

Automated Customer Inquiry?



Delight Your Customers and Close Cases Faster.

Software Companion Toolsmith



Amy Guerra
ByStorm Software

More Information

For a personalized ROI evaluation or a **free, 45-day trial** of Software Companion Toolsmith, visit www.bystormsoftware.com, email sales@bystormsoftware.com, or call 877-BYSTORM

Section One: The Opportunity for Automation

The Forlorn Forty-Six Percent

Forty-six percent of the time, you could have a serious problem. That's the average percent of technical support cases not being closed on the first call. These are the issues that cost you, both in company resources and your customers' satisfaction. The more complicated the supported products are, industry wide that forty-six can go up into the forlorn fifties and sixties.

Partially, we're seeing this problem because of new technologies that are being used well. User access to robust self-support options takes the simpler, documented-solution cases out of the open case mix, which is a great benefit. The calls that you actually field ARE going to take longer. Ideally, the number of calls you're fielding should be lower to compensate.

Unfortunately, the number of incidents overall is also up—some studies say as much as 27%. The simpler issues are being handled more efficiently and in greater volume, but the complicated issues aren't going away, and are stacking up as costly backlog. The future success of your support department lies in streamlining your total service delivery process—with a special focus on that open forty-six percent.

The Bottleneck

Automation of support processes has spread from hardware advances, to administrative efficiencies, to the relatively new resolution process solutions. Table One shows some common aspects of support case processing and tools being used to assist in each area.

For most support organizations now, 75-80% of costs lie in the resolution process. Knowledge Management systems and Resolution Delivery tools are examples of solutions helping address this area. But within the resolution process, 40% of the cost is in troubleshooting and issue determination. Diagnosis of the problem is the single largest time element—because it involves customer inquiry.

Table Two shows support case processing again, with the customer input areas included. Customer information-gathering processes can be the most time-consuming element of troublesome support cases, a weak spot for introducing bad information into the mix, and a common source of customer dissatisfaction.

Support issue information-gathering processes can be the most time-consuming element of troublesome support cases, a weak spot for introducing bad information into the mix, and a common source of customer dissatisfaction.

Automation and Assistance Tools for Support Case Processing

Table One

Customer Education and Case Logging	Diagnostic Wizards and Web Issue Submission
Incoming Cases & Routing	Web and Speech-Enabled Processing, etc.
Customer Administrative and Management	CRM / Help Desk Systems
Case Management	CRM / Help Desk Systems
Problem Search and Diagnosis	Internal KM System / CRM / Collaboration Tools
Internal Escalations / Outside Analysis	CRM / Collaboration Tools / External KM System
Solution Delivery and Solution Recording	KM & CRM Systems



What is NOT Automated

Table Two

Customer Education and Case Logging	Diagnostic Wizards and Web Issue Submission
Incoming Cases & Routing	Web and Speech-Enabled Processing, etc.
Customer Administrative and Management	CRM / Help Desk Systems
Case Management	CRM / Help Desk Systems
CUSTOMER INQUIRY	NOT AUTOMATED, AGENT AND CUSTOMER
Problem Search and Diagnosis	Internal KM System / CRM / Collaboration Tools
CUSTOMER INQUIRY	NOT AUTOMATED, AGENT AND CUSTOMER
Internal Escalations / Outside Analysis	CRM / Collaboration Tools / External KM System
REPEATED CUSTOMER DISCOVERY	NOT AUTOMATED, AGENT AND CUSTOMER
Solution Delivery and Solution Recording	KM & CRM Systems

Diagnostic Efficiency Makes a Huge Impact

ServiceXRG found in a 2004 study that reducing diagnosis time alone can reduce the **total** service delivery cost by 10-15%--showing the largest impact of any recommendation offered. To reduce diagnosis time, you have to know the right questions to ask, know which resources are relevant and necessary, and be able to collect the corresponding accurate environment and configuration information from the customer. In short, you need to streamline customer inquiry.

Asynchronous Communication

Looking back at the forlorn forty-six, the customer inquiry bottleneck really comes into focus when you consider that over ½ of cases not closed on the first call take 2-5 days to close, and over 1/3 of them are taking 5 days or more. A mere 10% of cases open beyond first contact are getting closed within 24 hours. Is it really taking someone 3 days worth of research to pinpoint the issue? No.

What is different in problem cases from the ones being closed on the first call is

that you are now dealing with an asynchronous process. There is a “stop and start” workflow introduced, where investigation gets only so far before a determination is made about new information needed, which means a new request for information from the customer, and more waiting. Everyone suffers from inefficient use of time and unnecessary frustration.

In an interview with a support director about IT administrative customers, he cited on average 16 phone minutes for an agent to explain how to get the log files needed for a case, three return calls to the customer to get the information, and a three-day turnaround to receive the files.

Developers working on escalated issues complain of time they’ve wasted because of inaccurate or incomplete customer information. Customers, who prize problem resolution above all else in rating support services, are being repeatedly bothered for more information while their issue remains unresolved.

Streamlining Customer Inquiry

Taking out the waiting and reliance on customer reporting makes a significant leap toward having one succinct, accurate diagnosis process without the stops and starts. At heart, this relies on getting everything you would ever need from the customer up front. The

challenge is that you have to know everything you need to solve the problem—before you know what the problem is.

This “Crystal Ball” magic solution has simply not been realistic in the past. Since you don’t know the problem, the only way to insure you would have what you need for diagnosis would be to have access to literally everything. From a capture standpoint, even if attempted this was impractical for a host of reasons, including unwieldy file sizes, non-standard data locations, transmission difficulties, and issues with viewing the information on other systems. There was simply no way for a user to transmit that amount of information--short of shipping in their computer.

Of course, there are technical ways for the agent to go to the user’s computer. But even with screen sharing or remote access capability, finding and evaluating the information is a manual or multi-step process for the support agent. Also, the data, if “collected” during a session, is generally not saved in a form that is interactive for upstream investigation in escalations.

This is not to discount the idea of having access to the user’s machine, which is really everyone’s idea of a best case scenario. But aside from the fact that

In a normal support case discovery phase, there is a “stop and start” workflow introduced. Investigation gets only so far before a determination is made about new information needed, which means a new request for information from the customer, and more waiting for both parties.

remote access isn't available in most support scenarios, you don't really need the WHOLE machine. What you do need are all the pieces of data relevant to solving the case, without having to search all over to find them wherever they may be on each different user's system. This is where the opportunity for effective automation is found.

Selective Information Capture

A real, working solution lies in highly customized, product-specific diagnostic tools. These tools would automate customer inquiry by gathering—not everything—but everything needed to diagnose problems with the product being supported, and arrange it all in a standard, user-friendly format.

Many software companies have tried for this by developing some sort of diagnostic gathering tools to accompany their products in the field. Focused on very specific information that is often needed in resolution, these tools are helpful because they are designed by the people who know better than anyone else what is needed. There is a catch, though. For a company to develop, distribute, and then have to support its own diagnostic discovery tools for each product can add as much, if not more, cost than it saves.

Engineering must develop the tools, with input from Support and QA, hopefully with testing cycles. This not only takes time, it takes time away from revenue-

bearing products. Since these tools are usually outside of the company's core competency, they are never going to be as well or as efficiently built as other products. Can they be counted on to be reliable in a support scenario?

Also, each product team is likely to build a very different solution, making maintenance and support of these tools across a large multi-product organization a thorn in the side of the very department the tools are seeking to help. Lastly, when the output from these tools isn't complete, isn't in a standard format, or is unwieldy because of size or viewing requirements, it doesn't completely solve the problem. Because they don't capture EVERYTHING needed, there is still asynchronous communication, so the impact of their efficacy is not anywhere near as high as it could be.

Automating Customer Inquiry

For any initiative to automate customer inquiry to succeed, it would have to achieve all of the following goals:

Success Goal #1:

Gather and provide the complete information necessary to solve the problem—before you know the problem.

Success Goal #2

Save and present the data in a way that is accurate, compressed, and in a standard format that can be read on any machine.

A real, working solution lies in highly-specified, user-friendly capture tools that you design—but that you don't have to build and support.

Success Goal #3

Be “immediate” for both your customers and your support agents.

Success Goal #4

Capitalize on your expertise about the information you need, but not use your engineering or support resources to build and maintain tools.

The answer to achieving all of these goals is highly-specified, user-friendly capture tools that you design—but that you don’t have to build.

ByStorm Software saw this need and is supplying the solution: an inexpensive, simple, and feature-rich “diagnostic tool-maker,” Software Companion Toolsmith.

Software Companion Toolsmith

Software Companion Toolsmith is a “virtual programmer,” allowing you to build product- or project-specific companion tools for the field without writing a single line of code. ***The resulting tools are wizard-based, remote diagnostic capture solutions***

specifically customized to your needs.

Each field tool you create runs with no installation and no footprint on your user’s machine. Gathering routines are run with just a click of a user’s mouse. The gathered data is saved in an encrypted and highly compressed file. Built-in package delivery options make the whole process fast, simple, and reliable for both you and your end-users. Once received, the contents of this Companion File are investigated in a specially-designed diagnostic discovery tool. You can even load two Companion Files at once and get a color-coded comparison of the differences.

With no engineering effort, you have the ability to create and use custom diagnostic companion tools for the products you support. Meeting the five established criteria for success, our solution will show measurable ROI in both cost savings and customer satisfaction.

The Four Criteria for Successful Automation of Customer Inquiry:

1. Gather the complete information to solve the case.
2. Save the information in a compressed, standard format.
3. Be an “immediate” process for your customers and agents.
4. Capitalize on your design expertise, but not take your engineering resources.

Section Two: Building Success with Software Companion Toolsmith

Success Goal #1: Powerful Information Gathering

Gather and provide the complete information necessary to solve the problem—before you know the problem.

Creating Companion Tools with Software Companion Toolsmith is easy—and can take as few as 10 seconds. Based on the most commonly-needed information for defect resolution, quite a bit of the gathering power of your new field diagnostic tool is standard in every tool you build. This information is quickly grabbed whenever your new field application is run and compresses to a file that can be as small as 20K.

An Overview of the Standard Information Gathered by SCT Remote Companion Tools:

Type of Information Gathered	Details
Running Tasks	Sorted by module. Author, version, copyright information, etc. available.
Loaded Modules	Sorted by process. Commonly-used details available, as well as file path.
Screen Resolution and Depth	In addition to settings view detailed information on display driver.
System Up-Time	Displays uptime to the second.
Virtual Memory Settings	Lists “total” and “free” space.
Load Address	Available for modules, relocation addresses also shown.
OS Version	Version details available.
Processor Information	Speed, identifier, and name.
Physical Memory Information	Lists “total” and “free” space. Also available for paging file memory.
Active Window Screen Capture	Takes snapshot of active window at the time of gather (can be turned on or off).
TCP/IP Information	System-wide DNS information.
Network Adaptor Information	IP address, NIC description, physical address, Subnet mask, WINS server information, and more.
Installed Software	Installed software list based on uninstall registry key.
Installed Microsoft Updates	Lists components for which updates are possible and all their installed updates. Whatever details provided for each update are shown.
Drive Information	Drive letter, type, free space, free to user, total size, total used, UNC.
Time Zone Information	International time zone setting of machine.
Information on Services	Display name, dependencies, logon account, start up type and path, status.
IE Settings	Domain information, specific protocol defaults, IP ranges.
Major System Events	If available from the OS, lists all recent system events

Of course, what separates Software Companion Toolsmith’s diagnostic tools from other capture tools is custom, configurable content. Automating customer inquiry rests on gathering everything relevant surrounding an issue into one file. For this reason, the field diagnostic tools compress the data by around 90%, so that anything that MIGHT be needed can be captured—the first time. Any additional files, event logs, registry data, or anything else you desire are captured in full—and saved into the same Companion File for delivery.

Detailed choices governing how the data is located and what you choose to collect make this possible. For example, if you needed to capture your product’s log files, these files might be:

- Stored in a different place on every user’s machine
- Really big
- Available for hundreds or thousands of days when you only need yesterday

You can overcome all of these potential issues when designing your new Companion Tools. Using Software Companion Toolsmith, you’ll add a “File Capture” task to your diagnostic tool design to capture the desired logs and save them with the rest of the output. You’ll then tell the diagnostic tool to locate the file you want via information from the registry telling it where the directory is when run on any system. Limiting the capture by size “x”, you’ll then designate whether to capture the first “x” K or the last “x” K of the file. Age-checking allows the gather process to just get yesterday’s or last week’s files and not the whole list.

This file capture is just one gathering task—and you can design your tools to run as many tasks as you can dream up. When you add in the ability to embed scripts, the information gathering power of your new tools becomes very flexible and robust. Here is a list of the configurable gathering options in SCT:

An Overview of the Custom Gathering Options in SCT Remote Companion Tools:

Type of Information Gathered	Details
Specific Files	Capture an actual file, its location information, and all of its details. You can also capture partial files, or select files based on criteria such as size and age of file. Locate the files for capture via a set path or via the registry. Capture as many as you need.
File Lists	Capture a directory without the actual files but with all of their details.
Registry Values	Value with name, type and data.
Registry Hives	Complete hive with all associated details.
Custom Output from Running Other Applications or Embedded Scripts	Launch, run, and gather information from OTHER programs. For example, PING.exe will run and the results will be captured. Or, your script you have already written to do very specific diagnostic tasks can be embedded and run right from our product.
Event Logs	Capture specific event logs. Filter by criteria you choose.
Global Groups and User Information	Each member of every group and their user details. Sorted by group or user. This can be turned on or off depending on whether you think the information is needed.
Local Groups and User Information	Each member of every group and their user details. Is turned on and off with the Global Group Command.

Software Companion Toolsmith Customer Success Story:

An Example of How Time and Resources are Saved: Capturing Log Files for the Support Process
ByStorm Software Customer's Sample Data: 1785 cases where log files were requested from end-users over a three month period:

	End-User Gathers by Hand		End-User Employs our Tool	
	Total Hours	Min./Case	Total Hours	Min./Case
End-User Time	595	20	29.75	1
Support Agent Time	386.75	13	22.31	.75

- Our Customer saves 565 hours of their End-Users' Time and Frustration
- Our Customer saves 364 hours of Support Agents' Time
- Cost Savings for Agent Hours:
 - \$17,472 for the Sample Period or \$69,888 Annually

A Log File is just one item of many often needed in issue resolution. Using Software Companion Toolsmith allows you to capture ALL of the necessary data without having to educate your end-users on where to find it, wait for them to send it, and then determine whether or not it is accurate. Your end-users will use the time you've saved them to tell others how great you are!

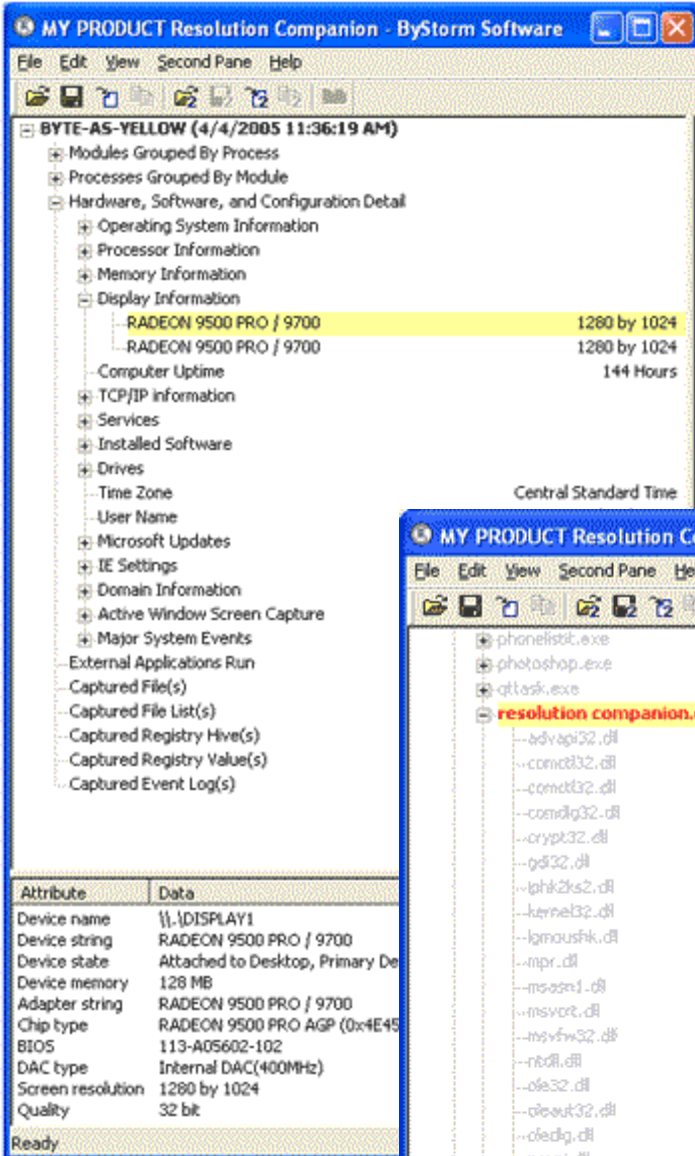
Success Goal #2: Standardized Presentation

Save and present the data in a way that is accurate, compressed, and in a standard format that can be read on any machine.

No matter how much information is collected by the field diagnostic tool, such as multiple files, registry data, or event logs, it is ALL saved into one, highly-compressed Companion File. Companion Files can be easily transmitted or archived, and are in essence a robust snapshot of the user's computer at the time of capture, but in a form that can be investigated. Because the diagnostic field tools gather the information directly from your users' systems to your specifications, you know the data is correct.

Companion Files (file type .sqa) are viewed in another no-installation, no-footprint diagnostic discovery tool that accompanies the solution: Resolution Companion. All of the collected information is presented in tree format for easy drill-down, with detailed information presented in the bottom properties screen (examples on next two pages).

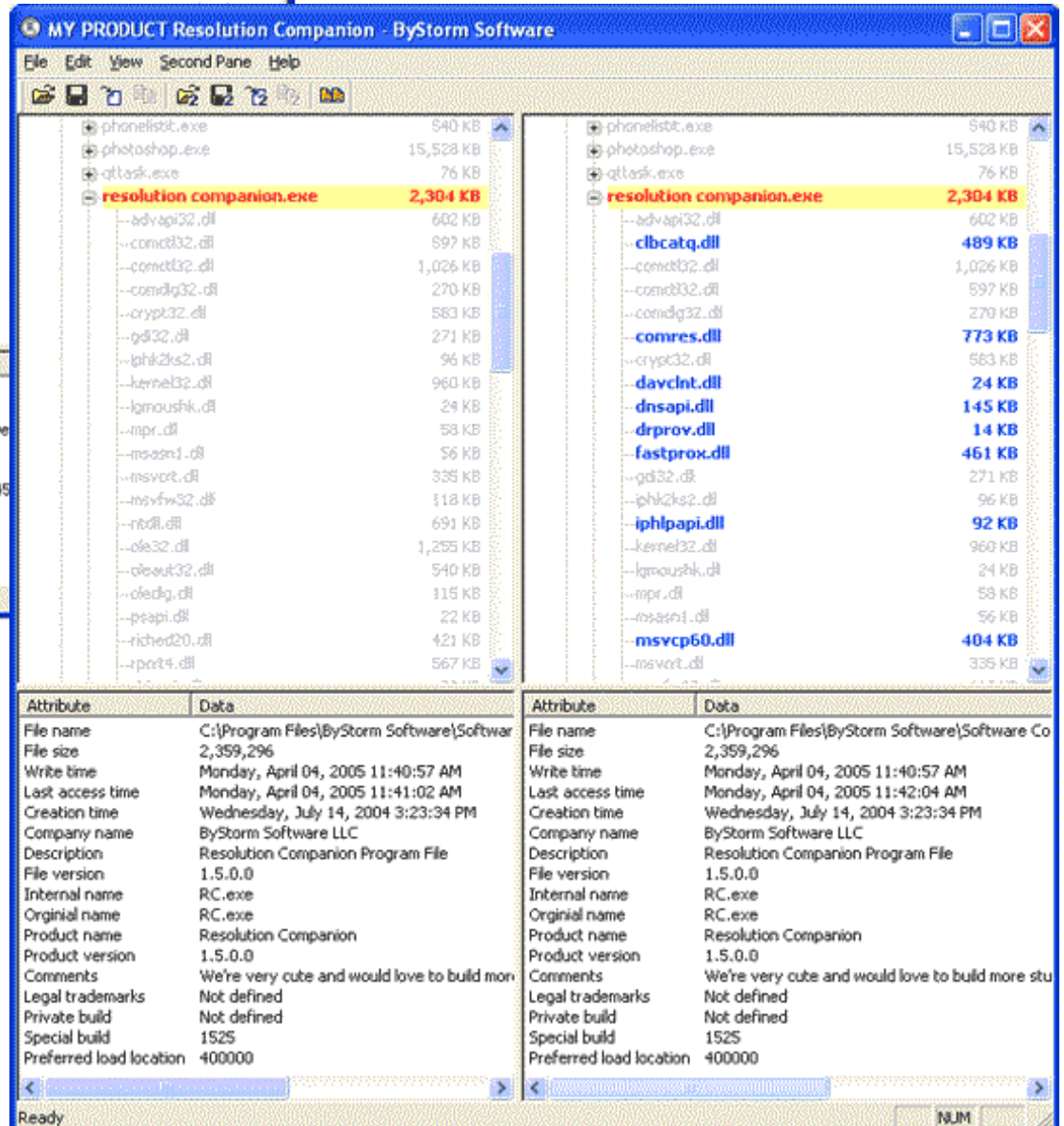
Every Companion File, no matter what the contents, is rendered in exactly the same format in Resolution Companion, so agents supporting multiple products always know where to look for what they need. Also, when SMTP delivery is used, much of the information from the Companion File can be inserted into the text of the delivery email via merge fields and even pulled to auto-populated the fields of a CRM system.



Here are examples of Companion Files being viewed in Resolution Companion, both in a single and a double pane view. All the panes of Resolution Companion are resizable.

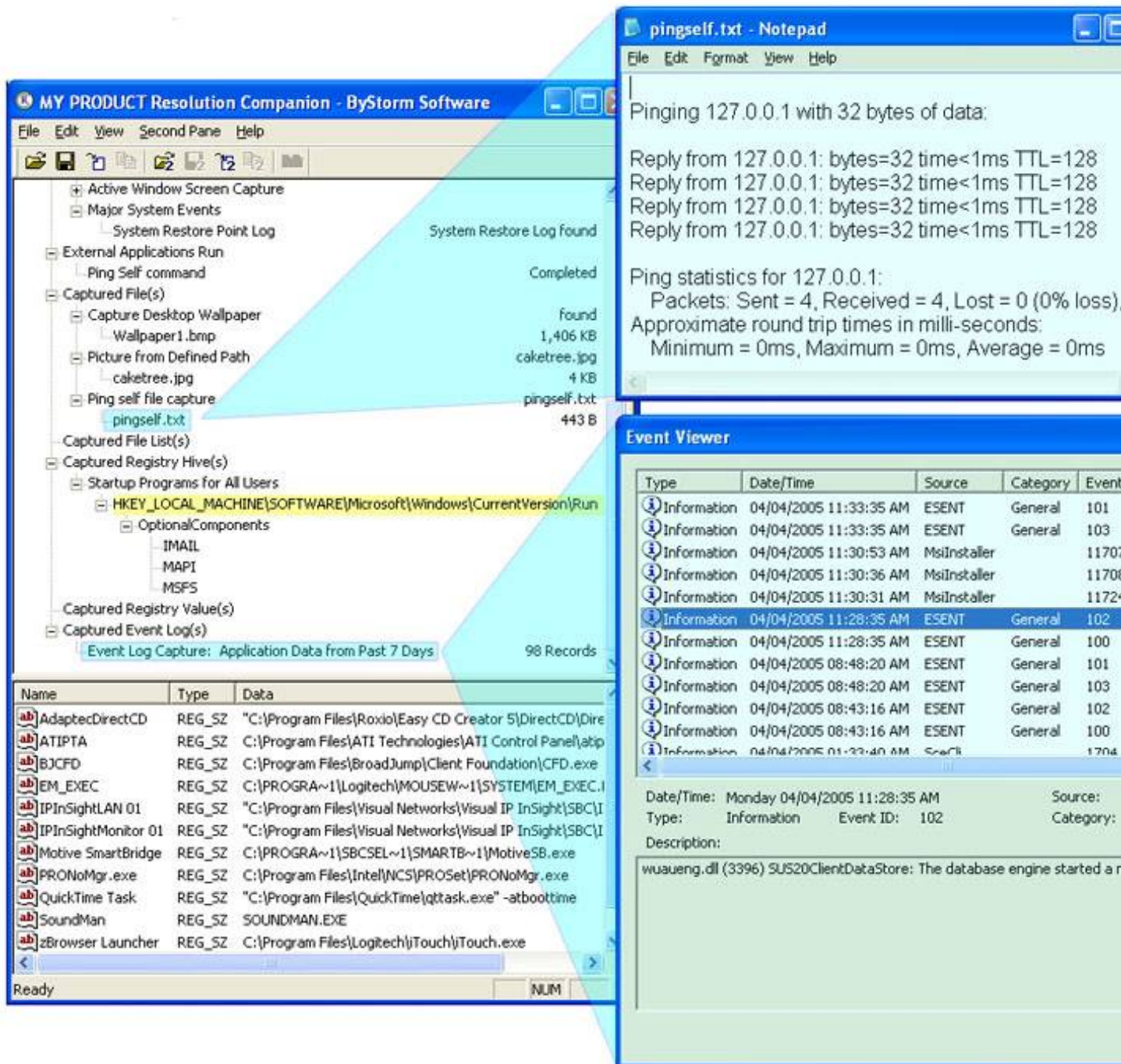
To the left, the details of the video card are being displayed in the bottom properties window.

Below, two Companion Files have been loaded side by side for a comparison. For the running process selected (in this case, Resolution Companion), many DLLs are being used on the capture on the right that are NOT present in the capture on the left, hence the bolded blue entries.



Viewing custom-collected information is just as simple. While registry, file list, or major system event information can be viewed in the bottom properties screen, actual files captured launch on double-click in whatever program is needed to read them (provided it is resident on the viewing machine). Screen captures and event log information launch on double-click in a built-in viewer.

Below, the results of running Ping.exe are being viewed in notepad, the contents of a registry key are being viewed in the bottom properties window, and seven days worth of application data from the event log is being shown in the built in event log viewer.

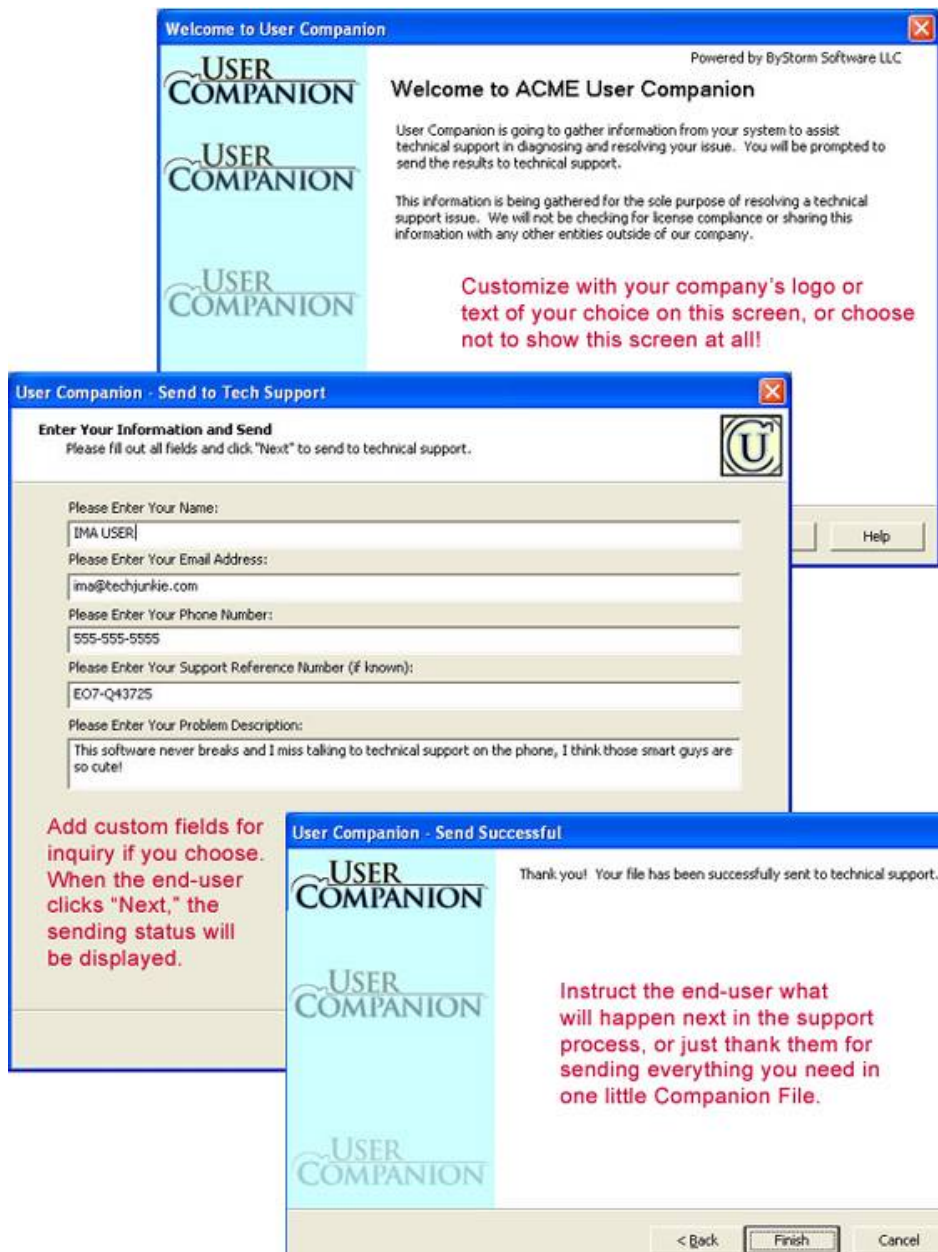


Success Goal #3: Immediacy

Be “immediate” for both your customers and your support agents.

Complex data retrieval used to rely on customer education and then customer action. By removing the need to teach customers where to find data, and making the process so easy that it can happen with just a couple of clicks, you make your job easier and your customers happier at the same time. Information that used to be requested for escalations and received at a later time can be received during the first call.

Software Companion Toolsmith’s field tool that you create, called User Companion, is designed in an easy-to-use wizard format. While your users just enter simple information such as a case number or email address, the complex gathering happens automatically and the whole package is sent to the address you have set.



In the prior example of the support director with IT Administrative customers, he was stating sixteen minutes of customer education and a three-day wait for log files was the normal timeframe for information gathering. Using SCT, this went down to about 45 seconds for the support agent to explain what to do and one minute for the customer to run the field tool, User Companion.

Software Companion Toolsmith Customer Success Story:

An Example of Increased Customer Satisfaction: Multiple Requests for Information

ByStorm Software Customer's Data: Complaints from End-Users and Sales Representatives.

Prior to the use of Software Companion Toolsmith, customer satisfaction was poor by in-house standards. One of the most often-heard complaints was about repeated requests for information in case resolution--whether it was the same information being asked for more than once, or repeated requests for new information over the course of the diagnostic process.

Previous end-user "complaints" are now accolades about easy data collection.

Cases are being completely resolved in the same time it used to take just to get the first round of requested information--and customers only have to send their data once.

Maintenance contracts make up more than a third of most software companies' revenue today. Keeping customers happy is crucial to ensure high-margin, highly-profitable maintenance contracts are renewed.

Success Goal #4: You Design It, We Support YOU

Capitalize on your expertise about the information you need, but not use your engineering or support resources to build and maintain tools.

The true strength of SCT is that it is a simple tool that may be easily introduced into any situation. No large investment in training, money, hardware, or change management is needed.

Creation of these tools actually requires no time from development engineers, aside from putting their heads together with support to determine everything that should be gathered. Custom, branded tools can be made by anyone. When there is a question with how a diagnostic tool is performing, you can use our resources to help you achieve the required result, not burden your own engineers to fix the tools that are supposed to be helping fix the problem.

This simple design interface doesn't mean that you are short-changed on control over how the tools work. Aside from the information gathered, you have a fairly wide amount of power over the design of your remote tools.

- *Presentation Options* dictate the appearance, for example, whether you include your company logo and license agreement.
- *Communication Settings* allow you to control delivery-related details. How many and what user-entry fields do you want to appear (i.e. name, email address, problem description)? Will the file be sent via SMTP, or saved to desktop for some other delivery method? Will you allow the end user to view the data before

it is sent or even choose to remove items that could be security risks? You can even add or remove screens from the wizard and add custom text or links to different screens to assist the user in the support process.

- *Performance Options* define whether or not the remote tool is deleted from the end user's system upon next reboot, whether screen shots or group membership information are captured, and more.

All of your choices are saved as a specification file, which you can open and edit at a later time to easily create new remote tools.

Software Companion Toolsmith Customer Success Story:

An Example of Overall Success: Case Closure Time

ByStorm Software Customer's Sample Data: Six Weeks of Use on One Product Line

Our customer shaved 1.6 days off of their case closure time--a 23% decrease across the board.

With staff salary and overhead making up 70% of support delivery cost, time is money. While everyone's costs are different, imagine your savings in resources, case backlog, and customer satisfaction if you cut your open case time by 23%.

Bottom Line ROI

Whether you're measuring success by decreased costs or increased customer satisfaction, this solution will show a great return on your investment.

Although every organization's service delivery challenges are different, the sample ROI information on the next page shows how the cost-saving value of Software Companion Toolsmith is realized in two areas: diagnostic efficiency and (often) a reduced burden on engineering.

For a sample company with 20 support agents and 10 products, over \$500,000 in annual savings can be realized. More importantly, these very same cost-saving activities actually INCREASE customer satisfaction.

Achieving diagnostic efficiency and shortening the total service resolution timeline rest on addressing the bottleneck: customer inquiry. Let ByStorm Software partner with you to realize the impact of automated customer inquiry in your support efforts.

Powerful, Standardized, Immediate, Personal . . . Software Companion Toolsmith meets all of these goals while saving you money and delighting your customers.

More Information

For a personalized ROI evaluation or a free 45-day trial of Software Companion Toolsmith, visit www.bystormsoftware.com or email sales@bystormsoftware.com.
877-BYSTORM

Software Companion Toolsmith Sample ROI

Section A: Projected Cost Savings from Using Custom Diagnostic Tools for Support

Example: Company X with 20 Support Agents, 7200 Cases per Month and 10 Products Supported.

	Cases	Close Rate	Cost per Case	Total Monthly
New Monthly	7200			
First-Call	3859.2	53.6%	\$ 55.00	\$212,256.00
Within 24 Hours	338.4	4.7%	\$ 65.00	\$ 21,996.00
Longer than 24 Hours	3002.4	41.7%	\$ 95.00	\$285,228.00
Total Monthly				\$519,480.00

Targets for use of new diagnostic tools:

- Move 30% of cases open longer than 24 hours to "solved within a day"
- Move 15% of cases being resolved within 24 hours to first-call closure

Results:

	Cases	Close Rate	Cost per Case	Total Monthly	
New Monthly	7200				
First-Call	3909.6	54.30%	\$ 55.00	\$215,028.00	Monthly Savings: \$27,504.00
Within 24 Hours	1188	16.5%	\$ 65.00	\$ 77,220.00	
Longer than 24 Hours	2102.4	29.2%	\$ 95.00	\$199,728.00	Annual Savings: \$330,048.00
Total Monthly				\$491,976.00	

Price for Using Software Companion Toolsmith

Includes updates, maintenance, use by anyone at your company, as many diagnostic tools as you can make, and unlimited use with your end-users! Create custom tools in minutes.

Annual Enterprise Subscription Tier Based on 20 Support Agents: ~\$15,000.00

Total Saved Through Shortening the Overall Service Resolution Process:

Net Annual Savings of \$315,048,

Section B: Projected Cost of Diagnostic Tools without Software Companion Toolsmith

Development of Diagnostic Tools—per Product. Weeks used x weekly salary.

	Weeks-Dev	Weeks-QE	Weeks-TS	Salary-Dev	Salary-QE	Salary-TS
Totals	7.25	7	1.25	\$ 13,942.31	\$10,769.23	\$1,442.31
Total Man Weeks: 15.5 Total Cost: \$26,153.85 Total Time in Weeks: 7.25						

Update of Diagnostic Tools—per Product. Weeks used x weekly salary.

	Weeks-Dev	Weeks-QE	Weeks-TS	Salary-Dev	Salary-QE	Salary-TS
Totals	3.75	3.5	0.75	\$ 7,211.54	\$ 5,384.62	\$ 865.38
Total Man Weeks: 8 Total Cost: \$13,462.54 Total Time in Weeks: 3.75						

Initial Cost for 10 Products: \$261,538.50 Two Annual Updates x 10 Products: \$269,250.80
Cost to Develop Diagnostic Tools with Software Companion Toolsmith: INCLUDED

In Summary, Company X Can Realize an Annual Savings of \$315,048 in Support Costs and well over \$250,000 in Development Costs for an Annual Total of:

Over \$500,000, while simultaneously increasing customer satisfaction and securing future maintenance revenue.