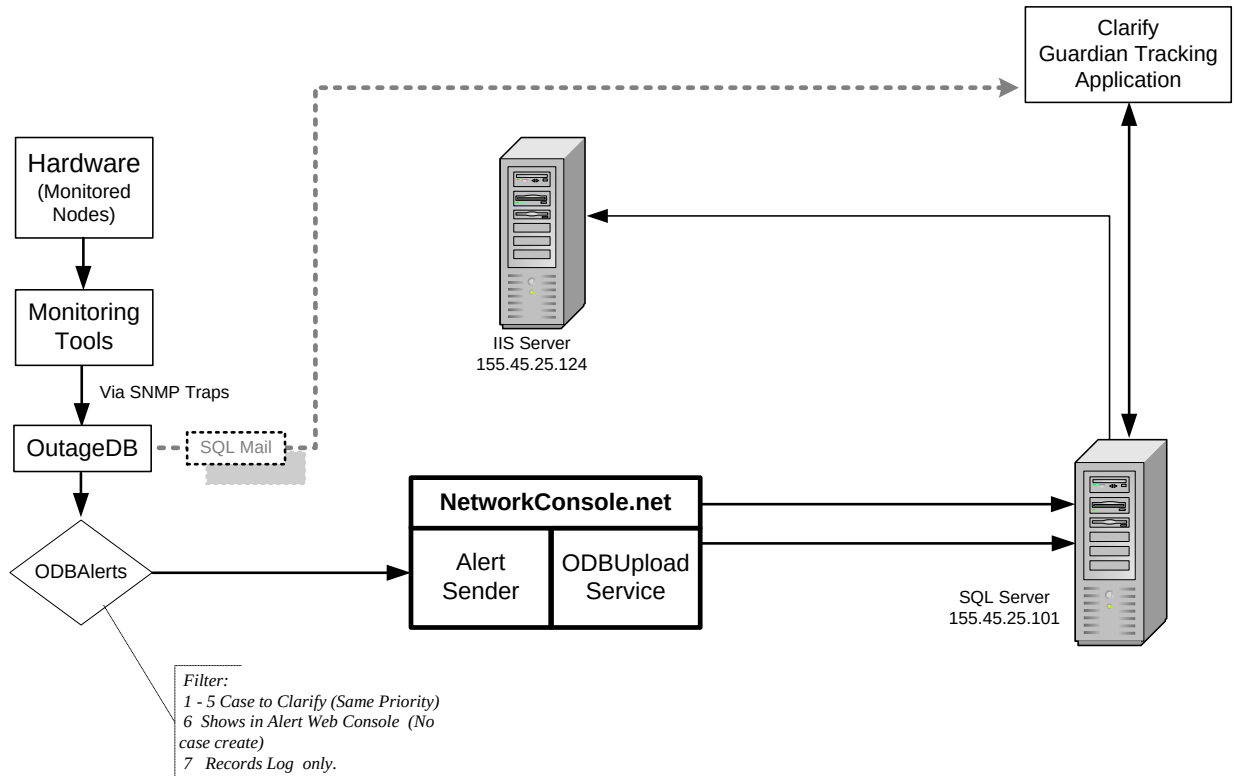
- Create Case – allows the user to create a case. The alarm will be packed into a template using SMTP (simple mail transfer protocol) and a text email will be sent to the Clarify Guardian system to have a case created. The Clarify system will

automatically return a case number back to the Outage Database via SMTP and post the case number into the alert record.

- Acknowledge – this feature will take the record off the HOME console page. By executing this feature the technical support engineer's WAN ID and time acknowledge will post to the outage database. This shows accountability of the alarm by a specific support engineer.
- Close – After the incident/alarm has been cleared, the technical support engineer will mark the alarm as closed. This will mark the incident closed in the database. If a new alarm occurs for a device that has been closed the count to the far left will reset to zero.

Visual Overview of Application

Alert Web Console is a set Visual Studio.net Applications (VB/ASP .net) with a SQL Server 2000 database



(Alert Sender/ODBUpload Service – Designed/Programmed by G. O'Shaughnessy, IET Student))

Figure 8. Application Overview

SQL Database

The SQL Database has 4 tables. Tables are shown below with a brief description.

Table relationships are also displayed below in Figure#5.

- ODBCcustomer – This table contains a record for each Network Services customer. The table includes tracking system account identifier, account name, and other fields required for auto case create in the SBS Tracking tool Clarify Guardian.
- ODBOutage – This table contains information about a current outage. Each event receives a unique ID from the ODBAlerts table.
- ODBAlerts – This table receives the SNMP trap from the tools. The alert is assigned an Alert ID and then based on type the alarm is displayed the network console.
- ODBTool – Table contains information about the specific tool that the trap was sent from.

SQL Table Relationships

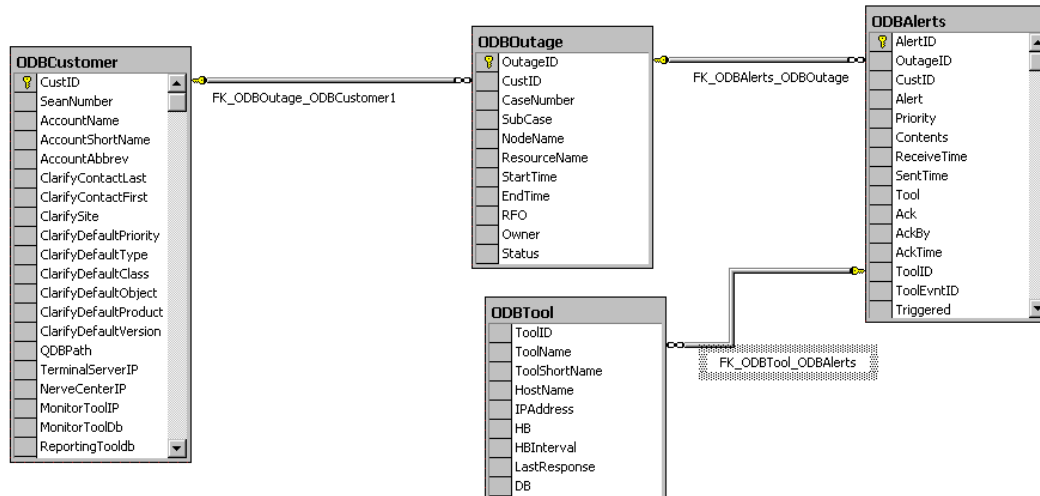


Figure 9. SQL Table Relationships

SQL Table Details

ODBAalerts

Key	<i>AlertID</i>	Bigint(8)
	<i>OutageID</i>	Bigint(8)
	<i>CustID</i>	Bigint(8)
	<i>Alert</i>	Char(255)
	<i>Priority</i>	Char(10)
	<i>Contents</i>	Char(1024)
	<i>ReceiveTime</i>	Datetime(8)
	<i>SentTime</i>	Datetime(8)
	<i>Tool</i>	Char(100)
	<i>Ack</i>	Bit(1)
	<i>AckBy</i>	Char(255)
	<i>Rowguid</i>	Uniqueidentifier(16)
	<i>ToolID</i>	Bigint(8)
	<i>ToolEventID</i>	Bigint(8)
	<i>Triggered</i>	Bit(1)

ODBCustomer

PK *	<i>CustID</i>	Bigint(8)
	<i>Seamnumber</i>	Char(10)
	<i>AccountNumber</i>	Char(5)
	<i>AccountShortName</i>	Char(10)
	<i>AccountAbbrev</i>	Char(4)
	<i>ClarifyContactLast</i>	Char(50)
	<i>ClarifyContactFirst</i>	Char(50)
	<i>ClarifySite</i>	Char(50)
	<i>ClarifyDefaultPriority</i>	Char(10)
	<i>ClarifyDefaultType</i>	Char(10)
	<i>ClarifyDefaultClass</i>	Char(30)
	<i>ClarifyDefaultObject</i>	Char(30)
	<i>ClarifyDefaultProduct</i>	Char(30)
	<i>ClarifyDefaultVersion</i>	Char(30)
	<i>QDBPath</i>	Char(40)
	<i>TerminalServerIP</i>	Char(15)
	<i>NerveCenterIP</i>	Char(15)
	<i>MonitorToolIP</i>	Char(15)
	<i>MonitorToolDB</i>	Char(40)
	<i>ReportingToolDB</i>	Char(40)
	<i>AutoTicket</i>	Bit(1)
	<i>Rowguid</i>	Uniqueidentifier(16)

ODBOutage

PK *	<i>OutageID</i>	Bigint(8)
	<i>CustID</i>	Bigint(8)
	<i>CaseNumber</i>	Char(15)
	<i>SubCase</i>	Char(15)
	<i>NodeName</i>	Char(50)
	<i>ResourceName</i>	Char(40)

	<i>StartTime</i>	Datetime(8)
	<i>EndTime</i>	Datetime(8)
	<i>RFO</i>	Char(1024)
	<i>Owner</i>	Char(50)
	<i>Status</i>	Char(10)
	<i>Rowguid</i>	Uniqueidentifier(16)
ODBTool		
Key*	<i>ToolID</i>	Bigint(8)
	<i>ToolName</i>	Char(255)
	<i>ToolShortName</i>	Char(10)
	<i>HostName</i>	Char(255)
	<i>IPAddress</i>	Char(15)
	<i>HB</i>	Bit(1)
	<i>HBInterval</i>	Bigint(8)
	<i>LastResponse</i>	Datetime(8)
	<i>DB</i>	Char(255)
	<i>rowguid</i>	Uniqueidentifier(16)

Figure 10. SQL Tables

6. Testing Procedure

Alarm testing will run parallel to current system. SNMP traps will be forwarded to the NOC Web Alert console while continuing to be sent to Skytel pagers. The testing period will run 4 weeks. During the phase engineers will be documenting any problems with the web interface, providing continuous feedback. At the end of the test phase the project will be reviewed with SBS management to implement the application into production

7. Conclusions and Recommendations

7.1 Conclusions

This project was created in response to a need to bring alarm logging, accountability and reduce cost of pagers for SBS. A team of SBS employees were brought together and used the Six Sigma foundations to gather data and design the

requirements of this project. The final project resulted in a Web Console for the technical support engineers of SBS. The project was developed using ASP.Net, Microsoft SQL Server 2000. Other aspects of the console were developed using VB.Net. The project was completed in February 2004. There was no cost associated with the project as SBS had all software and hardware requirements in place. The deliverables for this project were met. Testing was completed to ensure product functionality and usability.

7.2 Recommendations

This project started out as a Six Sigma Project to eliminate pager, bring accountability to alarms and logging so device trending could be established for SBS customer based. The first run on the project we were able to meet all of the requirements. The console has a simple interface and SQL Server is used to log all of the events for reporting required by our customers. Initially we assumed the reports the customer required and provided those as menu choices. The reports created were returned back to the customer and they actually had different requirements than provide in the reports menu. The Six Sigma Team had to work with the customers to retool the reports to their requirements. This did not create a lot of work but it was a simple step that was overlooked and we should have included the customers from the start of the project.

The web interface has a simple design that meets the needs of alarms being addressed. The console is used for pre-sales demonstrations and customers are always asking if any of the alarms change color or flash after they have passed a time interval, alerting engineers that the alarm is 15 or 30 minutes old. This is currently not in place

with the console, but customers make a good recommendation on the flashing. All alarms are critical and should be addressed and acknowledged within 15 minutes of posting to the console. Also customers have made mention of how simple the screen is. I could have dressed it up with the SBS company logo, but the current interface meets the need of alarms being addressed so it was not updated.

This was a joint project by me and Gene O'Shaughnessy. He is an IET student, but due the family requirements he was not able to go through the Senior Design classes to present the project with me. I would have liked to have worked with Gene and presented all elements of this project together. There are other elements of this tool more involved and presenting all features together would have made for a stronger presentation. Gene has interfaced the application with our tracking tool, provided trend analysis reports from the database and applied logic to suppress alarms we call noise makers. These are alarms for problems the customers won't fix, but the support engineers are not expected to do anything about. Gene has also linked the application to our knowledge base Livelink so engineers have immediate access to procedures.

Appendix A.

Code Snippets.

A.1 Main Console Home Page Code Snippet

Code from the main web form can be found below. This was created using ASP.net. The web page will search the SQL database every 60 seconds for any new records that have been received and post to the main console page.

Default.aspx

```
<%@ Register TagPrefix="Menu" TagName="Body" Src="MenuBody.ascx" %>
<%@ Register TagPrefix="Menu" TagName="Header" Src="MenuHeader.ascx" %>
<%@ Page Language='vb' AutoEventWireup='false' Codebehind=Default.aspx.vb
Inherits="NetworkConsole.Net.WebForm1" %>
<%@ Register TagPrefix="cc1" Namespace="DoubleListBoxControl"
Assembly="DoubleListBox" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>Default</title>
    <meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
    <meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
    <meta http-equiv=refresh content="<%=RefreshRate %>">
    <LINK href="Styles.css" type="text/css" rel="stylesheet">
    <%
      if bWarning and bSound then
    %>
    <bgsound src="Siren.wav" loop="1">
    <%
      end if
    %>
    <MENU:HEADER id="MenuHeader"
runat="server"></MENU:HEADER>
  </HEAD>
  <body MS_POSITIONING="GridLayout">
    <MENU:BODY id="MenuBody" runat="server"></MENU:BODY><br>
    <br>
```



```
  |
```

```

<a
href='ackall.aspx?ID=<%#
DataBinder.Eval(Container.dataitem,"OUTAGEID")%>'><img border='0'
src='images/Ack.gif' alt='Acknowledge All Alerts'></a>
<a href='close.aspx?ID=<%#
DataBinder.Eval(Container.dataitem,"Outageid")%>'><img border='0'
src='images/Close.gif' alt='Close Incident'></a>
<input type='checkbox'
name='ID' value='<%# DataBinder.Eval(Container.dataitem,"OutageID")%>'
</td>
</tr>
</AlternatingItemTemplate>
<FooterTemplate>
<tr>
<td align="left" class='strip'
colspan='7'>
Records:
<%# recordcount %>
</td>
<td class='strip' align="right">


</td>
</tr>
<tr>
<td class="strip" align="center"
colspan="8">
<p align="right">
 CREATE NEW CASES  ACKNOWLEDGE SELECTED ALERTS 
CLOSE SELECTED
INCIDENTS</td>
</tr>
</FooterTemplate>

```

```

                                </asp:repeater></table>
                                </form>
                                <SCRIPT type="text/vbscript">
sub Clicked(URL)
    Dim objElement
    Dim lngCurrentElement
    Dim strTempText

    Const TAG_NAME = "input"

    lngCurrentElement = 0

    For Each objElement In document.all.tags(TAG_NAME)

        ' Determine format of message to user depending on
        ' the element number.
        If lngCurrentElement = 0 Then
            strTempText = "first"
        ElseIf lngCurrentElement = _
            document.all.tags(TAG_NAME).length - 1 Then
            strTempText = "last"
        Else
            strTempText = "next"
        End If

        ' Report the HTML contents of the current <a> element and
        ' increment the element counter.
        If objelement.name = "ID" and objelement.checked then
            URL = URL & "ID=" & objelement.value & "&"
        end if
        lngCurrentElement = lngCurrentElement + 1

    Next
    URL = left(URL,len(URL) -1 )
    location = url
End sub

sub AckAllClick()
    Dim mURL
    mURL = "ackall.aspx?"
    call Clicked(mURL)
end sub

sub CloseAllClick()
    Dim mURL

```

```
mURL = "close.aspx?"  
call Clicked(mURL)  
end sub
```

```
sub CreateAllClick()  
Dim mURL  
mURL = "create.aspx?"  
call Clicked(mURL)  
end sub
```

```
sub SelectAllClick()  
Dim objElement  
Dim lngCurrentElement  
Dim strTempText
```

```
Const TAG_NAME = "input"
```

```
lngCurrentElement = 0
```

```
For Each objElement In document.all.tags(TAG_NAME)
```

```
    ' Determine format of message to user depending on  
    ' the element number.
```

```
    If lngCurrentElement = 0 Then
```

```
        strTempText = "first"
```

```
    ElseIf lngCurrentElement = _
```

```
        document.all.tags(TAG_NAME).length - 1 Then
```

```
        strTempText = "last"
```

```
    Else
```

```
        strTempText = "next"
```

```
    End If
```

```
    ' Report the HTML contents of the current <a> element and
```

```
    ' increment the element counter.
```

```
    objelement.checked = True
```

```
    lngCurrentElement = lngCurrentElement + 1
```

```
Next
```

```
end sub
```

```
sub DeSelectAllClick()  
Dim objElement  
Dim lngCurrentElement  
Dim strTempText
```

```

Const TAG_NAME = "input"

lngCurrentElement = 0

For Each objElement In document.all.tags(TAG_NAME)

    ' Determine format of message to user depending on
    ' the element number.
    If lngCurrentElement = 0 Then
        strTempText = "first"
    ElseIf lngCurrentElement = _
        document.all.tags(TAG_NAME).length - 1 Then
        strTempText = "last"
    Else
        strTempText = "next"
    End If

    ' Report the HTML contents of the current <a> element and
    ' increment the element counter.
    objelement.checked = False

    lngCurrentElement = lngCurrentElement + 1

Next
end sub

</SCRIPT>
</body>
</HTML>

```

A2. Login-in Code Snippet

Accountability has been an ongoing problem with the Support Engineers acknowledging and addressing alarms. Users are required to login to the Network Console Web page using their SBS Shared Infrastructure WAN ID. As alarms are received on the console users are required to acknowledge the alarms. The SQL

Database is updated with the users WAN ID so accountability is required should a problem arise with an alarm.

```
<!--#include file="ODBAalerts_connect.asp" -->
<%
Dim ID, logon, myArray, NTUserName, TimeStamp
    logon = Request.ServerVariables("LOGON_USER")
    'so we split the logon variable by the "\"
    myArray = Split(logon, "\", -1, 1)

    'take the upper bound of the logon variable
    logon = myArray(Ubound(myarray))

    'and you're left with just the NT Username
    NTUserName = logon
    'Response.Write NTUserName

TimeStamp = Now()
ID = TRIM(REQUEST("ID"))
IF ID <> "" THEN
    connect.execute "UPDATE ODBALERTS SET ACK = 1, AckBy = " &
NTUserName & ", AckTime = " & TimeStamp & " WHERE ALERTID = " & ID, ,128
END IF
%>
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title></title>
</head>
<body>
    <%
        ' Change the following line as appropriate for your script
        If 1 = 1 Then
            Response.Clear
            Response.Redirect "home.asp"

        End If
    connect.close
    %>
</body>
x</html>
```

A2. Console Refresh Rate

The default refresh rate on the console is 120 seconds. Users have the ability to reset the refresh rate from the user options on the home page of the Web Alert Console

```
<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="RefreshRate.aspx.vb" Inherits="NetworkConsole.Net.RefreshRate"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>RefreshRate</title>
<meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
<meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
</head>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
<SCRIPT type="text/vbscript">
Dim Rate
Rate = inputbox("Please Enter Refresh Rate in Seconds:", "Refresh Rate")
Location = "changesetting.aspx?Refresh=" & Rate
</script>
</form>
</body>
</html>
```

References

1. Dell.com. <http://www.dell.com>. January 2005.
2. Jupiter media Corporation. <http://www.asp101.com/>. January 2004.
3. Microsoft.com. “How to Buy”.
<http://www.microsoft.com/sql/howtobuy/default.asp>. January 2005.
4. Microsoft.com. “How to Buy”.
<http://www.microsoft.com/windows2000/advancedserver/howtobuy/pricing/default.asp>. January 2005.
5. Siemens Business Services, Inc. http://www.sbs-usa.siemens.com/Portfolio/ITOutsrc_index.asp. December 2004.
6. TechRepublic. <http://www.techrepublic.com>. January 2004.
7. W3Schools. <http://www.w3schools.com>. January 2004.