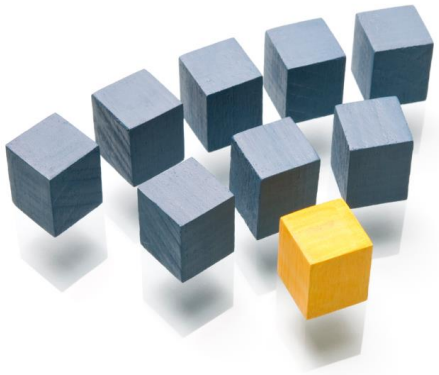
Reflex 80:20

White Paper



INSIDER TECHNOLOGIES

Table of Contents

Introduction

What Is Reflex 80:20

Reflex 80:20 is an Operations and Service Management Facility for the HPE NonStop arena. The package provides performance metrics and log monitoring for key hardware, operating system and application components.

Nominated error conditions can be corrected automatically or passed for manual attention via Console, Enterprise Management, mobile GSM or email technologies.

What This Document Provides

This paper provides a technical overview of the Reflex 80:20 product.

The functionality of each of the Reflex 80:20 software modules is described in detail.

Who Should Read This Document

The document is aimed at people with a technical background.

The document will provide an excellent introduction to new users at an existing installation, or to individuals who are considering a product evaluation and are looking for a more detailed description outside of the information provided by Insider's product literature.

Event Management Service (EMS)

The Reflex 80:20 product relies upon the native NonStop logging system EMS; the Event Management Service. Although you do not need to have EMS expertise to implement Reflex 80:20, this White Paper and our product documentation will refer to some of the terms commonly used in the EMS world.

Before we explore the facilities of Reflex 80:20 you will be better equipped to understand our product if you have a basic understanding of EMS. For those with knowledge of EMS skip this section and move onto the next section, Reflex 80:20 Architecture.

Introduction

The EMS subsystem is installed on every NonStop system, It is started when your machine is loaded and although it is something that you can tune, you cannot shut it down. It provides the native logging environment for your NonStop system. All the hardware, operating system and utility subsystems write their logging information to EMS. The log messages cover error conditions such as hardware failure but they also cover basic information such as a tape being loaded. As the log information covers more than just errors, each message is referred to as an event, hence the Event Management Service.

Although your application can still write log messages to disc files and spooler locations, it will be better integrated with the NonStop node if it too writes log messages to EMS in the required proprietary format. This task requires some programming expertise; however the Reflex 80:20 Gateway module can provide this interface into EMS for your application and we will look at this later in the document.

Already you may sense that the logs will be filled with events that could be ignored and that basing a Management regime for your NonStop node on EMS will require much research and development. Conquering this issue is the unique strength of Reflex 80:20 and as you progress through the paper, this will be illustrated to you.

Capturing and Processing Events

EMS event or log information is stored in a set of disk files. You are in control of where the files are held and how many you want to retain.

Any process wishing to write to these log files does not need to know where they reside as the files are fronted by a collector process. The primary collector process is known as $0, so if you write your log data to $0, then it is written to the appropriate disk file for you. To help spread the processing load you can have more than one collector and these extra processes are known as alternate collectors.

If you want to retrieve log messages so that you can process them, you use EMS distributors. These Distributors will ask a collector, e.g. $0 for events and you can go back in time for them or they can be forwarded on to you as soon as they are written to a log file.

Distributors can then be used to display retrieved log messages in a “pretty” format on your screen or they can be used to send retrieved events to EMS on other NonStop nodes. Finally, they can be used to send retrieved events to applications that will process them. This latter use of a distributor is the standard method that Event Management applications use to obtain their log information, and in this respect, the Reflex 80:20 application is no different.

As this stream of events is being retrieved by a distributor you can apply a filter to it to remove unwanted information. Reflex 80:20 generates and installs a filter for you and the filter’s event selection criteria is based on what has been configured in the Reflex 80:20 event database

Identifying Events

The older logging systems were based on writing strings of text to a disk file or printer. Whilst this format is the best possible for the human eye, it makes for a very inefficient format if this same log message is to be processed by a software module.

To overcome this, EMS event buffers contain numeric tags to help software process them. Each event buffer consists of 2 types of data: token codes and token values. Token codes are unique numbers that identify pieces of information and the token value is that information. For example the time that the event was generated might be token code number 1, so the buffer would contain:

```
1: (18:00, May 16th)
```

It is easier for a software module to retrieve information number 1, rather than scan a long piece of text trying to identify where the timestamp is. Although we have used number 1 in this example, the reality is that the value of these numbers is immense as they have to be unique.

As an application developer you can create events and providing you publish the values of your token codes, other application programs can retrieve your events and unpack the elements with a series of simple requests (“I’ll have information 100, 999 and 1002 please”) rather than having to parse the content of the message.

Every event has a standard set of information or header tokens. These include the name of the node where the event was generated, the time it was generated and the process that generated it.

There are some other important tokens that we will refer to in the document: the Subsystem Identity, the event number, the subject and the subject manager.

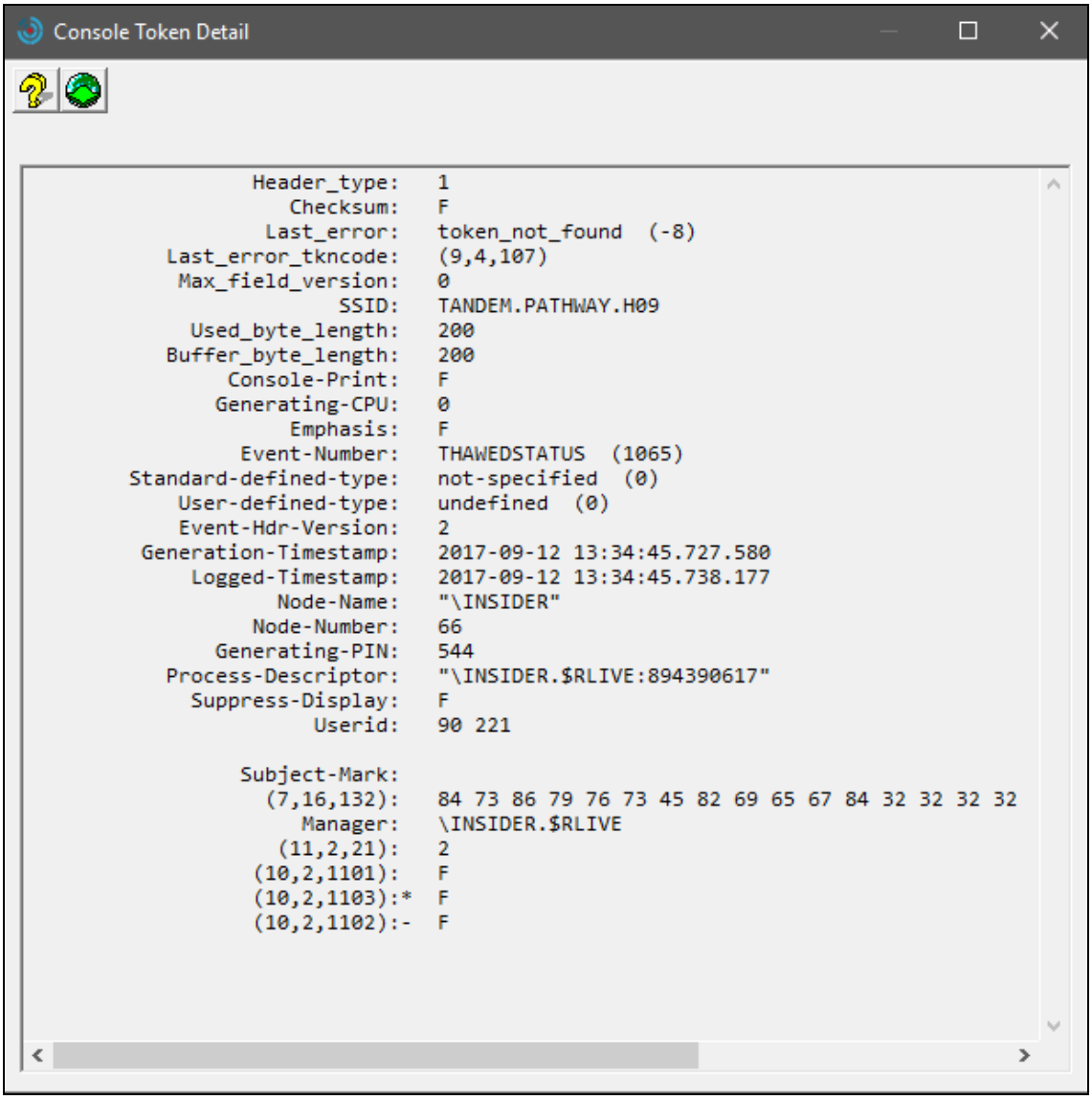
Each type of event needs a unique number, so that if you want to retrieve say all “PATHWAY Server frozen” log messages you can ask your distributor to obtain event “x” and all you will get is this message. This unique identity is provided by a combination of the Subsystem Identity (SSID) and the event number.

The Subsystem identity is made up of three parts; the Owner, the Value and the Version. The Owner of the event is usually based around the manufacturer of the event such as TANDEM or INSIDER. The Subsystem value is allocated on a per subsystem basis, so PATHWAY is number 8, TAPE is number 4 and TMF is number 10. The version is the Operating system version which is documentary and can be ignored, so TANDEM.8.0 are PATHWAY events and TANDEM.4.0 are TAPE events. As with earlier timestamp and node information, this information is held as a series of token codes and values and can be easily retrieved from the EMS event buffer.

The event number identifies the log message within the subsystem, so event 1065 within PATHWAY (TANDEM.8.0) is a Server being thawed but event 1049 is a User PATHWAY terminal being suspended. Using these standards, unique identities can be built up for every event that will be created on your NonStop node.

The subject value will represent the object that the event was for. In our example, it will be the Server Class that was frozen. The Manager is the process that controls our subject and in our example this is the PATHWAY monitor process. This is useful in the automation world as when we receive this event we can connect to the “manager” and ask it to restart the “subject”. Again subjects and managers are held as a series of token codes and values.

Finally, to illustrate the event buffer principle still further we include a screen shot of the Reflex 80:20 Console token map facility in use. The values down the left hand side of the screen are the token codes, the numbers have been converted to text values so that it becomes easier to interpret. The right hand values are the token values.



If further EMS background is required, the “HPE NonStop Technical Library” contains EMS related manuals

Reflex 80:20 Architecture Overview

In its most basic form, Reflex 80:20 can be split into two discrete functions: Reactive Monitoring and Proactive Monitoring.

Central to both of these threads is the NonStop Event Management Service (EMS). Reflex 80:20's reactive monitoring connects to this logging environment to retrieve and process events that you have nominated. In this way, hardware and application failures reported to EMS can be extracted and routed on to other Reflex 80:20 processing engines.

There are two options within Reflex 80:20 for reacting to nominated EMS events:

- Attempt to automatically correct the fault.
- Tell somebody about it. This can be achieved by either displaying the event detail in a graphical or text based Console, sending the event detail to a radio pager, mobile phone or email address, or by escalating the event detail to an Enterprise Manager through a supported interface, or via a low level SNMP trap. You can select more than one of these options.

Reflex 80:20's Proactive monitoring will operate in the background, tracking system performance, process and spooler availability and file attributes. If pre-configured thresholds are exceeded then the proactive monitoring software will raise an EMS event as an alert and this can then be processed by the Reflex 80:20 reactive monitoring processes.

The core database file that manages this process is the Reflex 80:20 event database. This file contains all the events that a Reflex 80:20 node is interested in monitoring and a matrix of how they should be processed.

EVENT SUBSYSTEM ID	EVENT NUMBER	GRAPHICAL CONSOLE	AUTOMATION	RADIO PAGER / GSM	EMS EVENT	ENTERPRISE MANAGER	EMAIL	SNMP TRAP
INSIDER.50.0	1001	Y	N	Y	N	N	N	N
TANDEM.31.0	5015	Y	N	Y	N	Y	Y	N
TANDEM.15.0	101	Y	N	Y	N	Y	Y	N
Etc...								

This database file also includes an alias, so for the third event in our table (TANDEM.15.0, 101) the more descriptive text "CPU-Down" is displayed on the appropriate screens and reports.

Key to the success of any Reflex 80:20 installation is the analysis and selection of the appropriate event log messages and the setting of the proactive monitoring thresholds so that the appropriate level of log information is generated.

To assist with this analysis, the Reflex 80:20 product is shipped with a pre-configured event database that contains the standard NonStop hardware "up" and "down" messages together with the events that will be generated by the Reflex 80:20 proactive monitoring software.

To complete the event table, the administrator of Reflex 80:20 needs to add events generated by the local application or monitoring macros. The event can be added by hand, or if it can be viewed in the Reflex 80:20 Console or the Reflex 80:20 Discovery statistics modules, then a database entry can be created at the click of a button. Alternatively, if the event has already been configured into a Reflex 80:20 database on another node it can be exported from that database into your node.

The Event Monitor

Once the event table is built and the reaction options are selected, the Reflex 80:20 Event Monitor will connect to the primary EMS collector $0, and to any nominated alternate collectors and await receipt of events as they occur.

The event database will be used to forward the details of a received event to the appropriate Reflex 80:20 React engine, although there are some circumstances when events will not be processed:

- The event has been received outside the cover period that was set for processing the event.
- The processing of this event within Reflex has been frozen. You may wish to do this if you are planning to reload a hardware module and want to suppress alerts.
- The event has broken the processing threshold set for it. You can say, for example, ignore this event if you receive more than 5 instances in a 60 second period.

The Event Monitor will only receive the events registered in the Reflex 80:20 database because a filter has been implemented to request just those events.

When you add new events to the database it is necessary to rebuild, recompile and reinstall this filter so that the new events are retrieved. All this can be done from menu options within the GUI and it is not necessary to stop the Reflex 80:20 software to complete this task.

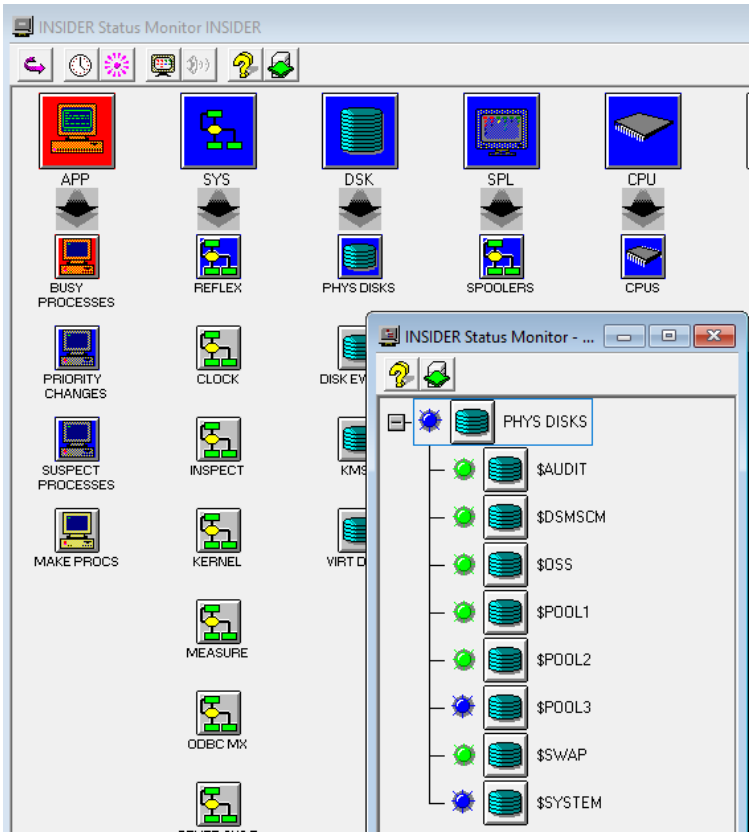
We will now look at the Reactive processing options in more detail.

Graphical Console Overview

The Reflex 80:20 Graphical Console is known as the OverDrive Status Monitor.

The view represents the overall status of a single NonStop node. OverDrive helps an installation to manage their NonStop nodes and active services by providing a graphical view of nominated exception conditions. Installations that need to manage their network other than “by exception” can witness a richer EMS event stream by using the OverDrive top-level views in conjunction with the text based Console facility that is described in a later section.

The OverDrive display is made up of columns known as Classes, and within classes, icons that can represent an underlying group of objects, or a single object. Classes can be built to represent subsystems, services or local user applications.



The structure of this display is completely at the discretion of the user. The most common approach is to create either status or service views. Status views are Operations driven where groups of “like” things, such as disk, spoolers, Pathways, are assembled into the same class. A service class however, could contain a set of process icons, a set of file icons, a spooler device and a specific communications line, all of which support a particular application. Either approach is acceptable and it is possible to implement both and have the same icon in more than one class.

Icons can represent physical objects such as a communications line or a state such as “TMF Hung Transaction”.

Icons in the OverDrive display can flash either blue (vulnerable) or red (down). Vulnerable is meant to illustrate that a service is continuing but that it is threatened. Losing a disk path is an example of vulnerability.

All red and blue icons move to the top left of the display and they continue flashing until acknowledged by a Reflex 80:20 user or until the underlying fault is corrected. This feature counteracts a weakness in traditional text based Consoles where important messages can scroll off the top of your screen.

Acknowledging a fault alters the icons colour to yellow. When the icon is reset, either manually or by a nominated EMS event, the icon assumes its normal grey colour.

If a sound card is installed on your PC then WAV files can be executed. Standard Reflex 80:20 sound files are shipped, but you can install your own.

The conditions that affect an icon are defined in the Reflex 80:20 event database. Once you have created an event in that database and selected the Status Monitor option as the reaction for it you will be presented with a further screen that allows the user to say which icon this event should be mapped to.

For example, you can state that the icon selected should represent the subject or maybe the manager token contained within the EMS event. It could be a combination of both tokens as in this PATHWAY Server Class: \INSIDER.$PMA.REFLEX-ADMIN. (Manager = \INSIDER.$PMA, whereas the Subject = REFLEX-ADMIN). A range of other event buffer tokens used in icon mapping are supported and they are available in a drop down list from the event database screen.

Alternatively, the mapping can be hard coded and not rely on the content of the event at all; for example, map this event to an icon called SERVER-FAILURES.

The OverDrive GUI is equipped with an intuitive GUI builder facility. Users can list out objects of a specific type or type/subtype, for example all PATHWAY server classes, and then drag and drop selected items into an allocated position in the OverDrive tree. This object list can be built by hand, or automation tools designed and written by Reflex, per customer requirement. Reflex 80:20 is also equipped with a range of autodiscovery modules that will detect the names of hardware, networks, IP, X25, PATHWAY, Spooler, Disk objects for example, as well as third party environments such as Base24/XPNET and MQ.

An installation can also add its own application objects into this database either manually or by creating an autodiscovery module.

Once the graphical view is implemented and active it will act as a focus for error conditions, however, clicking on an icon provides access to a range of other facilities:

- A fault can be acknowledged, the icon turns yellow and a record is written to the history database for this object. The acknowledgement will be seen by other users who have the screen active on their PC and details of who acknowledged it can be viewed.
- The icon can be reset and it will return to an up or grey condition. This facility is useful if there is no EMS “up” event that will reset this object icon automatically.
- The object history buffer can be listed to your screen so that first level trending can be completed.
- The user can select a GUI option from the object icon to start or stop the object by executing a Reflex 80:20 Task. This is discussed in more detail in a later section.
- You can drill into the object to view configuration information. Hardware, process, file, PATHWAY, IP, X25, MultiBatch, Base24 and MQ configuration views are supported but an API is available to help plug your local application in at this level. This view is known as the Status Agent view.

Network Monitoring

The OverDrive Status Monitor view represents the status of a single NonStop node. This section discusses how Reflex 80:20 is implemented across a NonStop network.

Reflex 80:20 needs to be installed on each of the NonStop nodes that will be monitored. Each Reflex 80:20 system will connect to its own local Primary EMS collector ($0) and selected alternate collectors, and listen and process local events based on the local event database.

Automation and escalation to external agencies such as an Enterprise Manager will be the responsibility of each of the local Reflex 80:20 systems.

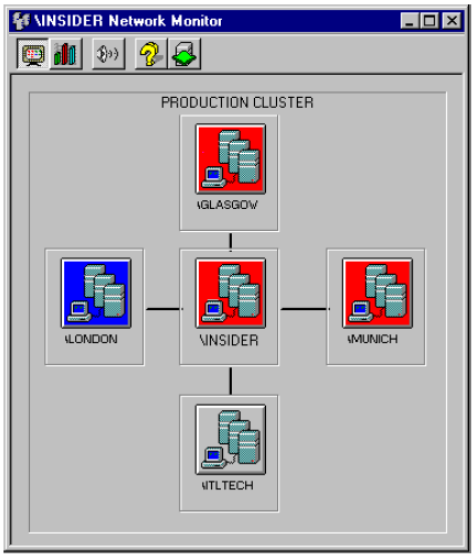
Each local Reflex 80:20 node will be responsible for maintaining a table of the status of the icons on display for its node. This will be updated as the Event Monitor passes new information to it as it arrives in the local EMS logs.

Outside of this, the Reflex 80:20 administrator needs to create a Network Hierarchy file that contains the names of all of the Reflex 80:20 nodes in the network. At least one of these nodes needs to be classed as the parent and typically it is the node that you have your Reflex 80:20 client connected to.

Each child node will then send regular status updates to the parent(s) stating that during the last poll period they have processed:

- no events, or
- at least one vulnerable event, or
- at least one down event.
- A fourth condition also exists, where the parent expects a message from a child and it never arrives.

The parent node has a state table for the nodes in the network and it is used to animate a display showing node icons. The icons will be shown as grey (no messages processed), red (at least one down), blue (at least one vulnerable) or orange (no child message received).



Clicking on the appropriate node icon brings up the detailed status monitor display for that node and the underlying fault can then be investigated further.

The interval that these child -> parent messages are sent is at the discretion of the Reflex 80:20 administrator. Even in a large multi-node environment therefore, network traffic is minimal but control from a single point is possible.

To provide a level of fault tolerance it is possible to configure your client program so that it has a list of alternative IP addresses and port numbers to connect to, in the event of a network failure. If an up to date network summary is available at this alternative address you will be able to view the latest status position.

EMS Event Text Console

As mentioned in the previous section, the Reflex 80:20 product is equipped with a graphical text viewing facility, called Console.

The product is based on the traditional principles of a rolling console such as ViewPoint but implemented with the benefits of a Windows application.

Users can display one or both of two event views:

- Critical events. These are events with the critical token set to TRUE in the event buffer.
- ALL events.

Critical and Action events are marked with red lights and can be acknowledged by the User.

Customisation of the event display is allowed:

- Compiled Filters or Filter Tables can be plugged into the display at two levels. You can restrict the information retrieved from the EMS logs by using an EMS Filter or Filter Table and then further restrict this information from being displayed on your Console. This latter facility is known as extended filtering.
- There can be a maximum of 16 different event views being assembled at any one time. A User selects their preferred view from a drop down list.
- Users can select the order that columns of information are displayed.
- Users can select most recent or oldest events at the top of the console display.
- Font and colour coding settings can be set on a per PC basis.

The Console view can be frozen and a user can then scroll around the display. When an event is selected from the Console display a number of attributes for that event can be marked, for example the subject and the event number, and Console will then on request display other recent events satisfying that search criteria.

Event Detail from the templates file or the Customised Event Detail file (EVENTCX) can be displayed for the event. This provides access to probable cause, effect and recovery data. Reflex 80:20 allows an EVENTCX database to be built from one of its GUI panels.

A technical token map can be displayed for the event. An example of this is shown in the EMS description.

One or more events can be copied to the Windows clipboard and made available to other Windows applications, such as Excel.

A nominated EMS event can be copied to the Reflex 80:20 event database where a User can then start to build Reflex 80:20 reactions for it.

A user can create and execute a search string based on any of the event tokens and values.

As well as building its own EMS filter, Reflex 80:20 also builds the negative filter which will show events that it does not process. Plugging this into a Console view and observing the results can help accelerate additions to the Reflex 80:20 event database.

Automation – TaskMaster

The automation engine of the Reflex 80:20 product is known as the TaskMaster.

The Reflex 80:20 Event Monitor can intercept EMS events that describe error conditions that can be corrected programmatically and without manual intervention and the details of this event can be forwarded to the Reflex 80:20 TaskMaster which will oversee the execution of the appropriate automated corrective action.

At some Reflex 80:20 installations, the TaskMaster is used to start up and close down applications based on a specific event being received from an external gateway, so this functionality can deal with more than just corrective action.

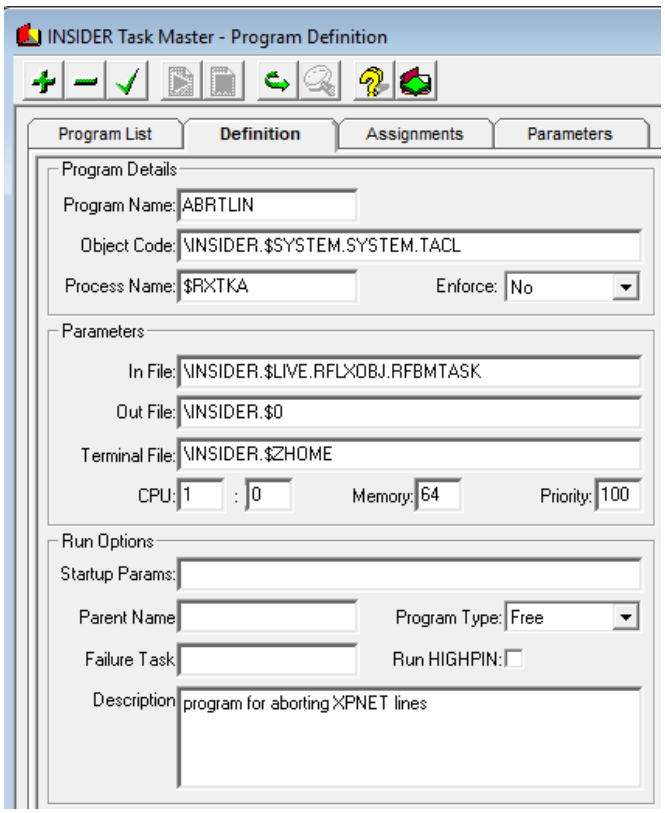
The TaskMaster interface to other subsystems is not based on using programmatic interfaces but on launching command line processes using low-level procedure calls.

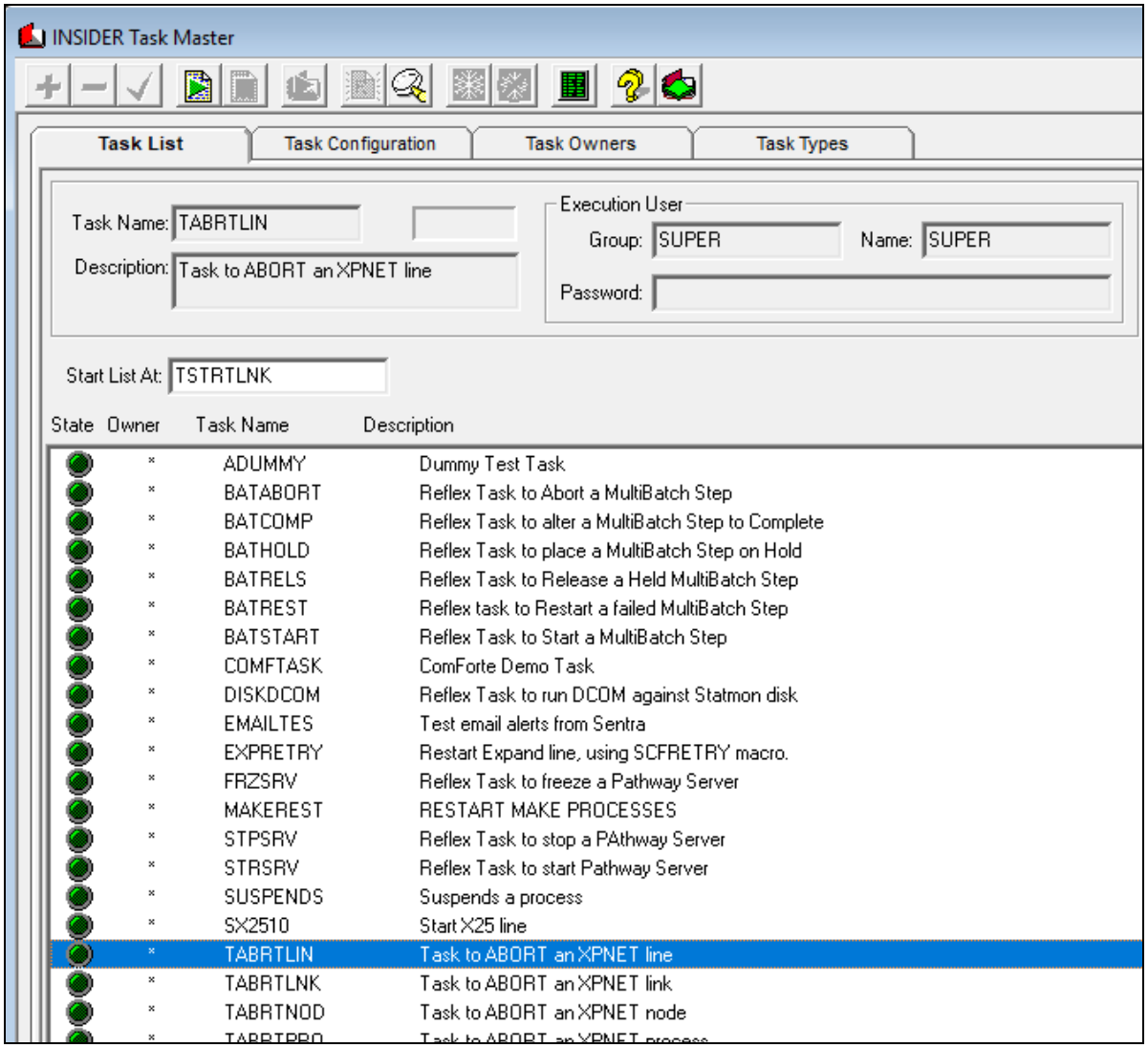
The command line instructions are defined in the TaskMaster “program” screen. Users have the option of specifying standard run options such as object code, IN, OUT, home terminal, CPU, memory, priority, start-up parameters, file assignments and PARAMs.

Users can also select the User Id that they want the program to run as.

Information from an event, such as the subject value, can be extracted and fed back into the command line. For example, the command SCF START ~subject~, could be used to restart a line automatically once an EMS event had been processed stating that the line was down. The subject value within the event, for example $X2510, would be extracted from the event and used in place of ~subject~.

Once programs have been defined, they can be linked together to form a Task if required.





Once built, the Task can then be assigned one or more owners by the Reflex 80:20 administrator.

If you are not an owner, then you cannot execute the Task. Executing a Task manually, updates the audit log with a record of who started which Task and when it was started.

Tasks can be executed a number of ways. They can be linked to an event in the Reflex 80:20 event database. If you select a Reflex 80:20 monitored event and then choose the Task option as your reaction, a further screen is displayed and you can select a Task from a pull down list. When this event next occurs, your appointed Task will execute.

As discussed in the OverDrive section, it is possible to execute Tasks from an icon button. Tasks can be mapped to object type and subtypes, so your Reflex 80:20 administrator can create a database that states that my start task for X25 lines is “STARTX25”. Stop Tasks and “other” Tasks can also be configured. When you click on the OverDrive icon you can select the start, stop or “other” option and the allocated Task for this type of object will execute.

Tasks can be executed manually from the Task configuration screen. This type of use can provide Operations staff with access to tested, secure and audited obey file scripts that they can execute as Users other than themselves.

Radio Paging and Email

One of the React choices that was highlighted earlier for nominated EMS events was the “inform somebody” option. For support staff who are off-site or for unattended overnight Operations, Reflex 80:20 provides a number of React Options:

- Mobile technologies such as radio pagers and telephones.
- Email

The earlier versions of the Reflex 80:20 product provided the radio paging functionality natively on the NonStop node. This involved dedicating a communications port on your NonStop system to this activity. Hayes compatible modems are still supported via this route along with a number of Radio Paging bureaus for which Insider has built customised interfaces. Further details of the supported Bureaus are available from Insider on request.

The standard TAPS protocol has also been implemented.

In keeping with requests from the user community, the current Reflex 80:20 architecture has been extended to provide the paging functionality through an outboard Windows Server. The Reflex 80:20 product is now shipped with a set of Windows Services that provide Paging and in addition, Email and Enterprise Management Gateway facilities.

Events that need to be routed to this set of Windows Services still need to be registered in the NonStop hosted Reflex 80:20 event database. Depending on your choice of escalation, further data entry screens are displayed that allow you to configure telephone numbers, email addresses and the event token fields that need to be despatched to your selected destination.

As a nominated EMS event is retrieved from the event logs by the Event Monitor it is despatched to a configured IP address and port number (a fall back address can also be configured) and the Reflex 80:20 Windows Service will then route the formatted details onward depending on the nature of the request.

- Email is provided by an Insider SMTP gateway running on the Windows Server.
- Paging, SMS and GSM access is provided by the Windows hosted ‘MessageMaster’ facility. This will be shipped and implemented as part of the Reflex 80:20 Windows Service installation.
- Insider supports Tivoli integration via either log file or TEC adaptor methods. Insider is a certified Tivoli partner.
- Reflex 80:20 supports integration with HPE Operations Center. Insider is a certified HPE Operations Center partner.

A final and more basic way of escalation is via an SNMP trap. Insider ships a MIB definition as part of the package. This can be loaded into an Enterprise Manager and it provides a richer trap definition than the native NonStop facility. Each EMS event has a unique trap number allocated and the traps variable bindings include subject, manager, criticality, description and any available cause effect and recovery information. This SNMP facility has been implemented on the NonStop node and does not require a Windows Server.

Reflex 80:20 Proactive Monitoring

Until this point, the White Paper has discussed how we analyse and nominate NonStop EMS events and then how Reflex 80:20 can process those events. This is the reactive face of the product.

For those instances where no EMS event occurs, Insider has implemented a set of facilities that will investigate potential issues and, based on simple thresholding rules, generate an EMS event that will drive Reflex 80:20's reactive facilities.

These background monitoring modules make up the Proactive Monitoring suite. It is imperative that as Reflex 80:20 is installed at sites, local monitoring macros that may have been executing for quite some time are reclassified as part of the Reflex 80:20 proactive monitoring environment. Any events that the macros generate can be registered in the Reflex 80:20 event database and this then makes them available to a wealth of reaction processing such as graphical consoles, automation engines and Enterprise Managers.

Some of the functionality that we are about to discuss here has been provided in the past using the NonStop Object Monitoring Facility (OMF) product. Conversion facilities are available that will create a Reflex 80:20 database from the contents of your existing OMF templates.

The existing Proactive Monitoring modules are for:

- Performance Metrics
- Processes
- File Metrics
- File Existence
- Specific Processes
- Spoolers
- Services
- TMF
- RDF

The events generated by this Reflex 80:20 software are already loaded into the Reflex 80:20 event database for you, although you can choose your own event numbers if you wish.

We will now look at some of these modules in more detail.

Proactive Performance Dashboard

The Reflex 80:20 Dashboard facility retrieves performance and capacity statistics from a variety of NonStop subsystems and processes them in a number of ways:

- The statistics are shown in a graphical form in a dedicated Dashboard display.
- The statistics are analysed and compared with pre-configured thresholds. If the thresholds are exceeded, then an EMS event is generated. If a value has returned below its threshold then a different EMS event is generated. This pair of “up” and “vulnerable” events can be used to animate an OverDrive Status Monitor icon and force the icon to turn blue (vulnerable) and grey (up).

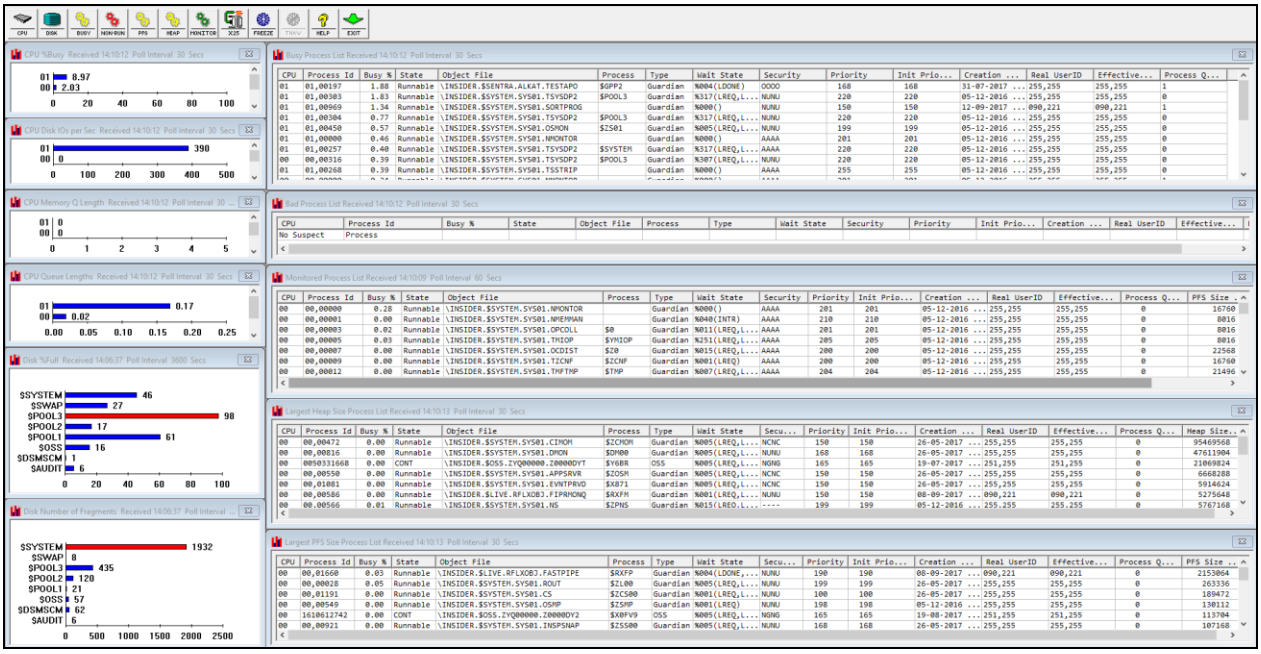
As with the Status Monitor facility, the Dashboard is installed on each Reflex 80:20 node and it retrieves performance statistics for that local node only.

Low level procedure calls are used to retrieve the latest set of statistics. The Reflex on-demand Measure facility can be used to retrieve attributes such as file reads, writes and locks that cannot be obtained by procedure calls.

You can choose to switch this facility off if there is concern about processing overheads and the poll period for each different type of collection can be set by the Reflex 80:20 administrator to help manage the impact on your system.

Newly collected sets of statistics are despatched to a threshold engine where the data is compared against values set by the Reflex 80:20 administrator. When thresholds are exceeded, exception alerts are created in the local EMS event log, the events can then be transferred to the top-level Network Monitor as described in an earlier section of this paper.

When the network node icon flashes, the Reflex 80:20 User has the option of selecting the OverDrive icon based status display or the Dashboard graph based performance display for that node.



A full list of the collected dashboard attributes is available in the product technical guide.

Proactive Process Monitoring

The next three proactive modules: process monitor, file metrics monitor and file existence monitor are known collectively as the Reflex 80:20 Heartbeat.

The Process Monitoring facility will track a list of nominated processes and raise an EMS event if the process does not exist. The module will then create a separate event once the process reappears. If a process terminates before it has issued a diagnostic message then this occurrence will be trapped by the Process Monitor software.

This pair of “up” and “down” events can be used to animate an OverDrive Status Monitor icon representing the process and force the icon to turn red (down) and grey (up). If the icon is positioned in a Service class as discussed in the OverDrive section then the impact on the application of losing this process can be assessed at a glance.

A range of process characteristics are also monitored and if any of them are exceeded a further event can be created. The process is still running but is classed as under threat so this is a “vulnerable” condition and consequently we can use this event to create a blue (vulnerable) OverDrive icon in the Status Monitor.

The attributes reviewed for a process are:

- Busy
- Memory pages
- Global Data Size
- Main Stack Size
- Native Heap Size
- % CPU Time
- Received Queue Length
- Process running time
- Process execution time

To restrict the number of messages created by this function you can monitor processes within a specified time period. This time period can be altered to suit the varying availability of your application and this is achieved by running a Reflex 80:20 batch utility to switch monitoring on and off for a group of processes. Some installations have automated the execution of this utility via the Taskmaster which is driven by an event created when the application closes down.

Proactive File Metrics Monitoring

The File Metrics facility will track a list of nominated disk files. It will create an EMS event if the file does not exist, a separate event if the nominated attributes of the file exceed preconfigured settings and a further event once the failed attributes drop back to an acceptable level.

This set of “down”, “vulnerable” and “up” events can be used to animate an OverDrive Status Monitor icon representing the file and force the icon to turn red (down), blue (vulnerable) and grey (up). If the icon is positioned in a Service class as discussed in the OverDrive section then the impact on the application of this file issue can be assessed at a glance.

The attributes monitored for a file are;

- % full
- % growth
- Index levels
- Number of records
- Extents remaining
- RWEP security and owner
- Timestamp today
- SQL valid
- ProgID
- Broken, corrupt and Crash Open
- Licensed/not licensed
- Audited/not audited
- Open/not open
- Empty/not empty
- Saveabend checks
- Subvolume size

To restrict the number of messages created by this function the Reflex 80:20 administrator can set the monitoring period depending on the criticality of the file. A file can be checked daily, every twelve hours, every hour or it can be classed as critical whereby the User specifies the poll period in minutes.

Proactive File Existence Monitoring

The File Existence facility will also track a list of nominated files. It will create an EMS event when the file is created and a separate event if the file does not exist. You can tie this monitoring to the Reflex 80:20 calendaring facilities so that checking is restricted to specific periods of the day.

Although this set of “up” and “down” events can be used to animate an OverDrive Status Monitor icon, this functionality is more likely to be connected to the automation TaskMaster facility so that when a file is delivered to your NonStop node from a remote platform a Task can be executed to process this file.

The attributes monitored for a file are:

- Timestamp today
- File is Empty
- File is Open
- Corrupt
- Broken
- RWEP security and owner
- File Age
- Modification Age
- Specific file
- First file with prefix
- All files with prefix

Spooler Monitoring

The Spooler Monitoring facility will track a list of nominated spooler supervisors and their components. Error conditions can be classed as “down”, for example, a print device is off line or “vulnerable”, or the collector data file is more than x% full. A further event will be created once the failed attributes return to an acceptable level.

This set of “up”, “down” and “vulnerable” events can be used to animate an OverDrive Status Monitor icon representing the affected spooler object and force the icon to turn red (down), blue (vulnerable) and grey (up). If the icon is positioned in a Service class as discussed in the OverDrive section of this paper then the impact on the application of this spooler issue can be assessed at a glance.

The spooler attributes that this module monitors are:

- Collector data file % full
- Devices on line
- Print processes running
- Collector process running
- Supervisor process running
- Number of print jobs in spooler

The Reflex 80:20 product is equipped with a spooler autodiscovery module that will create an object database and optionally, a graphical tree for the OverDrive status display. This object list and the tree will consist of spooler supervisors, collectors, print processes and print devices.

Application Monitoring

All of the proactive facilities that we have described so far vary little in concept from installation to installation.

CPU, process, file and spooler monitoring present the same challenge at most installations. Reflex 80:20 is equipped with autodiscovery tools that will analyse your NonStop node and create Reflex 80:20 database entries to reflect your local naming standards and the local hardware configuration.

Bulk configuration tools exist within the product to turn edit files containing the names of a large number of files and process names into Reflex 80:20 monitoring database entries.

The EMS events that the Reflex 80:20 proactive monitoring software generates are preconfigured in the Reflex 80:20 event database for you.

Once you have reviewed the threshold settings in the parameter database to your satisfaction, you can turn the Reflex 80:20 monitoring software on and any exception conditions will appear in the EMS event logs. These alerts can be forwarded to dedicated OverDrive icons. All of this can be achieved in a very short time.

Following on from this, attention must then be given to monitoring your application exception conditions.

If your application already creates EMS events or you have implemented macros to provide application health checking, then this paper has demonstrated in other sections, that processing your EMS events and creating local objects, graphical views, drill down configuration screens and automation rules is all feasible using the products standard features.

The remainder of the proactive section looks at the Reflex 80:20 options if your application is not instrumented this way.

Service Monitoring

Some of the applications that Insider has encountered write status information to a dedicated terminal where it needs to be monitored constantly, or the data is maybe written to a disk file where a report program retrieves it and displays it for review. A classic case of this type of monitoring is within a queue manager which could be processing electronic mail or financial payments.

To help automate the monitoring of this type of data, Insider has developed the Reflex 80:20 Service Monitor.

At the heart of this facility is the Service rules engine. Within the Reflex 80:20 GUI Users can build rules that automatically test the values of your data as it changes. For example;

```
IF QUEUE > 20 THEN "service default"
```

More sophistication can be built into the rule if required:

```
IF QUEUE1 > 10 AND IF QUEUE2 > 20 THEN "service default" or

IF ONE OF QUEUE1 OR QUEUE2 OR QUEUE3 > 20 THEN "service default"
```

You can also apply calendars if you require this:

```
IF QUEUE1 > 20 AND PEAK-TIME THEN "service default"
```

The rule can process different types of data if required:

```
IF QUEUE1 > 20 AND (FILE = "$D.QSV.QFILE" AND FILE.FULL > 95%) THEN
"service default"
```

Once the rule has been broken you can animate an icon on a dedicated Service Monitor display or you can create an EMS event which as always can be processed by the Reflex 80:20 Reaction modules.

This latter EMS event provides a very powerful facility as it allows the Reflex 80:20 software to alert only when a predefined set of previously unrelated conditions happen together.

How does the Service Monitor receive this application data and how does it know what it is when it arrives? The answer is that the installation needs to write a software module to extract the data from its current location and write it to the Reflex 80:20 Service Monitor facility.

Within the Service Monitor, you will also need to register the structure of your application information. This is a once only task that requires you to map out the structure of your data buffer stating the order, number, type (strings, numbers or binaries?) and lengths of the individual fields.

This may seem like more work, but the benefits are that your status information will be constantly reviewed for you and the alerts generated can be routed to centralised manned consoles and mobile alert technologies for attention.

Gateway

For those applications that are generating good quality diagnostics but as text rather than in the proprietary EMS format, then there is also a Reflex 80:20 solution for this problem.

The issue with text messages is that if they are written directly to EMS unformatted instead of as a series of tokens as described in our EMS overview, then EMS recognises this and creates a default event on behalf of the originating process with identical token settings.

The event buffer includes the same event number; TANDEM.12.0 event 512 and no subject. This makes filtering this information at best an expensive processing exercise and sometimes it can be impossible.

Instead of sending this diagnostic information directly to EMS, the Reflex 80:20 Gateway facility can take receipt of the text messages and based on some preconfigured rules generate a fully tokenised EMS event in a nominated EMS collector instead. This event can then be routed to the Reflex 80:20 Reaction engines.

Consider the following simple example text messages:

> Bank Of Insider: Interface $BOI25.#TERM1 ABORTED 140
>
> Bank Of Insider: Interface $BOI25.#TERM1 RESTARTED
>
> Bank Of Insider: Interface $BOI25.#TERM1 CONNECTED

We can create three different rules for this data.

Rule 1 – If the characters “Bank Of Insider: Interface” appear in positions 1 to 26 and the characters “ABORTED” appear in positions 42 to 48 then generate the event INSIDER.99.0 140 with the text as it is and a subject of whatever is in positions 28 to 40, i.e. $BOI25.#TERM1. This would process message 1.

Rule 2 – If the characters “Bank Of Insider: Interface” appear in positions 1 to 26 and the characters “RESTARTED” appear in positions 42 to 50 then generate the event INSIDER.99.0 999 with the text as it is and a subject of whatever is in positions 28 to 40, i.e. $BOI25.#TERM1. This would process message 2.

Rule 3 – If the characters “Bank Of Insider: Interface” appear in positions 1 to 26 and the characters “CONNECTED” appear in positions 42 to 50 then ignore this text message. This will ignore message 3, it is information and requires no further reaction processing.

If it is required, you can edit the text, deleting redundant information such as date and time and adding new data such as severity or the name of the originating process.

The text to EMS translation rules are built within the Reflex 80:20 GUI and require no programming expertise. If text messages arrive in the Gateway and there is no rule built, then an alert is generated and you can create and implement a rule for the new text without closing down the software.

The two new events (INSIDER.99.0 140 and INSIDER.99.0 999) are registered in the Reflex 80:20 database for you as part of the translation exercise and they can now be made available to all the Reflex 80:20 reaction modules.

As well as processing text from applications that are not instrumented for EMS there are two other common uses of the Gateway.

Applications sometimes generate EMS events of poor quality. They may include the subject of the event in the body of the description but not as a separate subject token. This means that if an Operations Management product wants to process this event and ascertain the name of the device affected by this error, then it has to parse the text. In this respect it is only slightly better than the original text message.

To illustrate this we can return to our earlier example:

If the message “Bank Of Insider: Interface $BOI25.#TERM1 ABORTED 140” was being raised as event INSIDER.99.0 140 rather than as text by the application and it had no subject, then we can use a print distributor to extract the event from EMS and send the template formatted text of this event to a Gateway process…

EMSDIST TYPE P, COLLECTOR $0, FILTER IN99140F, TEXTOUT $RFGW

…where it can be translated into a new event with a subject and then resubmitted into the Event Management System.

Careful use of this technique can significantly improve application management.

The other use of Gateway is to provide immediate EMS integration for those applications ported to the NonStop platform and which will run under the OSS personality. If the OSS process can be altered to write its log output to a nominated Guardian process, i.e. the Reflex 80:20 Gateway, then text to EMS rules can be built for these UNIX application text messages.

If the OSS application can only write to SYSLOG, then the EMS events generated by the OSS SYSLOG (TANDEM.143.0) can be retrieved, translated and resubmitted using the print distributor method described earlier.

Other Gateways

There have been instances in the past of Reflex 80:20 being used as an Enterprise Manager. The rationale has been that as the hardware is the most “available” technology in the network then it is a good place to site a management platform.

To meet this requirement, Reflex 80:20 is equipped with a variety of other gateways that will allow inbound SNMP traps and text messages sent via Async and Telnet linked devices to be routed to EMS and then on to Reflex 80:20 React engines.

Further information about these modules is available on request from Insider Technologies.

Miscellaneous Items

This section briefly describes those Reflex 80:20 features that do not fit into either the infrastructure, proactive or reactive categories of the product.

Discovery

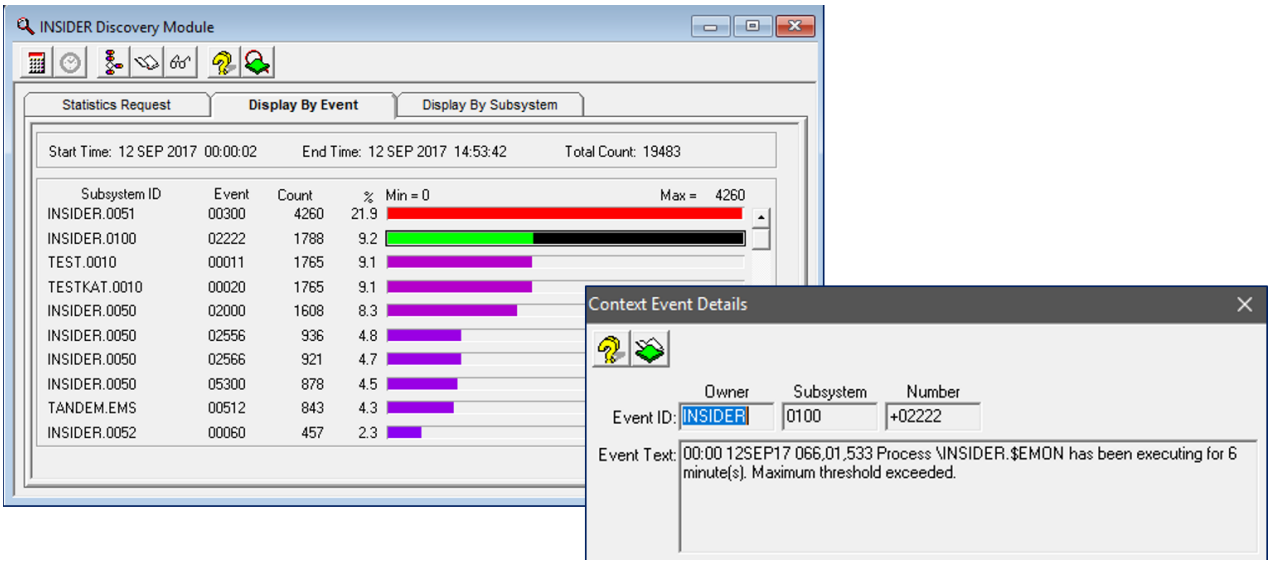
Reflex 80:20 contains a module, called Discovery, which will analyse your EMS event logs and produce results that can be viewed as either a printed report or as a histogram in the Reflex 80:20 GUI.

You can specify time periods and the name of an EMS filter as parameters to this exercise.

From the GUI histogram you can transfer a nominated event into the Reflex 80:20 database where you can start to build Reflex 80:20 reactions for it.

The Discovery facility is useful if you are attempting to build an EMS filter for your Operations Bridge and need to see which messages to suppress first. You can then subsequently run Discovery with your Operations filter attached to see which batch of event messages to suppress next. Using this technique you can usually suppress in excess of 95% of your EMS output in the course of a few days.

Conversely, you can use this facility to ensure that an event has not occurred. This may be useful if you want to check that a recurring hardware fault has been corrected.



Reporting

The RFLXCOM TACL utility will enable Users to perform some of the GUI based maintenance tasks from the command line or from a Batch scheduler. These tasks include table archiving, filter compilation and “warmbooting” the software.

Maintenance

The Reflex 80:20 product is equipped with a variety of reporting tools to help list out your database tables and ease maintenance.

Migration

Any React rules that are built in one database can be exported to an intermediate transit file which can then be copied to another node and imported into another Reflex 80:20 database. In this way the standard develop, test and release cycle can be followed.

Disaster Recovery

Any React rules that are built in one database can be exported to an intermediate transit file which can then be copied to another node and imported into another Reflex 80:20 database. In this way the standard develop, test and release cycle can be followed.

Technical Details

Reflex 80:20 Software Modules

The Reflex 80:20 product is comprised of a PATHWAY environment that is used to configure and administer the Reflex 80:20 database and it also provides persistence for some processes that have been configured as Server Classes with the “Autorestart” function set.

The key monitoring modules have been implemented as a set of NonStop process pairs.

Both Enscribe and NonStop SQL databases are used.

Insider Technologies is part of the HPE run time SQL license program, so if necessary, prospective customers do not need to purchase a full SQL license to run this product.

The GUI is a Windows User Interface application. The client/server connectivity is provided by Insider’s own FastPipe product which is based on IP sockets.

An installation macro can be used to deploy and load the Reflex 80:20 NonStop software.

Hardware and Operating System specifications for the NonStop and PC resident software are available on request from the Insider Help Desk at **support@insidertech.co.uk.**

Performance of any Management application is always an area of concern. It is difficult to quantify exactly the load that Reflex 80:20 will place on your system. The number of events that your application writes to EMS is a factor, the number of events that you have selected for monitoring needs to be considered as does the poll periods set for the proactive monitoring, the number of processes and files you build into the monitoring database and the number of concurrent Console sessions executing. As a general rule, the Reflex 80:20 software will occupy less than 1% of each CPU on your NonStop node.

Summary

Reflex 80:20 provides a single solution for your NonStop Management needs.

- The product can correct faults automatically, using the context of the original EMS event if required.
- Escalation of alert conditions can be to a local graphical console, OverDrive, or to “mobile” technologies.
- You can use Reflex 80:20 to integrate your NonStop network with your installations choice of Enterprise Manager.
- Legacy applications can use Reflex 80:20 gateways to ensure that they participate in your centralised Management strategy.

Although recognising that managing the performance of your NonStop node is a crucial part of delivering a Service, Insider also recognises that managing the infrastructure that the application relies upon and analysing and interpreting the data that the application generates is equally crucial to Service Delivery.

All three aims can be achieved by using the Reflex 80:20 product.

For further details about Reflex 80:20 or any other Insider products please contact the Product Development Centre on telephone number +44 161 876 6606 or at Email address support@insidertech.co.uk.

Insider Technologies is a UK-based software and services company, operating in the Financial and Messaging markets.

It provides Service Management, Tracking, Bespoke Software and Information Mediation solutions.

A cross section of our customers include Banking and Financial Services, Telecommunications Providers and Government and Military Institutions.

For details about the full range of products and services available from Insider Technologies Limited, please contact our Product Development Centre at:

Insider Technologies Limited
2 City Approach
Albert Street
Eccles
Manchester
M30 0BL
United Kingdom

Tel: +44 (0)161 876 6606
Fax: +44 (0)161 868 6666

e-mail: support@insidertech.co.uk
Website: http://www.insidertech.co.uk

Microsoft Partner
Gold Application Development