



CITRIX TECH BYTES

OCT 2, 2017

Reboot Schedule Internals



Andrew Ogle

20



Support for Reboot Schedules in the current generation of XenApp & XenDesktop products was introduced back in version 7.0, however these schedules work differently from those in XenApp 6.5. **In this post, I hope to add some technical detail to how these schedules operate in the 7.x products and perhaps dispel a few urban myths relating to them.**

You do not need this information to use reboot schedules, and it's most definitely not a User Guide, but if you want to understand why they behave as they do, this article might help.

Reboot schedules in 7.x support many configurations all requiring differences in behaviour, but here we'll cover a common case that roughly corresponds to a reboot schedule in XenApp 6.5: a scheduled reboot of a delivery group containing power-managed RDS (Server OS) VDAs.

It's important to distinguish a reboot *schedule* (the delivery group to reboot, when, and over how long) from a reboot *cycle* (a single occurrence of that schedule – for example, the one that started at 2am last Wednesday). This terminology is used in the SDK, as in **Get-BrokerRebootSchedule[V2]** (the V2 variant was added in 7.12 to support **multiple schedules for a delivery group**) and **Get-BrokerRebootCycle**.

In this example, we'll consider a scheduled reboot of a delivery group containing 60 power-managed RDS VDAs where the scheduled duration is 2 hours. Ten of these VDAs are off at the start of the cycle.

When a reboot cycle starts for this schedule the interval between VDA reboots is calculated using the scheduled duration and the total number of machines in the group, or the total number having the correct tag if the schedule is restricted by tag (in 7.12 or later), in our case this is $120 \text{ minutes} / 60 = 2$ minutes.

The VDAs to be rebooted are equally divided between two phases, with the most preferred VDAs in phase 1 and the rest in phase 2. Factors that make a VDA preferred include having fewer user sessions, currently being registered, and not being in maintenance mode. VDAs that are powered-off are not included and are immediately counted as having been skipped. In our example this leaves 50 VDAs with 25 in each phase.

We now start to reboot VDAs in phase 1. Nothing happens to VDAs in phase 2 yet.

Cycle Phase 1

Firstly, all VDAs in the phase are put into drain mode; this prevents new sessions being launched but allows existing sessions to be reconnected. The overall status of a VDA in a reboot cycle and in particular whether it's in drain mode can be seen in the **ScheduledReboot** property of the **Get-BrokerMachine** cmdlet.

Now, in our example, every 2 minutes for the next 50 minutes, one VDA in the phase is picked for reboot. Each selection uses the same criteria used to divide the VDAs into phases, but the preferred order of VDAs within the phase is reevaluated on *every* selection so that the 'most preferred' VDA at that time is always chosen. This results in a major difference from XenApp 6.5 where the VDA reboot order is fully determined ahead of time.

As each VDA is picked, the following occurs:

1. If, and only if, there are user sessions present *and* the schedule includes a warning message, the message is sent to the VDA (possibly repeatedly) until either the warning duration is reached or all sessions have logged off.
2. A shutdown request is sent to the VDA's hypervisor.
3. When the VDA has powered off it is taken out of drain mode and a power-on request sent to its hypervisor.

This work typically overlaps and runs concurrently for multiple VDAs within the phase. Once all VDAs in the phase have been picked (after 50 minutes in our example) the phase is complete. Many, but typically not all VDAs have completed their reboots and are available for use again at this point.

Cycle Checkpoint

Once phase 1 is complete, a check is made to see if at least one VDA has successfully re-registered and is available to accept new sessions. If not, the cycle pauses until at least one is available again. If none are available after a timeout period the cycle is abandoned (VDAs in phase 2 aren't rebooted).

This check avoids the possibility of rendering all VDAs in a delivery group unusable by, for example, rolling-out a faulty image using MCS. By default we'll wait a maximum of 30 minutes plus the configured warning duration for a VDA to become available.

Cycle Phase 2

Once a VDA from phase 1 is available again, phase 2 starts. All VDAs in phase 2 are placed in drain mode and each is processed in turn as described for phase 1 above.

Thus in our example, a VDA is picked every 2 minutes until after 50 minutes all VDAs in phase 2 have been selected, and many would typically have rebooted and be available again. At this point phase 2 is complete.

We now wait to allow outstanding operations to complete so that we have accurate counts of successfully rebooted VDAs, and of reboots that failed. These counts are shown in the output of **Get-BrokerRebootCycle**. Again, by default we'll wait a maximum of 30 minutes plus the configured warning duration for operations to complete.

At this point the cycle is complete.

Pictorial Example

The picture below shows a smaller reboot cycle example with only ten VDAs. VDAs 1 and 2 are rebooted first because they have no sessions present (no message is sent), and both are available again when the checkpoint is reached so phase 2 starts immediately. In this case, because the warning duration is

longer than the interval between reboots, both VDAs 9 and 10 do not start to reboot until after the nominal end time of the cycle.



Variations on a Theme

While our example covers power-managed RDS VDAs, physical RDS VDAs can also be rebooted in a reboot cycle (the VDA restarts Windows at the request of the Controller). However, physical VDI (Desktop OS) VDAs cannot be rebooted.

Reboot cycles for shared VDI VDAs are generally similar to our RDS example, but here the VDAs are not powered-on again once shutdown. It is left to power management policies of the delivery group to restart the VDAs as required.

Any Questions?

What happens to VDAs in Maintenance Mode?

VDAs in maintenance mode are always the least preferred ones when picking one to reboot and if they remain in maintenance mode are left to the end of their phase. If at that time there remain VDAs in maintenance mode they are skipped over and not rebooted. So VDAs in maintenance mode for a short period during a cycle are likely to be rebooted, but if they remain in maintenance mode throughout then they are not.

Is Drain Mode the same as Maintenance Mode?

No. A VDA can be placed in maintenance mode by the administrator to prevent automatic power management (including that of reboot cycles). Drain mode is different and is exclusively controlled by a reboot cycle. It cannot be set or cleared by the administrator.

Reboot cycle drain mode is also distinct from Terminal Services Drain Mode supported by Windows RDS machines (which is not used).

Why are VDAs with least sessions rebooted first?

The algorithm is designed to minimise disruption to end users. By rebooting unused or lightly loaded VDAs early it tries to ensure that later in the cycle, as users are forced off more heavily used VDAs, as many newly restarted VDAs as possible are available to accept new sessions.

Why are VDAs that are off ignored?

The objective of a reboot schedule is to periodically restart the VDAs' Operating Systems. This resets their memory to a clean state, possibly resets disk images, potentially aborts hung processes and removes unused sessions. However, if a VDA is already off then no action is required as it will be in a clean state when it next starts.

Powered-off VDAs are counted though when initially calculating the interval between reboots. This means that the reboot *rate* is consistent even if a large number of VDAs happen to be off, with the cycle potentially finishing earlier than its scheduled duration.

Why are reboot cycles sometimes shorter or longer than the scheduled duration?

A cycle may be shorter than scheduled because some VDAs are off, but a warning duration, the checkpoint at the end of phase 1, or the wait for outstanding operations to complete at the end of phase 2 may extend a cycle.

Accelerated performance.
Increased flexibility.
Revolutionary security architecture.

Available
NOW!

CITRIX®
XenServer 7.1



Tagged under:

Citrix Tech Bytes

Citrix Virtual Apps & Desktops (XenApp & XenDesktop)

20 Comments

Citrix Blogs

1 Login ▾

Recommend 3

Tweet

Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name



Brian • a month ago

I agree with Nick below. These are important features and we need to be able to control things a bit more than we can right now. We are attempting to do things with scripts but have been told that Citrix doesn't support anything that is scripted. At a minimum we need:

- Longer Drain Period, 24 hours would be nice
- Bi-weekly schedule option
- Ability to place machines in Drain Mode administratively if the reboot schedule ignores machines that are in Citrix Maintenance mode.

^ | ▾ • Reply • Share ›



Nick • 3 months ago

Good article but Citrix clearly have some work to do to achieve the elusive "6.5 feature parity" we're continually promised. It's pretty shocking that customers paying software maintenance have lost this (and other) functionality (critical for 24x7 Xenapp shops). IMHO Xenapp 6.5 support should have been extended until at least 2008r2 EOL. We're struggling to replicate the dual Xenapp 6.5 restart schedule we setup to minimise user disruption for 24/7 staff. In short approx 10%* of our servers reboot at 9pm. *This is sufficient capacity for our night shift staff who logon after 9:30pm. We use the 6.5 scheduling load evaluator option to prevent connections to these servers from 11am the same day. This means they do useful work during the day (hosting sessions

logged on before 11am) but are usually free from active sessions by 9pm. Another schedule restarts the remaining 90% of servers between 2-3am (and another load evaluator ensures these don't take any new sessions from 9:30pm -> 3:30am).

I've looked at the CVAD 7 1811 restart functionality and it's still lacking (no pre-reboot drain period, limited scheduling options:- daily/every day but not specific days, max restart window 5hrs). Now we have multiple schedules I don't understand why the two phase approach described above persists. With tagging you could have servers rebooting on alternate nights but an unlucky user could be logged off a (draining)

[see more](#)

^ | v • Reply • Share ›



Citrix Luke • 3 months ago • edited

I would like to see this change: "Drain mode is different and is exclusively controlled by a reboot cycle. **It cannot be set or cleared by the administrator.**"

^ | v • Reply • Share ›



Patrick Dyke • 4 months ago

It would be nice if you could specify a duration longer than 5 hours (which leaves 2.5 hours per cycle based on your description.) We typically drain servers for 8-12 hours (one shift) to avoid interrupting users. We have plenty of capacity to have some percentage of servers OOS at any given time, but we do not want to impact users which are 24/7. It seems this framework for reboots could accommodate that, if we have full control over all the parameters, including restart duration, number of cycles, whether to complete a restart if users are still logged in.

^ | v • Reply • Share ›



Giri • 5 months ago

Could you please post the Back end script for the Schedule reboots for both Phase 1 and Phase 2

^ | v • Reply • Share ›



Denis • 8 months ago

As part of the VDA reboot, do sessions get logged off gracefully or forcefully ? or are sessions simply ignored after the notification period expires and the VDA simply "resets" ?

^ | v • Reply • Share ›



Billy C • a year ago

Would be nice to have a feature to force users sessions off if they are still logged on beyond the requested reboot duration.

^ | v • Reply • Share ›



Chenxue Wu • a year ago

In your example, DDC will pick all powered on machines within 2 hours.

What if some of the VDAs had not complete the reboot task?

Will it stay in power off status or keep doing restart task?

^ | v • Reply • Share ›

**Frank Esch** • a year ago**@Manu Panicker** No, this works also with not power managed VDAs.

^ | v • Reply • Share ›

**Manu Panicker** • a year ago

Does the servers have to be power managed?

Would i be able to do it for my machine catalogs set manually using AD?

^ | v • Reply • Share ›

**Jayakumar P** • 2 years ago

Is that possible to reboot my Controller (DDC) from the delivery group restart schedules?

^ | v • Reply • Share ›

**Erlend Westervik** ➔ Jayakumar P • a year ago

Then who will orchestrate the rebooting :D?

The answer is no.

^ | v • Reply • Share ›

**Niels Geursen** • 2 years ago

Hi Andrew, I was wondering if it would be possible to cancel the scheduled reboot cycle for 1 Server. For example when 1 user is logged on to server X and the reboot schedule is already started voor Server A is there a possibility to cancel only the reboot for server X?

Also assume that the user doesn't have the time to logoff and logon again to logon to a rebooted server.

Thanks Niels

^ | v • Reply • Share ›

**Chenxue Wu** ➔ Niels Geursen • a year ago

I also want to know if it would be possible to cancel the scheduled reboot cycle for 1 Server.

But seems the smallest unit for a Reboot Schedule is one Delivery Group...

^ | v • Reply • Share ›

**Chenxue Wu** ➔ Niels Geursen • a year ago

In XenApp and XenDesktop 7.12 and later, we offered a new PowerShell command to create multiple reboots schedules within one delivery group.

You can set the specific machine with a RestrictTag wo perform different restart schedules.

See <https://www.jgspiers.com/cr...> <https://citrix.github.io/de...>

<https://support.citrix.com/...>

^ | v • Reply • Share ›

**Citrix Luke** ➔ Chenxue Wu • 3 months ago

Exactly this. Use New-BrokerRebootScheduleV2 to create a schedule and limit it using the "-RestrictToTag " switch. Then you can just untag the server you want

the server you want.

^ | v • Reply • Share ›



Kenan Cakmak • 2 years ago

Very useful information. Thank you for sharing!

^ | v • Reply • Share ›



Stefanos Evangelou • 2 years ago

Hello,

Thank you for the useful information. How about Citrix User Profiles? I need to be able to gracefully cleanup any local copies of the user's UPM profiles before actually rebooting each VDA. Is there a way to incorporate a local profile cleanup mechanism (e.g. delprof tool) before each VDA reboot in the Citrix Reboot Framework?

^ | v • Reply • Share ›



Ron Krogel • 2 years ago

I am curious if there is a manual way of setting to something longer for reboot cycle other than a week. Say a registry setting?

^ | v • Reply • Share ›

YOU MIGHT BE INTERESTED IN

CITRIX TECH BYTES

NETWORKING

6 HOURS AGO

Citrix ADC for Kubernetes: Smarter canary deployment with Spinnaker

Please note, this blog post describes the beta version of the solution, which is still undergoing final testing...





Stay Informed.
Subscribe Today.

email address

SUBSCRIBE

Yes, I'd like
to hear
about offers

and services from Citrix by email.

☐ Yes ☐ No

Citrix will process your data according to our [Privacy Policy](#)

CITRIX TECH BYTES

NETWORKING

JUL 26

Identify the best links to transfer your data with Citrix SD-WAN

Citrix SD-WAN uses an algorithm to select the links through which data packets are sent, and it's expected...



Marcos Negrão



CITRIX TECH BYTES

NETWORKING

JUL 23

Citrix SD-WAN programming with Windows PowerShell – Part 1

Citrix SD-WAN has available a powerful NITRO API that enables configuration and monitoring of the product programmatically. NITRO provides its...

**Shoaib Yusuf**

0

[CITRIX.COM](#)[ABOUT](#)[CONTACT](#)[TERMS OF USE](#)[COOKIE PREFERENCES](#)

© 2019 Citrix. All rights reserved.

