



# DBA SURVIVOR GUIDE

YOUR FIRST 100 DAYS

by Thomas LaRock

A complimentary excerpt of *DBA Survivor: Become a Rock Star DBA* from Apress, Inc.

## FOREWORD

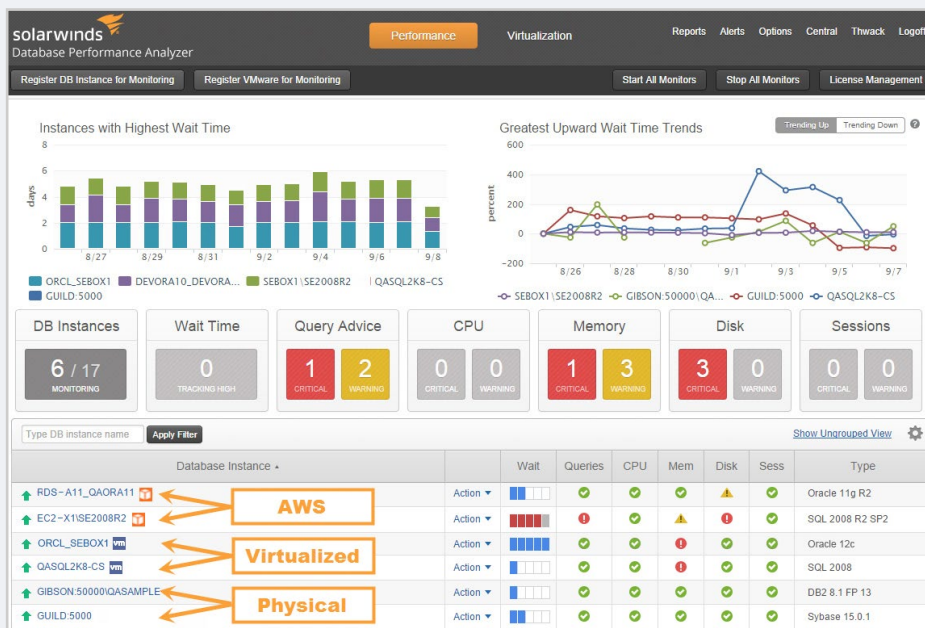
## FROM SURVIVE TO THRIVE: WHY PROACTIVE APPLICATION PERFORMANCE MONITORING IS CRITICAL

By Gerardo Dada, Vice President, Product Marketing and Strategy, SolarWinds

When Thomas LaRock published DBA Survivor Guide: Become a DBA Rock Star, it offered much needed guidance and advice for newly minted database administrators. Even for experienced DBAs in charge of hundreds of production databases, it can be daunting just to keep everything running. Generally, their day is a constant exercise in prioritizing the most critical items on the list, so optimizing for faster performance is often left until something is truly broken.

Slow performance, however, has become much more than a nuisance—it's become a business-critical challenge. How critical? Consider that 40% of users will abandon a website that takes more than three seconds to load. And when things slow to a crawl, the database is often blamed. One recent study reports that 88% of respondents said database performance issues are a significant contributor to application performance (Gleanster, 2015). But, determining the cause of slow performance is a time-consuming task, and challenging to assess which of many factors: memory, storage, network, CPU, impacts performance the most.

SolarWinds Database Performance Analyzer (DPA) offers new and veteran DBAs and database developers a simple way to determine the true cause of slow database performance.



*Pinpointing Root Cause of Database Performance Issues  
with SolarWinds Database Performance Analyzer*

With one glance, DPA provides a visual correlation of performance with server, storage and virtualization and database resources. With DPA you can easily, expose, understand and share information about the root cause of slow-downs with DBAs, developers, IT managers, system administrators, QA and data architects.

In short, DPA makes it possible to move beyond survival, and turn database performance management into a powerful tool for the business.

We hope you enjoy this complimentary excerpt from Tom's fantastic book, and share your journey to rock star DBA with us via [LinkedIn](#), [Twitter](#) (@SolarWinds) or [Facebook](#).








## FIRST THINGS FIRST

Now that your dream has come true and you are the resident database administrator, what do you do first? Like any other job in the world you need to get started on... well... something, right? But what? Where do you focus your energies in order to best demonstrate your value to your organization? Knowing where to begin is crucial for your success and this Survivors' Guide is going to help you best determine how to get started, help you plan for your first 100 days, and even help you decide who you should eat lunch with on a regular basis.



**TIP:** You are going to be constantly judged by your tangible results from this point forward.

In this eBook we will discuss the following:

-  Putting together your initial checklist
-  What to do with the information you have gathered
-  Responding to unfamiliar alerts
-  How to break the ice
-  Mr. Right vs. Mr. Right Now



## WHAT YOU HAVE IN COMMON WITH THE PRESIDENT

Do you really have anything in common with the President? Yes. More than you probably realize. First, about half of the people around you doubt whether you are qualified to actually hold the job you have been given. Second, every time you make a decision or plot a course of action you will constantly be criticized, even by your supporters. And third, you are going to be judged by what you accomplish in your first 100 days, good or bad, even if it was something outside your control.

Every four years we elect a new President, and the person in office is always subject to approval ratings. You will have your own version of this fact of life—it's called your annual performance review. Come review time, you want your approval rating to be as high as possible.

Sound awful? Perhaps, but it really is not all that bad as long as you are aware of these things when you start. The most important objective for you is your plan of action when you first arrive. If you think you can show up, grab a cup of coffee, and ease into your new position then you are mistaken. Your cup of coffee can wait until after you start gathering the information you need in order to do your new job effectively.

And what information is that? How about some of the basics first: What servers are you responsible for? What applications are you expected to support? What time of day are the applications used? Who are your customers? Are the databases being backed up properly right now? How would you know if the backups were failing? With so many items to check, it can become very overwhelming, very fast. That is why you need to put together a checklist of the bare essentials to get started. Then, after you are able to get a handle on your environment you can start making some short-term plans for improvements. Before you know it, your first 100 days will be behind you and you will be able to look back and see just how far you have come in a short amount of time.

Trust me, it is easier than it sounds, you just need to be organized.



## YOUR INITIAL CHECKLIST

By now you should be sitting at your desk on what we will call Day One. Your initial meetings with HR are over, you have gotten a tour of the place, and you are making certain you have the access you need to at least get started. Things like email, for example. Oh, and access to the database servers they expect you to administer.

That is the very first piece of information you need, right? What servers and systems are you responsible for? Without that little nugget of knowledge it is going to be very difficult to make any headway as you start your long, slow, journey upstream.

Your initial checklist is divided into sections. Why? Well, because I like to put things into lists and categorize the lists if possible. It just helps me remember the bigger picture and not worry about missing any particular detail. I would encourage you to try the same thing and see if it helps—but don't worry if you want to start your list differently.

Now, the checklist has three main sections: One section pertains to gathering information on what I simply call, your stuff. Another section deals with finding information on the customer's stuff. Still another section you need to consider is what I call your action plans. Those three areas are where you start to focus your efforts on Day One—find your stuff, find your customer's stuff, and start making an action plan.

So, a sample of a checklist might look like this:

1. Create a list of servers
2. Check database backups are running
3. Spot check and verify you can do a restore from one of those backups
4. Build a list of customers
5. List the "most important" databases
6. List upcoming deliverables/projects
7. Establish Environmental Baselines
  - Server configuration check
  - Database configuration check
8. Compose your recovery plan (not your backup plan, your recovery plan)



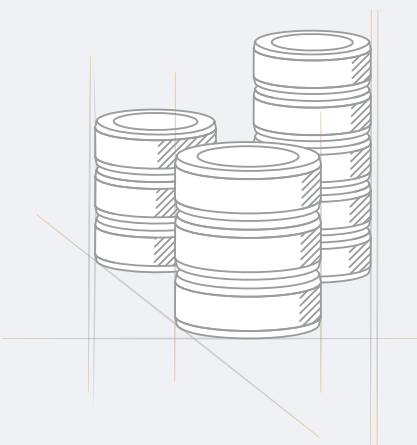
Notice that my checklist is missing a few things that a lot of people will tell you are a must for DBAs to address daily. It doesn't include things like, routine index maintenance, performance tuning, reviewing event logs, etc. Sure, all of those things are necessary, but we are still on your list of items for Day One. Everything I have mentioned so far is going to take you more than a few days to gather. If you get tied up troubleshooting some stored procedure on Day One, then you are setting yourself up for a massive failure should a disaster hit and you haven't had time to document your recovery plan.

Would you rather be a hero for telling that developer to stop writing cursors, or be a hero for informing a customer that you can have their database back up and running in less than 30 minutes? I know which choice I would make so soon after taking a new position.



**TIP:** These tasks will all take longer than a day to perform, but do not delay in getting the checklist started.

On your first day, explain to your manager that you'll be gathering this inventory data first. By taking the initiative to perform due diligence first, you're showing them that your first mission is to safeguard their data, your job, and their job too. They probably won't be able to produce the inventory for you, and they're going to want it even more than you do. You will have plenty of time later on for the other stuff and it will fall naturally into your environment baseline and subsequent action plans as you bring standards to your enterprise.



#### AN "OSHA" MOMENT

“ At one company, when I laid out this plan for my first two weeks, my boss said out loud: “Holy sh\*t! We don't have this information for the rest of our servers either.” He then pulled his sysadmins in, told ‘em to drop what they were doing, and to do the same thing I was doing—but for the file servers, Exchange servers, etc.



**Brent Ozar**  
reviewer

Let's look at why each of the items in the checklist is important, and important to address from Day One. If today is your first day, you want to begin these tasks right now.



## 1. CREATE A LIST OF SERVERS

Not sure I really need to explain this one, but you best get an idea of exactly what you are expected to administer. Chances are your initial list will not be complete, but it gives you an immediate baseline of reference. Trust me, at some point some person will walk up to you and start talking about a server you never knew existed. And they will be very confused as to why you have never heard of it, since they work with it all the time. There is a database there, and you are the DBA, so you should already know all of this, right?

Do your best to gather as much information as possible, right away. That way you will know more about what you are up against and it will help you when it comes time to formulate your action plans, which will be very different if you have five or five hundred instances to look after.

I know what you are asking yourself. You are asking “Self, how do I find out what I am responsible for?” I suggest you start with your immediate supervisor and go from there. The trail may take you to application managers and server administrators. For example, your boss might say that you are responsible for the payroll databases. But what are the “payroll” databases? You may need to run with that initial bit of information and do some detective work to track down the specific databases involved. Along your journey, you will be given an overview of the complexity that is your new home. Any detective work that you’re forced to do will pay off handsomely by deepening your knowledge and understanding of where you work.

If you are looking for a technical solution to finding databases, there are a handful of ways to get the job done. One of the simplest ways is to use the SQLCMD utility with the `-L` parameter to return a list of database servers that are broadcasting on the network. Since it is possible that some servers may not be found with that tool, you would be wise to still ask around when putting together your list.

And where do you keep such a list? I like to keep a lot of notes written down in a notebook. When I say “notebook”, I’m not talking about a computer. I’m talking about a physical, paper book that I can hold in my hands. Others prefer to put everything into word documents and store them on their computer. The paper notebook works best for me because I have found that I learn better by rote than by typing. What I write with a pen stays with me better than what I type with a keyboard.



One more thing to mention would be the importance of having a list of servers you are not responsible for. There is a chance that you have some systems in your environment that are maintained strictly by vendors. If something goes wrong with one of those servers it is important to know who is responsible for what. And if someone tells you that you do not need to worry about a server my advice would be to get that in writing. Believe me, when disaster strikes, you had better be able to provide proof about the systems that are your responsibility.

If you are storing your information in a spreadsheet or similar document, then you can go back over time to better track how your environment is changing. Are you administering more servers, or fewer? More databases or less? Are all your jobs running? Backups working? This information will surely help with your approval ratings. If nothing else, you'll have the backup detail to be able to clearly show your boss how your job is changing, and how your responsibilities are increasing. You'll be better able to document your value to the company.

### ON SELLING YOUR VALUE

*“You’d think your boss would know your value, and would know what it is that you do all day long. My own experience though, is quite opposite that. Early in my career, I suffered a painful layoff. Looking back on that event, I’ve come to believe it was in part because I failed to manage upwards in a way that kept my boss apprised of my true value to the company. My boss was local, in the same building. My internal clients were all over the country, with some even in Europe. I worked closely with those internal clients, delivering results—but my boss was outside the loop. He and I didn’t interact much. We never really needed to interact on a daily basis, so we didn’t. Guess who got the boot when the mandate came down to cut head-count?*

*I learned many painful lessons from that first layoff. One is that bosses are busy people. Your boss has his own set of responsibilities. He may have only a dim idea as to what exactly you do. Make sure that you do a better job than I did at keeping your boss “in the loop” and apprised of your good work. Someday your boss will be in a meeting with HR about cutting staff. When he’s in that meeting, you want him to be able to recall from the top of his head all the good that you do for the company.*



**Jonathan Gennick**  
editor



## 2. CHECK DATABASE BACKUPS

Now that you know what servers you are responsible for, your next question had better be: Are the backups running? Do not assume that everything is working perfectly. Dig into the details yourself and verify that each and every instance is being backed up. Oh, and that includes the system databases (master, model, msdb) as well as all user databases. Check that the backup files exist, that the directory they are being stored in is on a disk that has adequate space, and check to see if there have been any recent failures.

You will also want to note the backup schedule for the servers and databases. You can use that information later on to verify that the databases are being backed up to meet the business requirements. You would not want to find out that a business is expecting a point-in-time restore for a database that is only being backed up once a week.

I cannot stress this enough, but if there is one thing and only one thing for you to focus on as a DBA, it would be to ensure that you can recover in the event of a disaster. And any good recovery plan starts with having a reliable database backup strategy.





### 3. VERIFY THAT YOU CAN RESTORE

Every now and then I like to do a spot check of my backups by choosing some and attempting to restore. For me, I like to attempt a same server restore as well as a restore to a different instance. Note that this is not the same as just restoring the file header in order to verify that the file is readable. No sir, I want to restore the entire database, it's just my preference. I would not typically have time to verify each and every backup file, and would not attempt to do so. I just want to select a few backup files from a few servers and ensure that there are no issues.

Now, I said a "same server restore" in the last paragraph and I want to make something very clear regarding this action.



**CAUTION:** Be mindful when performing a same server restore to a production server.

There, now I feel better. Same server restores should be done only when you know it is safe to do so. How do you know if it is safe? Well, you could always ask someone. Or you could look to see if people are currently connected to the instance. Or if your database is fairly small and manageable and the restore should only take a few minutes, at most. And of course, if it is a non-production server, then you could feel safer.

Which databases should you verify can be restored? You could focus your efforts on just about any group or set of databases. The real goal here is for you to become familiar with the restore process in your new shop, as well as verifying that the backups are usable. Make certain you know all aspects of the recovery process for your shop before you start poking around on any system of importance. It could save you some embarrassment later on should you sound the alarm that a backup is not valid, and what turns out to not be valid is really just your understanding of how things work.



**TIP:** Another good reason to practice restores is so that you can perform one when the call comes in at 3:00 a.m. the morning after you've stayed up late three nights in a row. You'll be tired. You won't be thinking clearly. The practice that you've previously put in will serve as your brain's "muscle memory". Practice makes you better able to do the right thing under pressure and when fatigued.



## 4. BUILD A LIST OF CUSTOMERS

If you know what servers you are responsible for, then start asking who the customers are for each of those servers. Note that this line of inquiry can result in a very large list. And with shared systems, you could find that everyone has a piece of every server.

The list of customers is vital information to have. For example, if there is a need to reboot a server, it is nice to know who you need to contact in order to explain that the server will be offline for five minutes while it is rebooted. And while you compile your list of customers, it does not hurt to know who the executives are and which servers they are most dependent upon.

When you start listing out the customers you should also start asking about the applications and systems those customers use, and the time of day they are being used the most. You may be surprised to find some systems that people consider to be relatively minor are used throughout the day while other systems that are considered most important are used only once a month.

There is an extra benefit to building this list—you get your chance to begin building a relationship with your customers. Chances are they will be flattered that you care enough to seek them out. Putting together your list gives you an excuse to reach out to your customers. Think of it as an icebreaker, an easy excuse for the two of you to meet.

### INFREQUENTLY USED CAN STILL BE CRITICAL

“Tom’s time-of-day comment made me think back to when I used to work managing a data warehouse. We had a database that could be offline three weeks out of the month, and nobody would notice. It was a critical database, with a specific purpose. The workload just happened to be such that the database wasn’t used three weeks out of the month. It sure was used during week #1 though. During week #1, it had better be up 24x7. The remaining weeks, it didn’t matter so much. The moral? Don’t assume a correlation between how much something is used and how truly important it is.”



**Ken Simmons**  
reviewer



## 5. LIST THE “MOST IMPORTANT” DATABASES

While you gather your list of important customers, go one step further and find out what their most important databases are. This could be done by either (1) asking them or (2) asking others and then (3) comparing those lists. You could be very surprised to find that they do not match. You will be even more surprised to find how many people can forget about some of their systems and need a gentle reminder about their importance. As a DBA, we treat all databases with equal importance but we do recognize that some databases are indeed more important than others, especially given any particular time of day, week, or month.

For example, you could have a mission-critical data warehouse and everyone in the company could tell you that this system is vital. What they cannot tell you, however, is that it is only used for three days out of the month. So, the database could be offline for 21 days and no one would say a word.

Another example would be a trading platform. This system could be used heavily for nine or ten hours each business day. But for the remaining hours of the day it is not used at all. Does that mean that when these systems are not used they are not important? No, it does not. What it does mean is that you are gathering more details about the systems. If 17 different groups mention some small tiny database, but they consider the database to be of minor importance, you could consider it very important because it is touched by so many different people.

Another factor here is recoverability. If the business requires you to be able to recover a database to a point in time, then you should consider that to be an important database as well—even if that customer sits across from you and says “it’s not that important.” I have lost track of the number of times a customer has told me offhand that “the system really is not a big deal, we need to recover to within five minutes, and we use it all day long, but you don’t have to worry about it.” Yeah, right. Until a disaster happens, of course, and you are on the hook to put everything back, and quickly!



## 6. LIST UPCOMING PROJECTS AND DELIVERABLES

If there is someone around who can help you see the current and upcoming projects, it will be a big help for you to know what is about to be dumped in your lap. You want to minimize the number of surprises that await you—and knowing what projects are currently planned helps you understand how much time you will be asked to allocate for each one. And do keep in mind that you will be expected to maintain a level of production support, in addition to your project support, in addition to the action tasks you are about to start compiling.

You'll also want to know which servers will be decommissioned in the near future so that you don't waste time performance tuning servers that are on Death Row. Keep an eye on this angle as well.



## 7. ESTABLISH ENVIRONMENTAL BASELINES

Baselining your environment is a function that gets overlooked frequently. The importance of having a documented starting point cannot be stressed enough. Without a starting point as a reference, it will be difficult for you to chart and report on your progress over time.

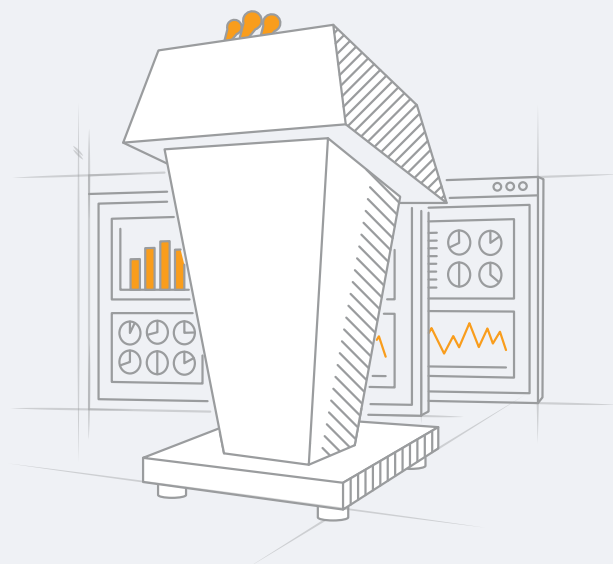
The idea of gathering baseline information on the servers you administer is to simply find out how many deviations you have from an ideal state. You have already done one baseline item; you have evaluated your database backups. Since this checklist item can become a little cumbersome in a short amount of time, try to focus on the basics.

For example, on a standalone database server with direct attached storage, your data files should exist on a separate drive from your log files. How many servers have that same configuration? And what about those disk drives? Is there a standard RAID configuration? You might be surprised to find that servers may not always be physically built the same. How about the memory settings or the number of processors for the instances? How big are the data and log files and what is the average CPU utilization?

Focus on a handful of big-ticket items to gather as your baseline information and keep the information going forward. You can add additional items over time as needed.

I would also advise you to streamline the ways in which you collect this information. For example, if you have 100 servers you really do not want to remote to all 100 servers just to examine the memory settings for each one. You will need to learn how to efficiently gather your information, much of which can be collected using tools such as System Center Configuration Manager (SCCM) or Operations Manager. If your environment does not have those tools, then you may need to script out your baseline queries using PowerShell, T-SQL, or some alternative.

Alternatively you could use a defined Central Management Server in SQL 2008 to run queries against multiple instances at the same time. Or you could look to use Policy Based Management to report on and possibly enforce configuration options. No matter what method you choose the goal remains the same—to track changes over time. Once you've captured a picture of how your environment looks, now you can start to document how to return servers to their original state if something goes wrong.





## 8. COMPOSE YOUR RECOVERY PLAN

Notice how I said “recovery” plan as opposed to backup plan. In your checklist so far you have already verified your database backups are running, started to spot check that you can restore from your backups, and got an idea of your important databases. Now is the time to put all of this together in the form of a disaster recovery (DR) plan. For each one of those databases listed as important, you should write down the exact steps involved to recover should a disaster happen.

Make no mistake about it—should a disaster happen, your job is on the line. If you fail to recover because you are not prepared, then you could easily find yourself reassigned to “special projects” by the end of next week. The best way to avoid that happening to you is to practice, practice, practice. Your business should have some scheduled DR tests (perhaps once a year), but that should not prevent you from doing your own smaller DR tests on a more frequent basis.

Document each system and all the steps required to recover. Is the database in full recovery mode? How frequently are you doing transaction log backups? Write down the backup schedule so that it is clear where the restore points are. If your customer expects recovery to the minute and you are in simple mode, then you are heading for a true disaster.

And don’t forget about recovering from past days or weeks. If your customer needs a database backup restored from two months ago make certain you know every step in the process in order to get that job done. If your company uses an offsite tape storage company, and if it takes two days to recall a tape from offsite, then you need to communicate that fact to your users ahead of time as part of your DR plans.





## YOU HAVE YOUR INFO, NOW WHAT?

Everything until now should have kept you busy until at least lunchtime on Day One. OK, maybe it will take you a few days to get all of the information on your checklist. You may or may not have a lot of that information already available—so the amount of time it takes to gather everything will depend upon the current structure of your particular department in your shop. (By the way, get used to hearing and saying “it depends”, more on that later.) No worries though, because you can transition into the next phase while you continue to gather your data.



## MEET WITH YOUR MANAGER

While gathering the data, you should meet with your manager to discuss your preliminary findings and work together to prioritize the work that needs to be done. If you have found one server that is not running any backups, or the backups are failing, then that server must take priority over everything else. I don't care which one of your “most important” customers screams at you on Day One to fix a query they wrote, which filled up a 70Gb log drive, you must ensure you can recover from a disaster.

Continue to review your checklist with your manager and determine which items need to be done, which items need the most resources and effort, and which ones can wait for now. It is very important that you and your new manager agree upon the tasks and their priority before you get started. This will serve both of you very well—and it gives you a chance to be measured from this point going forward.

By the way, should you and your manager disagree regarding your priorities, make sure you have that documented. It will serve you well when someone comes around asking why something specific has not been done yet and you politely explain why you have been working on something else. Then if there is still disagreement, you, the other person, and your boss can get together to re-evaluate your priorities.



**TIP:** Document, but not to be confrontational. Don't whack your boss upside the head with his old emails. Instead, use them as gentle reminders that you are committed to working on the priorities that your boss lays down. Your boss will appreciate that you are putting your efforts towards his priorities.

After you have met with your manager, you will be constantly measured. Everyone is going to want or need you for one thing or another. Every interaction you have with others will serve as a form of measurement and will be verified by the tangible evidence of actions you've taken. If you have that checklist, and your manager is aware of the checklist, you now have something tangible to deliver on starting from Day One.

Think about how important this will be for you. In three or four weeks, someone, somewhere, will ask your manager how things are working out with the new DBA. Your manager will be able to quickly affirm all the great work you have been doing (let's just assume you have been doing great work and not sitting on your arse all day). You will be able to track your progress easily and eventually you can prepare a brief summary or presentation that details your progress over your first few months.



## MEET WITH DEVELOPERS

If your company has in-house development teams, take some time to meet and greet them. They are going to figure most prominently in your success in the coming months. Find out what projects they are working on, what roadblocks they are currently facing, and see if there is an opportunity for you to offer assistance.

There is no question that the developers are going to keep you very busy. The developers will be the ones who look to push the limits with the technologies at their disposal. The end result will be an abundance of little training opportunities for you to take advantage. Fill up the transaction log? Let me understand more about what you were trying to do. Filled up the entire log drive? Your remote query has bad performance? Your stored procedure with 27 nested cursors is taking too long? Here, let me help you with that, and perhaps we can find a better way to get the job done.

Will every developer want to be your friend? Absolutely not. It is rare to be somewhere where everyone gets along at all times. If you come across a developer that believes they know more than anyone else in the room, you can sit back and chuckle because you already know that the DBAs are the smartest—otherwise you wouldn't always be called to fix their problems all day long.



## MEET WITH THE SERVER ADMINISTRATORS

Depending on your shop, you may or may not have server administration duties. Many DBAs focus only on the administration of the server instances and leave the O/S administration to others. If that is the case for you, then you need to make an effort to get to know this team. The server administrators have their fingers on the pulse of the whole company. They know which systems are most important. They know where to find some extra hardware when you are in a pinch.



### SYSADMINS ARE YOUR FRIENDS

“ I agree really strongly on this point that Tom makes. It’s a huge help if you are on friendly terms with the SysAdmins that manage your servers. If a DBA can be comfortable sitting down with a SysAdmin to troubleshoot a problem together, well, I’m not sure it gets any better than that.



**Jonathan Gennick**  
editor

System administrators are also going to be the ones to build out your infrastructure. It is vital that you are aware of the hardware you are using as well as the hardware that might be coming in the door. If your company looks to implement a SAN, they will not be asking you how to configure it, they will be asking someone on the server team. Since a SAN can, and will, have dramatic impacts on database servers, you need to be involved in those discussions. Ideally, you will be invited to the table to discuss such things early on, and the more time you spend with the server team the better chance you have of knowing about what changes are coming your way.

You are also going to be needed frequently to help the developers and the server administrators talk to and understand each other. Why is that the case? Well, think of yourself as the universal translator. Developers know code, they know various facets of the business, and because they focus on those items daily they are not spending time racking servers, installing routers, replacing hard disks, allocating space on the SAN, etc. Server administrators do those things, but they do not know how to code or build applications that will drive the business forward at the same speed as a developer.

Unfortunately these two groups rarely interface unless there is a problem, or when there's an opportunity to point fingers at each other. That is where you come in. The developers can point their fingers at you and explain what the roadblock is at the moment and you can turn and speak to the server administrators in a language they can understand. Without you, that does not happen. Frustration levels rise, and your shop ends up spending far too much money on hardware to solve issues that could be remedied by altering a few lines of code.

So, by meeting with both groups, you continue to brush up on your language skills and help to keep the peace—and ultimately you can also help to reduce costs.



## MEET WITH YOUR CUSTOMERS

By now you have met with almost everyone possible, except for your actual customers. Actually, when you are a DBA, everyone is your customer. But let's focus on the actual end users of the applications and servers—these are the people who are going to give you a much different view when compared to any other group.

Why is it so important to meet with them? Well consider this story. One day a man answered an ad to rent a bedroom above a garage at a family's home. The man was well kept, had a job (but no one knew what it was), but was also always around. He was a nice enough fellow, never was late with his rent, and the family liked having him around. The family indulged the man who always liked to go with them for rides in their car. It did not matter the destination, the man would ask "mind if I take a ride?", and the family would let him.

Every time he rode in the car he would ask one or two questions about the car, but not too many to lead the family to think anything was strange. He would simply work the questions into a conversation as they drove. "So, think they have enough cup holders in here?"—which would lead to all sorts of comments about the design and functionality of the car.

Eventually the man told the family it was time for him to move out, he had found a new place to live. Before he left, he finally told them what he did for a living—he worked for a car company. The reason he asked those questions was a way to get honest feedback about their car's design. The feedback they provided was much more reliable and honest than any survey could hope to be. The man observed the family actually using the product!

And so you need to do the same with your end users. Start understanding more about how they are using their tools, what their frustrations are, what little things would help them, and therefore make your business more efficient and productive. Go and ask your customers if you can take a ride in their car. You will be amazed at the things you will find.





## IS THAT ALERT SERIOUS?

So, you have collected your info. You have meet with all sorts of people. And now you get back to your desk to find about 1,500 emails waiting for you. What to do? Some of those emails might be database alerts. Better look into those.



**TIP:** Setting up rules in your email to file alerts automatically is a wonderful thing, but only when done carefully and correctly.

Is a specific alert important? I don't know. It could be. Or it might be safe to ignore. If it is really important, then someone will start yelling at you soon enough, right? Well, technically that may be true, but let's not let it come to that, shall we? Here is a basic rule for you to follow when it comes to alerts.



**REMEMBER:** If you don't know what they are for, then you need to find out why you are getting them.

Simple enough to remember—and following that rule will be the difference between you being thought of in a positive way versus a negative way. Consider the following scenario:

*It is a lazy Sunday afternoon. You are relaxing at home. Your Blackberry receives a message that has some cryptic message. You do not understand what the message means. You assume that someone else must have gotten the same message and knows what to do about it, so you will be able to find out tomorrow what it meant. The next morning at work you have users complaining about one of the important databases. Reports and batch loads should have finished hours ago and are still running. The users want to know what the problem is and how soon before you fix things. You have no idea what the problem is, but you do remember that cryptic message from yesterday. You do some research, find out where the alert came from, find the message itself, start to understand what it might mean, and finally isolate the problem. You start to get the system back into working order but by then it has essentially been out of the water for most of the day. No one else received the page, just you, and therefore all blame lies at your feet and your feet alone.*

Sound like good times? It wasn't, and it describes an event that happened to me in my first few weeks on the job. Since that day I have made it my personal mission to make certain that any alerting system of mine would only provide value. Too many times I hear about shops where they are flooded with emails. Some are alerts, some are just informational. I am always looking for things to be efficient, and that is why I only want alerts for items that are actionable.

So, to start let's assume that every alert that comes your way is something that requires action on your part. Note that I did not include this in the initial checklist, but only because responding to alerts should come secondary to the tasks above. How much sense would it make for you to spend hours responding to an alert for one of your least critical systems while the most critical system goes without a database backup for yet another day?

But now is the time to ask someone, "Hey! What is our alerting system?" If possible you should review the different types of alerts, and see if they have a way to mark as critical versus informational.

How can you judge the seriousness of the alert? How do you know if it is truly important? Simple. If it relates to one of the databases or customers that you are responsible for, then there is no question you should look into the alert. If it is not listed above, there is no question you should look into the alert. I hope that clarifies matters for you. In other words, if you don't know the answer, then the answer must be yes.



**REMEMBER:** If you do not know how serious an alert is, then treat it as if it is very serious.





## SHOULD I LOOK INTO THAT ALERT?

Now that we know any alert of unknown origin should be considered serious until proven otherwise, the real question becomes: What next? If you have other people on your team, then your first stop would be to ask them. If it is just you, then ask your manager. If those two stops lead to dead ends, then you have a few other options at your disposal.



## REVIEW THE ALERT SYSTEM

Start digging into the alert system to see if you can find out why the alert in question was generated. In doing so, you will learn more about the alert system, making it easier to understand future alerts that may come your way. If the alert you're worried about is for something straightforward like, Database Mail sending you an email based upon a failed job in SQL Agent, then your research effort should be easy. If the alert system is a little more complex as in Operations Manager, then it may take a little longer to ascertain the details—but the extra effort now will be worth it in the long run.

Also, while reviewing the alert system, make certain it is not sending emails to one mailbox (yours) as opposed to a distribution list. Always use a distribution list to send your alert notifications, unless you plan on being available and responsive every minute of every day.

But what if you do not know how the email was sent? What then?



## ASK THE DEVELOPERS

If you have already gathered details on the server or database involved in the alert, then go ask the developers if they have additional knowledge on the nature of the alert. They may be able to shed some light on events—perhaps a batch load failed or they were doing some testing. Worst case, it is a sign of something important and they will be very glad that you have come to them before notifying anyone else.

But what if they have no additional details? What then?





## ASK THE SERVER ADMINISTRATORS

The next logical step is to approach the server administrators. One of them might be familiar with the alert system or the nature of the alert itself. They may also have configured some email rules that might be of benefit to you.

Ask around to see—again you may be uncovering something more serious than you realize and the server team would be very glad to have you come to them before you were to contact your customers.



## ASK YOUR CUSTOMERS

When all else fails, go ask the customers themselves to see if they are aware of any issues at the moment. You may find that they are having some troubles already and they will be happy to see you. By helping them with any possible issues you may be able to also uncover the nature of the alert. And if they are not having any issues at the moment, it will be beneficial if you raise awareness now so that they can contact you again later should any abnormalities arise.

Every time an alert comes in, you are being given the opportunity to learn something new. Embrace that and learn to love it—because if you do not it will quickly become the bane of your existence to the point you want to throw your Crackberry against the wall.

And when that time comes, you will need to commiserate with someone, preferably over some food and beverages.



## HI, WANT TO GRAB SOME LUNCH?

By this point you have done a basic health check on the systems, you know your customers, and you know which systems are most important. So now is the time you start learning more about the people you will be working with most closely in the coming weeks and months.



**TIP:** When starting a new job, everyone is a stranger. You need to learn how to meet others in order to develop mutually beneficial relationships.

Now, I am a very shy person by nature. Most people do not believe me when I tell them that, but it is very true. As such, I have a hard time meeting new people, even people I work with. This makes it difficult for people to get to know me, and for me to know them. In turn, that can make it difficult to have a strong, successful working relationship. If there is one thing I have learned over the years, it is the importance of managing relationships with those up and down the chain from you.



## MEETING BY EATING

For people who are shy or naturally introverted, how are they able to build and manage relationships with others? It is not an easy thing to do. In fact, it can seem overwhelming at times. But I have found one very common trait that everyone shares—everyone eats.

It is true. At some point everyone will eat. You may find them sitting down to have lunch. You may find them going to get a cup of coffee in the morning. Chances are you are going to see everyone you work with, at one time or another, eating. And there is also a good chance that you tend to eat from time to time as well. What's to stop you from eating together every now and then? To break the ice, go to someone else's office or cubicle around 11:45 and introduce yourself as the new guy. Ask if they know of any good places to grab lunch in the area.

I take a walk to get a cup of coffee every morning. I do not always buy a cup for myself, but at least one person in the group does. So why do I go? Because the ten minutes I spend with them gives me ten more minutes of interaction with them outside of work, which I would not have had if I sat in my cubicle.

Every now and then I find myself having lunch, or walking to get lunch with others. And on days when I feel like I have been eating too much I head to the gym with a different group of coworkers. In other words, I have been able to find ways to interact with different groups of people, in different ways, in an effort for us to get to know each other a little better. Does that mean we always get along? Absolutely not. What it does mean is that we make an effort to get along.



## POLITICKING, OR NOT?

I must warn you against getting mired in office politics. That is a no-win situation for anyone. Aim for good, healthy relationships with coworkers. Healthy relationships are not about conspiring together and beating others down. Healthy relationships are about working together and building others up.



**CAUTION:** Do not get mired into gossip or office politics.

Do not get involved in conjecture on others, gossip, or rumors. Stick to known facts only, not hearsay, and do not knowingly spread false information. Office politics can easily drag down moral, create divisions, and just make life miserable.



## DEALING WITH INTROVERSION

Not all technical people are introverted, but many are. The nature of our work calls for detail-oriented people comfortably working alone for long periods of time. Those traits are in stark contrast to the outgoing salesperson who can meet 15 new people in 10 minutes, and be bosom-buddies with all of them. We are the people who can sit in a conference lecture with 100 other people, and walk out not having met a single one of them.

Yet career success demands that we make ourselves visible. Sometimes we need to step up and lead, or present on what we know. We need to be comfortable with meeting people and working together. And we need to do all these things while being true to ourselves and respecting the personalities that we were born with.



## ACCEPT THAT NOTHING IS WRONG

If you find that you're the sort of person who wants to have lunch alone, do not despair. The very first thing that you should do is:



**TIP:** Realize that you are not put together wrong

Yes, it is true. If our entire society were made up of social bubbleheads running around and meeting each other, we'd all be in a world of hurt. We need those outgoing people. Your company needs its sales force, for example. But we need all the other personality types too. Those salespeople need products and services to sell.

The key is to know thyself. Know your own comfort zone. Play to your strengths. Develop skills that you lack. You can actually treat extroversion as just another skill to master. Doing so will pay handsomely in both personal and job satisfaction.



**TIP:** A good book on personality traits is *Please Understand Me*, by David Keirsey and Marilyn Bates.



## TAKE CARE OF YOURSELF

The defining trait of an introvert is sometimes described like this:

***An introvert draws energy from being alone, and expends energy to engage in social situations.***

Does that sound like you? If it does, then don't be afraid to lay down some boundaries. For example, when travelling to conferences, my editor generally blocks out breakfast for himself. That's his "me time", when he can charge his batteries for the day. He also takes himself off the clock for lunch. He'll reach out aggressively to engage you if you walk by the Apress booth, but at lunch he switches off the "selling mode" and just enjoys a good meal.

When travelling with colleagues, don't feel obligated to spend every waking moment with them. There is nothing wrong with telling your coworkers that you need some time alone, and that you'd rather not go barhopping until 2:00 a.m. Be up front about why you're doing it. Your colleagues will understand. Some may admit to being in the same boat.

Did I just say to eat lunch with coworkers? I'm going to contradict myself now by giving you permission to eat alone if you need some space. Meals are a great venue in which to engage coworkers and get to know them better. But you don't have to spend each and every meal with them. Take care of yourself first. If you need lunch alone to get your head together for the afternoon, then take it.



## BE OPEN AND STRAIGHTFORWARD

Don't try to be something you're not. Don't be afraid to admit that you're not a super-outgoing person. Sometimes when I want to meet someone, I just walk up and tell that person I'd like to meet them. It's a straightforward approach lacking in subtlety, but why not? If you take an interest in people, and attempt to befriend them, most people will respond positively regardless of how clumsy your own efforts are.



## JOIN SOME ORGANIZATIONS

Organizations provide structure. And that structure makes it easier to step outside of one's comfort zone. That's why organizations such as the Professional Association for SQL Server (PASS) are so valuable. They provide a socially safe venue in which to make professional contacts. The fact that the other members are there for the same reason only helps.

Organizations also provide opportunities to lead, and to develop speaking and presentation skills. You can start small. You don't need to make your first talk in a 1000-seat room. Start with a local user group. Work your way up as your comfort and confidence grow.

Don't make it all about work either. The skills and confidence that you develop working in a community or church organization easily translate into the professional space. If you can work with people to make a difference in your community, then you know you can do the same in your job.



## ACCEPT THAT NOTHING IS WRONG

Didn't we just have this heading? We sure did, but the point is so important it bears repeating. It is a sad truth that society in general tends to smile upon bubbly and outgoing people, and to conversely frown upon those who are quiet and withdrawn. That's an unfortunate and a wrongheaded attitude. Do not get sucked into it!

Everything I've said about developing people skills is intended with just that one goal in mind—to develop your people skills. Don't try to make yourself something you are not. Your personality is a big part of what makes you good at your job. Don't fight that goodness. Play to your strengths. At the same time, work to develop your skills in other areas. There is no contradiction in doing both those things.

### CONVERSATION STARTERS AND SMALL TALK

*“The art of making small talk in social situations is one that did not come easily or naturally to me. Even in elementary school I would notice that some people could sit down to the lunch table and almost instantly be engaged in conversation with everybody around them. What's more, the conversation would drift from topic to topic in ways that I seldom could follow. Just when a topic got interesting and I was ready to dive in deeply with a comment, the group would suddenly be talking about something else. I was always a half step off.*

*In business, it's helpful to have a few tricks for starting conversations. One of my favorite ways to begin a conversation is with a question. If you ever catch me manning the Apress booth at a conference like PASS, the question you're most likely to get hit with is: “Do you have any of our books?”. It's a logical question that usually leads to some back-and-forth conversation about specific books and what's good and bad about them.*

*My question is a good conversation-starter, but it's sincere too. I truly want to know what readers like and dislike about our books. I want to know which topics resonate with our audience, and which do not. I'm hungry for help, really, in doing my job better.*

*I might ask other questions too. Sometimes a company name will strike me as interesting. Other times I see names of cities that I've visited or lived in. Questions are a great way to get a conversation rolling, but be sincere. Don't just **show** interest. **Be** interested.*



**Jonathan Gennick**  
editor



## TRACK YOUR PROGRESS

Remember that checklist? Good, because you need it in order to chart your progress over the coming weeks. As a DBA, a lot of your work is done behind the scenes. In fact, people will often wonder what it is you do all day since much of your work is never actually seen by the end users. Your checklist will serve you well when you try to show people some of the tangible results that you have been delivering.

No matter how many people you meet and greet in the coming weeks, unless you can provide some evidence of tangible results to your manager and others, people will inevitably wonder what it is you do all day long. If your initial checklist shows that you have 25 servers, six of which have data and logs on the C: drive, and two others had no backups at all, it is going to be easy for you to report later that your 25 servers now have backups running and all drives are configured properly.

In the end, it is not your effort that people will remember. It is the end result. Make certain you keep good track of your progress so that the facts can help provide people a way to understand exactly what you have been delivering.





## GET PROACTIVE

Now that a lot of your research is complete, you should have a very clear picture of what you are up against in your new environment. As time goes on, you will be establishing your own identity in your new role. Most DBAs will fall into one of two categories: Mr. Right, or Mr. Right Now.

Which one of the following would best describe how you handle your work tasks or how you want to be perceived?

**Mr. Right Now:** This is the guy who is always available to solve any problem. He is very visible, everyone knows his name, and everyone knows what he does. He fixes things. It doesn't matter what time of the day or night, he is always there. When there is a problem, any problem, you call "Mr. Right Now" and he finds a way to solve that problem. He rarely fails to deliver and everyone praises his hard work and effort. However, when he solves a problem, he solves it just enough to move on to the next emergency.

**Mr. Right:** This is the guy that you rarely see. When there is a problem he pokes his head around the corner—but since there are few problems people never really know that he is around. In fact, people often ask "So, what is it you do around here?". But when "Mr. Right" solves a problem, he solves it for good, making sure that he never has to touch that server again to address that issue. Then, he proactively does that same fix against the rest of his servers before they experience a similar issue, thereby solving problems before they happen.

Which one would you rather be? Would you rather be the guy that is very visible and that the end users love because he is always there to fix things? Or would you rather be the guy that people rarely see and have no idea what you do for a living?

How about you answer this question: Which one (Mr. Right or Mr. Right Now) would you consider to be a junior DBA versus a senior DBA?



**TIP:** Being able to fix things does not make you a senior DBA. You become a senior DBA by making certain that things do not break in the first place.

While "Mr. Right Now" may be getting slaps on the back from the end users for all of his efforts, the ultimate question must be asked: Is he fixing problems that are being caused by his fix of an earlier problem?



There are two reasons for someone to be a “Mr. Right Now”. The first is, that there are far too many problems at the onset of their tenure and they are going to be fighting a long battle to get all of the fires under control. The second reason is that they are a junior-level administrator who knows just enough to fix the current issue.

“Mr. Right”, however, is the guy who not only puts out the fires as they happen, but has the knowledge and experience to put controls into place to make certain that the problems have a lower likelihood to happen again. So, they not only know enough to fix the current issue, but they know enough to solve the root cause of the problem. The solutions they put in place do not complicate the environment, they help to lessen their workload.

As you start your new role, there is a very good chance that you will need to be “Mr. Right Now” for an extended period of time, even if you are at a senior level. It could be the case that there is more work than one person can handle. Of course, if you have your checklist and are well organized, you’ll know where to focus your efforts first. But as you go down that checklist you need to keep thinking about solutions, as opposed to a quick fix.

Over time, you really want to be “Mr. Right”, and have your environment configured in a way that practically heals itself. And do not despair if you feel like you are losing out on an opportunity to show people how valuable you are to the company. As you continue to configure your environment and are proactive in heading off potential trouble, you will find you have more time to be visible in different ways.



**TIP:** Work smarter, not harder. Find ways to be proactive in solving issues.

Would you rather be stuck in a server room all weekend? Or, helping someone learn to write better T-SQL code? Would you rather be rebooting boxes as quickly as possible during the business day as you troubleshoot some issue? Or, would you rather be helping someone in finance understand how to configure and build their own reports using SSRS? Trust me, there are plenty of ways for you to show your value without having to run around the place as if it were on fire.

Back in college we were always told to study smarter, not harder. The same principal applies here—work smarter, not harder.

## Become Mr. Right! Compliments of SolarWinds

Get your free trial of Database Performance Analyzer  
and get proactive with your database management.

[www.solarwinds.com/dpa](http://www.solarwinds.com/dpa)

A complimentary excerpt of *DBA Survivor: Become a Rock Star DBA* from Apress, Inc.