# Solving Problems Remotely: The Challenges and the Heartaches!

**John Brimmer**
**Remote Account Support Engineer**
**Hewlett Packard Company**
**2124 Barrett Park Drive**
**Kennesaw, Ga. 30144**
**Phone: 770-795-5568**
**Fax: 678-355-7006**
**john_brimmer@hp.com**

**Abstract:**

The Hewlett-Packard Response Center is a large, multi-talented group of people dedicated to remotely solving problems. Getting the right information to the right resource within the Response Center is the key to timely resolution of a problem. The proactive services organization has spent much time and energy developing ways to capture system information and use it to resolve problems. This paper is an attempt to communicate some of these techniques.

The value of one strategically gathered piece of information that is communicated to the right resource can be demonstrated with real world scenarios. By using the same principles that are used to support high availability applications across the whole IT environment, problem resolution can be facilitated. Understanding the groups that make up the Response Center and the way a problem can be solved using collaboration between engineers and information will aid the system administrator in the proactive collection of useful information.

Gathering information about systems with in today's complex IT environments can be a preventive measure. Support Levels also play an important role in the amount of information collected. Knowing what information has been already collected can facilitate problem resolution and save the system administrator time. Information is the key to remote problem resolution.

## Introduction

When anyone is trying to solve a problem remotely there is a whole new set of challenges. The fact that the person is not physically there means they need to take a somewhat different approach to gathering information to solve problems. By looking at some real world scenarios we can see these challenges. What follows is some real world scenarios that should shed some light on the processes behind remote problem solving that could be used by anyone or any organization.

## The Trojan Horse

Ann called me one Tuesday morning about the patch bundle I had sent her the week before. Ann was the administrator for among other things two identical K580's running the same application. Ann had installed the patch tape on one of her K580's with out any problems at all. Every patch installed properly and was configured correctly and the K580 booted up with out any problems. Both of these K580's were running 11.0 and were loaded for 64-bit operation. There were nine patches in the bundle. On the other K580 during software distributor's analysis phase only five patches were selected for installation. Of the four that were not selected two didn't even show up and two had messages as if they didn't even apply to this installation. We tried several things and finally decided to bring in more help. We opened a call within the Response Center and worked with a member of the SYSADMIN team. We created several depots and tried installing from them using different options in software distributor to no avail. Next we collaborated with a senior member of the SYSADMIN team. He suspected that the problem was within the IPD or Installed Product Database for software distributor. He checked for corruption no luck. He engaged a member of the expert center and we convened a conference call about 4pm in the afternoon. During the call we outlined all the trouble shooting that had been done so far. We went over the fact that we had two identically configured systems in everyway. We looked at the way we were installing the patches from tape or from depot. We discussed the Software Distributor options we had tried and the results of each. The discussion turned to how these systems had become K580's. They had both been upgraded to K580's and 11.0 64-bit were installed at that time. Then came the question. It seemed innocent.

The senior person from the SYSADMIN team asked, "What is the model string?"

Ann ran the command and said, "K580".

" Is that a upper case K or lower case k? "

"Lower Case k"

Lights went on, bells rang, with several people on the phone. Software Distributor didn't understand that this was a 64 bit OS! The model string needed to have a capital K within it to recognize the system as one capable of handling the 64 bit OS. A check of the other system and it had the proper model string. How did they get to be different. The OS couldn't have been installed with the lower case k. Ann was able to enlighten us there. Shortly after the upgrade there had been problems and a CPU needed to be changed and when the CE changed the CPU he had to run ssconfig to update the model string for the replacement processor and there was a typo lower case k instead of upper case K. Simple mistake! Easy to understand from the perspective of hindsight!

I had spent the greater part of a day dealing with this, and it took minutes to solve with the right piece of information in the right persons hands. Since that time I have checked the model string on every 64 bit OS. Odds are I will never see this problem again but…. you never know. I have also become quite aware of all our tools we use and whether they collect this information. I have noticed since that the model string is captured in most of the collection scripts I have seen.

- **Collect system configuration information regularly**
- **Collect system configuration information after repairs and changes**
- **Make comparisons between old information and newly collected information**

## To Recover or not to Recover that is the question!

One of my customers Paul had placed a call with the Response Center on Monday involving what seemed like some missing operating system files. Paul just had PSS level support so I didn't know he had placed the call till Tuesday morning when he called me along with his manager. Monday had been a busy day for Paul as could see as I reviewed the call he had opened. Paul and the engineers he worked with had approached a series of problems each one at a time and they had fixed each one. Paul had talked to the SYSADMIN team, the NETUX team, and the disk hardware team to verify the disk hardware. All with in the same call! Each team had fixed a problem and afterwards it seamed Paul was on the road to a solution.

So here we were on Tuesday morning with some more problems. I was on the line now with Paul, his manager, and several other members of the HP support team from the local area. It was decided to involve the Systems Interrupt team or SIT for short. After a brief look at the situation Tom from the sit team came to me and said John my recommendation here would be to restore the operating system from scratch. Tom had seen many situations like this and knew that we could fix every file and problem with an expenditure of time and maybe even determine the root cause or we could reload the OS and start over in a much shorter period of time. So we convened a conference call to discuss our options with Paul and his manager. We were soon to find that the painful part was yet to come. This system had been in production at least 2 years. It had a sister system 100 miles away. The painful part was gathering the CDs, patch bundles, and system information to restore the system. Several hours later we had found all the resources we needed and by the next day we had recovered the system and we were back in production. This drove home to me several things in hindsight one if we had had system configuration information already collected we could have made the recovery determination quicker maybe even Monday before I was involved and we may have prevented fixing so many individual problems and gone for the main problem. We had examined logs and never could determine what caused the problem. The system information combined with a method of recovery such as an "Ignite" tape or an "Ignite Golden Image" could have formed the basis of what I like to call a "Recovery Roadmap" that can recover a system and also serve as a source of information during troubleshooting.

- **Always have a recovery road map**
- **Leverage technologies such as Ignite**
- **Be aware of time needed for recovery versus repair of a problem is**
- **Test recovery before you need it**

**Lets Fly South**

Y2K was an interesting event for the Response Center. There were many questions and concerns along with many upgrades and migrations taking place with in the IT world. One sticks out in my mind among others. This particular group of data centers was moving from 9.04 to 10.20 during the last half of 1998. What stands out in my mind was that the application vendor had worked out the entire upgrade process for OS and there application. This was unusual but not unheard of. I soon found that the challenging part was these were all to occur in successive weekends over a 4-month period. The reason for the special process was most of these centers only had a 3 hour maintenance window on selected Sunday mornings from about midnight to three am and they absolutely had to be up and running by 4 am. Another complicating factor was these were all on the 9.04 OS using switchover and when they migrated to 10.20 they were staying on switchover because the application vendor had yet to certify the Service Guard product. Another driving factor was that in January 1999 HP was discontinuing support for 9.04. The data centers were spread over four states also with different system administrators and different HP account teams involved. I soon found that I was the link between each of these different upgrades.

The data center that was the southern most was the first to attempt this process. It was very enlightening to say the least. The systems ranged from K460's to K580's all with mirrored root disks and running switchover. The plan was to use a dual boot scenario and do one upgrade with a process that was developed in the field and wasn't really supported but it worked. This way they could always go back to 9.04 if the process failed and this happened several times. There were problems with every upgrade and early in the process I started collecting every shred of configuration information and data on our problems. Every problem centered on the disk configurations. As we kept moving north with the upgrades we kept looking southward.

### By The Way

I have had many times when that crucial piece of information came as part of the information we gather routinely as part of Mission Critical Support. Since I am notified every time one of my customers is paged I am in a unique position to see the problems across a data center. I remember one situation where I was paged and it had been determined that a fiber channel card connected to the SAN had failed and was to be changed. When I am paged I also receive an email with the same information. This all seemed fairly routine. Since it was 3 am I didn't worry much about it until several days later. A database crashed on another system using the same SAN. During the trouble shooting process it soon became apparent that the failure several days ago and this crash were related but no one but myself new about the fiber channel card failure. By the way the way the other day we changed…this piece of information sense of all the error messages and problems we were seeing with the database.

- **Keep records of everything**
- **Have people who are focal points and who are notified when things happen**

### Times have Changed

Over time the way information for troubleshooting things remotely is gathered has changed considerably. When I started in my present position several years ago it seemed like each team in the Response Center had a custom script to collect information that was needed to troubleshoot problems in their area of expertise. The SYSADMIN team might care about LVM, syslogs, and other such information while the NETUX team may be more concerned with networking configuration items. With the advent of higher support levels there arose the need to gather information proactively. When you looked at the troubleshooting process you found a lot of time was wasted moving information back and forth. It made more sense to collect this information in advance and then verify if it changed when you have a problem. As I have worked in these environments I have come to the realization that anyone can collect this information proactively and have a plan to use it when the need arises.

There are many scripts and tools currently available.  Some come as part of higher level support but several are available to anyone.

| **Mission Critical Tools**<br>**(used with CSS and BCS support levels)** | **Freely available Tools** |
|---|---|
| **Customer Operational Profile**<br>(Web based Reporting tool)<br>-getconfig script(system info)<br>-collect.sh(collects patch info) | Cfg2html –converts configuration info to html<br>http://members.tripod.de/rose_swe/cfg/cfg.html |
| **Sysinfo scripts**<br>Created by HP field resources to collect system information | **HP-UX System Administration Handbook and Tool Kit,** by Marty Poniatowski<br>Chapter 8 HPUX system auditing<br>Prentice Hall PTR, isbn  0-13-905571-1 |
| **High Availability Observatory**<br>HP workstation supplied by HP and configured to collect system information onsite and transmits to HP | **Individual HPUX commands**<br>swlist, bdf, iosan, etc |

No matter which tools you use you need to be able to collect information on a regular basis and compare the information for changes. This is done automatically within the High Availability Observatory.  The output from other tools can be processed in different ways to see if things have changed. Things as simple as the model string changing could be caught in this way.  Keeping this information over time can be valuable in determining where problem started or catching one before it develops.

## Recovery Road Maps

When the decision is made to recover a system for whatever reason the challenge is gathering everything you need.  I have worked with customers where we have put together a whole strategy for recovering their systems. Ignite is a powerful two edge sword that can be used to help recover a system but it can also provide information about your configuration. You also have to be aware of everything that has been changed on a system since that Ignite image has been created and be able to incorporate that also into your strategy for recovery.  Creating software depots for applications and patches is a useful way to maintain a recovery roadmap or strategy. In the example case about recovery an extra day was added by having to find everything needed to restore the system. One thing I failed to mention in that example was that tapes and CD's had to be sent overnight to do the recovery of that system.

## Conclusion

Solving problems over a telephone through someone else or using a dialup connection can be challenging.  If you find yourself in that position nothing can replace proactive planning and collecting information. When a crisis is emerging is the wrong time to collect information. It is much easier to have configuration information, system history, information how to contact people, software, backups, system images using Ignite, and other resources at your finger tips ready to use.