NATIONAL TRANSPORTATION SAFETY BOARD
Office of Research and Engineering
Washington, D.C. 20594

April 8, 2002

Specialist's Computer Study

SCADA Control System

A. ACCIDENT

Location: Bellingham, Washington

Date: June 10, 1999

Time: 1530 Pacific Daylight Time

Pipeline: Olympic Pipeline Co.

NTSB #: DCA-99-MP-008

Chairman: James R. Cash
Electronics Engineer
National Transportation Safety Board

C. SUMMARY

On June 10, 1999 at approximately 1530 local Pacific Daylight saving time a 16" pipeline transporting liquid gasoline owned by the Olympic Pipeline company ruptured. The pipeline had a remote supervisory control and data acquisition (SCADA) system installed. During post accident interviews with Olympic personnel, including one of the on-duty controllers, they reported that the system became unresponsive and sustained several outages during the accident time. These host computer files and the SCADA system was investigated to see what effect its operation had on the pipeline accident.

D. BACKGROUND

The SCADA system that was in operation at the time of the accident was a SCADA Vector system, manufactured by Teledyne Brown Engineering of Huntsville, Alabama. The software running at the time of the accident was Vector system 3.6.1

The overall SCADA software function can be broken down into five high level functions. To accomplish each of these functions, there are many associated sub-processes or programs running on each host machine. On an operating SCADA system, there might be over 30 of these individual sub-processes resident on both computers and running on each of the two computer system. Each of these sub-processes accomplishes a particular task that makes up the whole SCADA system. The general functions are listed in order of decreasing priority within the SCADA system.

1. Communications function: This set of sub-process manages the communication with the data sensors through programmable logic controllers (PLC) or remote terminal units (RTU) located along the pipeline. Depending on the particular sensor and what it was measuring, the sensors are queried by the communications sub-process at a rate of once every several seconds to once every several minutes.
2. Information Store function: This set of sub-process manages the database that stores all of the data used by the SCADA system. This sub-process coordinates the storage of the incoming data from the communication sub-process and supplies data to other sub-process requesting data.
3. Display and Control function: This set of sub-process manages the bi-directional human interface to the SCADA system. It's functionality is closely tied to the communications processes. It processes the data that is sent to the controller's screens. In addition, it also receives commands issued by the controllers at their individual workstations and sends the command information to the appropriate device in the field.
4. Data archiving function: This set of sub-process ensures that all of the data used by the SCADA system is stored in permanent archive records.
5. Miscellaneous Operation and Housekeeping Function: The system contains several sub-processes that run at scheduled intervals that accomplish routine housekeeping functions such as moving data or printing daily/hourly reports. There was also a sub-process running that extracted data out of the SCADA system and supplied it to a separate leak detection software program.

Communications Function

The communication sub-process function is the main backbone of the SCADA system. This function collects and manages the data that is coming in from several hundred individual sensors through a number of PLC's and RTU's scattered all along the pipeline. In addition to the collection function, the communication sub-process also manages the various commands that are sent from the SCADA system to the field units.

The Olympic system was configured with two types field data collections units. The first type is a "dumb" (RTU) collection unit. This system is classified as "dumb" because it contains no local computer decision software. It contains no programming that would enable it to make local decisions without first communicating with the controllers through the main SCADA computer system. The main function of these units is to assemble the individual parameter data from its local site and communicate that data to the main SCADA system. The unit would also process any commands that come from the main SCADA system and executes those commands locally. For example, if this unit was located at a remote pumping station it would collect the 20 or 30 sensor data parameters associated with the station. These parameters could be incoming pipeline pressures, outgoing pressures, flow rates, status of individual valves, pumps and other control devices. The unit would then assemble all of these individual sensor parameters and coordinate the transmission of this data over a public or leased telephone, radio or some other communication link to the main SCADA computer. The unit would also accept operator commands from the main SCADA system over the same communication links. The local unit would then attempt to carry out these commands such as turning on a pump or changing a valve at the local site. The main SCADA computer would get verification that the command was carried out successfully when it received its next batch of parameters from the site. It would check to see that the status of the pump or valve changed appropriately in accordance with the command that was given. Even though this unit is classified as "dumb" it can contain some basic safety features. It can have 'never to exceed' limits programmed into it. If a sensor exceeds one of these safety limits it will shut down a pump or do some other function locally to protect the facility even if it was not in contact with the main SCADA system. These safety features are used to protect the basic facility in the event of a communication failure when the local controller cannot communicate with the main control system. In addition to this capability, many stations have local controllers embedded in their motor controls which are designed to protect the station equipment and nearby downstream piping. At the time of the accident, the majority of the field units that were installed on the Olympic system were of this type.

The second type of field unit that was used on the Olympic pipeline system was a "smart" or PLC field controller. This type of controller carries out the same basic functions as the "dumb" controller but has the added advantage of being able to make local decisions based on its programming without having to rely on the main SCADA system to tell it what to do. This more expensive and more complex unit is usually employed at the larger terminal and pumping facilities. These facilities usually contain many more sensors and control options that need tracking. The advantage of this type of controller is that it can automatically perform pre-programmed tasks without the intervention of the main SCADA system. An example of this might be to monitor a tank level and cycle a pump or a valve to maintain a preprogrammed level. Even though the local "smart" controller is responsible for carrying out a pre-programmed task, it still communicates with the main SCADA system and continuously provides raw data from the local site. The main SCADA system has the ability to override some of the automatic PLC functions. This unit may also have basic 'never to exceed' safety limits programmed into it. These limits are again set to protect the local facility in the event of lost communication with the main controlling system.

The communication functions at the SCADA computer consists of a bank of data receiving units. These receiver units perform two functions. They interface with the local telephone or radio communications link and temporarily store the incoming field data. They then provide the field data to the SCADA computers when they are ready to receive it.

The SCADA data collection sub-system can be set up in several ways when it comes to receiving data. Some parameters, especially ones that are very dynamic and are constantly changing, are sampled at a pre-programmed constant rate. This sample rate might be as high as every few seconds for a very dynamic parameter or it might be as low as several times per minute for a less active parameter. The second sampling scheme requests that the parameter value be transmitted only when it has changed by a set amount. This sampling technique improves communication link efficiency drastically reduces the amount of data that needs to be communicated and processed by the SCADA system. On some parameters that are very stagnant, there is no need to continuously send the same value over and over every several seconds. The designers predetermine a dead band threshold, where a varying data value must exceed the threshold, up or down, before it is necessary to be reported. Periodically a new value is automatically sent even though it hasn't changed just to make sure that the system has the correct value and to verify that the sensor is still alive and active out in the field.

As part of the communications front end of the SCADA system, many parameters must go through a conversion and scaling processor. This processor takes the raw values that are sampled in the field and scales and converts this raw data into engineering units[1] that are familiar to the human controllers. For example, the raw sensor in the field might measure pressure of the pipeline in linear counts from 1 to 1000 corresponding to a pressure of 0 to 500 pounds-per-square-inch (PSI). This count value is transmitted to the SCADA system. If the raw count value were displayed to the controller, he would not have a good idea of what that count value really meant. He might be looking at several hundred different pressure values and each one is represented by a different count value. The communications processor takes the raw values and converts them into the correct pressure in PSI and displays that value to the controller. In that way all of his pressure values are scaled to the same display units.

The final function of the communications processor is to take the converted engineering unit data from the field and put them into the information store of the computer. This is the basic raw data that the other sub-processes use to perform their functions.

Information Store

The information store of the SCADA system consists of several sub-processes that are constantly moving data to and from various locations. The Information store's main function is to manage the centralized storage location used by the SCADA system. For efficiency reasons, most of the current data being used by the SCADA system resides in the physical memory of the host computer. While this scheme speeds up the processing of the system it also leaves the system venerable to losing the data in the event of a failure. To prevent such a loss the SCADA system maintains a duplicate copy of the "in-memory" data on the computers hard disk drive. In the event of a system failure, upon re-starting, the

[1] Engineering units are common, easily understandable measurements units. I.E. motor current in Amps, flow in barrels per hour, pressure in pounds-per-square-inch etc.

SCADA would read this disk file and should be able to pick up operation where it left off prior to the failure.

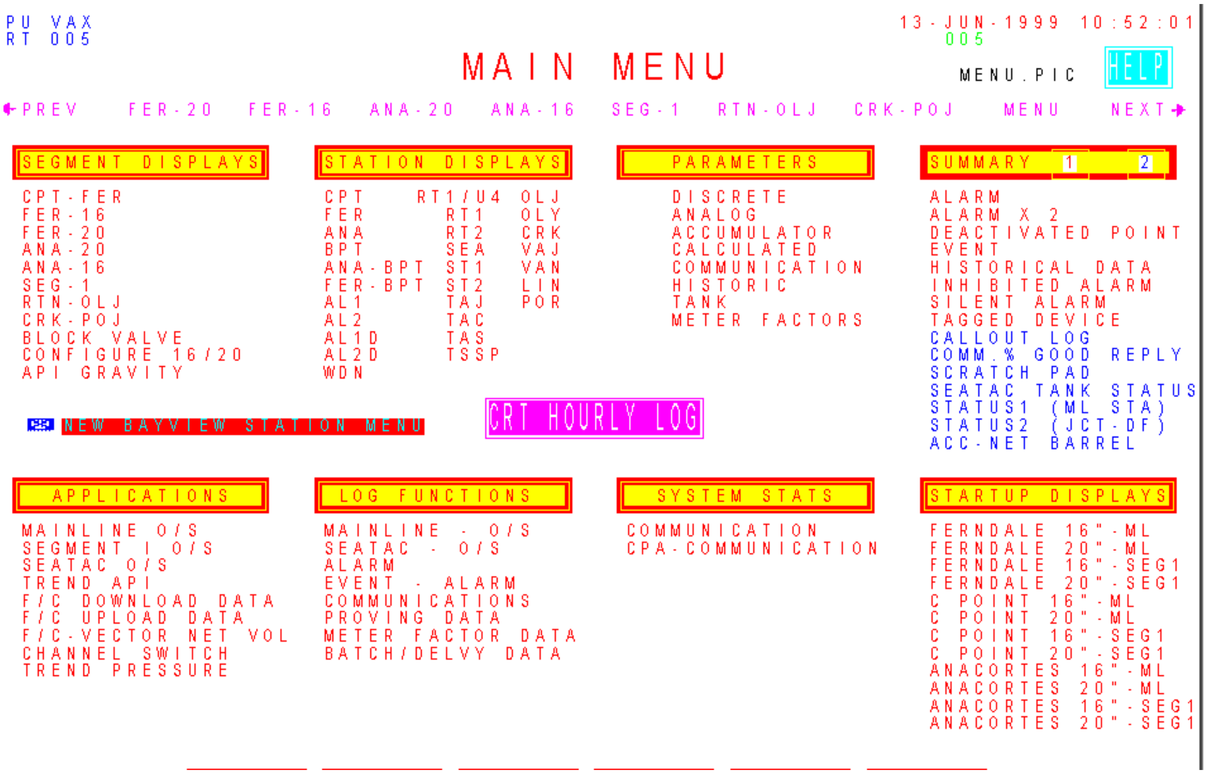
Another function of the information store is to communicate with the back up computer to make sure that both computers have the same data. This again is a backup/safety feature of the SCADA system. In the event of a failure of the primary computer, the backup system will assume the role of primary and the overall SCADA system will continue to function. If the information store function is operating properly the primary and backup computers, depending on where it is in the update process, should have nearly the same data and the change over should be relatively transparent. In the Olympic system to test this automatic fail over, the primary computer is stopped once a week. The system is monitored to make sure that the backup system assumes the role of primary and that the operation of the SCADA system continues with no loss of data or functionality. The two computers switch roles for the week when the current primary system is intentionally failed and the other system assumes the primary role for the next week.

Display and Control

The function of the display and control portion of the SCADA system is to provide a user-friendly interface between the computer and the operating controllers. The SCADA system collects hundreds of individual parameters from the various field sensors. This data needs to be presented to the human controller in a fashion that allows an individual to make intelligent decisions on that data. In turn it takes those human decisions, in the form or commands, and passes them back to the field units for execution. As part of this human interface various symbols, lines, colors, and text are used to represent the state of the field data. The controllers in the Olympic system had over 40 different pre-programmed screens that were available. Some of these screens covered large portions of the pipeline and some of the screens focused on certain aspects of the operation. Additionally there were specific display screens that were developed for use in startup or shutdown of the pipeline or for the transfer/storage stations. Any of the 40 screens were available on any of the computer workstation but each workstation is only able to display one SCADA screen at a time. In addition to the pre-programmed screens, the controllers could construct “on the fly” custom displays if the need arose. These custom screens are constructed by the controller issuing several commands to the SCADA system. This feature would be used to view and trend information that wasn’t normally available on a pre-constructed screen. A common example of this capability is when a custom trend display is constructed which would plots a system parameter or multiple parameters over some time period. Graph 1[2] is an example of what the main menu Olympic Pipeline controller’s SCADA screen looked like at the time of the accident

[2] The colors in the graph do not represent the original colors that were shown on the controllers workstation display. The original color scheme used primary symbol colors on a black background. To include these pictures in this report, the original color scheme was inverted to display colored symbols on a white background.

GRAPH 1



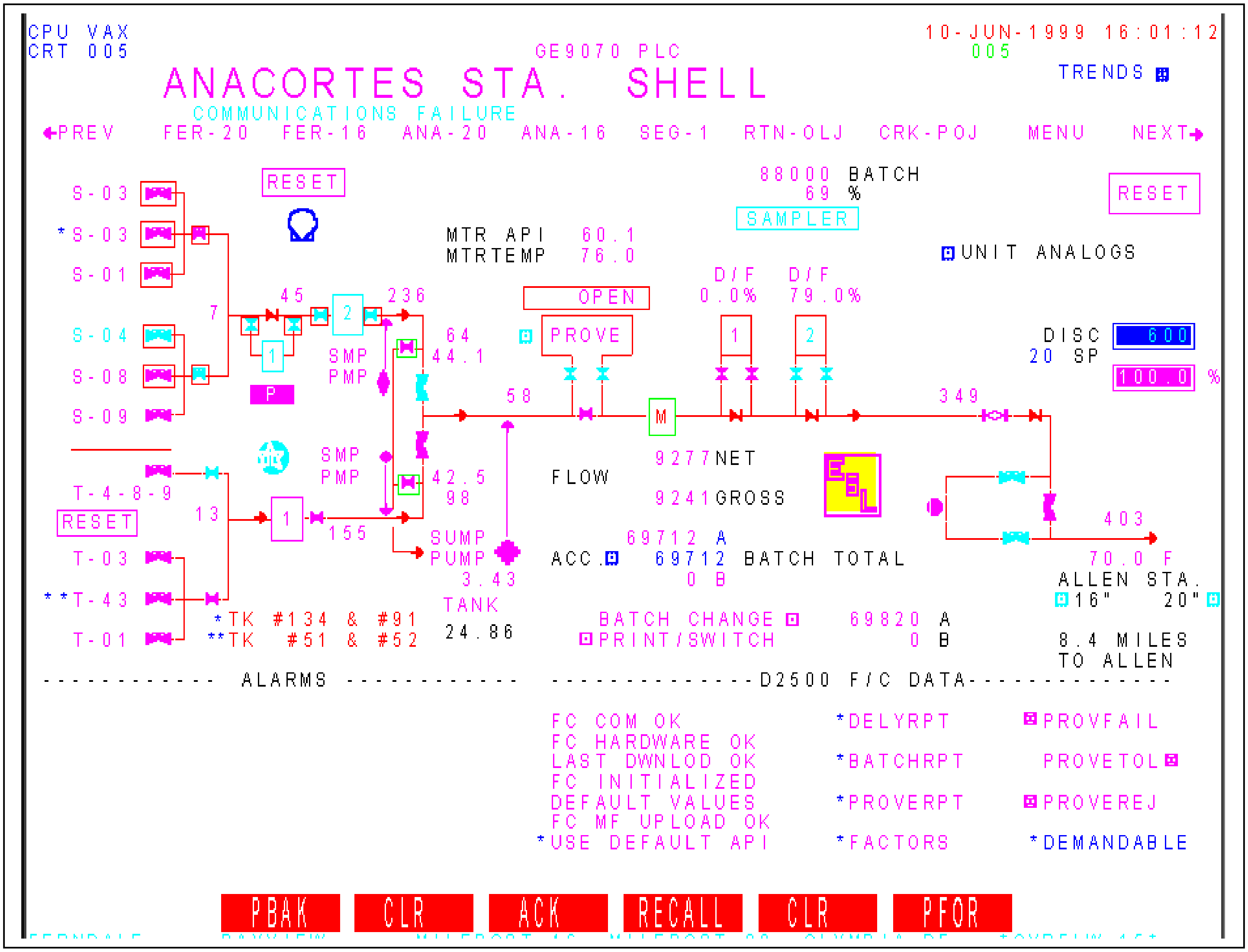
The main menu screen is divided into several sections. Each section leads the controller to another similar sub-menu. The "New Bayview" station sub-menu can be seen in red in the middle of the menu. It should be noted that not all of the selections were active. Several choices, especially in some sub-sub-menus were not yet defined and produced an error message when they were selected.

Graph 2[3] represents a "typical" station display. The symbols and lines shown represent the various valves, pumps, tanks etc.. that reside at the station. The numbers next to some of the symbols represent the corresponding pressures, flow rates, temperatures, pump motor currents, tank levels, etc.. that are associated with the station's operation.

In addition to displaying the information the controller can click on various symbols to enter in a new setting for that specific parameter. For example by clicking on a valve symbol, the operator is given the choice of changing the position of that valve from its current position to either open to close. Pressure set points could be increased or decreased, pumps could be turned on or off, all by clicking on the representative symbol on the menu.

[3] The colors in the graph do not represent the original colors that were shown on the controllers workstation display. The original color scheme used primary symbol colors on a black background. To include these pictures in this report, the original color scheme was inverted to display colored symbols on a white background.

Graph 2



FC COM OK
FC HARDWARE OK
LAST DWNLOD OK
FC INITIALIZED
DEFAULT VALUES
FC MF UPLOAD OK
*USE DEFAULT API

To strengthen the human interaction with the displays, the Vector SCADA system design incorporated several features. One feature that could be enabled is the color of the displayed data or symbol could change depending on the data associated with the symbol. For example when a pressure would reach a particular point the data symbol might turn from white to yellow. As the pressure continued to rise it might change from yellow to red and then if it continued to rise it would start flashing. These features if programmed, help to alert a controller to a changing condition prior to the condition exceeding a limit. While these and many more features were designed into the system, it was up to Olympic's management as to how many and what features would be used.

In addition to the changing symbols, definite warning and alarm limits could be set in the displays. When a value exceeded one of these limits, the SCADA system would issue a alarm message to the controllers workstation. There was usually an associated bell or attention getting buzzer that would sound to alert the controller that there was an alarm message being displayed. Each individual parameter could be configured to use any or all of the available features of the SCADA software.

The Olympic system in general did not utilize all of the human-machine enhancements that were available in the SCADA system. There was no clear pattern of what features were being used or not used. It was not clear from a review of the screens or from the documentation or interviews of the Olympic personnel why they chose not to use certain features.

An additional function of the display sub-process was to take the commands issued by the controllers and to pass those commands through the system and to the appropriate field device.

Data Archiving

One major design philosophy of the Vector SCADA system was that selected data that was used by the system and all of the commands issued by the system should be stored in permanent records. To accomplish this task there are many permanent disk records kept which contain copies of the incoming data as well as copies of all the commands issues by the controllers. These records are organized by type and by date. Every day, just after midnight local computer time, an entire set of historical records is created for that day. As the day goes on these files are then appended with the various data that comes in. At midnight of the following day another new set is created.

The primary purpose of data archiving is to ensure that there is a permanent record of the day's events. In addition to this function the stored data is also used for other functions. The operational controllers use this stored historical data to generate plots that "trend" data spanning many hours. The SCADA system is designed to give the controller a picture of what is currently happening in the field. The computer's memory contains the current data by only keeping the most recent value. If the operator wants to know what happened previously, then the system needs to process this historical data for that information. (In the Olympic system the leak detection software used this stored historic data to monitor for potential leaks). In addition to operational uses, the pipeline managers use this data for various accounting, billing and strategic planning programs.

Miscellaneous Operation and Housekeeping

The miscellaneous functions comprise all of the other background sub-processes of the SCADA system. At any given time, there might be several sub-processes that might be running. While the jobs do run on the current operational system, they normally run in the background at a lower system priority then the primary operational tasks.

VAX/VMS Operating System

The Digital Equipment Corporation (DEC) VAX/VMS operating system is a multi-user operating system that was designed for the VAX family of computers back in the late 1970's. The operating system is a true multi-user system in that it contains all of the features that allow many people to be connected to the system all running independent jobs or processes. The operating system is designed to keep all of the individual users functioning independently and insulating their data from one another.

The VMS operating system installed on the Olympic system, had received numerous updates since its initial inception. These updates were in the form of both major and minor revisions. The VAX family is composed of a wide range of computer platforms. They range from small mini-computers to large mainframe computers all running the same operating system (VMS). The DEC VAX-VMS family of computers and operating system have received the highest security rating from the Department of Defense, allowing it to be able to store secret and top-secret data. Historically, the VMS operating system and VAX hardware platform have shown to be very reliable. A properly written application might run continuously for months or years without needing to be reset or re-started.

System Organization

The VMS operating system storage is organized into three distinct functions or areas. These areas are 1) the core operating system, 2) the user or application area and 3) a common area.

The VMS core area is where all of the files that make up the operating system are stored. This area is normally not accessible to the casual user. Any modifications or changes to these files require that the operator have the highest level of authorization or system administrator privileges. This is the area where all of the accounting, error, security, and system log files are kept.

The user area of the computer system is where the individual user is allocated a specific portion of the computer for their use. The user area could be as simple as a folder where the user stores data. It also may consist of many folders and may form a large complex program area. The SCADA system as setup by Olympic Pipeline resembled the later example. In this installation the entire SCADA system was setup in a user area called "vector". This area contained the executable programs that make up the SCADA system as well as the data areas that stored all of the data and log files that were used and created by the SCADA system.

A common area in a VMS system is created so that many people can run a common program or application against their specific data. This concept cuts down on the number of duplicate, usually large, applications that are stored on the system. The operating system is constructed to permit many users to share the same application at the same time. As far as the individual user is concerned it appears that he has un-impeded use and access to the program. In reality, many individual users may be running the same program at the same time. The operating system is sophisticated enough to keep track of sometimes hundreds of users, or subroutines, all simultaneous running the same application.

Process Management

To manage all of the individual programs running on the VAX computer the system needs to allocate its resources carefully. Even on a large mainframe computer system that is supporting hundreds of users there is only one central computing unit. This core-computing unit is capable of running only one program at any given time. In order to support all of the programs that need to run on a large system the management unit swaps process in and out of the core-computing unit. This swapping happens so fast that it appears to the user that he has exclusive use of the machine when in reality his individual process might have been swapped in and out of the core unit thousands of times. This swapping is managed by assigning priorities to all of the process that are waiting for their turn in the core-computing unit. The priorities are set from 1 to 15, where 1 is the lowest and 15 being the highest. Most system level processes run at a high priority of 13 to 15. They have priority over the normal user assigned priority of 5 to 10. Other non-time critical jobs like printing and batch jobs usually run at a low 1-5 priority. It is possible for a user function to run at a higher priority but this must be managed carefully in order to balance the necessary system functions with the user program.

The configuration of the Olympic system was that the major key components of the SCADA sub-processes were running with a high VMS system priority of 12-15. This included the communications and information store sub-processes. Other less critical SCADA sub-processes were running at a high-medium VMS priority of 8-10

In addition to the real-time interactive processing of programs, the VAX/VMS computers also have a queue manager that scheduled low priority batch jobs for execution. There are several queues usually setup on a VAX computer. One queue is set up to run the printers. Another queue is setup to run batch jobs. Batch jobs normally are large jobs that if run in the real-time environment would consume vast amounts of resources and would dramatically slow down the computer response to other users waiting for their results. The queue manager breaks down these large jobs and schedules them for execution so as not to interfere with normal interactive jobs. These jobs can also be scheduled to run late at night when other demands on the system are low. Batch jobs usually run at the lowest system priority, and it is the job of the queue manager to make sure that the batch job doesn't interfere with normal operational response of the system. This normal balance can be upset by a batch job that has been intentionally submitted with a higher than normal priority.

File Structure

The VMS operating system has a file structure very similar to that, which is found on a conventional PC computer. Each individual disk drive is organized as a multi-level file , under which many sub-directories may be established. The disk file structure appears to be an upside down tree containing one main trunk with hundreds of branches off of the main trunk. Each sub-directory may contain individual files but they also may contain additional sub-directories. This organization of sub-directories under sub-directories is limited only by available disk space. The VMS file structure contains several additional features that are not found on a conventional PC disk drive. Because the VAX has to support many users, there are additional security and accounting restrictions that accompany all of the files and directories on a VMS system. Each file or directory in the system belongs to a unique owner. In addition to who owns the file, the date and time that the file or directory was created and when it was last backed up or modified is also kept with the file. The owner can control access to each file or directory. The owner can specify access to a unique individual user a group of users or permit the world or all users access. The access can be restricted even further by specifying that a file is read only, read-write access or read-write-delete access. Any combination of access restrictions can be assigned to an individual file or directory or to any particular group of files or directories.

Another feature that is unique to the VMS operating system is that many files with the exact same name can be created. VMS uses a similar file naming convention that a PC uses. It has a common name followed by a period with an extension. For example an error file might be called "system_error.dat". The system_error being the common name portion of the file and the .dat the file extension. In order to keep track of the same file names, VMS appends a version number to the end of the file name. The system appends a semicolon followed by an increasing number starting at 1 for each new version of the file. Each time the file is saved the version number increases by 1 from the last time it was saved. The same error file that was created multiple times or was saved multiple times would have a file name of "system_error.dat;1" for the first file and "system_error.dat;2" for the second one. The operating system would keep each copy or version of the file until the owner specifically deleted or purged old versions of file. By default the VMS operating system is set up not to automatically purge file versions. This default setting can be changed to only keep the most recent "X" number of versions. The keep version number (X) is configurable by an individual owner or the administrator can set it as a new system-wide default setting. As it was described above, VMS treats each version as a new and unique file and thus keep a record of when the file was created or modified for each of the versions The way that the operation system operates on files is that if a file was opened for read only access, a new file would not be created with a new version when the program was finished with it. If a file was opened with write or delete access, a new version of the file would be created when the file was closed.

The Olympic system was set up with the default VMS system file attributes. There was no automatic purge or version limits enforcement set in any of the configuration files. This means that the system by default would keep all version of a file, as described above. Additionally it would not purge or erase any older files after a period of time. It should be noted that the owner of the files or some other user with sufficient system privileges could manually delete files or purge older versions of files at any time. File purging can also be automated by a system administrator. A system command file could be periodically run to rid the computer of older or unnecessary files. This could be accomplished on a daily or a weekly schedule.

VMS System and Accounting Logs

Due to the multi-user aspect of the VMS operation system numerous system log files are kept. These files keep track of the amount of processing time a user consumes. This information might be used for billing a client for CPU time. The system also keeps track of any system operation that was accomplished. These operations include print jobs completed, batch jobs that were accomplished. The system also keeps track of any system errors that were encountered by a program. It might be as simple as a software error because a file was not found, when a program tried to access it. It also contains serious hardware failures, which could be associated with a disk drive fault or some other hardware failure inside of the computer. The system also makes generic placeholder entries in the system logs when the computer first starts up and opens the file, and at periodic intervals when the system is operating. Each entry in the log contains a short descriptive message along with the system time and date when the entry was made.

VMS also contains a security log that keeps a record of who is logged into the system. The security log would contain an entry if someone were attempting to break into the operating system. Each time a user types an incorrect user name or password a break-in entry is made in the security log. VMS, by default, allows the user 6 tries at getting their password correct when they login in. If a user tries more than 6 times that particular account is locked out for a period of time. The system is designed to thwart programs or users that randomly try likely passwords in an attempt to gain access to system.

E. DETAILS OF INVESTIGATION

According to testimony obtained from the SCADA controller and the programmer/system administrator that were on-duty at the time of the accident the VAX/SCADA control system had several periods where it appeared unresponsive to the controllers commands. The system was examined to determine the extent of the reported slowdown and what caused the reported unresponsiveness.

The configuration of the computer system, software, hardware and control room at the time of the accident was as follows:

Configuration at Time of Accident

Host Computer:

- Two identically equipped and configured Digital Equipment Corporation VAX model 4000-300 computers
- Computer operating system: Digital Equipment Corporation VAX/VMS operating system version 7.1
- Each computer had 256 Mb system memory
- Each computer had a RF-31 384 Mb system disk drive
- Each computer had a (2) RF-31 384 Mb user disk drives
- Each computer has a TK70 backup tape drive
- Each computer has a TU-81 9-track tape drive
- Network communications were provided by native VAX/VMS DECNET and 3rd party TCP/IP protocol

In addition to the 2 main SCADA computers an additional similarly configured Digital Equipment Corporation DEC Alpha 3000 computer running Alpha/VMS was used as a host for the separate leak detection software package.

SCADA System:

- SCADA Vector Software version 3.6.1
- 15 PC Control Terminals were configured
  - o 8 SCADA display stations running DOS vector display
    - CRT01 OLYPC1
    - CRT02 OLYPC2
    - CRT03 OLYPC3
    - CRT04 OLYPC4
    - CRT05 OLYPC5
    - CRT06 OLYPC6
    - CRT07 OLYPC7
    - CRT08 OLYPC8

  - 7 SCADA display stations running Microsoft Windows NT displays
    - CRT09 OLYPC9 VAX MOTIF
    - CRT10 REN_PC_DISP1 # Dispatch segment 1 workstation
    - CRT11 REN_PC_DISP2 # Dispatch segment 2 workstation
    - CRT12 REN_PC_SCADA # Computer room pc
    - CRT13 OLY03 DEC Alpha 3000
    - CRT14 PC14
    - CRT15 PC15
- Each VAX computer running identical Vector software: one computer in a primary and the other running as a "hot" backup

At the time of the incident the SCADA system was running on the computer called OLY02 (primary). The 2nd computer, called OLY01 (backup), was running in a "hot" spare configuration. This means that the pipeline data was being processed on the primary machine and was being passed to the backup machine. In this configuration, if the OLY01 machine had sensed a failure in the OLY02 machine it would have asserted itself as primary by taking control from the OLY02 machine. This changeover is quite fast and it is almost transparent to the controllers. The two computers are constantly exchanging operational data between each other. This exchange serves two purposes. First it keeps both machines "current" with respect to the pipeline data. If one should fail the other already has most of the data needed to carry on the operation of the pipeline. The second purpose of the data exchange is as a metric used between the two computers as a measure of proper operation. If the data is not exchanged in a timely manor the backup machine will assume that there was a failure in the primary computer and take control of the operation. As long as this exchange progresses in the time frame set up by the administrator of the system each computer is happy and each other is assumed to be okay. The two computers communicate with each other over an "Ethernet" style network connection using Digital Equipment Corporation VAX/VMS Decnet protocol.

Network

The network configuration that was in place at the time of the accident consisted of a basic Ethernet one-backbone network. This means that each device was connected to one common connection point and that there was only one path from any one device to another. Each VAX computer was connected directly to the one backbone. Each of the control room computers were also connected to this backbone. The network protocol that was being used between the various computers was a combination of Digital Decnet and Internet standard TCP/IP protocol. Each of these protocols can share the same physical Ethernet network. The PC's that were installed in the control room for the operators used both of the protocols.

Eight terminals, CRT1- CRT8 were running the DOS[4] version of the Vector display software. They communicated with the host VAX computers via Decnet protocol. The other 7 PCs were running Microsoft Windows NT software and used the windows version of the SCADA Vector display software. They communicated with the host computers via TCP/IP protocol. In addition to the two SCADA computers, there was an additional DEC Alpha computer that was dedicated to running the leak detection software. This computer was also communicating using Decnet protocol and was directly connected to the Ethernet backbone network.

Also connected to the Ethernet backbone was a bridge whose purpose was to connect the operational Ethernet segment found in the SCADA control room with the rest of the building network. The building network is reported to have some Internet connectivity. It was reported that there were several other departments in the Olympic Pipeline company that used data that was obtained from the SCADA system. This data was electronically transferred from the SCADA system computers to other department computers. A bridge device offers some protection and isolation of one segment from the adjacent segment. This protection is mostly from hardware network failures and faults. A hardware failure or fault in one segment will not affect the adjacent bridged segment. While a bridge does offer some software isolation from a casual intruder it does not offer the same protection that a full-featured intrusion firewall would afford.

Connected to each of the VAX computers in the Olympic SCADA system were 2 console terminals. These terminals are a combination keyboard and printer, similar to an older style teletype terminal. They don't have a display normally found in a more familiar PC computer terminal. Everything that is either typed on the keyboard or is sent from the computer is printed out on the printer. These devices are used to provide a permanent printed copy of all of the commands sent to the computer and a print out of all of the messages from the computers. Any information inputted or outputted on either of these terminals would produce a hard copy printout on the terminal. The ports that they are plugged into are special console ports. These ports are part of the main central processor and are the most direct connection to the heart of the computer. These ports are usually responsive regardless of any other program or operation that the computer may be executing at the time. Additionally this is the default location where all system generated message would be sent. If a hardware or software error were detected the computer would log the error in a file and send the same message to the console terminal. In addition to the log file this is where any VAX-VMS level user password failures or detected break-in messages would be sent.

The VAX computer systems also have additional terminals and workstations connected to them. These individual terminals and workstations are connected to the VAX's by using either network connections or to the VAX directly by using one of the several serial communications ports located on either VAX. Most of the day-to-day programming and development would be accomplished, using one of these remote terminals. There would be no key-stroke record of the commands entered via one of the remote terminals or workstations.

[4] Microsoft Disk Operating System DOS

In Addition dial-in access to the VAX computers was provided by a modem that was connect to the VAX communications port. This provided direct dial-in access to the VAX computers from the outside, provided the user knew the phone number and had an authorized dial-up account and password on the VAX system.

LOG Files

The SCADA system was reported as running normally throughout most of the day on June 10. The controllers reportedly did not notice any abnormal operation of the SCADA system. The system administrator was working on a terminal in the computer room programming some new reports that extract data from the SCADA database. He said in his testimony that he submitted to the batch queue manager on the operational computer the new reports that he had just composed. When he returned after a few minutes he reported the primary computer as unresponsive. In addition he stated that the console terminal did not respond to his commands. Additionally the controllers were reporting that the SCADA system was not updating the control screens like it normally does. Because of this apparent unresponsiveness, he reported that he “crashed” the primary computer in hopes that the backup would pick up the load and things would return to normal. He stated that it appeared that the same condition affected the backup computer because it didn’t pick up the load as he had hoped it would. After several minutes he “crashed” the backup (now primary) OLY01 computer. He said in his testimony that he had suspected that the problem with the computers was caused by an error in his new report that he had submitted a few minutes prior to the initial slowdown. He also stated that he surmised that the error had also infected the backup (OLY01) computer because they constantly share data. After OLY02 had rebooted, he stated that he deleted the new historical batch job and that after several minutes the system had returned to normal.

Backup tapes from the Olympic system were examined to see if they contained any information about the reported computer slowdown. The backup plan that Olympic had at the time of the accident was as follows: A daily backup was made of all of the new and modified files found in the SCADA area on both OLY01 and OLY02 computers. This backup was usually accomplished early in the morning around 6:00 AM. This backup was accomplished while the systems were operating. A weekly backup of the entire Vector system area was made of both machines every Monday. This Monday backup coincided with the weekly change over from one computer being prime to the other.

The backup tape files of the VMS operating system that were examined by the Safety Board, were made after the time of the accident on June 10, on July 14, 1999 and again on July 26, 1999.

The log files from the VMS operating system and from the SCADA system were the first files to be examined. The text report generated from the June 10 log files included in the appendix of this report. It can be seen from the VMS log files from both OLY01 and OLY02 computers that there were no unusual errors or activity until the systems were restarted at

1531 system time. A listing of the VMS log files showed that several versions of the Accounting.dat log files on both OLY01 and OLY02 computers were missing. Additionally several versions of the Operator log files on both OLY01 and OLY02 were missing. Table 1 lists the VMS operating logs that were examined from both the OLY01 and the OLY02 computers:

TABLE 1

Computer system	File Name	File version	File Size blocks	Backup File	File Creation date	File Creation Time
VMS						
OLY01	Accounting.dat	80	49	Oly1vms_990510	27-Apr-1999	16:00:00
OLY01	Accounting.dat	81	10	Oly1vms_990510	27-Apr-1999	17:47:00
OLY01	Accounting.dat	82	203	Oly1vms_990510	27-Apr-1999	17:43:00
OLY01	**Accounting.dat**	**83**	**missing**			
OLY01	**Accounting.dat**	**84**	**missing**			
OLY01	**Accounting.dat**	**85**	**missing**			
OLY01	**Accounting.dat**	**86**	**missing**			
OLY01	**Accounting.dat**	**87**	**missing**			
OLY01	Accounting.dat	88	72	Oly1vms_990614	10-Jun-1999	9:38:00
OLY01	Accounting.dat	89	47	Oly1vms_990614	16-Jun-1999	15:58:00
OLY01	Accounting.dat	90	136	Oly1vms_990626	14-Jun-1999	11:16:00
OLY01	Accounting.dat	91	58	Oly1vms_990626	21-Jun-1999	9:32:00
OLY01	Accounting.dat	92	48	Oly1vms_990626	23-Jun-1999	4:44:00
OLY01	Accounting.dat	93	47	Oly1vms_990626	24-Jun-1999	10:11:00
OLY01	Accounting.dat	94	53	Oly1vms_990626	25-Jun-1999	13:22:00
OLY02	Accounting.dat	70	43	Oly2vms_990510	27-Apr-1999	16:57:00
OLY02	Accounting.dat	71	122	Oly2vms_990510	27-Apr-1999	17:20:00
OLY02	Accounting.dat	72	83	Oly2vms_990510	3-Mar-1999	9:10:00
OLY02	**Accounting.dat**	**73**	**missing**			
OLY02	**Accounting.dat**	**74**	**missing**			
OLY02	**Accounting.dat**	**75**	**missing**			
OLY02	**Accounting.dat**	**76**	**missing**			
OLY02	**Accounting.dat**	**77**	**missing**			
OLY02	Accounting.dat	78	44	Oly2vms_990614	1-Jun-1999	10:02:00
OLY02	Accounting.dat	79	44	Oly2vms_990614	8-Jun-1999	10:52:00
OLY02	Accounting.dat	80	6	Oly2vms_990614	10-Jun-1999	15:30:00
OLY02	Accounting.dat	81	100	Oly2vms_990614	10-Jun-1999	16:30:00
OLY01	Operator.log	60	18	Oly1vms_990510	26-Apr-1999	17:44:00
OLY01	Operator.log	61	68	Oly1vms_990510	27-Apr-1999	19:17:00
OLY01	**Operator.log**	**62**	**missing**			
OLY01	**Operator.log**	**63**	**missing**			
OLY01	**Operator.log**	**64**	**missing**			
OLY01	**Operator.log**	**65**	**missing**			

OLY01	Operator.log	66	11	Oly1vms_990614	16-Jun-1999	9:34:00
OLY01	Operator.log	67	11	Oly1vms_990614	14-Jun-1999	9:57:00
OLY01	**Operator.log**	**68**	**missing**			
OLY01	**Operator.log**	**69**	**missing**			
OLY01	Operator.log	70	70	Oly1vms_990626	23-Jun-1999	4:16:00
OLY01	Operator.log	71	71	Oly1vms_990626	23-Jun-1999	4:40:00
OLY02	Operator.log	52	34	Oly2vms_990510	27-Apr-1999	17:16:00
OLY02	Operator.log	53	41	Oly2vms_990510	3-May-1999	9:06:00
OLY02	**Operator.log**	**54**	**missing**			
OLY02	**Operator.log**	**55**	**missing**			
OLY02	**Operator.log**	**56**	**missing**			
OLY02	**Operator.log**	**57**	**missing**			
OLY02	Operator.log	58	48	Oly2vms_990614	1-Jun-1999	9:55:00
OLY02	Operator.log	59	30	Oly2vms_990614	10-Jun-1999	15:32:00
OLY01	Error.sys	1	773	Oly1vms_990510	4-Nov-1998	9:18:00
OLY01	Error.sys	1	838	Oly1vms_990614	4-Nov-1998	9:18:00
OLY01	Error.sys	1	878	Oly1vms_990626	4-Nov-1998	9:18:00
OLY02	Error.sys	1	812	Oly2vms_990510	13-Nov-1998	12:45:00
OLY02	Error.sys	1	879	Oly2vms_990614	13-Nov-1998	12:45:00

The operator log from the OLY01 computer that covers the time frame of the reported slowdown is missing. Again OLY02 was the primary machine at this time. By referring to Table 1 it can be seen that the last version of the Operator log on OLY01 was created on April 27, 1999 this was version 61 of the log. The next Operator log, version 66 was created on the 16th of June. No explanation was obtained from any of the individuals interviewed, why the files were missing from the backup tapes. A complete listing of the files contained on the individual backup tapes is included in the appendix of this report.

The core VAX-VMS operating system and the programs that make up the SCADA Vector pipeline control system are highly sophisticated systems. While the various computer components do monitor and log some operator activity, they do not however record individual keystrokes inputted by the system users. The monitoring and logging that is accomplished is limited to recording higher level events such as when programs or jobs start and stop, when individuals login or exit. The system also records information associated with any abnormal hardware or software operation or failures.

It should be noted that when the operating system starts up, it checks to see that there is a valid log file for it to use. If it doesn't find an old file, it will create a new file at that time. While the system is operating, the current in-use log file is locked. This means that you may look at the log file's contents but you cannot edit or delete it. VMS does however provide a means of closing a currently active log file. When an authorized system administrator user issues the command, the operating system will release the old file, close it and create a new file with a version that is one higher than the old file. The Error.sys file in the above table illustrates this point. The operating system will continue to use the existing file if it is available. The only Error.sys files that were found on both systems were created in November of 1998. As long as the operator did not manually close this file, the system on every re-boot continued to use it. It would append any new data to the end of the existing file. Common system administration practices would be to periodically manually close the various system log files. This forces the system to create a new version of the files. This routine practice helps to keep the size of the log files to a manageable level and protects historical log activity from being inadvertently destroyed by being overwritten.

SCADA System Records

As part of the normal SCADA system operation several files are kept that contain various historical, error and operating information that the SCADA system generates. This information is kept in a series of files that are created every day by the SCADA system. For example, the historic files where all of the incoming sensor readings are stored are created just after midnight. The way that the system is configured, the SCADA system creates the files a day ahead of when it is going to be needed. For example on day June 10 1999 the system would create the files that were labeled June 11th. These files would sit there all day not being used until midnight when they would start to receive the data for that day. When the system changed over to the next day it would then create the files for the next day June 12. The SCADA system resembles the VMS operating system in that if the SCADA system restarts at some point during the day it will try to use the current days log files, if they exist. If they don't exist, the SCADA system will create a file for the current day at that time. It will also look to see if the files are there for the next days use. If those files don't exist it will also create them at the time of restart.

Table 2 lists the SCADA historical files that were found of the various backup tapes.

TABLE 2

OLY01 990610 Backup

Directory [OLY01_990610.VCS.SYSTEM.HIST]

AVG990607.HIST	1	34	7-Jun-1999	8:39:59.95	7-Jun-1999	8:39:59.95
AVG990608.HIST	1	18	7-Jun-1999	8:40:05.22	7-Jun-1999	8:40:05.84
INS990601.HIST	1	3582	31-May-1999	0:00:00.70	7-Jun-1999	8:11:27.62
INS990602.HIST	1	3768	1-Jun-1999	8:40:42.70	7-Jun-1999	7:29:20.07
INS990606.HIST	1	5028	5-Jun-1999	0:00:10.65	7-Jun-1999	8:03:21.66
INS990607.HIST	1	4440	6-Jun-1999	0:00:02.15	8-Jun-1999	0:05:05.60
INS990608.HIST	1	2940	7-Jun-1999	8:40:02.61	9-Jun-1999	0:05:13.84
INS990609.HIST	1	4542	8-Jun-1999	0:00:03.23	10-Jun-1999	0:05:05.92
INS990610.HIST	1	30	9-Jun-1999	0:00:12.61	9-Jun-1999	0:00:13.74

INS990611.HIST	1	30	10-Jun-1999	0:00:04.66	10-Jun-1999	0:00:05.82
MAX990607.HIST	1	34	7-Jun-1999	8:40:01.91	7-Jun-1999	8:40:01.91
MAX990608.HIST	1	18	7-Jun-1999	8:40:05.96	7-Jun-1999	8:40:06.57
MIN990607.HIST	1	34	7-Jun-1999	8:40:00.48	7-Jun-1999	8:40:00.48
MIN990608.HIST	1	18	7-Jun-1999	8:40:06.72	7-Jun-1999	8:40:07.34
ROC990607.HIST	1	34	7-Jun-1999	8:40:01.66	7-Jun-1999	8:40:01.66
ROC990608.HIST	1	18	7-Jun-1999	8:40:08.17	7-Jun-1999	8:40:08.82
SNP990607.HIST	1	34	7-Jun-1999	8:40:01.40	7-Jun-1999	8:40:01.40
SNP990608.HIST	1	18	7-Jun-1999	8:40:04.47	7-Jun-1999	8:40:05.10
SUM990607.HIST	1	34	7-Jun-1999	8:40:00.68	7-Jun-1999	8:40:00.68
SUM990608.HIST	1	18	7-Jun-1999	8:40:07.46	7-Jun-1999	8:40:08.07

OLY01 990611 Backup

Directory [OLY01_990611.VCS.SYSTEM.HIST]

AVG990610.HIST	1	34	10-Jun-1999	15:59:31.07	10-Jun-1999	15:59:31.07
AVG990611.HIST	1	18	10-Jun-1999	15:59:36.81	10-Jun-1999	15:59:37.44
INS990601.HIST	1	3582	31-May-1999	0:00:00.70	7-Jun-1999	8:11:27.62
INS990602.HIST	1	3768	1-Jun-1999	8:40:42.70	7-Jun-1999	7:29:20.07
INS990606.HIST	1	5028	5-Jun-1999	0:00:10.65	7-Jun-1999	8:03:21.66
INS990607.HIST	1	4440	6-Jun-1999	0:00:02.15	8-Jun-1999	0:05:05.60
INS990608.HIST	1	2940	7-Jun-1999	8:40:02.61	9-Jun-1999	0:05:13.84
INS990609.HIST	1	4542	8-Jun-1999	0:00:03.23	10-Jun-1999	0:05:05.92
INS990610.HIST	1	4458	9-Jun-1999	0:00:12.61	11-Jun-1999	0:05:07.75
INS990611.HIST	1	30	10-Jun-1999	15:59:34.57	10-Jun-1999	15:59:35.99
INS990612.HIST	1	30	11-Jun-1999	0:00:06.48	11-Jun-1999	0:00:07.64
MAX990610.HIST	1	34	10-Jun-1999	15:59:33.06	10-Jun-1999	15:59:33.06
MAX990611.HIST	1	18	10-Jun-1999	15:59:37.56	10-Jun-1999	15:59:38.20
MIN990610.HIST	1	34	10-Jun-1999	15:59:31.70	10-Jun-1999	15:59:31.70
MIN990611.HIST	1	18	10-Jun-1999	15:59:38.31	10-Jun-1999	15:59:38.92
ROC990610.HIST	1	34	10-Jun-1999	15:59:33.62	10-Jun-1999	15:59:33.62
ROC990611.HIST	1	18	10-Jun-1999	15:59:39.79	10-Jun-1999	15:59:40.39
SNP990610.HIST	1	34	10-Jun-1999	15:59:32.07	10-Jun-1999	15:59:32.07
SNP990611.HIST	1	18	10-Jun-1999	15:59:36.09	10-Jun-1999	15:59:36.69
SUM990610.HIST	1	34	10-Jun-1999	15:59:32.57	10-Jun-1999	15:59:32.57
SUM990611.HIST	1	18	10-Jun-1999	15:59:39.04	10-Jun-1999	15:59:39.68

OLY02 990610 Backup

Directory [OL02_990610.VCS.SYSTEM.HIST]

INS990605.HIST	1	5778	4-Jun-1999	0:00:03.24	7-Jun-1999	16:08:38.35
INS990606.HIST	1	5028	5-Jun-1999	0:00:02.11	9-Jun-1999	8:08:41.54
INS990607.HIST	1	4566	6-Jun-1999	0:00:03.20	9-Jun-1999	20:17:25.11
INS990608.HIST	1	2958	7-Jun-1999	0:00:01.86	9-Jun-1999	8:09:13.41
INS990609.HIST	1	4542	8-Jun-1999	0:00:11.10	10-Jun-1999	0:21:39.98

INS990610.HIST	1	312	9-Jun-1999	0:00:15.32	10-Jun-1999	0:26:04.47
INS990611.HIST	1	30	10-Jun-1999	0:00:08.53	10-Jun-1999	0:00:25.24

OLY01 990611 Backup

Directory [OL02_990611.VCS.SYSTEM.HIST]

AVG990610.HIST	1	34	10-Jun-1999	15:40:22.93	10-Jun-1999	15:40:22.93
AVG990611.HIST	1	18	10-Jun-1999	15:40:28.05	10-Jun-1999	15:40:28.69
INS990605.HIST	1	5778	4-Jun-1999	0:00:03.24	7-Jun-1999	16:08:38.35
INS990606.HIST	1	5028	5-Jun-1999	0:00:02.11	10-Jun-1999	8:53:01.44
INS990607.HIST	1	4566	6-Jun-1999	0:00:03.20	10-Jun-1999	8:55:30.38
INS990608.HIST	1	2958	7-Jun-1999	0:00:01.86	10-Jun-1999	13:39:24.60
INS990609.HIST	1	4542	8-Jun-1999	0:00:11.10	10-Jun-1999	13:54:58.11
INS990610.HIST	1	4518	9-Jun-1999	0:00:15.32	11-Jun-1999	0:37:15.26
INS990611.HIST	1	414	10-Jun-1999	15:40:25.89	11-Jun-1999	0:38:45.58
INS990612.HIST	1	30	11-Jun-1999	0:00:08.49	11-Jun-1999	0:00:10.40
MAX990610.HIST	1	34	10-Jun-1999	15:40:21.47	10-Jun-1999	15:40:21.47
MAX990611.HIST	1	18	10-Jun-1999	15:40:28.80	10-Jun-1999	15:40:29.44
MIN990610.HIST	1	34	10-Jun-1999	15:40:21.70	10-Jun-1999	15:40:21.70
MIN990611.HIST	1	18	10-Jun-1999	15:40:29.55	10-Jun-1999	15:40:30.22
ROC990610.HIST	1	34	10-Jun-1999	15:40:22.27	10-Jun-1999	15:40:22.27
ROC990611.HIST	1	18	10-Jun-1999	15:40:31.10	10-Jun-1999	15:40:31.73
SNP990610.HIST	1	34	10-Jun-1999	15:40:21.93	10-Jun-1999	15:40:21.93
SNP990611.HIST	1	18	10-Jun-1999	15:40:27.25	10-Jun-1999	15:40:27.93
SUM990610.HIST	1	34	10-Jun-1999	15:40:21.09	10-Jun-1999	15:40:21.09
SUM990611.HIST	1	18	10-Jun-1999	15:40:30.35	10-Jun-1999	15:40:30.99

The table lists the files that were recovered from the VAX backup tapes. The first column is the file name. The 2nd column is the version number of the files. The 3rd column is the size of the file in VMS blocks[5]. The 4th and 5th columns give the date and the system time that the file was created. The last 2 columns list the date and the time that the file was last modified by a process or user.

The main storage file that the SCADA system uses to store historic sensor data is the INSxxxxxx.HIST file. Where xxxxxx is the year/month/day portion of the file name. Most of the INS historic files sizes and file characteristics recovered are consistent with what would be expected for a normally running system. The files were created a day prior to their use at about 00:00:00 system time. They are all about the same size. It should be noted that

[5] One VMS block equals 512 bytes

when a file is in use, the size of the file is incorrectly reported by the VMS system. For example, in the backup from OLY01 that was created in the early morning of June 10. The system should be storing data in the INS990610.HIST file. The file should contain about 6 hours of data when the backup was made yet the file size that was recorded was only 30 blocks, a file size consistent with an empty unused file. If the next day's backup is examined, the file size that is reported on the same file is similar to previous days file size. One exception that should be noted is the file INS990608.HIST found on the OLY01_990610 and 11 backups. This file has a file creation time of 08:40:00 on the 8th of June. The size of the file is about half of the previous days record. This means that 1) the SCADA system was restarted on June 8th and 2) the original historic file for that day was not found when the system restarted, 3) the system created a new file at 08:40:00.

These SCADA historical files were examined to see if any abnormal operation of the system could be documented. As stated above the data that this file contains is the sensor data from all of the field sensors located on the various elements of the pipeline. This data is periodically placed in the files when the operating computer scans that sensor. This scanning can happen a frequently as once every few seconds to as long as once every minute. It varies for each sensor in the system. In general, the scanning rate is determined by the dynamic nature of the sensor and how critical its value is to the operation and management of the SCADA system. This is the job of the communications sub-process. It scans the individual sensors and places the current values in the historical database. Other sub-processes read these historical files and use this information to complete the other functions of the SCADA system.

A small sample of what would be contained in the historic file is shown in Table 3. It was not possible to include a complete historic record in this report because of the size of the file. On any given day, each daily historic file from the Olympic system contained over 102,000 individual entries collected repeatedly from the several hundred physical sensors installed on the Olympic pipeline system.

TABLE 3

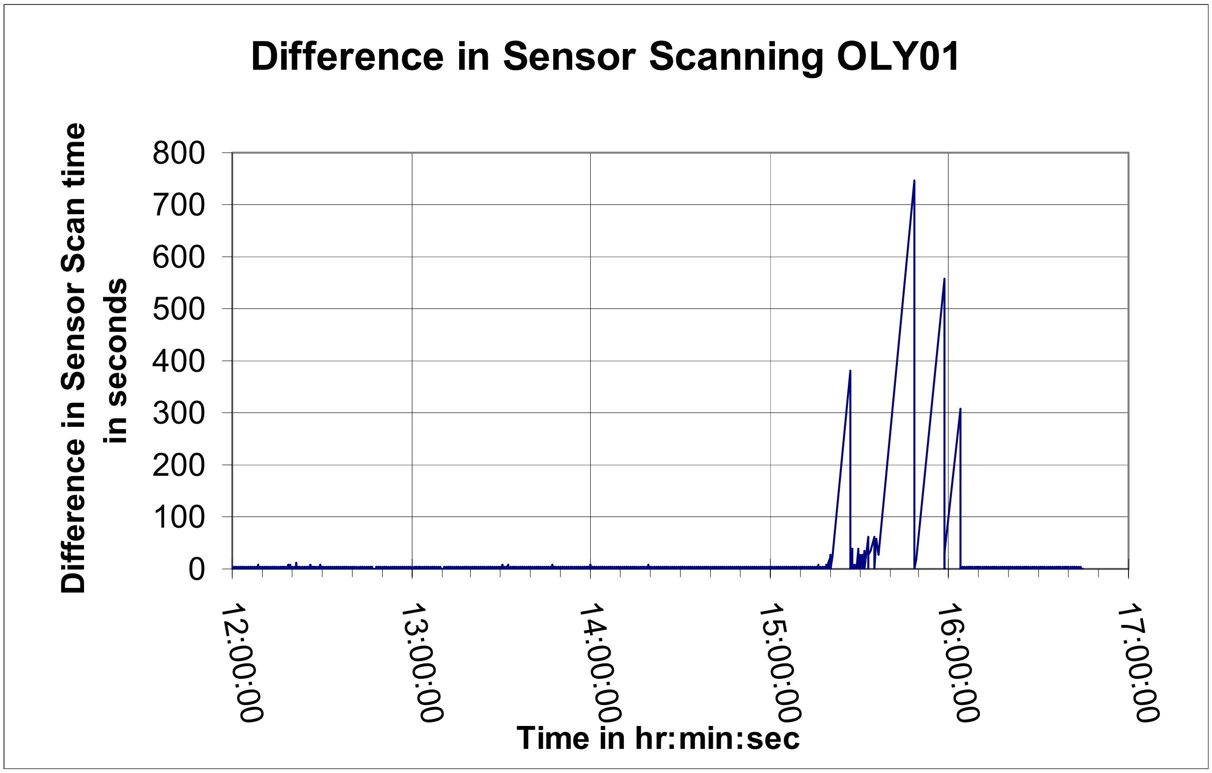
OLJHU2MVIB	15:07:27.19	0.258
RT1HSUMPGAGE	15:07:28.79	1.91
TASHU2MVIB	15:07:29.78	0.195
CHPHGRSFLOW	15:07:30.53	8710
FERHNETFLOW	15:07:31.65	8648
RT1HNETFLOW	15:07:33.46	8706
OLYHMTRTEMP	15:07:33.89	85.1221
OLJHU2PVIB	15:07:34.21	0.069
OLJHU2MVIB	15:07:34.21	0.271
RT1HSUMPGAGE	15:07:34.38	1.85
CHPHP1OBTMP	15:07:35.34	63
CHPHSUCTION	15:07:36.29	19
FERHNETFLOW	15:07:37.64	8657
BPTHLT2227	15:07:39.59	1.8

CHPHNETFLOW	15:07:40.54	8649
ST1HT115	15:07:41.51	23.67

The data shown in Table 3 was extracted from the Historic file as recorded on the OLY01 computer system for the day of June 10, 1999 at 15:07 local time. The 1ST column in the table lists the name of the individual site/sensor name. The basic naming convention that the Olympic system used in its development of the SCADA system was: the first 3 letters is the site where the sensor was located. The "H" as the 4th letter signifies that the value came from a historic file. The rest of the name is a shortened name that describes the individual sensors. The next column is the time that the value was taken and the last column is the value that was collected from the sensor. It should be noted that the sensor value that is stored in the historic database is not necessarily the value that was obtained from sensor. For example the sensor might report a flow rate as so many counts or some particular voltage reading. This reported value is converted to a pressure in PSI or a flow in barrels/hour by the input communications processor. For each field sensor that is installed, there is a entry in a conversion table. This table contains a conversion formula that is used to change the reported value into typical engineering units.

A examination of the historic log was conducted to see if there was any abnormal value or patterns detected, during the period prior to the accident. It was assumed that the periodic sensor scan rate of the computer would be relatively constant during the operation of the system. The Historic data was examined to see if the computer maintained the same polling pattern throughout this time period. A polling indicator was derived from the recorded log file parameter name and the time that the sample was taken. By calculating the difference in time between individual polling samples and plotting that difference vs. time, any changes in the polling period could be seen. Graph 3 depicts the difference in polled samples in seconds vs. time of day for the day of June 10, 1999.

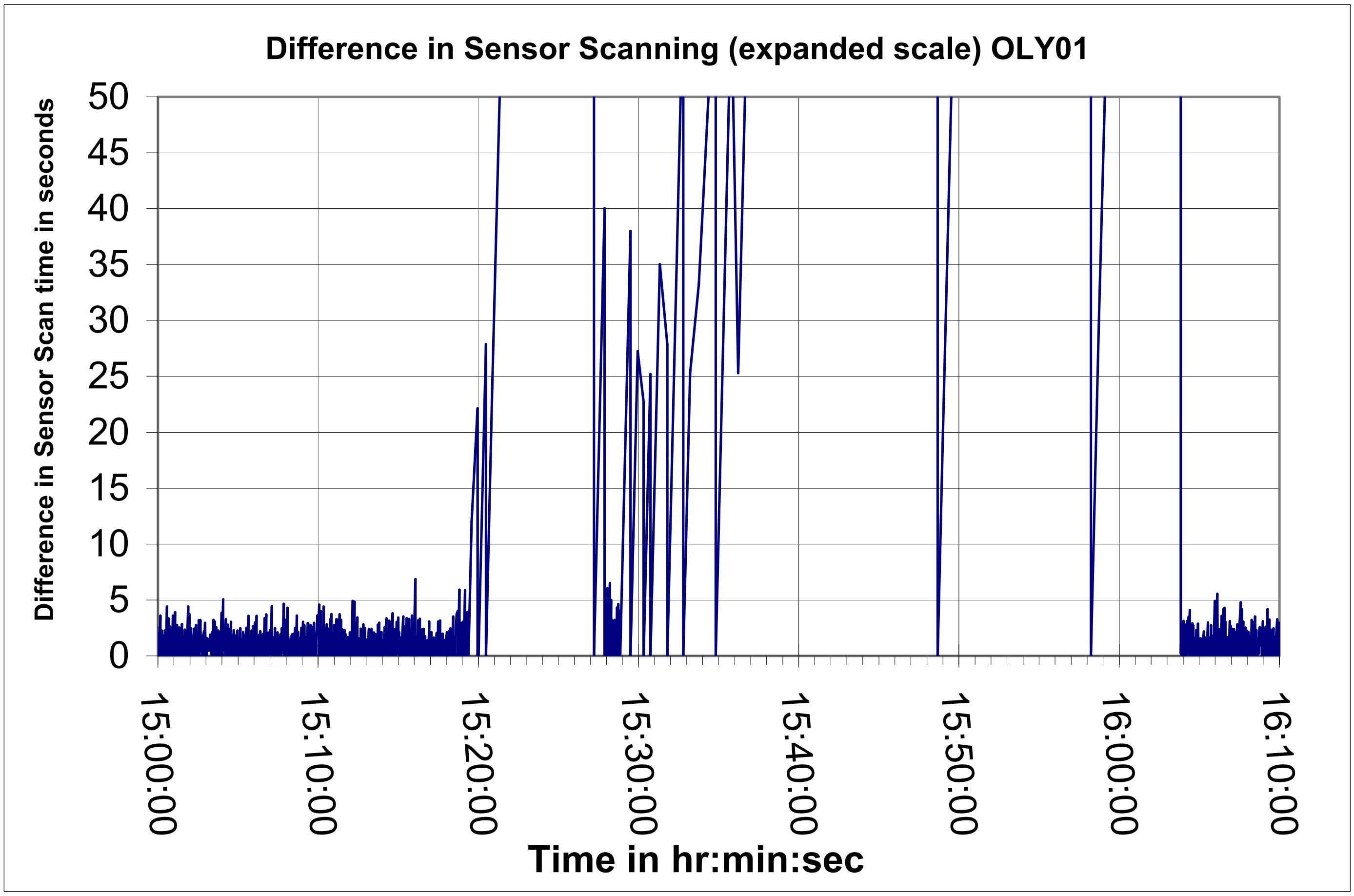
GRAPH 3



It can be seen from Graph 3 that the OLY01 system is running with a scan rate of about 2-4 seconds prior to the initial system problem which occurred about 15:20. To examine the time period just prior to the initial problem area, the time scale was expanded.

Graph 4 depicts the time period 30 minutes prior to the initial problem area of 15:20.

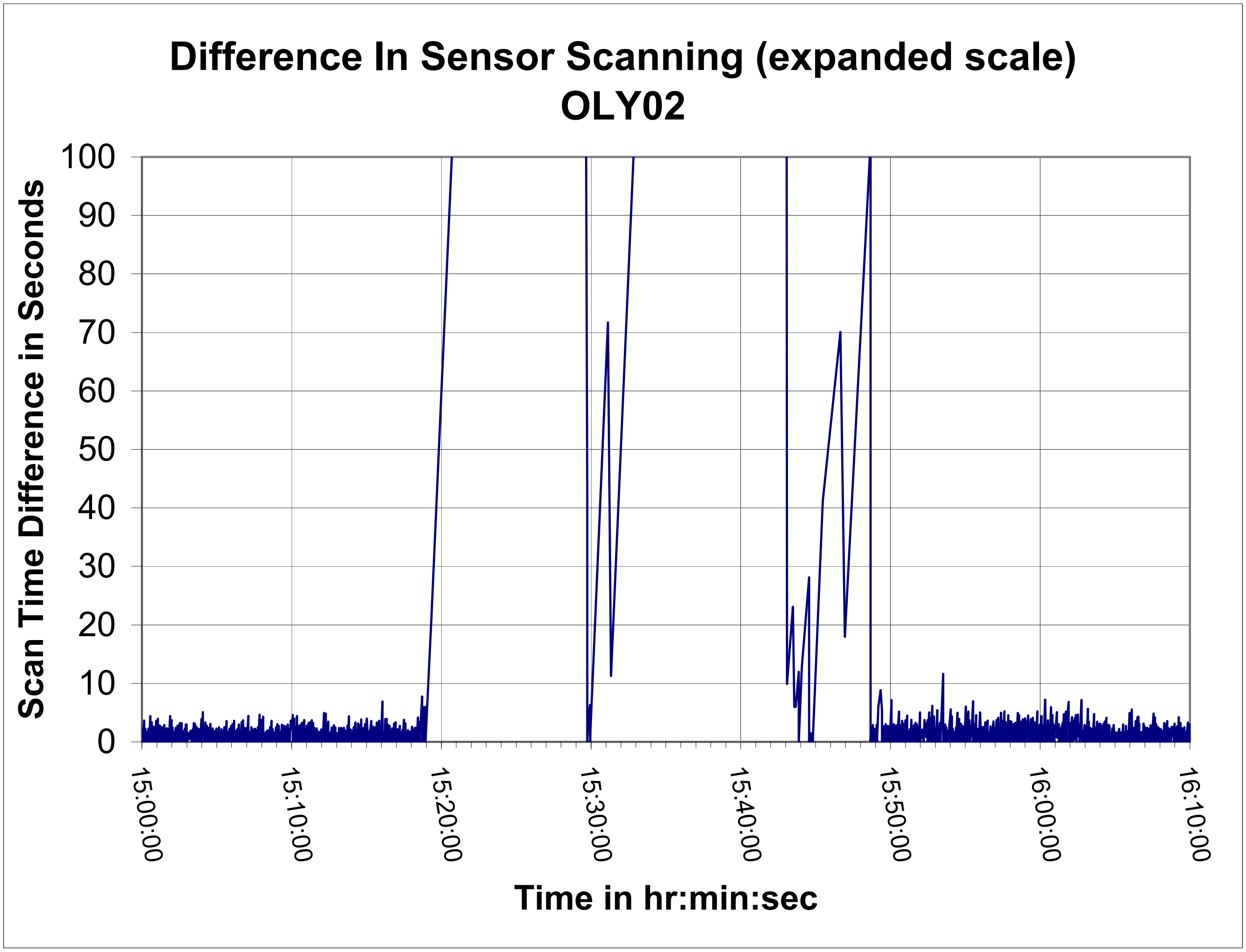
Graph 4



It can be see from this chart that the OLY01 system scan time is running at a consistent rate until 1519:20 when the delta time between samples goes from about 3 seconds to almost 30 seconds. This rate lasts for about 2 minutes after which time the rate suddenly jumps to over 400 seconds between samples. The system briefly recovers to a normal scan rate at 1527:00 for almost a minute after which time it virtually stops scanning any new data until 1604:00 when it returns to a normal 3-5 second scan rate. According to the VMS system log, the OLY01 computer was crashed at 1544:20. The first entry in the system log file when it was in the process of rebooting, was recorded at 1550:01. It is unknown exactly when the SCADA control portion of the system restarted. Those messages are normally written to the VMS accounting log on the host computer system. The accounting log for this time period on the OLY01 computer was missing from all of the backup tapes.

The same examination of the historical data that was recorded on the OLY02 computer was conducted. This data (shown in expanded time scale) for the time period just prior to the event is shown in Graph 5.

GRAPH 5



According to the VMS system log the OLY02 computer was halted at 1517:35. The first entry in the system log which indicated that the VMS system was restarting occurred at 1531:57. According to the Accounting logs, the SCADA system was restarted and running on OLY02 computer at 1545:31. There were no records found to document what was happening on the system from the time the VAX restarted until the time the SCADA system came back online.

This slowing/stopping of the sensor scan pattern was the only abnormality found that indicated that there was a problem with the SCADA computer control system. None of the host VAX computer logs or any of the error logs associated with the SCADA system captured any data that indicated that the system was having a problem completing any of the assigned tasks.

SCADA Historical Trend Displays

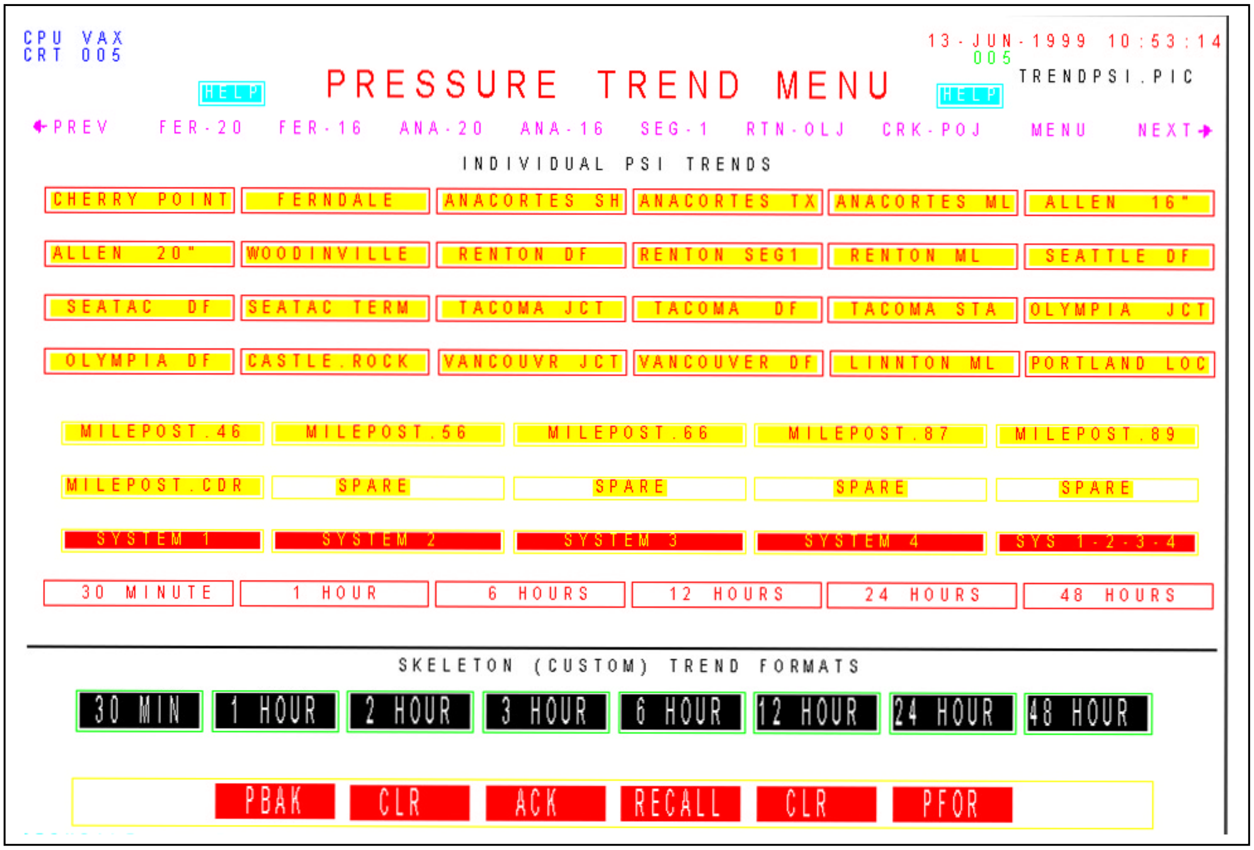
To document what historical information was available to the controllers during the accident sequence, several historical trends were run on the SCADA system. (See appendix of this report) These trends were manually generated by typing in the appropriate commands at the SCADA workstations. Up to 4 parameters can be trended on the same graph at a time. To generate the trend displays the computer will look up the specific parameter(s) from the historical record over the time period that the user has requested. In order to use the command line interface to the SCADA system, the user must know the specific name(s) of the parameter that he wants to trend. The computer then looks up the data for the specified time (30 minutes, 1 hour , 2 hour, 6 hour, etc.) period and draws a graph that depicts the data. The trend information uses the current time as time zero and goes back in time the specified amount.

It should be noted on some of the graphs that the current time is past the time associated with the pipeline rupture and the subsequent restart attempts. Care should be exercised when viewing these graphs. At the time of the accident, the controllers would not have the luxury of having trend information into the future. If they requested any trend data, their display would only contain data that represented the current time and data of time that had already passed. To correctly view the trend data graphs, the reader should enter the graph (horizontal axis) at the time of interest and any data to the left of that time represents what would have been available to the controller if they had the same display running. Additionally any time the computer system is not available to the controllers, they would have no way to view or access the historical trend data stored on the system.

Any time the computer is not executing the SCADA data acquisition program, it will not be acquiring new data and there will be gaps in the historical trend data corresponding to the time the computer was not running. If a controller displayed trend data from a time period that contained gaps, the trend graph would attempt to plot what data it did have. The graph would display the last value and would draw a horizontal straight line at that last value until a new value was available. Unchanging data (either because the acquired value is not changing or because the SCADA system did not acquire a new value) appears as a straight horizontal line on the trend graph. There was no special highlighting feature programmed to alert the controllers that they are looking at a graph that may contain gaps in the displayed data.

The SCADA system as configured by Olympic Pipeline does have several pre-built trend screens. These screens when selected present pre-configured trend data to the controllers. To select this data the controller only needs to click on the appropriate icon for the station and for the time period desired. No additional specific parameter names need to be entered to display a pre-configured trend. In addition to the pre-configured screens, there are several skeleton or custom trend displays that are available to the controllers. If the skeleton display is selected, the computer draws a blank trend display of the selected time period. The controller needs to then enter the alpha-numeric name of the desired historical parameter that he wants to trend (up to 4 historical parameters can be trended on one graph). Graph 6[6] depicts the Olympic pipeline “Historical Trend” controller page of their SCADA system as configured on the day of the accident.

Graph 6



[6] The colors in the graph do not represent the original colors that were shown on the controllers workstation display. The original color scheme used primary symbol colors on a black background. To include these pictures in this report, the original color scheme was inverted to display colored symbols on a white background

The top portion of the menu page are pre-configured pressure trends for some of the stations or pipeline sections. The bottom portion of the menu contains the skeleton or custom trend selections. (Actual trend displays around the time of the accident can be found in Appendix 1 of this report)

Appendix Attached

Appendix – 1 Selected Trend Graphs
Appendix – 2 VAX VMS System Logs
Appendix – 3 Olympic Backup Tapes listing
Appendix – 4 Olympic Computer System Upgrade Final Status Report dated Aug 6, 1999
Appendix – 5 Olympic Pipeline Control Methodology & SCADA Assessment dated July 2, 1999
Appendix – 6 Olympic Computer System Preliminary Status Report dated July 1, 1999
Appendix – 7 Oly1 Computer Console Printout
Appendix – 8 Oly2 Computer Console Printout

James R. Cash

Electronics Engineer