HI-TECH solutions

Advanced Image Processing Products - http://www.I-Cube.co.za

SeeUtilities

Technical Document

Jan 17, 2006 For AA

Table of Contents

1.Purpose & Scope

This document describes the set of utilities that support Hi-Tech Solutions' See$_x$ products (such as SeeGate, SeeTruck, SeeCrane or SeeLane). These utilities enrich our products, ease the technical support and cut the time to market.

The set of utilities include the following:

- Networking Utilities (SeeData)
- Maintenance Utilities (SeeMonitor)
- Operation Utilities (SeeService)
- Housekeeping utilities (SeeCleaner)
- Installation Utilities (SeeCal)

2.Referenced Documents

2.1 SeeData - Technical information document
2.2 SeeLane - Technical information document
2.3 SeeGate - Technical information document
2.4 SeeCrane - Technical information document

These documents are available in our home page; see last section for details.

3.Document Overview

This document is structured as follows:

Section 4 – SeeMonitor – maintenance utilities
Section 5 – SeeService – operation utilities
Section 6 – SeeCleaner – housekeeping utility

Note that SeeData is covered by a separate document (see reference #2.1), and SeeCal is described in its readme.txt file.

4.SeeMonitor

4.1.Overview

4.1.1. Utility objectives

SeeMonitor is a tool which is useful for maintenance, operation review and installation. It is especially designed to handle multiple See$_x$ applications (such as SeeGate, SeeCrane or SeeLane) connected via the network. This tool is usually installed on one or more central severs, or on one of the See$_x$ PCs. It requires a special HASP license, as indicated on the dongle label.

The objectives of the SeeMonitor utility are:

a. **Monitoring** the **status** of See$_x$ applications

b. Summarize the operation **graphically**

c. Enable a quick **access the events log**

d. Report on the status to external management systems.

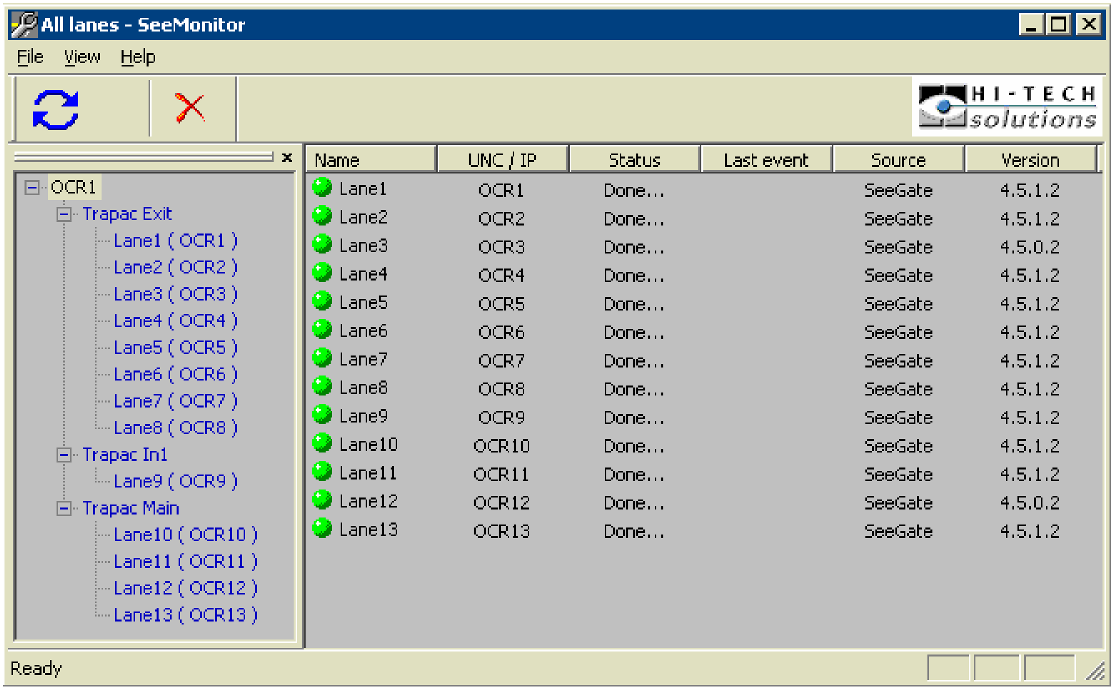
These functions are further detailed in the following pages.

4.1.2.Sample Displays

The following display examples show some of the utility's main displays.

4.1.2.1.Main status display

The following image shows a typical main display which shows the status of the 14 lanes in a SeeGate array. On this display you can see that all lanes are in normal status, as indicated by the **green** light status. Additional information may also be seen, such as the address name, the status of read, the last warning (**yellow**) or error (red, if fault exists), and the application version name and number.



- OCR1
  - Trapac Exit
    - Lane1 (OCR1)
    - Lane2 (OCR2)
    - Lane3 (OCR3)
    - Lane4 (OCR4)
    - Lane5 (OCR5)
    - Lane6 (OCR6)
    - Lane7 (OCR7)
    - Lane8 (OCR8)
  - Trapac In1
    - Lane9 (OCR9)
  - Trapac Main
    - Lane10 (OCR10)
    - Lane11 (OCR11)
    - Lane12 (OCR12)
    - Lane13 (OCR13)

Name	UNC / IP	Status	Last event	Source	Version
● Lane1	OCR1	Done...		SeeGate	4.5.1.2
● Lane2	OCR2	Done...		SeeGate	4.5.1.2
● Lane3	OCR3	Done...		SeeGate	4.5.0.2
● Lane4	OCR4	Done...		SeeGate	4.5.1.2
● Lane5	OCR5	Done...		SeeGate	4.5.1.2
● Lane6	OCR6	Done...		SeeGate	4.5.1.2
● Lane7	OCR7	Done...		SeeGate	4.5.1.2
● Lane8	OCR8	Done...		SeeGate	4.5.1.2
● Lane9	OCR9	Done...		SeeGate	4.5.1.2
● Lane10	OCR10	Done...		SeeGate	4.5.1.2
● Lane11	OCR11	Done...		SeeGate	4.5.1.2
● Lane12	OCR12	Done...		SeeGate	4.5.0.2
● Lane13	OCR13	Done...		SeeGate	4.5.1.2

Figure 4.1.2.1 Main status display

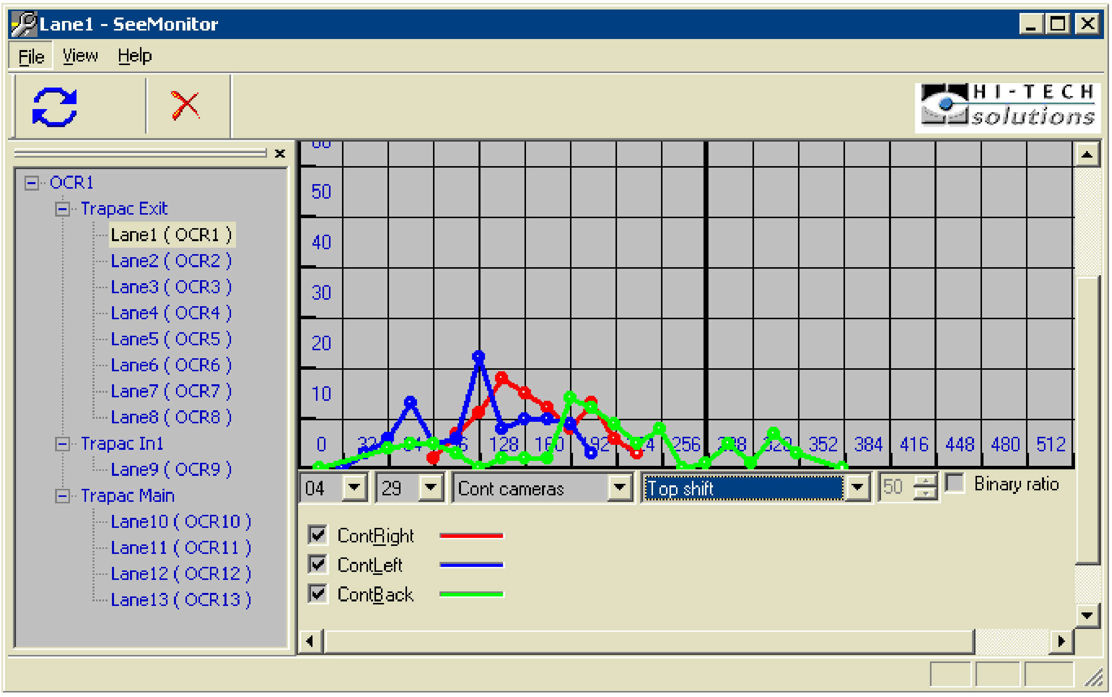
On the **lef**t side there is a list of all the lanes, and the user may select a specific lane to get additional graphical displays or to access the event log.

4.1.2.2.Sample graph display

There are several graphs available per lane. The following example shows the “Top shift” graph, which summarizes for an entire day (in this example, 04-29) for a selected subject (container camera) a histogram of the top shift (the position of the top-left corner of the container marking in relation to the top-left corner of the image).

There are 3 graphs (in blue, green and red) that show a histogram for each of the container cameras (red=right; blue=left; green=back).

The horizontal axis units are pixels (picture elements in the image). The vertical units are the percent of the total number of recognitions.



Lane1 - SeeMonitor

File View Help

- OCR1
  - Trapac Exit
    - Lane1 (OCR1)
    - Lane2 (OCR2)
    - Lane3 (OCR3)
    - Lane4 (OCR4)
    - Lane5 (OCR5)
    - Lane6 (OCR6)
    - Lane7 (OCR7)
    - Lane8 (OCR8)
  - Trapac In1
    - Lane9 (OCR9)
  - Trapac Main
    - Lane10 (OCR10)
    - Lane11 (OCR11)
    - Lane12 (OCR12)
    - Lane13 (OCR13)

04 | 29 | Cont cameras | Top shift | 50 | □ Binary ratio

☑ ContRight 🟥
☑ ContLeft 🟦
☑ ContBack 🟩

Figure 4.1.2.2 Sample graph display (lane 1 - top shift parameter)

In the above example the red graph (right camera) has a peak in about 140 pixels, so the average cases of recognition the container ID top-left corner was shifted downwards at 140 pixels.

This graph is useful to detect cases where the camera is looking **too low** or **too high**. If this happens the histogram will not be bell-shaped (a normal distribution) and rather will be cut at the left side (means the camera is looking too low and misses higher markings) or cut at the right side (means the camera is looking too high and misses lower markings).

This is a perfect tool to analyze the installation over time, and useful for both post installation and for maintenance.

4.2.SeeMonitor Data flow

4.2.1.SeeMonitor Inputs

a. Recognition log file (*.log, which contains ascii debug information on each event)
b. Recognition summaries in diagnostic data file (*.dat, a binary file with summary per hour)

c. Events logged (as information, warnings or errors) in the Windows application event log

Note: The ".log" and ".dat" files collect the debug and summary information required for SeeMonitor. Each See$_x$ (SeeLane/SeeTruck/SeeGate/SeeCrane) PC will write to the files according to the lane number (1 thru 6) where the recognition event happened.

The See$_x$ application requires that all the .log files are present in its directory prior to execution. Otherwise, it will assume that no debug information should be collected. Therefore, you need to set up the .log files prior to execution, or quit the See$_x$ application to start collecting.

Note that the files are not updated immediately since See$_x$ applications waits for 25 events before writing to the lane$_i$.log file(s), and one hour of operation for the lane$_i$.dat file(s).

4.2.2.SeeMonitor Controls

a. **State.ini** - a control file with description of warnings and errors and the state machine for determining the lane status (it is general for all See$_x$ systems).

b. **Monitor.ini** - a control file which defines the See$_x$ node and log/dat files (needs to adapt for each configuration)

4.2.3.SeeMonitor Outputs

a. Status "lights" for each lane (in red/yellow/green state, where: red=error, yellow=warning, and green=normal).

b. Event log access for each lane and display of last error/warning.

c. History graphs (recognition % per camera per hour of day; top/bottom, left/right shifts, etc.).

4.3.Running the application

4.3.1.Installation

System Requirements: PC 586 or higher with a minimum of 128MB RAM running Windows NT/2000.

a. Copy the directory to your disk.

Note: if you are copying from the CD, be sure to change the settings of all the files to read/write (unmark the read-only property).

b. Update the monitor.ini file for your network (see below)

c. Be sure that on each PC the log/dat files exist (see below).

d. In case of a central server, install the Hasp Driver (since it wasn't installed before). Insert the HASP plug into the parallel or USB ports (according to dongle type). *Note: the dongle must be ordered with the SeeMonitor option set into it, and the maximum number of units it monitors. Be sure to specify this in the purchase order. If it is not installed, the application will run for a limited trial period.*

e. Make the See$_x$ application directory sharable (using the right button on the Windows explorer), and assign it a name (for example : "Release").

4.3.2.Operation

a. Activate the SeeMonitor.exe file.

b. A window should open with a status display. For each lane you will see a 3-level status, the title of the lane, the network name of the See$_x$ PC, the connection status, and the last warning/error that caused the yellow or red to appear.

This display will refresh itself in a periodic rate defined in the configuration file (default: every minute). You can press F5 to refresh immediately, although it is recommended not to do so when there is a recognition event.

c. To check each lane graphs, click on the lane title on the left side. Then select the desired date (month and day), camera type, and graph type.

d. To check the lane's event log, press on the event log button in the bottom of the lane display.

4.4.Graph types

The following graphs currently exist:

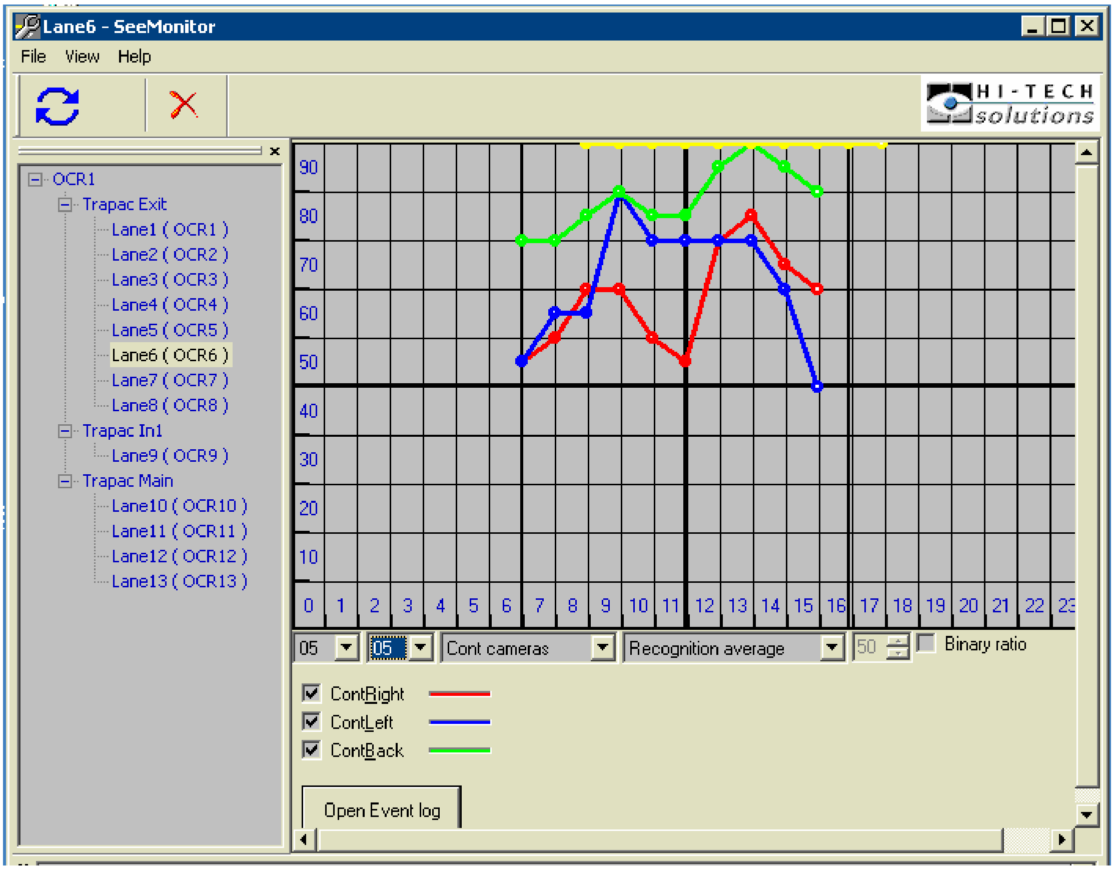
4.4.1. Recognition average:

This graph shows the daily recognition hourly summary (% per hour and day for each camera in that lane). The recognition % is the number of events with 'good' recognition divided by the total number of recognition events (which are listed per hour on the right side).

Note that 'good' recognition means that the application output any result, but there might still be wrong characters. However, this is a good indication of the quality of each camera - shown in red, blue (and optionally green) graphs.

An additional yellow graph shows the net recognition (all cameras combined), which is usually higher than each single camera.

A sample graph is shown below. In this graph the lane 6 recognition % verses time (24 hours on May 5) is shown in 4 colors: yellow is the overall recognition (100% during the whole day), green is the back container camera (80-100%), red is the right side camera (55-85%) and blue is the left container camera (50-90%).



- OCR1
  - Trapac Exit
    - Lane1 (OCR1)
    - Lane2 (OCR2)
    - Lane3 (OCR3)
    - Lane4 (OCR4)
    - Lane5 (OCR5)
    - Lane6 (OCR6)
    - Lane7 (OCR7)
    - Lane8 (OCR8)
  - Trapac In1
    - Lane9 (OCR9)
  - Trapac Main
    - Lane10 (OCR10)
    - Lane11 (OCR11)
    - Lane12 (OCR12)
    - Lane13 (OCR13)

05 | 05 | Cont cameras | Recognition average | 50 | □ Binary ratio

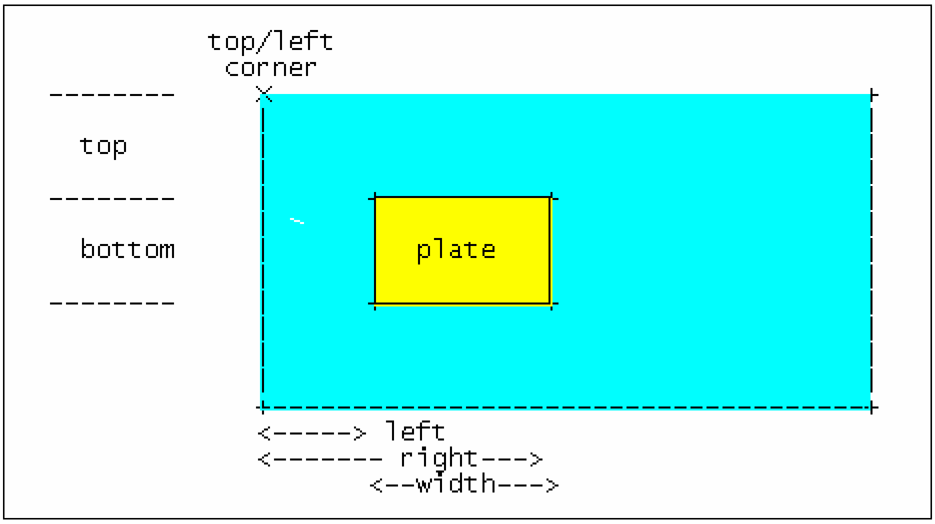
☑ ContRight 🟥
☑ ContLeft 🟦
☑ ContBack 🟩

Figure 4.4.1: Sample Recognition rate graph

4.4.2. Top/Bottom/Left/Right shift:

A histogram of all the recognitions, shows where the plate (or Container/Chassis ID marking) was found relative to the image top/left side.

The following illustration shows how these parameters relate to the image:



You can use these graphs to check for cases where the plate is cut outside of the effective field of view.

4.4.3. Width/Height

These graphs show the histogram of all the recognition events of the width or height of the plate/marking.

You can use these graphs to check that the average plate size is good (for example, 150 pixels for most widths for Car/Truck plates).

4.4.4.Capture start position

These graphs show a histogram of the index of the first recognized plate.

You can use this graph to verify that the first or last images in the capture set are within the proper range or field of view.

4.4.5.Number Recognized

This graph shows a histogram of the number of recognized images for each event.

You can use this graph to verify that the first or last images in the capture set are within the proper range or field of view.

4.4.6.Horizontal/Vertical Center

This graph is a histogram of the center of the plate- relative to the left/top side.

You can use this graph to verify the field of view is centered.

4.4.7.Brightness

This graph shows a histogram of the average brightness of the entire image.

You can use these graphs to verify that the proper settings of the Iris and the frame grabber settings.

4.4.8.Zone Contrast

This graph shows a histogram of the average contrast within the plate/ID zone, by calculating the standard deviation. This measure shows how well the fonts stand out in the background of the plate/ID. A higher value is preferred for best recognition. Usually a value of 50 grey levels is best.

You can use these graphs to verify that the plates are properly illuminated and that the captured images are in good quality.

4.4.9.Brightness/Contrast verses Hour of day

The average image brightness or zone contrast is plotted against the hour of day.

These graphs may be used to check for different installation checks, such as a camera with closed Iris or ineffective illumination. The graphs are very useful and should be checked from time to time to observe operation under different environment conditions.

4.5.Other tools

SeeMonitor includes other utilities that assist in the site management.

4.5.1.Expert System Analysis

A new feature added into SeeMonitor is ESA – Expert system analysis. This tool is based on a set of predefined rules that control its operation. The set of rules form a list of rules that are used to analyze the data, execute the rule operators, and then come up with the target results.

These rules cover the following topics:

- distorted histograms – due to wrong camera positions or bad triggering
- low recognition rates
- bad brightness values

More information on this tool will provided in future revisions of this document.

4.5.2.Autoprint Graphs

4.5.2.1.Plot per time period

A useful graph can be generated in the main status display options. This graph shows the daily change of a total recognition for the entire site. It was initially built for use in the terminals using an auto-print data flow, but it can be used also to calculate and display a total recognition value for the whole site.

More information on this graph will provided in future revisions of this document.

4.5.2.2. Daily Graph

This graph plots a number of factors (that build up the total recognition calculation) for each hour of a requested date. This graph is useful to observe which factor affects the total recognition and when.

4.6.Configuring the operation

4.6.1.Overview of monitor.ini

This monitor.ini configuration file is a Windows configuration initialization file, which defines the parameters that control the SeeMonitor tool.

The file contains several fields: [General], [Cluster_1],..[Cluster_n] where n is the number of clusters defined in the parameter "No_Clusters".

Each cluster is a separate set of gates which is important to separate them in the display (for example, 2 buildings of the same company).

4.6.2.General Section

a. **No_Clusters**=1: defines the number of "clusters" (default is 1)

b. **Update_Period**=60: defines the rate in seconds to update the status display (default value is 60 seconds)

c. **Offline=0**: defines working mode (0 - real time monitoring of remote systems, 1-offline charts drawing, default - 0)

d. **MonitorId=1**: defines the system ID that is transferred via network

e. **TcpPort=3002**: defines TCP/IP port for network communication

4.6.3.Cluster section (for each cluster)

a. **No._lanes**: defines the number of lanes in the cluster (e.g., 20)

b. **Title**: a string that is used as a title (e.g., North Gate)

c. For each lane i (where i=1 to No_Lanes):

1. **Address**i: the network name (e.g., Server1)

2. **Share**i:

   The shared directory name where the See$_x$ application is running.(e.g., Release"). Be sure that the directory is set as shared using the Windows Explorer (with the right button).

If the SeeMonitor is running on the See$_x$ PC, or there are no sharing restrictions in the LAN, you can leave this parameter empty.

3. **Titlei** : the title of the lane (e.g., west exit lane 3).

4. **Filei**: the log file in that lane (e.g., lane6.log). Note that the filename is always lanei.log where i is the lane number starting from 1.

5. **Diagi** : the diagnostic file name in the lane (e.g., lane6.dat). Note that the filename is always lanei.dat where i is the lane number - starting from 1.

4.6.4.Example:

A sample monitor.ini file is provided below.

The configuration below assumes 2 servers (server1 & server2) with 3 lanes each (each numbered lane1 thru lane3), and connected as follows:

```
+--- Server1
|         |
|         +------- Lane 1
|         +------- Lane 2
|         +------- Lane 3
|
|
+--- Server2
          |
          +------- Lane 1
          +------- Lane 2
          +------- Lane 3
```

A sample monitor.ini file :

```
[General]
No_Clusters=1
Update_Period=60

[Cluster_1]
No_Lanes=6
Title= Gate

Address1=Server1
Share1=
Title1=Entrance 1
File1=lane1.log
Diag1=lane1.dat

Address2=Server1
Share2=
Title2=Entrance 2
File2=lane2.log
Diag2=lane2.dat

Address3=Server1
Share3=
Title3=Entrance 3
File3=lane3.log
Diag3=lane3.dat

Address4=Server2
Share4=
Title4=Entrance 4
File4=lane1.log
Diag4=lane1.dat

Address5=Server2
Share5=
Title5=Entrance 5
File5=lane2.log
Diag5=lane2.dat

Address6=Server2
Share6=
Title6=Entrance 6
File6=lane3.log
Diag6=lane3.dat
```

4.6.5.Additional configurations

4.6.5.1.Change of camera names

You can change the names of the cameras from their default names, by adding parameters (in each cluster). This is useful for special systems such as SeeTrain or SeeCrane where the cameras are used for different functions, and the user would like to analyze the graphs with the proper functional names rather than the default names.

The syntax of the camera name change is:

```
   defaultname = newname
```

For example,

```
  CarFront1=FrontPlate
```

The original names that are used by SeeMonitor are:
CarFront1, CarFront2, CarBack, ContRight, ContLeft, ContBack, TruckUppRight, TruckUppLeft, Chassis1, Chassis2.

4.7.Remote Reporting

SeeMonitor can report its status to external management systems, according to a TCPIP socket protocol that is explained in this section.

This reporting interface may also be used to allow SeeMonitor to send messages to service technicians in order to alert on the system status.

4.7.1.The messages protocol

4.7.1.1.TCPIP socket

All communication between the two systems (External-Management-System and SeeMonitor) is performed through an Ethernet network with TCP/IP protocol. SeeMonitor listens on TCP port 3002 (a parameter that can be changed in the configuration file).

4.7.1.2.Message format

Message is in the following format in text strings:

[STX]$[SeeMonitorID]$[MessageLength]$[MessageType]$[MessageContent]$[ETX]

- Parameters in [Message Content] are separated by %.
- [Message Length] is the number of bytes (characters) of the entire message without STX and ETX.
- [SeeMonitor ID] is the ID of the SeeMonitor software, an optional parameter (see note below).

4.7.1.3.Multiple sites

Each SeeMonitor monitors an entire site (a number of lanes of different Seex systems, such as SeeLane or SeeGate). However, there may be more than one site, such as across the Country. All sites can be optionally connected to the same External-Management-System. The SeeMonitor ID is 1 for the first SeeMonitor and is incremented by 1 for any additional SeeMonitor.

4.7.2. Messages to SeeMonitor

	Format		**Description**
#	**Message Type**	**Message Content**	
1.1	InitReq		This message will be the first one to be sent to establish the connection. *SeeMonitor answers with message #2.1 (Init).*
1.2	CommunicationReq		External-Management-System tests the communication with SeeMonitor. The only response from SeeMonitor can be message #2.2 (CommunicationOK). If no message is sent back External-Management-System will consider it as a loss of communication and will keep sending this massage until it is answered.
1.3	LastErrorReq	[Lane Number]	Gets the last error from SeeMonitor (if any). SeeMonitor answers with message #2.4 (LastError) This message should be sent in 3 cases: 1.When the system starts. 2.After communication loss, when communication is established again. 3. Every N minutes. N can be configured and its default value is 5. If [Lane Number] is zero SeeMonitor will send LastError results for all lanes.

4.7.3.Messages from SeeMonitor

#	Format		Description
	Message Type	**Message Content**	
2.1	Init	‘2’ | ‘1’ | ‘0’ |	SeeMonitor sends this message once it receives message 1.1. Content: ‘2’ = In Progress, ‘1’ = OK, ‘0’ = FAIL
2.2	Communication	OK	SeeMonitor will send this message once it receives message #1.2
2.3	Error	[Lane Number]% [Error ID]	SeeMonitor sends this message whenever it detects an error. Note that some of the errors are success errors and cancel previously sent errors. [Error ID] is a string. MSB is the first byte.

2.4	LastError	[Number of Lanes]%
		{
		[Lane Number]%
		[Error ID]
		}
			SeeMonitor sends this message once it receives message #1.3 (LastErrorReq).
			This message specifies the last error (if any) for the lane specified by [Lane Number] or for **all** lanes if [Lane Number] in #1.3 was zero. In the latter case the structure between the brackets repeats itself for all lanes.
			If there are no errors - SeeMonitor sets [Error ID] to 0.
			[Error ID] is a string. MSB is the first byte.

2.5	MessageLost		SeeMonitor sends this message in case it has a failure in sending one of its messages between 2 successful #1.2 (CommunicationReg) messages. Note that if the failure was before an unsuccessful #1.2 message External-Management-System will move into communication loss state. *Once External-Management-System receives this message it will send message #1.3 to SeeMonitor, so it will get all errors.*

4.7.4. List of events

The following table shows a sample list of error events (indicate a lane problem, and correspond to the “red” status light in the display) and success events (that cancel the error events, that correspond to the “green” status). Note that the “yellow” status warnings are not reported.

This list may change from time to time. A separate document will detail the updated list.

4.7.4.1.Success events:

Success ID	**Description**
1	Application started. *This success event cancels other errors.*
3	The technician has enabled the sensors. *This success event cancels error 312.*
41	Recognition rate is normal. *This success event cancels error 542.*

4.7.4.2.Error events:

Error ID	**Description**
312	The technician has disabled the sensors. This is done for local testing of the system.
500	One of SeeLane DLLs initialization failed.
503	Memory allocation failed.
506	Manufacturer internal software error.
507	Internal error.
521	Memory allocation failed.
540	Memory allocation failed.
541	Unrecoverable error in one of SeeLane DLLs.
542	Recognition rate is low. The recognition percentage rate is less than the one configured in SeeLane settings.

5.SeeService

5.1.Main objectives

The main objective of the SeeService application is to provide support to other Company applications (See$_X$, such as SeeGate, SeeCrane or SeeLane). These objectives include:

1. Periodic application checks

   SeeService performs supervision of a specified See$_x$ application - by periodic check of normal application execution. This is useful to prevent the application to crash or get stuck.

2. Version control

   SeeService adds the capability to update some components of See$_x$ application automatically for a number of units connected on the network.

   The application checks from a specified update directory if there is a change of one or more files and if so - it will stop the current execution (within 10 seconds), update the files, start the application again, and write this event to the system event log file.

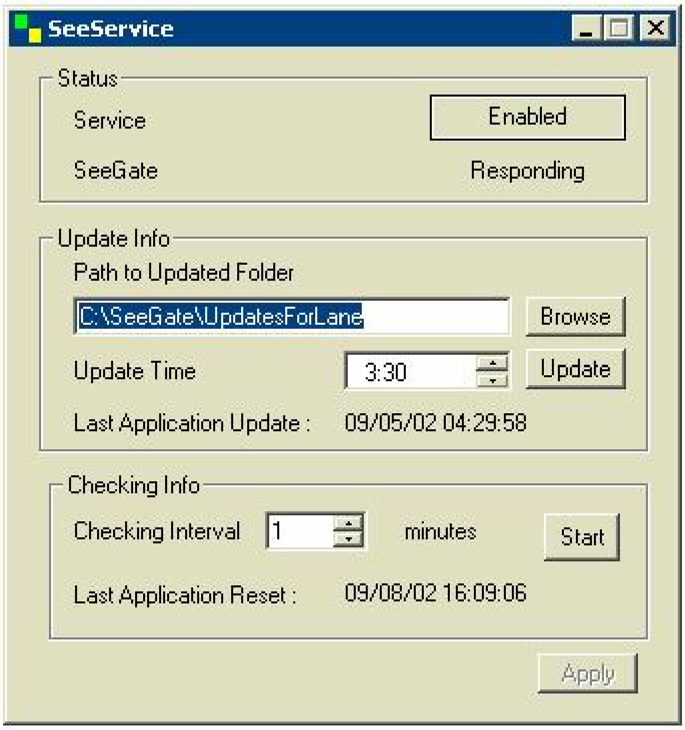
Notes:

1. Only one application can be serviced by SeeService.
2. SeeService runs as a background active application, and could be seen as an icon in the bottom application tray. To check the application - the icon should be maximized.
3. In case the technician wants to stop the supervised application, the SeeService must first be disabled – otherwise SeeService will detect that it stopped and will automatically rerun it.

5.2.Sample Display

A sample display of the application can be seen below. In this example the status of the SeeService utility is “enabled” (it could be disabled), and the supervised application is “SeeGate”.

The update path is specified, and it is checked automatically at 3:30 AM (or, it could be manually updated using the update button). The last application reset was 09/05/02 at 04:29:58.



Service: Enabled

SeeGate: Responding

Path to Updated Folder: C:\SeeGate\UpdatesForLane

Update Time: 3:30

Last Application Update : 09/05/02 04:29:58

Checking Interval: 1 minutes

Last Application Reset : 09/08/02 16:09:06

Figure 5.2: Sample display of SeeService

The SeeGate check is performed every 1 minute, and the last time it was resetted was 09/08/02.

5.3. Running the application

5.3.1.Operation

To execute the SeeService, activate the SeeService.exe file. Usually it should be activated from Startup.

5.3.2.Button control

'Enabled\Disabled - switch the service operation status. This is useful if a temporary halt is required to a supervised application.

'Browse' - define the updating directory (the source of the update files)

'Update Time' - edit this control to determine the periodic update time

'Update' - update immediately the application components (without waiting for the periodic update)

'Checking Interval' - edit this control to set time interval between successive checks of the supervised application (e.g., 1 for each minute)

'Apply' - confirm the changes (not relevant to 'Update')

5.3.3.System Requirements

PC 486 or higher, with a minimum of 16MB RAM, running Windows 2000/XP.
Win 95/98 are not supported on new releases and might result in run-time errors.

6.SeeCleaner

6.1.Main objectives

The objectives of the SeeCleaner utility are to provide periodic cleaning of accumulated debug and image files generated by a See$_X$ application (such as SeeGate or SeeLane). Thus, the disk space will be kept at reasonable size, while leaving the debugging and monitoring capabilities active.

The application either **trims the size** of the large debug and diagnostics files, or **clears** the contents of other log files (while backing up the files first in its backup subdirectory), or removes old image directories. The parameters for the cleaning are specified in the setup procedure.

6.2. Running the application

6.2.1.Installation

The entire package can be Zipped with WinZip, so it should be first opened if it is zipped.

Use the Setup.exe to install the application. It will guide you through a set of windows (which most require clicking 'Next'). The last window requires to set "Launch program" in order to let it start running.

The setup program can also be activated from the server disk (use the Windows explorer to double-click on the setup program residing on a server node).

Note that in case the application is already installed, you should first uninstall it (see below).

6.2.2.Operation

If you checked the 'Launch program' option at the installation, cleaning is performed automatically once per day . Note: you may set this functionality by running the utility from command line by 'SeeCleaner /s'.

In addition, you may run SeeCleaner utility manually:

a. Activate the SeeCleaner.exe file.
b. Determine desired parameters.

c. Press 'Save' to update the parameters.
d. Press 'Execute' to perform cleaning.

6.2.3.Optionally checking the operation

You can optionally check that the program is properly scheduled. Select in Windows in the following order: "Start" -> "Settings" -> "Control Panel" -> "Scheduled Tasks", and view if the utility is scheduled (normally at 600am).

After this time, you can see that the files have been either trimmed or cleared.

6.2.4.Uninstalling

Use the control panel add/remove programs to uninstall the program, or to upgrade the application.

6.2.5.System Requirements

PC 486 or higher, with a minimum of 16MB RAM, running Windows 2000/XP. Win 95/98 are not supported new releases and might result in run-time errors.