



Messenger—Technical Packet

**BYTWARE, INC.**

*Monitoring & Securing a Connected World.™*



## **MessengerConsole®/MessengerPlus® Technical Packet**

Revision September 2012

MessengerConsole® and MessengerPlus® are registered trademarks of Bytware.

IBM®, Power Systems®, and IBM i® are registered trademarks of International Business Machines.

Other brand and product names are trademarks or registered trademarks of their respective holders.

© 2012 Bytware. All Rights Reserved.



## Table of Contents

<b><u>I. Messenger Introduction</u></b>	<b><u>5</u></b>	<b><u>III. Events &amp; Events Console</u></b>	<b><u>32</u></b>
Solution Overview	6	Working with Event Queues	34
Justification & Benefits	7	Event Console	34
<b><u>II. Monitors</u></b>	<b><u>13</u></b>	<b><u>IV. Actions</u></b>	<b><u>35</u></b>
Pre-configured Monitors	16	Action Scripting	36
Message Queue Monitoring	17	Action Escalation	37
Job Monitoring	20	Action Scheduling	38
Messages from job logs and S/36 procedures	20	Action Shortcuts	38
Program error messages from specified interactive jobs	20	Working with Actions	38
System History Log (DSPLOG QHST)	21	Reply Management	39
Job Status Monitoring (WRKACTJOB)	21		
Total CPU Usage Monitoring (WRKSYSSTS)	23	<b><u>V. Paging &amp; Notification</u></b>	<b><u>40</u></b>
Job Queue Monitoring (WRKJOBQ)	23	About Paging	41
System Audit Journal Monitoring (QAUDJRN)	24	Supported Notification Types	41
Output Queue Monitoring (WRKOUTQ)	25	Paging Companies	44
Line Status Monitoring (WRKCFGSTS)	26	Message Formatting	45
Ping Monitoring	26	Escalation	45
Disk Monitoring	27	Scheduling	46
Restricted State Monitoring	28	Restricted State Paging	46
Creating Custom Monitors	28	Shares with Service Director/ECS/Fax Line & Modem	46
Scheduling Monitors	29	Modem Pooling	46
Securing Monitors	29	Pager Security	47
Define additional parameters for each Monitor	30	Add Paging to Your Programs	47
Working with Monitors	30	Network Paging/Message Dispatching	47
Working with Event Monitors	31	Automatic Clean-up	48



## Table of Contents continued


<b><u>VI. Reporting</u></b>	<b><u>49</u></b>
Event History	50
Action History	50
Paging History	50
QHST Analyzer	51
Usage Tracking	51
Monitor Configuration Reporting	51
<b><u>VII. Consoling</u></b>	<b><u>52</u></b>
Introduction to MessengerConsole	53
Benefits	54
Features	55
The Console	56
The Alternate Console	56
Distribution Lists	57
Monitor Distribution	58
Escalation	59
Actions	59
Integration	59
How it Works	60
<b>Appendix. Interactive Downtime Calculator</b>	<b>61</b>

# SECTION I. MESSENGER INTRODUCTION




## Messenger Introduction


**MessengerConsole and MessengerPlus provide advanced consoling, monitoring, and notification features.**

Since 1986, Bytware has maintained innovative leadership with the most asked-for consoling, systems monitoring and paging notification solutions: MessengerConsole and MessengerPlus. All of the features outlined in this Technical Packet are available in MessengerConsole. MessengerPlus is a slimmed-down version of MessengerConsole. Features not available in MessengerPlus will be identified with the MessengerConsole-only icon .

### Messenger Overview

Ideal for 24/7 systems management, Messenger is a fully automated monitoring, message management, paging/notification and reporting solution that combines monitoring of messages, CPU usage, disk space, WRKACTJOB, WRKJOBQ, WRKOUTQ, WRKCFGSTS, TCP/IP, QAUDJRN, and more, with flexible action escalation to deliver the best 24/7 protection for your System i.

Messenger performs pre-defined actions to automate response to reoccurring events such as resetting failed communication lines, incrementing full files, etc., and can be configured to send replies, run commands, send notifications/e-mail messages, and copy events to other users and systems . Reporting features collect usage information that is critical for statistical management.

Available only with MessengerConsole are the easy-to-implement centralized consoling features designed to provide centralized control of activities across local and remote systems, as well as across multiple partitions. From a single point of control, MessengerConsole lets you streamline monitoring, message management, reply automation, paging/e-mail notification, and statistical reporting for multiple systems and partitions which in-turn helps manage your System i in a more efficient and cost-effective way. .

When you buy a Bytware Messenger product you are buying an award winning solution that carries two decades of in-the-field, proven power and reliability that continues to evolve and grow with your company and the System i environment. We look forward to working with you.



## Justification & Benefits:

Bytware's Messenger products are powerful monitoring solutions with a focus to provide intuitive system monitoring and notification features that are easy-to-use, without requiring extensive training and set up.

*Do you need to know...*

- When a time critical job does not complete on time?
- When backups fail?
- When security policies are breached?
- When there is a delay in your high availability mirroring?
- When the system printer is waiting on a message?
- When important communication lines are down?
- When a batch job has been running for too long?
- When an inquiry message has not been answered within a reasonable time?
- When any of the above occur on a remote or mirrored system or partitions? ☒

*Bytware has a Messenger solution that will fit your needs.*

*"MessengerConsole caught the error right away. It notified me there was problem, and identified the job. In terms of the staff time it saved, MessengerConsole paid for itself within the first month."*

*— D. Sowers, Toyota Motor Sales*

*"MessengerPlus has made us more aware of what's happening on our system. By routing messages, we know when system and application errors occur before they become a serious problem. This is a survival product!"*

*— G. Scarvey, Jimmy Dean Foods*



## Justification & Benefits continued:

*Is systems management slipping out of control?*

- Are you able to manage the increased demands placed upon yourself, your servers, and operations staff?
- Is systems management becoming an endless cycle of sleepless nights, server babysitting, and unexpected or lengthy downtime?
- Is staff turnover, sick leave, vacations, etc. leaving you short handed?

*“MessengerPlus has been a savior. The IS department can leave the building at any time because they know the system is being monitored. The software package was very easy to implement and even easier to maintain and adjust. MessengerPlus is definitely one of our department’s best investments.”*

— Roy Cox, FTI

*Size does not matter.*

- It doesn’t matter how big your company is; or how small it is. An iSeries, AS/400, or System i generates and logs thousands of messages in a day to dozens of message queues, many vital to the health and status of the system and business operations. Manually monitoring these queues is impractical, especially when this process can be easily automated.
- Messenger products can efficiently be utilized by all sizes of companies, especially organizations with multiple and partitioned systems, and high systems availability concerns.





## Justification & Benefits continued:

*Reduce the high cost of downtime.*

- Automate monitoring.
- You specify what is monitored and when procedures are performed; send notifications with job logs or instructions attached, send replies, run commands.
- Copy events to other users or systems. ☒
- Automate routine messages that halt processing.

*Reduce the high cost of managing multiple systems with MessengerConsole.*

- Reduces the time and effort to monitor and diagnose problems by providing a consolidated view of the events from all your remote systems. ☒
- Improves the ability for a local operator to be notified and respond to remote events. Remote inquiry messages can be answered easily, without the need to manually sign on to the remote system. ☒
- Reduces the cost, setup and administration of remote monitoring by dispatching notifications from the central system. ☒

*“With today’s lean and mean operations we can’t afford to have people assigned to do console monitoring around the clock.”*

*— J. Huey, DFS Group Limited*



## Justification & Benefits continued:

*Improve operations  
productivity and  
systems management.*

- Automate procedures for reoccurring events, such as automatically resetting failed communication lines, increment full files, etc.
- Determine frequency of recurring problems by displaying the number of times an event has occurred over a period of time, such as the number of power outages, security breaches, program crashes, etc. Usage information is excellent for statistical management.
- Automatically route messages to the appropriate people. Route program errors to the HelpDesk or Programming staff, security errors to the security officer, etc.
- Eliminate the worry, hassle and necessity of evening and weekend call-ins once and for all! Simply let Messenger notify you when system events occur.

*“MessengerPlus is completely running our system. We never anticipated it to be so versatile. It is fantastic and everyone loves it. The time and money saved is unreal!”*

*— S. Fangman, Novartis Pharmaceuticals*



## Justification & Benefits continued:

### *Improve system response time.*

- Monitor the Work with Active Jobs display for jobs consuming too many CPU resources and automatically notify an operator, or put the job on hold.
- Monitor for jobs with poor response times and reduce their priority automatically.
- Monitor for jobs with too low a priority and increase their priority automatically.

### *Improve security and stay on top of important events.*

- Be notified automatically when a security breach occurs, such as a user signing on after hours or accessing unauthorized areas of the system.
- Be reminded of appointments and calendared events.
- Be alerted automatically about power failures and other critical events.
- Be paged if backups or other critical jobs do not start and/or complete on time.

*“We installed MessengerPlus to watch over our casino floor for jackpots, machine malfunctions and high rollers. Customer service and response time has never been better.”*

*— A. Metzger, Harrah’s Casino*



## Justification & Benefits continued:

### *Improve company-wide communications.*

- Graphical paging interface for all network users.
- Give your users the ability to page mobile workers, such as outside sales staff, doctors, delivery drivers, etc.
- Multi-user, Multi-line paging features are perfect for dispatch centers, such as fire and police, security, transportation, delivery, etc.

### *Know that you've made the right choice for your company.*

- Bytware innovated systems monitoring/paging in 1988.
- Thousands of customers globally.
- The #1 most asked for monitoring and paging solutions.
- Bytware devotes 100% of research, development and technical support manpower to providing new, innovative and cost effective means to manage system events.
- Bytware's Messenger products are consistently preferred by industry experts and chosen for easy-to-use flexibility, unsurpassed reliability and comprehensive features.
- Bytware provides 24/7, toll free technical support.

## SECTION II. MONITORS



## Monitors

System i can generate and log thousands of events in a 24 hour period to dozens of message queues, with many of these events vital to the health and status of the system and business operations. Missing any number of critical events such as a job failure or impending DASD failure could cost your company thousands of dollars in lost revenue as a result of computing and user downtime.

Ideal for 24/7 systems management, Messenger is a fully automated monitoring, message management, paging/notification and reporting solution that combines monitoring of messages, CPU usage, disk space, WRKACTJOB, WRKJOBQ, WRKOUTQ, WRKCFGSTS, TCP/IP, QAUDJRN, and more, with flexible action escalation to deliver the best 24/7 protection for your System i.

How can you automate message monitoring and feel comfortable in knowing you are getting only the most important messages? Implement Bytware's Messenger products to automatically detect and filter software and hardware events.

The flexibility of Messenger allows you to schedule monitors, define additional parameters for each monitor, define the monitoring parameters for your system, and specify events you want to monitor for.

### Monitors include:

- Pre-configured monitors.
- Message queue monitoring. Monitors and filters messages and events from any number of messages queues, including QSYSOPR and QSYSMSG by a wide variety of selection criteria:

- ◇ Message ID, Type and/or Severity
- ◇ Job name, or generic job name
- ◇ User, or generic user name
- ◇ Program, or generic program name



- ◇ Accounting code, or generic accounting code
- ◇ Message text and substitution variables
- ◇ Time periods
- Job monitoring for jobs that do not start or complete by a specific time.
- Messages from job logs and S/36 procedures.
- Program error messages from specified interactive jobs.
- System History Log (DSPLOG QHST)
- Job status monitoring (WRKACTJOB):
  - ◇ Jobs consuming excessive CPU resources
  - ◇ Jobs exceeding a specified length of time
  - ◇ Jobs that must be running during specified times
  - ◇ Jobs with poor response times
  - ◇ Jobs with too low priority
  - ◇ Jobs with a specific status (e.g. LCKW or MSGW)
  - ◇ Jobs with specific functions, ie. S/36 procedure abends; a user is running a specific program
- Total CPU usage monitoring (WRKSYSSTS)
- Job queue monitoring (WRKJOBQ)
- System Audit Journal monitoring (QAUDJRN)
- Output queue monitoring (WRKOUTQ)
- Line status monitoring (WRKCFGSTS)
- PING monitoring for TCP/IP device status: routers, firewalls, Windows servers, Netware servers, etc.
- Disk monitoring to detect common DASD storage problems:
  - ◇ Remaining disk space
  - ◇ Growth over a period of hours, days, weeks or months
- Restricted State Monitoring
- Custom: create your own monitors



## Pre-configured Monitors

Bytware leverages its extensive expertise in systems monitoring to pre-configure Messenger to automatically detect many types of common events and critical messages. These events can be easily changed, removed, or added to the monitoring list.

- Automatically detects many types of common events and critical messages.
- QSYSOPR, QSYSMSG and QHST pre-configured for easy set-up.
- Pre-configured schedules for unattended, attended, weekdays, week nights, and weekends.
- Pre-configured to automatically page on critical events.

### Pre-configured monitors include:

- RPG, CL and COBOL batch error messages.
- Backups not completed by a specified time.
- Jobs that end abnormally.
- Tape related messages.
- Printer related messages.
- Disk and hardware related messages.
- Communications error messages.
- Security related messages.
- All messages severity 80 and above, excluding printers.
- File Full messages.
- BRMS error messages.
- Changes to system values.
- Power failures and Uninterruptable Power Supply (UPS) related messages.
- Any job waiting on a message.
- Interactive jobs using over 30% CPU. This event monitor detects programs that may be “looping.”
- Batch jobs running longer than four hours.
- Batch jobs using over 50% CPU.
- Interactive response times over 30 seconds.
- Interactive run priority under 20.
- Batch run priority under 30.
- Jobs using more than 3600 seconds on CPU.
- Job queue QBATCH with 10 or more jobs.
- Output queue PRT01 with 50 or more spooled files.
- Over 20 different types of security and system related events from the i5/OS audit journal.





## Message Queue Monitoring

Messenger can monitor any number of message queues including QSYSOPR and QSYSMSG. It offers the most extensive user configurable filters on all monitors and can filter messages and events from any number of messages queues.

Do you need to know when a batch job halts due to an error? Messenger can filter messages from any number of messages queues using Message Filters. Message Filters allow you to easily customize the specific sets of messages you want to monitor by including or excluding messages by the following:

- Message ID
- Message type
- Severity
- Job name, or generic job name
- User, or generic user name
- Program, or generic program name
- Accounting code, or generic accounting code
- Message text and substitution variables (compare data)
- Time periods

You can easily create Message Filters to accommodate situations such as all severity 99 messages excluding printers. Messenger is pre-configured with many common Message Filter Sets that you can use immediately or customize to meet your specific needs.

**Message Filters can be combined**, so you can monitor for situations such as all severity 99 messages excluding printer alignment, etc. You can easily monitor your personal message queue and redirect messages to your notifications or e-mail to see messages from others when you are not signed on. You can also be notified when files and Office documents sent from other users or systems have arrived in your mailbox.



## **Message Queue Options**

The Message Queue Options display is shown when you are changing a monitor for type \*MSG. The following options allow you to define parameters for monitoring of a message queue.

### **Message queue and library**

The name of the message queue and library to monitor.

### **Severity filter**

The minimum severity of messages to filter from the queue. Messages with severities below the specified severity will not be monitored, even if they are specifically included. It is best to leave this parameter zero, unless you are sure you do not want any messages below the specified severity level. A slight performance improvement can be realized on message queues with many low severity messages (such as QSYSOPR). This is because messages below the specified severity are not read from the message queue and not filtered. You can filter messages by severity using the Work with Message Filters display.

### **Message delay**

Specifies the number of minutes messages on the queue must wait before being monitored. This allows time for a local operator to handle a message before Messenger will begin monitoring the message. If the message is inquiry, Messenger will not monitor the message if the message is answered within the specified time period by a local operator. If the message is informational, Messenger will not monitor the message if the message is removed within the specified time period.

### **Read back limit**

Specifies the number of minutes prior to starting Messenger that you want messages monitored. Messages that have already been monitored will not be monitored again (not valid for QHST).

### **Lock message queue**

Specifies if you want Messenger to lock the message queue to prevent users from answering messages on the message queue, bypassing Messenger's security and auditing controls.



## Working with Message Filters

The Work with Message Filters display allows you to define the specific messages you want to include and exclude from monitoring. Message filters allow you to include and exclude the messages you want by any combination of:

- Message ID
- Message type
- Severity
- Job name, or generic job name
- User, or generic user name
- Program, or generic program name
- Accounting code, or generic accounting code
- Compare data, or generic compare data, such as substitution variables or scan strings of text

**Message Filters can be combined**, so you can monitor for situations such as all severity 99 messages excluding printer alignment, etc. Using the Work with Message Filters display, you can:

- Add new messages to monitor, or exclude from monitoring.
- Change the parameters of a message filter, such as the message ID, job name, etc.
- Remove messages from monitoring.

You can create as many message filters as needed to define the event you are monitoring. Each message filter has several parameters. Specifying more than one parameter for a message filter are ANDed together: all of the parameters specified on a filter must be met. Specifying more than one message filter are ORed together: any of the message filters matched will log the event.

Many of the message filter parameters allow generic values. Generic values allow you to match a range of values by typing as much of the value you want to compare and end with an asterisk (\*). For example, to filter for DSP01, DSP02, DSP03 but not DSP10, you would specify DSP0\*.



## Job Monitoring for jobs that do not start or complete by a specific time

### *Do you need to know if a time-critical job does not complete on time?*

Sometimes a very serious problem is identified by the absence of an event. You probably have a job that runs every night that is critical to your business. Would you like to know if that job does not start or end by a certain time? Messenger can monitor for specified jobs not starting or ending on time and alert you to a potential problem.

Messenger can monitor for a specific message to occur by a specified time, such as backups or day end processing job completion message ('Job 123456/QSYSOPR/BACKUPS completed normally'). If the message does not arrive before the specified time, Messenger can page you to alert you of a potential problem.

Messenger is preconfigured to send a notifications to the default device if the message does not occur on time.

## Messages from job logs and S/36 procedures

Messenger can monitor for messages appearing in the joblog of another active job. You can also use the \*JOBLOG Monitor to monitor for error messages in System 36 procedures as well as messages occurring in CL programs that ignore messages due to the use of MONMSG CPF0000.

## Program error messages from specified interactive jobs

Messenger can also monitor error messages in interactive jobs such as the messages on the 'Program Messages Display'. Messenger allows you to monitor specific jobs without using the resources otherwise needed to monitor every job.



## System History Log (DSPLOG QHST)

One of the most powerful features of System i is its ability to log system activity and error messages to the history log QHST.

The problem with QHST is that it contains so many messages that without adequate filtering tools, QHST contains too much information to be utilized proactively using the DSPLOG command. The result is important messages logged to QHST are never seen. Messenger can monitor QHST using Message Filter Sets to automatically and proactively detect important messages. Examples of messages to monitor from QHST:

- When a specific job has started or ended.
- Jobs that end abnormally.
- Changes to system values.
- Changes to system pool sizes.
- Security messages.
- When a specific subsystem is started or ended.
- Other messages.

## Job Status Monitoring (WRKACTJOB)

Do you need to know when a specific job is using 90 percent of the CPU? Perhaps the most important command used by system operators is Work with Active Jobs (WRKACTJOB), the display used to monitor the throughput of the system.

**Messenger can pro-actively monitor the Work with Active Jobs display for:**

- Jobs consuming excessive CPU resources.
- Jobs exceeding a specified length of time.
- Jobs that must be running during specified times.
- Jobs with poor response times.
- Jobs with too low priority.
- Jobs with a specific status, ie. LCKW or MSGW.
- Jobs with a specific function, ie. S/36 procedure abends or when a user is running a specific program.



**Jobs are monitored using Job Filters. You can include or exclude the jobs you want by any combination of:**

- Job name, or generic job name.
- User name, or generic user name.
- Subsystem name.
- Job type: interactive, batch, etc.
- Job function.

Filters can be combined, so you can monitor for situations such as all batch jobs with user QSYSOPR, etc.

### **Working with Job Filters**

The Work with Job Filters display allows you to define the specific jobs you want to include and exclude from monitoring. Job filters allow you to include and exclude the jobs you want by any combination of:

- Job name
- User name
- Job type: interactive, batch, etc.
- Subsystem name
- Function

**You can test the selected job(s) for any of the following conditions:**

- Priority is lower than a specified value.
- CPU % exceeds a specified value.
- Response time exceeds a specified value.
- Status is a specified value: LCKW, MSGW , etc.
- Run length: Hours, Minutes, CPU seconds.

Job Filters can be combined, so you can monitor for situations such as all interactive jobs lower than priority 20 excluding the console, etc.



Using the Work with Job Filters display, you can:

- Add new jobs to monitor, or exclude from monitoring.
- Change the parameters of a job filter.
- Remove jobs from monitoring.

### **Job Filter Details**

The Job Filter Details display allows you to define the parameters for filtering jobs. Specifying more than one parameter for a job filter are ANDed together (all of the parameters specified on a filter must be met).

## **Total CPU Usage Monitoring (WRKSYSSTS)**

Messenger can monitor the average CPU percentage used over an elapsed period of time:

- Specific CPU percentage.
- Length of time used to calculate average CPU percentage.

## **Job Queue Monitoring (WRKJOBQ)**

Do you need to know if QBATCH has 10 jobs waiting? Messenger can monitor the Work with Job Queues (WRKJOBQ) display for job queues that have a specified number of waiting jobs, are held/not held and perform the appropriate actions upon them, such as paging, releasing, moving to another job queue, etc.

### **Job Queue Details**

The Job Queue Options display is shown automatically when you press Enter from the Event Monitor Details display when the monitor type is \*JOBQ. This display allows you to define parameters for monitoring:

- Specific job queue and library name.
- Number of jobs.
- Status of job queue.



## System Audit Journal Monitoring (QAUDJRN)

The Security Audit Journal (QAUDJRN) is the primary source of information about security-related events on your system. You can use the journal monitor to filter events from the audit journal and alert administrators by alert or e-mail:

- Changes to auditing.
- System management changes.
- Changes to system values.
- Authority failures.
- Programs changed to adopt authority.
- Users obtaining adopted authority.
- Authority changes.
- Authority changes during restore.
- Restoring programs that adopt authority.
- Use of system service tools.
- Profile swapping.
- Invalid passwords.
- Command string audits.
- User profiles changed, created, restored.
- Actions that affect jobs.
- Objects created, deleted, moved, renamed, or restored.
- Audited object accessed.

### Journal Details

The Journal Details display is shown automatically when you press Enter from the Event Monitor Details display when the monitor type is \*JRN. This display allows you to define parameters for monitoring of journals.





## **Working with Journal Filters**

The Work with Journal Filters display allows you to define the specific journal entries you want to include and exclude from monitoring. Journal filters allow you to include and exclude the entries you want by any combination of:

- Entry type
- Job name
- User name
- Program name
- Object name
- Entry specific compare data

**Using the Work with Journal Filters display, you can:**

- Add filters for journal entries to monitor, or exclude from monitoring.
- Change the parameters of an existing filter.
- Remove filters from monitoring.

You can create as many filters as needed to define the event you are monitoring. Each filter has several parameters. Specifying more than one parameter for a filter are ANDed together: all of the parameters specified on a filter must be met. Specifying more than one filter are OR'd together: any of the filters matched will log the event.

## **Output Queue Monitoring**

Do you need to know if the system printer is held? Messenger can monitor the Work with Output Queues (WRKOUTQ) display for output queues that have a specified number of waiting spooled files, are held/not held and perform the appropriate actions upon them, such as paging, releasing, moving to another output queue, etc.



## **Output Queue Details**

The Output Queue Details display is shown automatically when you press Enter from the Event Monitor Details display when the monitor type is \*OUTQ. This display allows you to define parameters for monitoring an output queue:

- Specific output queue and library name.
- Number of spooled files.
- Status of output queue.

## **Line Status Monitoring (WRKCFGSTS)**

Do you need to know when a critical communications line is down? Messenger can monitor the Work with Configuration Status (WRKCFGSTS) display for lines, controllers, devices and network interfaces for abnormal statuses. Simply define the configuration object names and their normal status. If the status changes to any that are not checked, Messenger will log an event and perform the specified actions, which can include resetting the line and/or paging an operator.

## **Configuration Status Details**

The Configuration Status Details display is shown automatically when you press Enter from the Event Monitor Details display when the monitor type is \*CFGD. This display allows you to define parameters for monitoring of communications lines, controllers, devices, and network interfaces.

## **Ping Monitoring**

Do you need to know when another server is down? Unlike the SNA environment, System i does not generate an error message when TCP/IP devices are down. Messenger can detect the up/down status of TCP/IP devices anywhere on your TCP/IP network, such as routers, firewalls, Windows servers, Netware servers, etc. If your System i is connected to the Internet, you can detect the status of virtually any device in the world. Simply define the host names or IP addresses and Messenger will quietly interrogate those devices at regular intervals. If a device fails to respond, Messenger will log an event and perform the specified actions, such as paging or e-mailing a network administrator.

## **PING Details**

The PING Details display allows you to define parameters for monitoring of TCP/IP devices including monitoring of applications running on a specified port of any server.



## Disk Monitoring

The \*DISK monitor checks the growth of disk storage on an hourly, daily, weekly, or monthly basis. When specified thresholds are reached, the software triggers an alert that tells the administrators of the impending disk disaster.

- Monitor disk on an hourly, daily, weekly and monthly basis.
- Monitor for disk space remaining in percentage or megabytes.
- Monitor for disk space growth in percentage or megabytes.
- Monitor multiple ASPs.

### Disk Filter Details

ASP. Type the number of the Auxiliary Storage Pool (ASP) you want to monitor. To see a list of available ASPs, use the command DSPASBRM. The system ASP is 1.

- **Last ASP MB Used.** The amount of DASD used at the start of the monitoring period. This value is captured by the product at the start of the monitoring period, and is for display only. This value represents the current baseline for calculating measurements.
- **Threshold value and type.** The amount of DASD that can be used before triggering an event, expressed in either megabytes or as a percentage of total DASD. Threshold monitors detect when disk usage reaches a specified amount, either in megabytes used, or a percentage of disk (i.e. 90%).
- **Growth value and type.** The amount of storage utilization that can increase over a period of time before triggering an event, expressed in megabytes or as a percentage of total DASD. Growth monitors detect disk usage exceeding a percentage over a period of time (Hourly, Daily, Weekly, Monthly).
- **Growth period.** The length of the monitoring period: Hourly, Daily, Weekly, Monthly.
- **Detect spikes.** Specifies if the product should create events if spikes are detected during the growth period. When this value is on (Y=Yes), the product will create an event whenever the threshold is exceeded, and not just at the end of the monitoring period. The product will automatically acknowledge existing events if the condition clears.



## Restricted State Monitoring

Want to know when the SAVLIB \*NONSYS needs another tape? Did you know you can be notified about inquiry messages on your cell phone, pager, or e-mail during a Save Entire System (Option 21 on the GO SAVE menu) or when using Backup and Recovery Media Services (BRMS)? Messenger can monitor any message over severity 70 while in restricted state. This includes device and tape error messages, as well as inquiry messages such as a SAVSYS asking for another tape. And Messenger can send a notification to one or more devices while your system remains in a restricted state. Unlike competing products that only page during restricted state by requiring you to embed a send pager message command into the restricted state function, Messenger has restricted state monitoring features ready to detect errors and page you.

## Custom: Creating Your Own Monitors

Messenger allows you to create Monitors to detect many types of critical events occurring on your System i server. The following types of monitors are provided:

<b>CPU Percentage</b>	CPU monitors allow you to detect when the total CPU % of the system exceeds a specified threshold (WRKSYSSTS)
<b>Disk</b>	Disk monitors allow you to detect DASD events, such as low disk space remaining, or disk usage growth exceeding a specified threshold
<b>Joblog</b>	Joblog monitors allow you to detect messages occurring in interactive jobs
<b>Job run</b>	Job run monitors allow you to detect whenever specific jobs are not running
<b>Job</b>	Job monitors allow you to detect conditions about specific active jobs, such as their status, or CPU usage
<b>Job queue</b>	Job queue monitors allow you to detect when a job queue has exceeded a specified number of jobs (WRKJOBQ)
<b>Journal</b>	Journal monitors allow you to detect events that appear in the system audit journal (QAUDJRN)
<b>Line</b>	Line monitors allow you to detect the changes in status of System i lines, controller and devices (WRKCFGSTS)
<b>Message</b>	Message monitors allow you to trap message events that appear in System i message queues (WRKMSGQ)
<b>Output Queue</b>	Output queue monitors allow you to detect when an output queue has exceeded a specified number of spooled files
<b>Ping</b>	Ping monitors allow you to detect changes of a network IP device status. You can also detect changes in specific ports, such as FTP, Telnet, SMTP and POP3.



## Scheduling Monitors

Want to monitor only during nights and weekends? You can create schedules to define time periods that you can attach to monitors and actions. When a schedule is attached to an event monitor, the events are monitored only during the scheduled time. When attached to actions, the actions are performed only during the scheduled time. This allows you to perform different actions during different time periods.

- Pre-configured schedules for unattended, attended, weekdays, weeknights, and weekends.
- Create your own schedules.
- Apply schedules to monitoring or actions.

Using schedules, you can specify the days of the week and a starting and ending time period. For more complex scheduling, you can create schedules that contain a list of days and times.

### When you are working with:

**Monitors**

**Event Monitors**

**Actions**

**Event Queue Lists**

### Schedules:

Allow you to specify time periods the monitor is searching for events.

Allow you to specify time periods the event is monitored.

Allow you to specify time periods the action is performed.

Allow you to specify time periods that events are escalated to an Event Queue.

## Securing Monitors

Messenger allows you to easily prevent unauthorized users from changing monitoring setups. You can define an iSeries authorization list to secure changes to the monitor. Additionally, you can define menu and command authorities, such as Work with Monitoring Objects, and assign public or private authorities to various functions within Messenger.



## Define additional parameters for each Monitor

Allows you to define additional parameters for any monitor, pre-configured or custom. Parameters include:

- The scheduled time periods when to monitor for the event.
- How the events for that monitor should appear on the screen (color of actual data on screen, if the workstation alarm should be sounded, etc.)
- Attach instructions to event text where they can be viewed on the event console or viewed when sent as a pager message.
- The text of the event. For example, you may want to reword the system message “Checksum suspended on disk DSK01” to “Disk DSK01 failed, contact Operations Manager immediately!”
- The actions you want performed. You can perform actions to page, email, run commands, send replies, escalate to other users, etc.
- The severity of the event. For messages, you can override the severity to a higher or lower value.
- The allowed replies, which can be secured by an authorization list.
- The actions to perform (if any) when a specific reply is issued.
- If the event is informational or inquiry. For messages, you can override an informational message to an inquiry message.

## Working with Monitors

Allows you to define the monitoring parameters for your system:

- Create new monitors.
- Change the parameters of an existing monitor, such as the scheduled time periods, authorities, etc.
- Delete monitors from your system.
- Start and stop monitoring for particular events.
- Access the Work with Event Monitors display, which allows you to manage the individual events for a monitor, such as power failures, jobs ending abnormally, etc.
- Display the number of events logged by each monitor.
- Access the Work with Events display, which allows you to work with the events logged by the monitor.
- Display and reset the last time a monitor has logged an event and the number of events that have been logged since the last time it was reset.
- Print reports of the monitor’s configurations and details.



## Working with Event Monitors

Allows you to define the specific events you want to monitor:

- Create new events to monitor.
- Change the parameters of an existing Event Monitor, such as the scheduled time periods, color coding, etc.
- Delete Event Monitors from your system.
- Display the number of events logged by each Event Monitor.
- Start and stop monitoring for particular events.
- Access the Work with Actions display, allowing you to define the actions to perform for a particular event.
- Access the Work with Events display, allowing you to work with the events logged by the Event Monitor.
- Access the Work with Message Filters, if you are working with a message monitor.
- Access the Work with Job Filters display, if you are working with a job monitor.
- Access the Configuration Status Details display if you are working with a configuration status monitor.
- Access the PING details display if you are working with a TCP/IP monitor.
- Access the Display Journal details if you are working with a Journal monitor.

## SECTION III.

# EVENTS & EVENTS CONSOLE





## Events & Events Console

The Work with Events display shows you the events that have occurred on your system. If you are using the optional MessengerConsole module, the display can also include events from other System i servers. Newest events are shown at the top of the display with older events scrolling off the bottom. Unanswered events always remain at the top of the display until they are answered or acknowledged.

Using the Work with Events display, you can:

- See the events that have occurred on your system. As new events occur, they appear on the screen automatically. Events are displayed using the attributes specified using the Event Monitor Details display, which allows you to specify settings for color, alarm, severity and message text.
- Send a copy of the event to another Event Queue. If you are using the optional MessengerConsole module, the event can be exported to another System i.
- Acknowledge an event to remove it from the display and all other displays with a copy of the event. If you are using the optional MessengerConsole module, the event is removed from all remote systems.
- Display the history of the event. The event history shows you when the event occurred, the actions that were performed and the users that have a copy of the event.
- Exclude the event from further monitoring, providing you are authorized to make changes to the monitor that logged the event.
- Display a subset of events, such as those from a specific job, user, etc.
- Access the Work with User Actions display, which allows you to associate a 2-character option code with actions. The actions are performed when the option code is typed next to an event, similar to the OS/400 Programming Development Manager (PDM). This allows you to create options to work with jobs, vary lines, send notifications, etc. User Actions greatly extend the capabilities of the Work with Events display by allowing you to create shortcuts to the commands you use most often to resolve problems.



## Working with Event Queues

The Work with Event Queues display allows you to manage Event Queues. Event queues are similar to message queues but are more powerful; they contain all types of events, not just messages. Additionally, Event Queues can:

- Break on your screen in more than one location at a time.
- Show events in colors and sound alarms.
- Can be distributed across a network of System i servers.
- Allow multiple users to access the same event. When an event is acknowledged, the event is automatically removed from all Event Queues.
- Run actions whenever a new event is added to the queue.
- Secure other users from displaying or changing events in the queue.
- Escalate to other Event Queues after a specified time period.

## Event Console

The Messenger Event Console™ shows you the list of messages and events that have been logged as a result of the monitoring you have defined. From the console within Messenger you can:

- Link directly from an Event to the Event Monitor where you can customize the setup of the event.
- View all monitored messages and events from all systems and partitions on a single screen.
- Automatically refreshes to show new events.
- Color-codes and alarm important messages and events by severity, job and user.
- Easily exclude unwanted messages from further monitoring.
- Show the history of actions that were performed.
- Allow you to easily run frequently used actions on events using user-defined actions, similar to PDM (the OS/400 Programming Development Manager).
- Easily change the filtering of monitored messages and events, excluding or modifying monitoring and actions as you go.
- Subset events by system, event type, severity, etc.
- Copy or move events to other centers and systems. 🔄

## SECTION IV. ACTIONS



## Actions

Messenger can perform unlimited actions for an event. With Action Scripts, Action Escalation and Action Scheduling, you define how errors are responded to. You can create Actions to:

- Send pager and e-mail messages.
- Send messages to users or workstations.
- Send job logs, system printouts, or IFS files as e-mail messages.
- Run CL commands.
- Send replies to automate routine messages.
- Send events to other systems.
- Reply to messages.
- Escalate unanswered messages.
- Acknowledge informational and inquiry messages.

## Action Scripting

*When a line fails do you vary it off and leave it off, or do you vary it off and back on again?*

Messenger allows you to specify as many actions as needed to perform a task. This is important because in many cases you need more than one action to correct a problem. For example, you need two actions to recover from a failed line: one to vary the line off, and another to vary the line on again.

### Substitution Variables

Messenger allows you to pass parameters from the event to the action, such as the message ID, job name, etc. You can use this information in your own event handling programs, or as part of an action that Messenger will perform:

- &MT The text of the message. For non message events, the default text is assigned by Messenger.
- &MQ The message queue name.
- &ML The message queue library name.
- &MI The message identifier.



- &MD The first 256 characters of the message data.
- &MJ The job name.
- &MU The user name.
- &MN The job number.
- &MK The 4 character message key.
- &MP The program name.
- &MV The severity code.
- &MY The message type code.
- &MH The first 256 characters of the message help.
- &MF The message file name.
- &MB The message file library name.
- &MS The name of the system the event originated.
- &1 .. &99 The message data variable from the message description.

**Example 1: Automate a CPI593D (line &23 failed) message by varying the line off and on:**

```
VRYCFG CFGOBJ(&23) CFGTYPE(*LIN) STATUS(*OFF)  
VRYCFG CFGOBJ(&23) CFGTYPE(*LIN) STATUS(*ON)
```

**Example 2: Automate procedures to hold a job using 90% CPU and page an operator:**

```
HLDJOB JOB(&MN/&MU/&MJ)  
*PAGE TOPGR(QSYSOPR) MSG('Job &mn/&mu/&mj using 90% has been held')
```

## Action Escalation

When a program is looping and filling a file, you wouldn't endlessly answer the file full message with an "I" every time. If you did, the program would fill up all available disk space and your system would probably crash. Yet many competing products do exactly that, blindly answering the same message with the same reply every time. You can configure limits on the number of times an action is run. This is important because you would never want to repeatedly run an action that is not correcting a problem. For example: you may want to send an "I" (Ignore) reply to a file full message, but after the third try within 60 minutes, send a "C" (Cancel) reply or notify a pager.



## Action Scheduling

When a job queue has too many jobs in it waiting to run, do you want a notification every time, or just on weekends and during unattended processing? Messenger gives you the option to attach schedules to actions, so you can specify when actions are performed. For example, if a job queue is overloaded you can specify that a notification is sent during unattended operations and a break message is sent during normal hours.

## Action Shortcuts

You can create powerful time-saving shortcuts using abbreviated codes that run one or more actions. When you type the code next to an event, the actions associated with the code are performed. You can create user actions to execute commands, send notifications, replies, escalate to other users and systems, etc.

**Example:** Instead of typing: DSPJOBLOG JOB(123456/QSYSOPR/DSP01), you can create a shortcut command called JL to display the job log of the job with the error message.

## Working with Actions

The Work with Actions display allows you to define actions to perform for events. Examples of actions are:

- Sending pager and e-mail messages.
- Sending replies.
- Running commands.
- Copying the event to an Event Queue.
- Automatically acknowledging an event.

**Actions can be specified at several levels, depending on when you want the actions performed. You can attach actions to:**

- **Monitors.** Performs the specified actions whenever the monitor logs an event. For example, attaching a pager message action to the QSYSOPR monitor will send a pager message whenever a message you are monitoring for occurs on QSYSOPR. Actions attached to monitors can be considered default actions; they are performed only when there are no actions specified at the Event Monitor level.



- **Event Monitors.** Performs the specified actions whenever a specific event occurs. For example, attaching a reply action to a 'Verify alignment' Event Monitor will answer the message. When you specify an action at the Event Monitor level it overrides any actions specified at the monitor level (actions at the monitor level will not be performed).
- **Event Queues.** Performs the specified actions whenever an event is added to the Event Queue.
- **User Actions.** Performs the specified actions whenever the user action code is typed next to the event using the Work with Events display.

Using the Work with Actions display, you can:

- Add new actions to perform.
- Change the parameters of existing actions such as adding a delay to the action to allow for operator intervention.
- Remove actions.

## Reply Management

Every message waiting for a reply results in lost minutes of processing, causing operational crises. Not only is it important to answer messages quickly but correctly. Answering a message with the wrong reply could cause an even larger crisis. Messenger can:

- Automate procedures for reoccurring events.
- Answer routine messages automatically and quickly.
- Reply by pager.
- Reply by e-mail.
- Reply by GSM/SMS messaging.
- Only display valid replies to guarantee messages are answered correctly.
- Assist you in answering non-routine messages.
- Allow you to schedule when replies are sent.

## SECTION V. PAGING & NOTIFICATION





## Paging & Notification

Getting messages is reliable and easy. By far the industry's easiest-to-use and most flexible paging system, Messenger takes care of all the complexities of communication for you.

### About Paging

Because System i can generate thousands of messages in a day, many vital to the health and status of the system and business operations, it is critical that your paging solutions be sophisticated enough to deliver only the important messages and not bog you down with repetitive paging over and over and over again. Messenger is the industry's easiest and most flexible paging solutions for System i. Whether it is automatically sending a vital message or end-users are using our graphical "Send a Pager Message" interface, getting a message from System i is reliable and easy.

Messenger offers the most inclusive paging functionality available today. Supporting all TAP/IXO, SMS/GSM protocol paging services, international GSM and SMS pager device types, E-mail and workstations.

### Supported Notification Types

#### Alphanumeric

A type of device that is capable of displaying text messages. Includes extensive alphanumeric paging support (1-way and 2-way pagers), using a variety of industry standard international protocols. Most cellular phones that have text messaging capabilities can receive messages just like an alphanumeric pager.

#### 2-way

A type of device that is capable of 2-way messaging. Want to answer system messages directly from your 2-way pager? Messenger can send 2-way messages to certain types of 2-way pagers. You can reply to the message directly from the pager by choosing one of the multiple choice responses or type a custom reply using the pager's data entry capabilities. 2-way pagers offer the following benefits:

- You do not have to sign on to the system to answer a message. Replying to a system message from the pager answers the message on the message queue.



- Users are immediately notified of your response. Your responses are sent back to the user's message queue.
- Choose to acknowledge the message only, which does not answer the message but stops the message from automatically escalating to another pager.
- You can integrate two-way paging capabilities with your applications using the Send a Pager Message (SNDPGRMSG) and Retrieve Pager Message (RTVPGRMSG) commands.
- Pager message delivery is guaranteed because the paging network will automatically hold your pager messages for up to 72 hours when your pager is out of a service area (depending on service provider).

### **GSM/SMS**

Messenger sends to global GSM/SMS pagers, allowing simple and quick wireless distribution of messages to any cell phone in the world without the need for a paging provider, phone lines, or receiving modems. Messages are received anywhere across the globe in seconds. In addition to enabling easier wireless messaging, the GSM/SMS paging also eliminates the compatibility issues that can affect international users. In many areas of the world, variations in phone service can make it difficult to successfully set up automated notification to mobile devices. By sending messages phone-to-phone using standardized text-messaging protocols, Messenger now eliminates these issues. All that is needed is a number and a phone capable of receiving a text message. This protocol also allows the capability to reply back to system messages from your wireless device.

### **E-mail**

Messenger will send the pager message to the specified SMTP Internet e-mail address. Messenger can send e-mail messages to your desktop, pager, cell phone, or PDA.

#### **2-Way e-mail**

- Gives you the ability to answer system messages from any internet based e-mail.
- Allows administrators to answer system and application messages from any of the fast growing array of wireless Internet devices, such as Palm Pilots, cellular phones and pagers.



- Answer system and application messages via desktop e-mail clients such as Eudora, Netscape, Outlook, Notes, etc., eliminating the necessity for access to a green screen.
- Allow applications to communicate with mobile workers.
- Attach IFS files, system print-outs such as WRACTJOB and WRKSYSSTS, or joblogs to your e-mail messages.

## **Workstations**

When Messenger sends a page to a \*MSGQ pager, the message appears in the specified message queue which can break on your workstation. Replies are automatically sent back to the sender. For system messages, answering the message on your workstation will automatically answer the system message on QSYSOPR. Answering a message automatically stops escalation of the message to other users.

## **Group**

A type of pager used to broadcast a single message to multiple pagers at the same time. Want to send a single message to multiple pagers at once? If so, you'll like the ability to create Group pagers. When you send a pager message to a group pager, all of the pagers in the group will receive the same message.

## **SNMP**

Sends an SNMP trap. Messages sent to this type of pager are sent as an SNMP network trap that can be monitored using an SNMP manager such as HP Openview. Messenger sends numerous OIDs with each trap. In addition, a trap MIB (Management Information Base) document is available that can be loaded into a trap manager to allow system managers to pick and choose which objects (fields) to display on a enterprise console. You might choose to see the system, date & time, severity and message text. You could open the trap, and see additional details such as the message ID and job name, for example.

## **SNPP**

Sends a text message to wireless devices using the SNPP protocol through the internet.



### **Numeric**

A type of pager that is capable of displaying numeric characters only. Supports all numeric pagers and can send up to 16 digits. For system messages, Messenger sends a code that you can specify with the type of situation that has occurred. For example, 400-911 for a power outage, 400999 for a program failure, etc.

### **Telephone**

A type of pager that sends a tonal message to a standard telephone. Can send a tonal message to a telephone. This method can be used to alert someone that does not have a pager, or leave tonal messages on voice mail systems. This method is a great alternative if your pager or cell phone does not receive a strong signal in certain areas (i.e. poor reception at home or in a computer room located in a building's basement).

### **Schedule**

A type of pager that automatically selects the appropriate pagers (or lists) based on the current date and time. Would you like to specify which pagers receive system messages during certain time periods? You can create a Schedule Pager that automatically pages the appropriate pager(s) based on the current time of day. When you send a message to a Schedule Pager Messenger will determine the pager names to receive the message based on the time periods you have specified. You can also attach a schedule to a pager, or groups of pagers, that will determine the time periods the pager is eligible to receive messages. Messages sent to pagers outside scheduled times are held and then delivered at the appropriate time.

### **Escalation**

A type of pager used to escalate messages to other pagers after a specified time period. Want to automatically page another pager if a problem goes unresolved? Messenger provides the most extensive escalation capabilities in the industry. When you send a message to an escalation list pager, Messenger will automatically page the next pager on the list at specified intervals. You can specify the time it will wait between pages, allowing the flexibility to escalate more critical messages faster. The escalation process ends when the message is answered or after a specified number of tries.

## **Paging Companies**

- Popular paging company details are pre-configured
- Supports multiple paging companies



## Message Formatting

Messenger allows you to easily personalize the format of the messages sent to you by using any of the following special variables that you can mix and match in any order:

- Message text
- Message help text
- Job name that sent the message
- User name that sent the message
- Number of the job that sent the message
- Number of the pager message assigned by the system for your pager. The inclusion of this number tells you if any pager messages have been missed since your last message from the system.
- Date and time of the message
- Message lengths up to 999 characters
- Custom

For example, the message “Load next volume on TAP01 (CG)” can be formatted to read:  
SYSA:QSYSOPR/BACKUP, Load next volume on TAP01 (CG).

Extended message lengths. Allows you to send up to 999 characters, even if your pager allows less. Most alpha pagers have restrictions on the message length, usually 80 to 240 characters. The messages are numbered (1/3, 2/3, and 3/3) so that you are aware of the total number of messages making up the whole message.

## Escalation

To escalate an unanswered message to other users, you can create a list of pagers, which can include any combination of other pagers. An \*ESC pager list will escalate messages to the pagers on the list one at a time at the defined intervals. At any time during the escalation period, if the message is acknowledge or answered then the escalation process is ended and the remaining pagers will not be paged. You can specify the order the pagers will be paged, the number of times to page the specified pager, and the number of minutes between pages.



## Scheduling

Want to send a page only during nights and weekends? You can create schedules to define time periods that you can attach to pagers. When a schedule is attached to page actions, the actions are performed only during the scheduled time. This allows you to perform different notifications during different time periods.

- Specify time periods, such as first shift, second shift, etc.
- Create your own schedules.
- Apply schedules to pagers.

Using schedules, you can specify the days of the week and a starting and ending time period.

## Restricted State Paging

Did you know Messenger can notify you about inquiry messages via email, cell phone, or pager when the operating system is in restricted state? Messenger can send notifications or e-mail messages to one or more devices while your system remains in a restricted state. When a message arrives on QSYSOPR, Messenger retrieves the message and, if the severity is 70 or greater, it will create an event and end an alert message to notify you. You can send pages from within your restricted state program, from BRMS, or when using GO SAVE menu option 21.

## Shares with Service Director/ECS/Fax line and modem

If you are using OS/400 V4R1 or later and the IBM 7852-400 Options modem, Messenger can automatically share the modem with synchronous applications such as Service Director, ECS and remote dial-in access, saving you valuable V.24 ports. You do not have to manually switch the modem between uses.

## Modem Pooling

If your paging volume is very high, you will appreciate the ability to utilize multiple modems simultaneously to deliver thousands of pager messages per hour.



## Pager Security

Need to prevent users from accessing unauthorized pagers? The pager security features allows you to attach authorization lists to pagers so you can prevent unauthorized users from viewing, changing or sending messages to unauthorized pagers. Non-administrators cannot see pager numbers. You can define whether the public can change the pager; can send messages to the pager but cannot change it; or cannot display, change or send messages to the pager. Messenger tests authority using the user's ID, then the user's group profile name, then \*PUBLIC.

## Add Paging to your Programs

Want to page from your application? The Send Pager Message (SNDPGRMSG) API can be easily called from your RPG and CL applications. You can send 1,000 characters to one or many pagers without having to deal with the complexity of communications protocols. You can even designate your message to be sent at a later time. Using pager lists and schedules, your messages can escalate to other pagers automatically and page on-call staff.

## Network Paging/Message Dispatching

Whether you use our graphical Windows Send a Pager Message interface or green screen pop-up window, they both guide you through sending messages to any number of pagers or E-mail device types, complete with prompting and on-line help text. Unlike competing products that use CL commands, Messenger includes a CL command interface and user-friendly windows/GUI interface:

- Users can access the Send a Pager Message window from within any application.
- You can easily add the Send a Pager Message window to your application menus.
- Users can select one or many pagers from a list of authorized pagers.
- Eliminates retyping of frequently used messages by allowing you to save and resend messages again later.
- Unlike competing products, each user can send and check the status of their own messages. Users cannot see other users messages. Only system operators and administrators can see all messages.
- Eliminates the need for setting up each PC with emulation software, modems, etc.



## Automatic Clean-up

Messenger's System Options allows you to define the number of days to keep pager messages and the number of days to keep events. You can specify the number of days that pager messages are retained and stored. Messages older than the specified number of days are automatically deleted. You can also specify the number of days events are retained in the Events database. Events older than the specified number of days are automatically deleted. This prevents any possible storage issues relating to Messenger and it allows you to retain an audit trail if needed.



## SECTION VI. REPORTING



## Reporting

### Messenger has extensive reporting capabilities including:

- Event History
- Action History
- Paging History
- QHST Analyzer
- Usage tracking: Easily view the number of times a specific message or type of message has occurred over a period of time.
- Monitor Configuration Reporting

## Event History

You can display the history of the event. The event history shows you when the event occurred, the actions that were performed and the users that have a copy of the event.

## Action History

Messenger allows you to easily display the history of the actions that have been performed for an event. It maintains a detailed audit trail of the actions performed. For any given event, you can quickly view the actions that were performed, the users that were notified, the replies that were entered, etc.

## Paging History

Messenger maintains a detailed history of paging and E-mail activity. Users can print reports of their pager messages showing the date and time entered, the message text, which pager(s) the message was sent to, and the dates and times of those transmissions. Filtering options allow you to restrict the report to specific pagers, date ranges, message text, etc. You can view or print a list of messages that meet specified criteria:

- The messages that were sent
- Date and time
- Job name
- Users that were notified and when



- Paging company responses
- Replies from 2-way pagers
- Communications detail (ICF) allowing you to diagnose any paging problems
- The number of messages waiting to be sent
- The number of messages waiting to be sent for a Paging Company
- The number of messages waiting to be sent for a pager

## QHST Analyzer

Contains a utility to analyze your system history log (QHST) and display the number of occurrences of messages. This utility is useful in assisting you in determining the QHST messages you want to monitor.

You can use the severity column to identify the number and types of messages you would receive at different severity levels. The Message ID column can be used to identify specific messages to include or exclude from monitoring.

## Usage Tracking

Need to know how many times a program has crashed this year? Messenger maintains a counter for each filter you define, allowing you to easily view the number of times a specific message or type of message has occurred over a period of time. The following statistics can be reported:

- The number of times an event has occurred by type (power outage, job failure, job queue limits, etc.) over a period a time.
- The date and time of the last time the event occurred.
- The number of unanswered events by type of event, such as printer messages, batch errors, etc.

## Monitor Configuration Reporting

Messenger includes the option to print configuration reports to easily document your monitoring setups. You can even generate test messages to see how your monitors would respond to a specific system message should it occur. This ability to simulate a system message includes testing with message ID, message queue and library, message type, and specific message data.

Additionally, Messenger will log when a Filter was created so you can track when configuration changes occurred.

## SECTION VII. CONSOLING

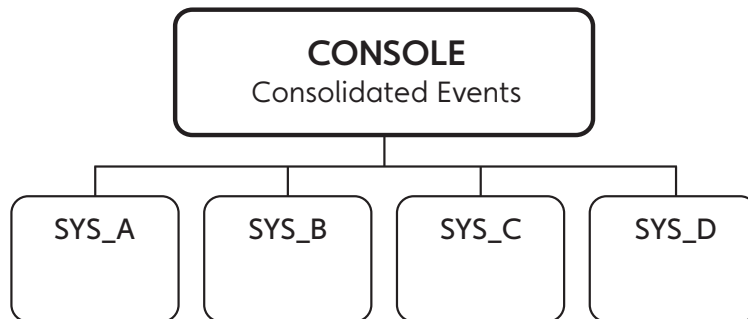


## Consoling

*The features on the following pages are specific to MessengerConsole and are in addition to the features found in MessengerPlus.*

### Introduction

Event Consoling is the process of collecting events from multiple systems to a central location where they can be viewed and managed centrally, called a Console. With MessengerConsole you can quickly see the events that have occurred on remote systems and manage these events, as well as manage monitoring, filters, actions/replies, and notification all without the need to manually sign on to the remote system.



A major hurdle to automating remote monitoring is the amount of time and effort required to setup and maintain the remote monitoring software at each site. Even if you initially setup your remote systems by restoring the software from a pre-configured system, the ongoing maintenance becomes a major problem over time.

MessengerConsole solves this problem by allowing you to create and maintain all of your monitoring setups from a single central system. Any changes you make from the central system are distributed to remote systems automatically. Once you have defined the events you want to monitor, you can distribute those definitions to remote systems without the need to manually recreate those setups for each and every system.



## Benefits

- Reduces the time and effort to setup and maintain the remote monitoring software.
- Improves the efficiency of the network by reducing the transmission of unimportant events from remote systems.
- Ensures consistency of the monitoring policies implemented at each system.
- Secures distributed monitors at the remote sites from being changed by remote users. All administration and maintenance must be performed from the console. Remote users can continue to create new monitors of their own (providing they are authorized), but they will not be able to change or delete the monitors you distribute.
- Reduces the time and effort to monitor and diagnose problems on remote System i servers by providing a consolidated view of the events from all your remote systems.
- Manage multiple remote System i servers from a central site, reducing the necessity for 24 x 7 technical staff at remote locations.
- Reduces downtime associated with unanswered remote messages by escalating these messages to the centralized console. Remote messages can be answered easily, without the need to manually sign on to the remote system. Unanswered console messages can be further escalated to e-mail and pagers.
- Reduces the cost, setup and administration of remote monitoring by dispatching pager messages from the central system. You do not need V.24 ports, modems and phone lines for each system.



## Features

- Remote monitoring and automation continues to run in real time while events are being transferred to the console system, because the transferring of events is handled asynchronously by a separate batch process. Remote monitoring does not require or depend on continuous connectivity to the consoling system.
- Remote events will queue when console connectivity is unavailable and will be automatically sent when connectivity is restored.
- An alternate console can be utilized to serve as a temporary console when the primary console is unavailable. The switching can occur manually or automatically from the primary console or the alternate console.
- From the console you can easily define which events are automated at the remote sites and which events are sent to the console system for handling or automation.
- Events can be escalated to the central console after a specified period of elapsed time or during specific time periods. This allows a remote operator time to respond to an event. For remote unattended systems, the events can be escalated immediately.
- Events can be escalated to the central console after a specified threshold, such as after the third “I” reply to a file full message.
- Works with SNA and TCP/IP communications. The communication between systems is accomplished using OS/400 Distributed Data Management (DDM) or TCP/IP sockets. You have control over the parameters used to create the DDM files. DDM can communicate over SNA, TCP/IP with AnyNet, or TCP/IP.
- Automatic Checking of Remote System Processes. MessengerConsole provides an MCPING command (works with SNA and TCP/IP) that allows you to determine if the required monitoring processes are running on one or more remote systems. You can run the command interactively, or automate the command to run at regular intervals, to test the communications and monitoring processes. Status messages can be logged on both the sending and receiving systems, where they can be monitored for abnormal conditions and automated to appear on the console and/or notify an administrator by pager/email.



## The Console

The Work with Events display shows you the list of messages and events that have been logged from all remote systems.

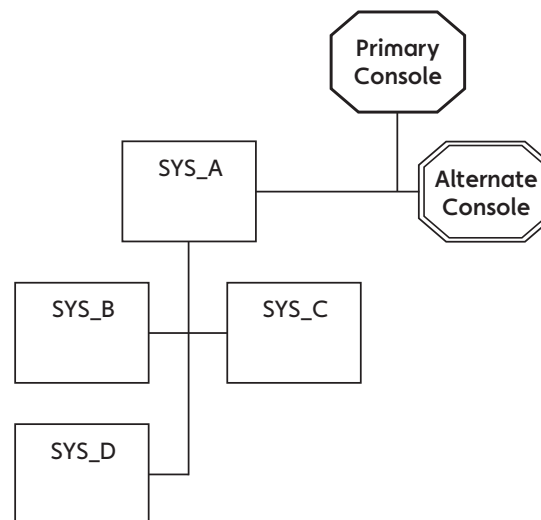
- Messages waiting for a reply are sorted to the top of the display and displayed in reverse image.
- Displays important messages in color and alarms.
- Answered and acknowledged messages are automatically removed from the console display, even if the reply occurs at a remote site.
- Allows replying to remote messages without signing on to the remote system.
- Automatically refreshes to show new events without pressing F5.
- Easily exclude unwanted messages from further monitoring.
- Subset events by system, event type, severity, etc.

## The Alternate Console

MessengerConsole can mirror itself to a backup, or alternate console.

The alternate console can become the primary console at any time in the event of a failure or planned outage of the primary console.

- The Start System Redirection (STRSYSR) and End System Redirection (ENDSYSR) CL command can be easily integrated with a job scheduler or your custom CL program to start and end redirection at scheduled times.
- The redirection commands can be run from the Alternate Console in the event of unplanned downtime of the Primary Console.
- Remote systems will automatically redirect events to the Alternate Console instead of the Primary Console until redirection is ended.
- You do not have to run redirection commands from the remote systems, this is handled for you automatically.

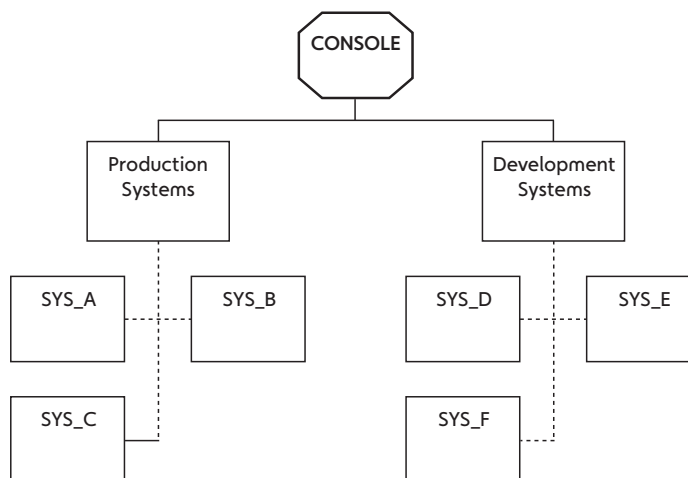






## Distribution Lists

MessengerConsole allows you to group your remote systems into logical groups called Distribution Lists, which are simply groups of systems that you want to monitor in the same way—such as Production, Development, Manufacturing Plants, etc. You can configure a monitor to run across a distribution list, and any changes you make to the monitor are distributed to the systems in the list.

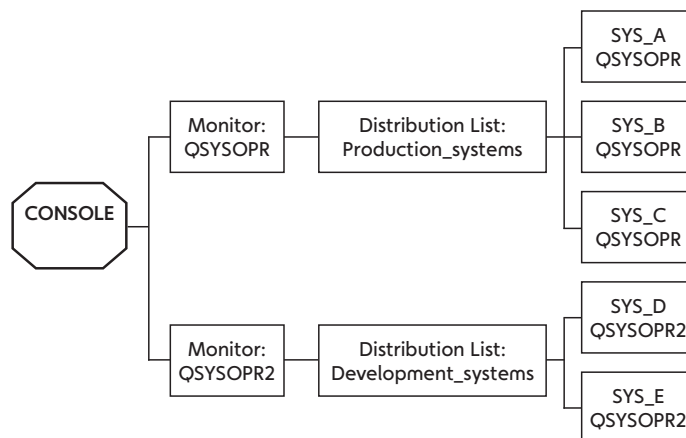


- A system can be a member of one or many lists.
- A distribution list is applied to a monitor.



## Monitor Distribution

Monitor Distribution is the process of distributing a set of policies (monitors) to one or more systems. Once you create your distribution lists, you specify the monitors that will use the distribution list. MessengerConsole will automatically distribute those monitors to the proper systems for you. Any future changes you make from the console to a distributed monitor will be automatically updated at the proper remote systems.



- Distribution occurs as a background process. If connectivity to a remote system is unavailable, the distributions will automatically queue until remote connectivity is restored.
- The distributed monitors are protected from changes at the remote systems, ensuring integrity and security. All administration to a distributed monitor must occur from the console.



## Escalation

Events can be escalated to the central console after a specified period of time. This allows a remote operator time to respond to an event. For remote unattended systems, the events are escalated immediately.

## Actions

MessengerConsole can perform actions on consoled events from the central system and at remote systems.

- Send pager and e-mail messages (supporting all protocols, including 2-way pagers)
- Send messages to local and remote users, workstations, or systems
- Run CL commands
- Reply to inquiry messages
- Send events to other users
- Escalate unanswered message

## Integration

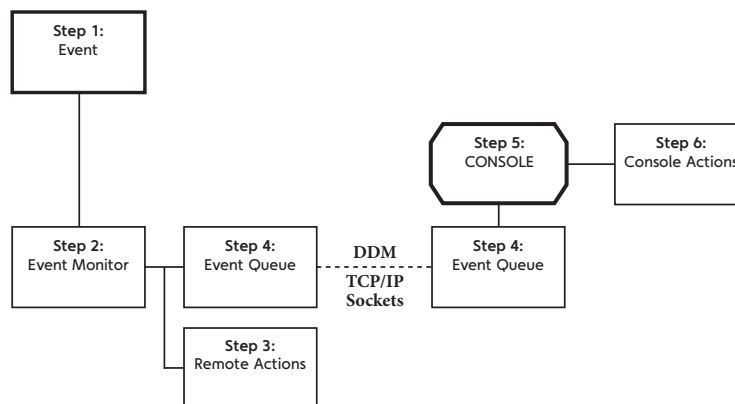
MessengerConsole can be integrated with Enterprise Monitoring Solutions, providing a more complete monitoring solution for enterprise environments. This is made possible by sending SNMP network traps to an enterprise trap manager.



## How it Works

The following high level diagram shows the processes involved in consoling a remote event.

- **Step 1**  
An event occurs, such as an error message or critical job is not running.
- **Step 2**  
The event is detected by an Event Monitor (which was previously defined at the console system and distributed to the remote system). The Event Monitor contains the filters that define the event, the actions to perform when the event occurs, and the name of the Event Queue (an Event Queue is simply a list of events/messages, similar to an e-mail inbox).
- **Step 3**  
If any remote Actions were defined for the Event Monitor (such as an automated reply, sending a pager or e-mail message, running a command, etc.), they are performed on the remote system at this time.
- **Step 4**  
If an Event Queue was specified, the Event is copied to the specified Event Queue. If the Event Queue is defined on the Console, the Event is transmitted to the Event Queue on the console system.
- **Step 5**  
The Event appears on the Console.
- **Step 6**  
If any console Actions were defined for the Event Monitor they are performed at this time (such as paging, e-mailing, running a command, etc.). If you are sending pager messages from the Console system they would be performed in this step.





## How much money do you lose when you have system or job failures?

Calculating and understanding the cost of downtime can be complicated. “Downtime” refers to when any critical application or system is unavailable to your users, vendors, and/or customers. Downtime can range from the obvious—loss of power or disk crash—to the not so obvious—message waits, job failures, communication failures that all lead to lost processing time. This simple worksheet will calculate your downtime costs.

Your Company				Example
<b>Calculate your average hourly revenue:</b>				
1	Annual revenue	▶ \$		\$ 50,000,000.00
2	Number of work days per year	▶ ÷		÷ 260
3	Equals daily revenue	\$		\$ 192,307.69
4	Number of hours worked per day <small>(The total hours per 24 hour period employees are reporting to work and/or requiring access to the system).</small>	▶ ÷		÷ 10
5	Equals hourly revenue	\$		\$ 19,230.77
<b>Calculate your average hourly labor costs:</b>				
6	Number of employees	▶		250
7	Average cost per employee <small>(The average cost of employee labor includes hourly wage, benefits, overhead, training.)</small>	▶ ×		× \$30.00
8	Equals hourly labor costs	\$		\$ 7,500.00
<b>Calculate your hourly cost of systems downtime:</b>				
9	Hourly revenue (from Line 5)	▶ \$		\$ 19,230.77
10	Hourly labor costs (from Line 8) <small>(Percentage hourly your company relies on System i for business processing, revenue generating, etc.)</small>	▶ +		+ \$7,500.00
11	Equals hourly revenue and labor costs	\$		\$ 26,730.77
12	Hourly percentage of System i reliance	▶ ×		× \$0.75
▶ <b>Your hourly cost of system downtime</b>		\$	◀	<b>\$ 20,048.08</b>
<i>Not included, but equally important, are factors such as lost processing time, recovery and overtime costs, lost goodwill with vendors and customers, and employee moral and job satisfaction.</i>				



**For more information about Messenger or to arrange a technical walkthrough, please contact us at 775.851.2900.**

**Additional information about Messenger is also available on our website at [www.bytware.com/products](http://www.bytware.com/products)**

Bytware  
6553 Flying Cloud Drive, Suite 200  
Eden Prairie, Minnesota 55344  
USA

MessengerConsole® and MessengerPlus® are registered trademarks of Bytware. IBM®, Power Systems®, IBM i®, System i®, iSeries® and AS/400® are registered trademarks of International Business Machines. Other brand and product names are trademarks or registered trademarks of their respective holders.

© 2012 Bytware. All Rights Reserved.