Citrix XenApp is an application virtualization product from Citrix. It allows users to connect to their corporate applications from various computer systems and even mobile devices. Together with tight integrations with other systems, such as different third-party authentication services, troubleshooting XenApp has become more complicated.

This book teaches you how to approach troubleshooting complex issues with XenApp deployments, understand the problem, find a fix or workaround, determine the root cause, and apply corrective steps wherever applicable. The book progresses to give you an idea of the many supportive components that play an important role in XenApp’s application delivery model and that should be considered while troubleshooting XenApp issues. At the end of this book, you will see how and where to use supportive components that help minimize XenApp issues.

Who this book is written for

This book is for Citrix administrators and engineers who are currently managing Citrix XenApp in a production environment and want to learn how to troubleshoot XenApp issues in the shortest possible time. It is assumed that you have a basic understanding of XenApp's components.

What you will learn from this book

- Get to know about troubleshooting methodologies and perform a root cause analysis of the problem
- Understand the individual components and interactions required for a successful XenApp environment
- Identify and troubleshoot issues around the core Citrix components
- Identify XenApp components and their roles and learn how they interact with communication channels
- Gain tips and tricks to optimize Citrix deployments
- Monitor and optimize the XenApp environment

In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter 1 'Basic Troubleshooting Methodology'
- A synopsis of the book’s content
- More information on Troubleshooting Citrix XenApp®
Dragos Madarasan works as a cloud support engineer for one of the largest IT companies in the world. After previous stints as a freelance IT consultant and working for a managed services provider, he now enjoys tackling complex scenarios and using his knowledge to help clients who have taken their business to the cloud.

Dragos publishes interesting cases on his personal blog and whenever time permits, he enjoys taking part in community-led events as a technical writer and speaker.

When not in front of a laptop, Dragos enjoys running and reading books on his Kindle.

Dragos has previously worked as a technical reviewer for *Microsoft SCCM High Availability and Performance Tuning* and *XenApp 6.5 Cookbook*, Packt Publishing.
**Suraj Patil** is an accomplished virtualization consultant with 8 years of experience in the information technology industry. He is a specialist in designing, building, maintaining, and optimizing Citrix, Microsoft, and VMware oriented infrastructures for large enterprises and mid-sized organizations. He holds a bachelor's degree in Information Technology and has many certifications from vendors such as Microsoft, VMware, Citrix, Red Hat, and Cisco.

Suraj is a Citrix Certified Professional—Mobility (CCP-M), VMware Certified Professional—Data Center Virtualization (VCP5-DCV), VMware Certified Professional—Network Virtualization (VCP6-NV), Microsoft Certified Solutions Expert (MCSE)—Private Cloud, and Cisco Certified Network Associate (CCNA).

Suraj currently lives in Mumbai and works for a Fortune 500 company as a Citrix consultant.

You can visit his blog at [www.v12nsupport.com](http://www.v12nsupport.com).
Preface

In today's world, every organization has at least one or two applications that run their entire business.

Accessing these applications becomes mission-critical if the organizations want to sustain their businesses. Any users to get their job done need access to these critical applications.

Citrix XenApp with its FlexCast technology offers a flexible solution to mobilize Windows applications with a highly secure delivery model.

As with a large number of applications nowadays, XenApp requires minimal configuration and installation decisions and an experienced administrator can configure an infrastructure in a matter of minutes or hours. Particularly because installation is a simple process, it is troubleshooting that sometimes becomes difficult. Troubleshooting is not a science, it's an art form; and behind every issue there is a cause, so you must plan for the situation.

With this book, we will cover troubleshooting preparation, general processes, and real-world examples to resolve any XenApp issue in a proper manner.

By the end of this book, you will have enough knowledge to maintain and optimize your own Citrix XenApp environment.

What this book covers

Chapter 1, Basic Troubleshooting Methodology, covers understanding problems, breaking down problems into their affected components, and finally, testing problems.
Preface

Chapter 2, Understanding the Citrix® Components, introduces you to the supportive components that are part of the XenApp infrastructure. It will describe the process of starting a published application and how each component comes into play.

Chapter 3, Troubleshooting XenApp® Issues, explains standard troubleshooting processes and how to follow them to troubleshoot complex XenApp issues in a mission-critical environment.

Chapter 4, Troubleshooting Other Issues, covers troubleshooting with provisioning services, NetScaler Gateway, Citrix Storefront, and other infrastructure components.

Chapter 5, Monitoring and Optimizing, explains using Citrix Director, EdgeSight, and NetScaler Insight Center to optimize XenApp infrastructure.
XenApp has grown into complex software with ever-expanding infrastructures in place. Together with tight integrations with other systems, such as Remote Desktop Services, Active Directory Domain Services, and other third-party authentication services, troubleshooting XenApp has become more complicated.

This first chapter will cover basic troubleshooting methodologies, how to approach troubleshooting complex issues, and what the full process entails—understanding the problem, finding a fix or workaround, determining the root cause, and applying corrective steps where applicable.

In this chapter, we will cover:

- Basic troubleshooting guidelines and methodologies
- Breaking down problems into affected components
- Resolution testing
- Root cause analysis and corrective actions

**Troubleshooting 101**

As with many software nowadays, XenApp requires minimal configuration and installation decisions, and an experienced administrator can configure the infrastructure in a matter of hours.

Particularly because the installation is a simple process, it is the troubleshooting that sometimes becomes difficult.
Basic Troubleshooting Methodology

It is important to note that a solid grasp of XenApp components, interaction, and workflow is needed before performing troubleshooting.

Most times troubleshooting can be easy, either the solution is straightforward, perhaps because the administrator has experienced this problem in the past, or a simple Internet search for the particular error message will yield a Citrix knowledge-based article or blog post for that particular problem.

In all other cases, troubleshooting needs to be performed in an organized fashion so the solution is reached in the shortest amount of time possible, since many times the problem could involve downtime for a large number of users.

Although seemingly unimportant, one of the most important aspects of troubleshooting is producing a comprehensible problem statement:

- How is the problem manifesting itself?
- Who is facing the issue?
- When did the issue start?

Without clear answers to these questions, an ambiguous problem can undermine efforts for a solution.

Consider the fact that most of the time an issue is generally logged by a service desk or call center (first line of support), who might escalate it to a desktop support team (second line of support), and who will in turn escalate it to a Citrix team (third line of support).

If any piece of information is misunderstood by the analyst logging the incident, this in turn can be propagated to the Citrix team with the information being completely irrelevant to the troubleshooting process or even incorrect.

Consider the following scenario: a user working in the finance department calls the helpdesk and complains that an accounting application stopped working in Citrix. The application was working fine last week. The help desk agent performs a series of basic troubleshooting steps and escalates the problem to the next line of support without requesting additional information.

Consider the following questions:

- How many users are affected? Has the application stopped working for other users?
- What is the expected behavior of the application?
- Are you in the same location as last week or a new office?
While each question in itself might not directly lead to a solution, it can narrow down the problem considerably.

For instance, a positive answer to the first question might indicate this is a server or network issue as it affects multiple users.

A positive answer to the third question might indicate this is a network error; the next logical step would be to check whether there are any networking restrictions applied to subnets or IP addresses in the current location.

The fifth question is meant to check whether the issue is specific to a user, machine, or location.

**Breaking down problems**

When troubleshooting difficult cases, after making sure you have understood the problem (and that the information provided is correct and relevant), you must ensure a systematic approach to problem solving.

One strategy that can be used is *divide and conquer*, where you break down a problem into individual, easily solvable problems.

Considering the previous example where a user calls the helpdesk, one way of breaking down the problem is testing each *sub-system* individually, for example:

- Are the Citrix servers online and healthy? Check the monitoring systems.
- Is the network link reliable? Run a continuous ping and check whether websites load correctly.
- Is the problem easy to reproduce on any machine or does the problem follow the user?

For instance, in the case of XenApp 7.5/7.6, the following components can be considered:

- Server/desktop operating system machines and virtual delivery agents
- Delivery controller
- StoreFront
- Citrix receiver
- NetScaler Gateway
Basic Troubleshooting Methodology

Going back to our example, one or more components can be causing a problem. For instance, there might be a problem with the **Virtual Delivery Agent (VDA)** on the server/servers hosting the finance application. This prevents the controller from being able to use the broker agent part of the VDA to communicate with the server.

Another possibility is that the issue is related to authentication. The StoreFront or the NetScaler Gateway (if the user is outside the corporate network) might have problems authenticating users to site resources.

It is important to quickly rule out as many components as possible. For instance, we could quickly test if the Citrix web page is accessible internally (where only the StoreFront component is used) and externally (where we might be reaching a NetScaler Gateway first). If the webpage is accessible internally but not externally, we would need to focus our attention on the NetScaler Gateway.

Alternatively, if, in both scenarios, the webpage does not load, we might focus our attention on the actual servers and/or delivery controllers.

Let’s take another example: several users complain that during the day, applications published in XenApp start to become slow every morning.

The users mention that the slowness has been happening for some time, but it has only started to impact them recently.

Consider the following questions:

- How long has the initial slowness been observed (several weeks or months)?
- Around what hour is the impact noticeable?
- How long is the impact—several hours or the entire day?
- How often does the problem occur—one a daily basis or only on specific days?

Answers to these questions can be tremendously important when dealing with performance-related issues. For example, it is important to establish whether the performance is affected during specific hours/days (to help to isolate whether a scheduled operation is causing the issue) and whether it is consistent (for example, happens every day of the week or happens only on specific dates/days).

Further breaking down the problem could consist of:

- Determining whether there is any correlation between systems tasks (antivirus, backup, web filtering, and so on) and the start of the slowness
- Determining whether the impacted application(s) can be tied to a group of servers, users, or user locations
- Analyzing past monitoring data for any negative performance trends
Use NetScaler Insight Center to collect information about traffic, performance data, and session information for NetScaler Gateway.

Resolution testing
Before describing how resolution testing should be done by administrators when troubleshooting a XenApp environment, there are two terms that need explaining. In software development terms, resolution testing is known as the process of retesting a bug once the development team has released a fix.

Regression testing is another methodology where test cases are re-executed for previously successful test cases.

Both testing methods are an important part of testing a software solution, as sometimes fixing one bug can cause regressions in other parts of the solution leading to new bugs.

Citrix administrators need to think in the same manner as testers do. Once the problem has been understood and a fix has been identified, then the fix or workaround can be applied. Once the fix is applied, the next step is to attempt to reproduce the initial issue. If this is not successful, it would generally mean the initial issue is resolved and most of the time that is the case.

However, besides testing for the initial issue, a Citrix administrator should also perform a number of tests to ensure that the fix does not negatively affect the XenApp infrastructure in another manner, for example, another application might stop working.

Root cause analysis
Once the problem has been correctly understood, and a fix applied and tested, the next step would be to determine the root cause and apply corrective actions if needed.

The Root Cause Analysis and Corrective Actions (RCCA) is the final step in troubleshooting a problem and involves determining the root cause of the issue and outlining any suggestions and recommendations for actions that can be implemented to prevent the reoccurrence of the underlying issue.
Most of the problems encountered in the Citrix world can be grouped into three categories:

- Performance issues, for example, applications are slow to start, the network is unreliable, and so on
- Incorrect configuration, for example, XenApp is not properly configured during the initial installation or a subsequent change
- Broken code leading to unexpected behavior from XenApp or underlying components—these are trickiest to debug and probably the least encountered

Most root cause analysis reveal either a performance issue or an incorrect configuration.

Where a root cause is deemed to be performance related, tackling them usually requires improvements in the infrastructure—bigger bandwidth, more servers, faster disks, and so on. The real challenge is determining how much to scale the infrastructure so that performance falls back within acceptable parameters without spending a large amount of money.

Preventive steps for these types of problems could be:

- Ensuring a capacity management process is in place
- Monitoring Citrix infrastructure for active usage
- Creating an easily scalable Citrix architecture

Incorrect configurations are usually self-evident; for example, if an administrator performs a change that negatively affects the Citrix infrastructure, again usually almost immediately. The root cause analysis, therefore, focuses on the following questions:

- Has the change management process been followed?
- Have the risks been properly established and highlighted?
- Have actions been considered to minimize the risks?
- Is there a backup plan in place in case a rollback is needed?
- What is the impact of a failed change and how will it affect users or production environments?

Changes where the risks have been appropriately highlighted ("Changing X setting has the risk of bringing down the Citrix site for 15 minutes"), where the change is performed out of hours (minimizing risks) and has a proper rollback plan in place are perfectly acceptable.
Most changes have the potential of causing downtime, but if the proper change management process is followed, the risks are minimized and the potential outage reduced.

Preventive steps for this type of problems could be:

- Ensuring the risks have been correctly identified and presented to the business
- Ensuring steps to minimize the risks have been identified
- Ensuring there is a clear backup plan in place

Finally, during troubleshooting, a number of changes might need to be done before the final fix is found. It is, therefore, a good idea to keep a track of these changes while the troubleshooting process is actively ongoing.

Once the correct fix has been identified, a retroactive change request should be logged in the IT system. Although, in this instance, the change hasn't followed the standard change management approval process, it is still useful to have changes logged in the system in case they need to be looked up in the future as part of troubleshooting previous changes.

**Summary**

In this chapter, we covered the basic methodologies of troubleshooting. We've described troubleshooting as first understanding the problem, breaking down the problem into its affected components, and finally, testing. The problems are solved once the fix or workaround is identified.

We highlighted the fact that sometimes, problems can be traced back to scheduled changes in the infrastructure, and that keeping track of changes is important as it can help in identifying the problem and mitigating or resolving it.

Finally, we discussed the root cause analysis, the process of determining the root cause of the issue (not just the fix/workaround) and preventive steps to minimize the reoccurrence of the issue.

In the next chapter, we will cover the Citrix XenApp/XenDesktop components, identifying and describing each one briefly. We will then talk about how the components interact and which communication channels are used during the interaction.
Where to buy this book
You can buy Troubleshooting Citrix XenApp® from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.