Microsoft

Troubleshooting Active Directory Lingering Objects

Analysis and Troubleshooting

Microsoft Windows Server 2012 R2 Datacenter

Hands-on lab

This lab walks you through the troubleshooting, analysis and resolution phases of commonly encountered Active Directory lingering object issues. You will use ADREPLSTATUS, repadmin.exe and other tools to troubleshoot a five DC, three-domain environment.

TechEd
Europe 2014

This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Acknowledgments	
Author	Justin Turner
Bio	Justin is a Support Topic Lead and Senior Support Escalation Engineer with the Identity (Directory Services) team based in Irving, Texas, USA. He has created or contributed to many training courses, knowledge base and TechNet articles for Microsoft over the past 14 years. He teaches Microsoft employees and customers new product architecture, is a charter Microsoft Certified Master (MCM), Microsoft Certified Trainer (MCT) and holds an M.S. degree in Computer Education and Cognitive Systems (Instructional Systems Design).
Contributors	Special thanks to the following contributors: ***Ken Brumfield*, *Arren Conner, David Everett, Rob Lane, Bill Long, Salih Karagoz, Herbert Mauerer*** *and* ***Tim Williams***
Reviewers	Many thanks to the following individuals that spent their own time reviewing and providing feedback: **Arren Connor, David Everett, Salih Karagoz, Rob Lane, Wayne McIntyre** and **Dean Wells**
Content authors	Thanks to the following for writing foundational content on lingering object troubleshooting: Arren Conner, David Everett, Jasmin Hashmani, Rob Lane, Glenn LeCheminant, Herbert Mauerer

Introduction

Estimated time to complete this lab

75 minutes

Objectives

After completing this lab, you will be able to:

- Understand the cause, identify the symptoms, and identify ways to resolve lingering object issues.
- Accurately determine the full scope of a lingering object problem, document which cleanup methods to use to resolve the issue and are able to able to explain how an Active Directory Administrator can avoid lingering objects in the future.

Prerequisites

Before working on this lab, you must have an understanding of the following:

- Active Directory logical model (core components)
- Active Directory replication model
  - Active Directory replication concepts
  - Active Directory replication topology
- Experience troubleshooting Active Directory replication
  - See "Troubleshooting Active Directory Replication Errors" lab
- Experience using repadmin and LDP

However, detailed step-by-step instructions are included, so those new to Active Directory lingering object troubleshooting will be able to follow along.

More: The appendix contains a lot more detail, background information, sample log output, references and information on how to reproduce the issues in a lab. Ensure you save off the document for later reference.

Overview of the lab

In this five DC, three-domain lab environment you will work through one of the most challenging Active Directory replication problems seen by IT professionals globally: Lingering object identification and cleanup.

In the lab, you will be given everything needed to eradicate lingering objects from your environment. Included free of charge: all the tools, background information and timesaving techniques needed to save the day on your next lingering object-induced Active Directory outage. We will work through the symptom, cause and resolution phases of lingering object troubleshooting. Several scenarios and cleanup methods are used along with a full description when alternate cleanup methods are needed in the comprehensive lab guide.

Scenario

Active Directory replication problems are one of the top support call generators for Microsoft. Lingering object issues are the most challenging Active Directory replication issue to resolve and are routinely escalated through multiple levels of support. On average, it takes twice as long to resolve a lingering object problem than it does the average AD replication issue as a result of the complexity involved in its troubleshooting.

Lab Activity Overview

Exercise 1: Lingering Object Fundamentals

During this exercise, you will review terminology, symptoms and analyze replication metadata of lingering objects.

Estimated time to complete this exercise: 5 minutes

Exercise 2: Lingering Object Discovery

During this exercise, you will generate diagnostic data via repadmin, ldifde and replfix. You will then analyze that data and document all lingering objects in the environment.

Estimated time to complete this exercise: 10 minutes

Exercise 3: Lingering Object Removal Methods

Task 1 - Lingering Object Removal Using LDP

During this task, you will remove a lingering object using LDP

Estimated time to complete this exercise: 10 minutes

Task 2 - Lingering Object Removal Using Repadmin

During this task, you will remove lingering objects from the environment using repadmin /removelingeringobjects

Estimated time to complete this exercise: 5 minutes

Task 3 - Lingering Object Removal Using REPLDIAG

During this task, you will most lingering objects from the environment using Repldiag.

Estimated time to complete this exercise: 5 minutes

Task 4 - Lingering Object Removal Using Lingering Object GUI tool

During this exercise, you will remove the remaining visible lingering objects using Lingering Objects Liquidator.

Estimated time to complete this exercise: 5 minutes

Task 5: "Live" lingering object (abandoned deleted object) remediation

During this exercise, you will identify and re-animate live lingering objects.

Estimated time to complete this exercise: 15 minutes

Exercise 4: (Optional) Lingering Link identification and cleanup

During this exercise, you will identify all lingering-linked values in the environment. You will them remove them in order to ensure group membership consistency amongst DCs.

Computers in this lab

This lab uses computers as described in the following table.

Virtual Machine	Role	IP Address	DNS Client settings
DC1.root.contoso.com	Domain controller in the forest root domain, DNS, GC, All FSMO roles	192.168.10.1	192.168.10.2; 127.0.0.1
DC2.root.contoso.com	Domain controller in the forest root domain, DNS, GC	192.168.10.2	192.168.10.1; 127.0.0.1
ChildDC1.child.root.contoso.com	Domain controller in a child domain in the forest, DNS, GC, Domain-wide FSMO roles	192.168.10.11	192.168.10.1; 127.0.0.1
ChildDC2.child.root.contoso.com	Read-only domain controller (RODC) in the child domain in the forest, DNS, GC, MinShell	192.168.10.12	192.168.10.11; 127.0.0.1
TRDC1.treeroot.fabrikam.com	Domain controller in a tree-root domain in the forest, DNS, GC, Domain-wide FSMO roles	192.168.10.21	127.0.0.1; 192.168.10.1
WIN8Client.root.contoso.com	Windows 8.1 administration workstation in the forest root domain	192.168.10.5	192.168.10.1; 192.168.10.2

All user accounts in this lab use the password adrepl123!

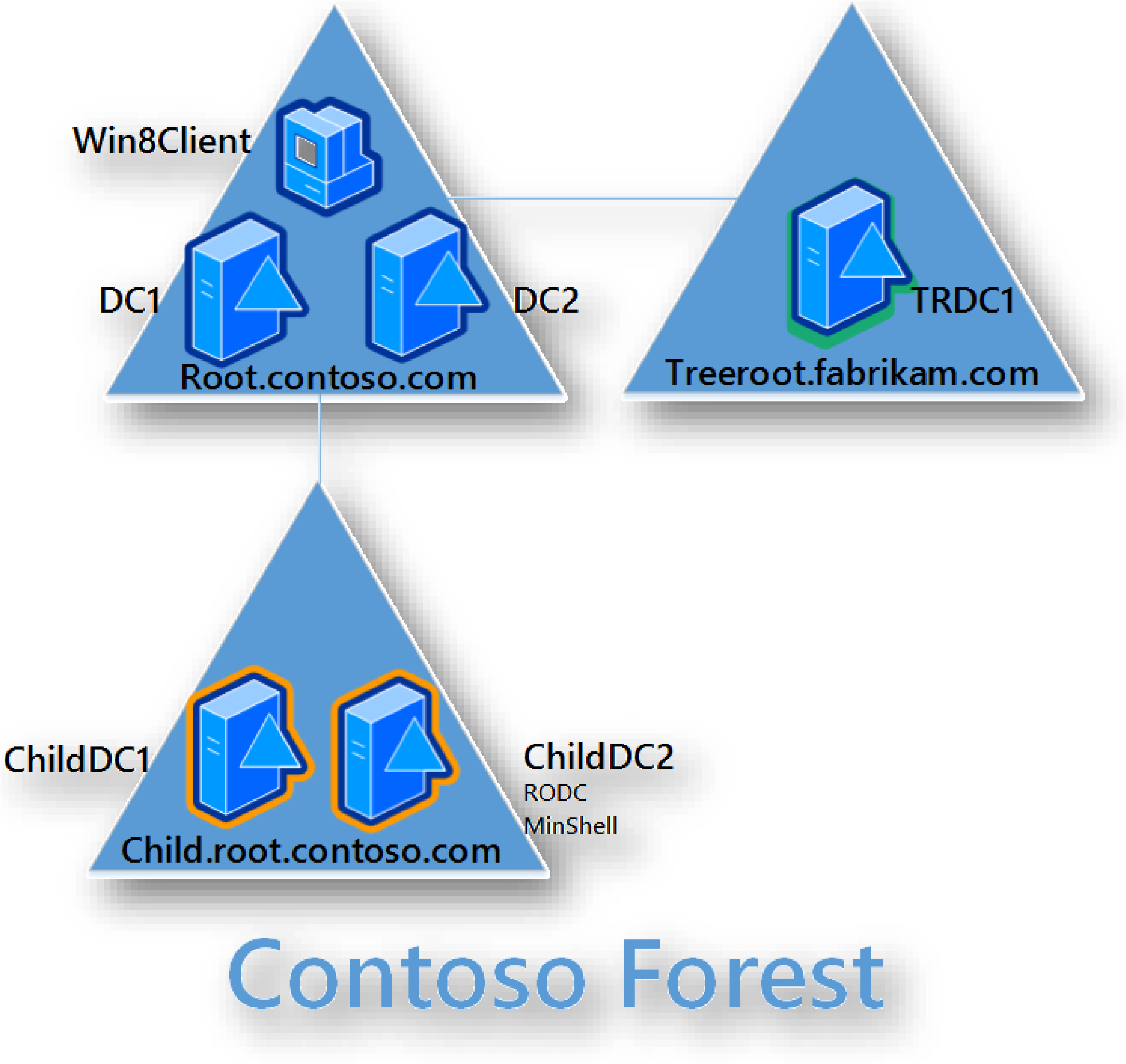


Figure 1 Lab environment

Exercise 1: Lingering Object Fundamentals

In this Exercise:

1. Lingering Object terminology
2. How to prevent a lingering object problem.
3. Understand the cause and identify the symptoms of Lingering Objects

In this exercise, you will review lingering object terminology, prevention methods and use ADREPLStatus, repadmin.exe and the Directory Service event log to identify symptoms of lingering objects.

More:

- Lingering object: An object that is present on one DC, but has been deleted and garbage collected on one or more DCs.
- AD replication error status **8606** is logged when the source DC sends an update of one or more attributes for an object that does not exist on the destination DC.
- Event 1988 is logged in the Directory Service event log when strict replication consistency is enabled
- Event 1388 is logged in the Directory Service event log when loose replication consistency is enabled. An AD replication error status is not logged for loose replication consistency since lingering objects are reanimated.

Task 0 - Lingering object terminology

Refer to Table 1 Lingering as needed for a description of the various terms mentioned throughout the lab document.

Tip: This section is jargon intense, a **Lingering Object Glossary** is provided for your reference.

Lingering object terminology

Table 1 Lingering Object Glossary

Term	Description	Notes
Abandoned delete / Live lingering object	An object is deleted on one DC. The deletion is never replicated to other DCs hosting a writable copy of the NC for that object. The deletion replicates to DCs/GCs hosting a read-only copy of the NC. The DC that originated the object deletion goes offline prior to replicating the change to other DCs hosting a writable copy of the partition.	Symptoms: GCs report source DCs have lingering objects in source DC partition: Root.contoso.com: DC1 and DC2 Child.root.contoso.com: ChildDC1 ChildDC1 replicates Root partition from DC1 and replication fails with error 8606
Abandoned object	An object created on one DC that never got replicated to other DCs hosting a writable copy of the NC but does get replicated to DCs/GCs hosting a read-only copy of the NC. The	Discovery of this object type is challenging.

	originating DC goes offline prior to replicating the originating write to other DCs that contain a writable copy of the partition.	1. Look at all objects in partition (or to make it not so complicated – just pick a single object) 2. Look at USN in object's replmetadata for originating create 3. Look at UpToDatenessVector in /showutdvec output for object partition on all R/W DCs for Originating DSA GUID reported in #2 4. Alert on object where #2 is higher than #3
Lingering link	A linked attribute contains the DN of an object that no longer exists in Active Directory. These stale references are referred to as lingering links.	
Lingering Object	An object that is present on one replica, but has been deleted and garbage collected on another replica.	

Loose Replication Consistency	With this behavior enabled, if a destination DC receives a change to an attribute for an object that it does not have, the entire object is replicated to the target for the sake of replication consistency. This undesirable behavior causes a lingering object to be "reanimated."	Warning: This setting will cause the undesirable behavior of reanimation of lingering objects.
		Event 1388 is logged in the DS event log of the destination DC when a source DC replicates changes for a lingering object
		For all domain controllers, type:
		`repadmin /regkey * -strict`
		For all global catalog servers, type:
		`repadmin /regkey gc: -strict`

Strict Replication Consistency	With this behavior enabled, if a destination DC receives a change to an attribute for an object that it does not have, replication is blocked with the source DC for the partition where the lingering object was detected. Event 1988 is logged in the Director Services event log on the destination DC and AD replication error status 8606 is logged for the last replication failure status message (visible in repadmin /showrepl output).	• Defines how a destination DC behaves if a source DC sends updates to an object that does not exist in the destination DC's local copy of Active Directory. • Destination DCs should see USN for creates before object is modified • Only modifies for lingering objects arrive for object not on destination DC • Only destination DC's enforce strict replication and log events • Destination DCs stop replicating from source DC's partitions containing LO's • Lingering objects are quarantined on source DCs where they can be detected

		• End-to-end replication may be impacted for partitions containing lingering objects
		• Administrators must remove lingering objects to restore replication
		• For all domain controllers, type:
		`repadmin /regkey * +strict`
		• For all global catalog servers, type:
		`repadmin /regkey gc: +strict`

Tombstone	An object that has been deleted but not yet garbage collected This object is retained in the database for the tombstone lifetime so that other DCs have an opportunity to learn of the object's deletion	
Tombstone Lifetime (TSL)	The amount of time tombstones are retained in Active Directory before being garbage collected and permanently purged from the database.	

Deleted object	When AD recycle bin is enabled, an object that is deleted (deleted object) is recoverable with a full set of attributes using a PowerShell command (2008 R2) or via PowerShell and a GUI- based tool (ADAC) in Windows Server 2012). The object remains in this state until the **deleted object lifetime** expires and then it becomes a **recycled object**.	IsDeleted = True
		IsRecycled = <not set>
		Stored in the **Deleted Objects** container in most instances (some objects do not get moved on deletion).

Deleted object lifetime	The deleted object lifetime is determined by the value of the **msDS-deletedObjectLifetime** attribute. • By default, tombstoneLifetime is set to null. When tombstoneLifetime is set to null, the tombstone lifetime defaults to 60 days (hard-coded in the system). • By default, msDS-deletedObjectLifetime is also set to null. When msDS-deletedObjectLifetime is set to null, the deleted object lifetime is set to the value of the tombstone lifetime. If msDS-deletedObjectLifetime is manually set, it becomes the effective lifetime of a system state backup.	CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=<mydomain>,DC=<com> Attribute: **msDS-deletedObjectLifetime**

Garbage Collection	A process that permanently deletes tombstone objects or recycled objects • runs on DCs every 12 hours by default / 15 minutes after restart Can be manually initiated with LDP, LDIFDE or other LDAP tool	`Repadmin /setattr "" "" doGarbageCollection add 1"`

Recycled object	After a deleted object lifetime expires, the logically deleted object is turned into a recycled object and most of its attributes are stripped away.	IsDeleted = True
		IsRecycled = True
		Can only be recovered if *toggle recycled objects* flag is used during the authoritative restore process.

Tombstone	Generically, this is an object that has been deleted but not garbage collected. Prior to the introduction of the AD recycle bin, this is the term for a deleted object.
	If AD recycle bin is enabled:
	An object that is deleted retains all of its attribute values and does not become a recycled object until the **deleted object lifetime** expires.
	If AD recycle bin is not enabled:
	A deleted object immediately becomes a tombstone and is stripped of most attribute values.
	To recover a tombstone with a full set of attributes, you must perform an authoritative restore.
		If AD recycle bin is not enabled:
		IsDeleted = True
		IsRecycled = True
		If AD recycle bin is enabled and the object is within the deleted object lifetime:
		IsDeleted=True
		IsRecycled=not set
		If AD recycle bin is enabled and the object is now a recycled object:
		IsDeleted=True
		IsRecycled=True

Tombstone Lifetime (TSL)	The number of days before tombstones or recycled objects are eligible for garbage collection. By default, tombstoneLifetime is set to null. When tombstoneLifetime is set to null, the tombstone lifetime defaults to 60 days (hard-coded in the system). This is also the effective lifetime of a system state backup. If msDS-deletedObjectLifetime is manually set, it becomes the effective lifetime of a system state backup.	CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=<mydomain>,DC=<com> Attribute: tombstoneLifetime

How to prevent a lingering object problem:

The root cause of most lingering object problems are long term AD replication failures that have been allowed to persist beyond the tombstone lifetime number of days. The best way to avoid and prevent lingering object issues:

1. Proactively monitor AD replication with a tool like ADReplStatus.
2. Correct AD replication problems within the tombstone lifetime number of days
3. Prevent large jumps in system time from occurring on domain controllers

⚠ **Important:**

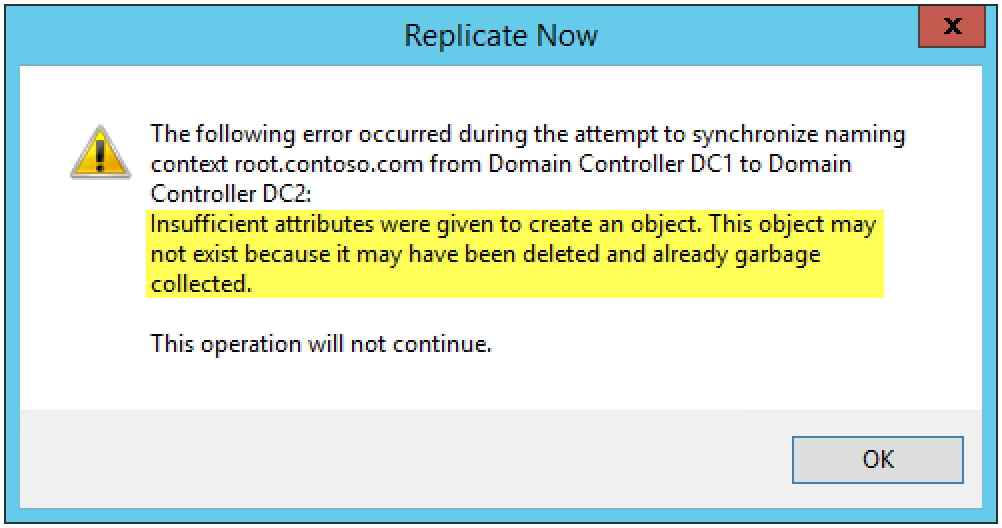
- Resolve replication failures within TSL # of days
- Ensure Strict Replication Consistency is enabled
- Ensure large jumps in system time are blocked via registry key or policy
- Don't remove replication quarantine with the "allowDivergent" setting without removing LOs first
- Don't restore system backups that are near TSL number of days old
- Don't bring DCs back online that haven't replicated within TSL
- Do not allow a server to replicate that has experienced a USN rollback
- Ensure originating changes are replicated out to other DCs in the same domain before forcefully demoting a DC or restoring a VM checkpoint of a Windows Server 2012 DC VM guest

Task 1 - Lingering object symptoms and identification

AD replication status 8606 and event ID 1988 are good indicators of lingering objects (when the DCs are configured for Strict Replication Consistency). It is important to note, however, that AD replication may complete successfully (and not log an error) from a DC containing lingering objects since replication is based on changes. If there are no changes to any of the lingering objects, there is no reason to replicate them and they will continue to exist without logging any noticeable errors. For this reason, when cleaning up lingering objects, do not just clean up the DCs logging the errors; instead, assume that all DCs may contain them, and clean them up as well.

Scenario

- AD replication of the Root partition from **DC1** to **DC2** fails with error, "Insufficient attributes were given to create an object".



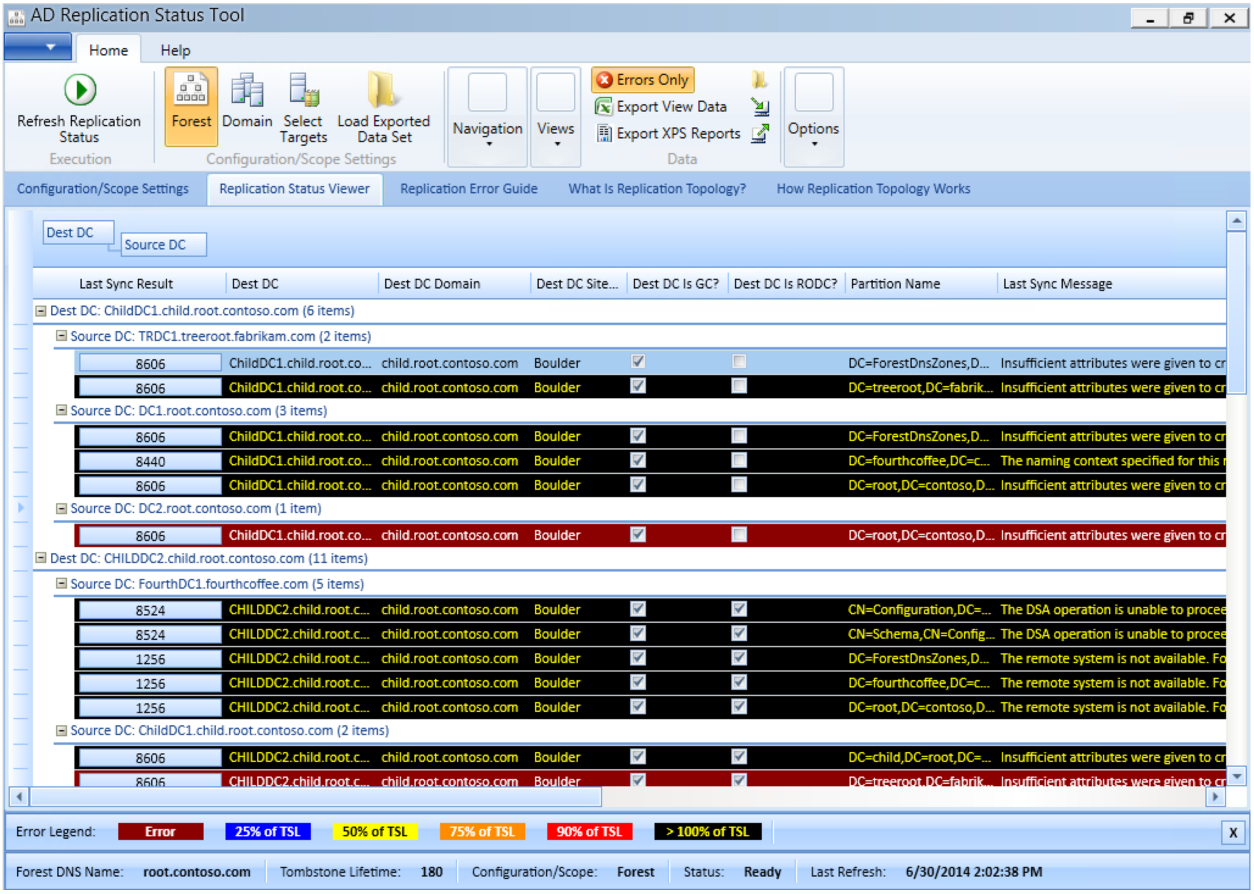
The following error occurred during the attempt to synchronize naming context root.contoso.com from Domain Controller DC1 to Domain Controller DC2:
Insufficient attributes were given to create an object. This object may not exist because it may have been deleted and already garbage collected.

This operation will not continue.

- All DCs have lingering objects in almost all partitions
- DC2 reports error 8606 replicating from DC1

A. Use the AD Replication Status Tool to get forest-wide AD replication status

1. Connect to **Win8Client**.
    - The ROOT\Administrator account is already logged on to this machine.
    - Note: Domain admin privileges are not needed for this task, but these privileges are required in later exercises.
2. On **Win8Client**, double click the **AD Replication Status Tool 1.0** shortcut on the desktop.
3. Within the AD Replication Status Tool, click **Refresh Replication Status**.
    - The tool will take one to two minutes to check the AD replication status.
    - You will know data collection is complete when the **Status:** prompt changes from **Running** to **Ready** and the focus is switched to the **Replication Status Viewer** tab.
4. Click the **Errors Only** menu option on the Data section of the ribbon to see a detailed view of all replication errors in the forest.


AD Replication Status Tool
Home
Help
Refresh Replication Status
Execution
Forest
Domain
Select Targets
Load Exported Data Set
Configuration/Scope Settings
Navigation
Views
Errors Only
Export View Data
Export XPS Reports
Data
Options
Configuration/Scope Settings
Replication Status Viewer
Replication Error Guide
What Is Replication Topology?
How Replication Topology Works
Dest DC
Source DC

Last Sync Result	Dest DC	Dest DC Domain	Dest DC Site...	Dest DC Is GC?	Dest DC Is RODC?	Partition Name	Last Sync Message
⊟ Dest DC: ChildDC1.child.root.contoso.com (6 items)							
⊟ Source DC: TRDC1.treeroot.fabrikam.com (2 items)							
8606	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=ForestDnsZones,D...	Insufficient attributes were given to cr
8606	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=treeroot,DC=fabrik...	Insufficient attributes were given to cr
⊟ Source DC: DC1.root.contoso.com (3 items)							
8606	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=ForestDnsZones,D...	Insufficient attributes were given to cr
8440	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=fourthcoffee,DC=c...	The naming context specified for this r
8606	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=root,DC=contoso,D...	Insufficient attributes were given to cr
⊟ Source DC: DC2.root.contoso.com (1 item)							
8606	ChildDC1.child.root.co...	child.root.contoso.com	Boulder	☑	□	DC=root,DC=contoso,D...	Insufficient attributes were given to cr
⊟ Dest DC: CHILDDC2.child.root.contoso.com (11 items)							
⊟ Source DC: FourthDC1.fourthcoffee.com (5 items)							
8524	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	CN=Configuration,DC=...	The DSA operation is unable to procee
8524	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	CN=Schema,CN=Config...	The DSA operation is unable to procee
1256	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	DC=ForestDnsZones,D...	The remote system is not available. Fo
1256	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	DC=fourthcoffee,DC=c...	The remote system is not available. Fo
1256	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	DC=root,DC=contoso,D...	The remote system is not available. Fo
⊟ Source DC: ChildDC1.child.root.contoso.com (2 items)							
8606	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	DC=child,DC=root,DC=...	Insufficient attributes were given to cr
8606	CHILDDC2.child.root.c...	child.root.contoso.com	Boulder	☑	☑	DC=treeroot,DC=fabrik...	Insufficient attributes were given to cr

Error Legend: Error | 25% of TSL | 50% of TSL | 75% of TSL | 90% of TSL | > 100% of TSL

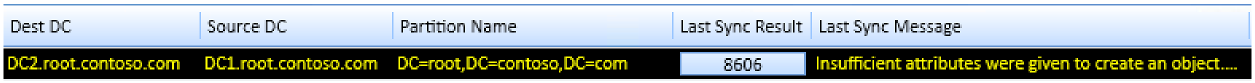
Forest DNS Name: root.contoso.com | Tombstone Lifetime: 180 | Configuration/Scope: Forest | Status: Ready | Last Refresh: 6/30/2014 2:02:38 PM

Figure 2 Replication Status Viewer pane

- The Replication Status Viewer is highly customizable.
  - Drag different columns to the top for different pivot options.
  - Add and remove columns of interest.

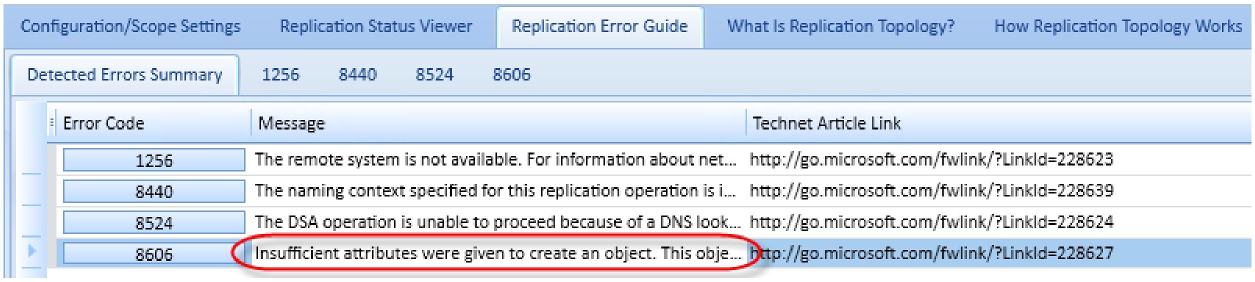
Later on, we will be investigating this one failure:

DC2 is failing to replicate the **root** partition from **DC1**.



Dest DC	Source DC	Partition Name	Last Sync Result	Last Sync Message
DC2.root.contoso.com	DC1.root.contoso.com	DC=root,DC=contoso,DC=com	8606	Insufficient attributes were given to create an object....

5. Click the **Replication Error Guide** tab for a quick summary view of all errors.

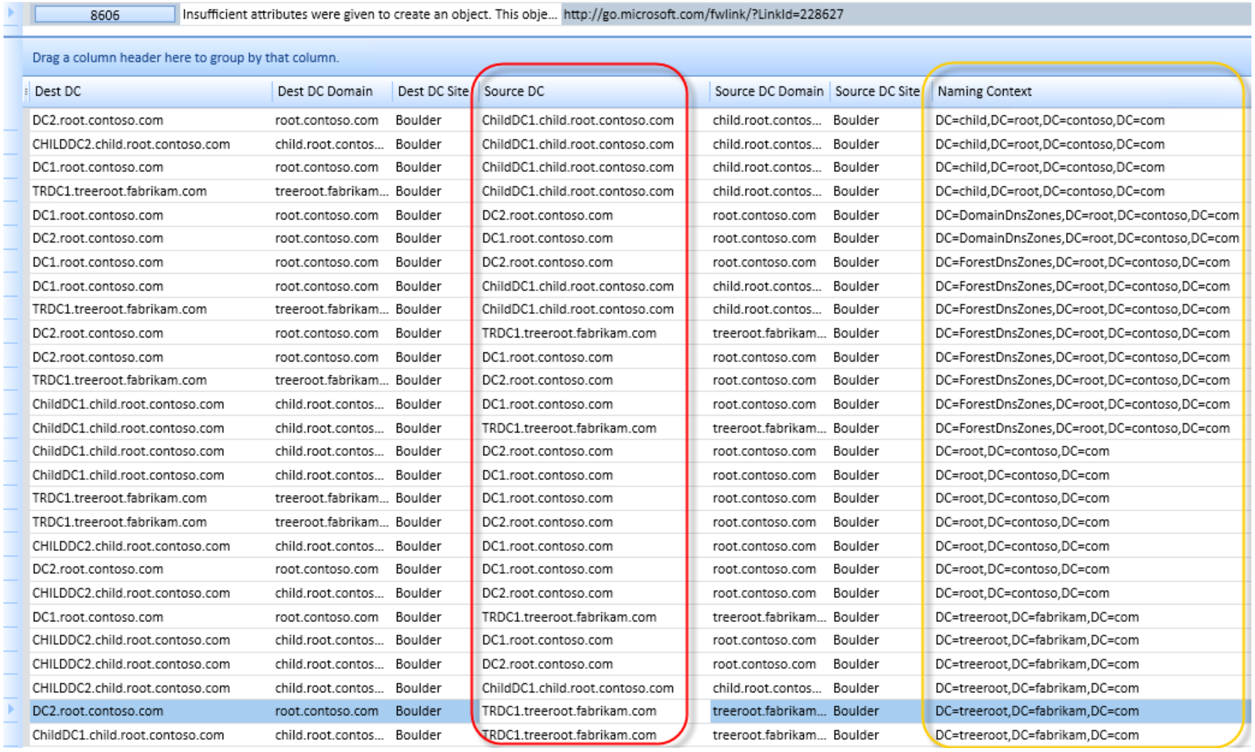

Configuration/Scope Settings
Replication Status Viewer
Replication Error Guide
What Is Replication Topology?
How Replication Topology Works
Detected Errors Summary
1256
8440
8524
8606

Error Code	Message	Technet Article Link
1256	The remote system is not available. For information about net...	http://go.microsoft.com/fwlink/?LinkId=228623
8440	The naming context specified for this replication operation is i...	http://go.microsoft.com/fwlink/?LinkId=228639
8524	The DSA operation is unable to proceed because of a DNS look...	http://go.microsoft.com/fwlink/?LinkId=228624
8606	Insufficient attributes were given to create an object. This obje...	http://go.microsoft.com/fwlink/?LinkId=228627

Figure 3 Replication Error Guide pane

6. Select the message text, "**Insufficient attributes were given to create an object…**" to see a sortable list of all DCs with this replication error.

Tip: The DCs listed in the **Source DC** column have at least one lingering object for the partition in the **Naming Context** column.



8606	Insufficient attributes were given to create an object. This obje...	http://go.microsoft.com/fwlink/?LinkId=228627

Dest DC	Dest DC Domain	Dest DC Site	Source DC	Source DC Domain	Source DC Site	Naming Context
DC2.root.contoso.com	root.contoso.com	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=DomainDnsZones,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=DomainDnsZones,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=ForestDnsZones,DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=treeroot,DC=fabrikam,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=treeroot,DC=fabrikam,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=treeroot,DC=fabrikam,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com

Figure 4 Replication Error Guide pane with focus on Error 8606 details

If you click on error **8606** in the **Error Code** column, our latest troubleshooting content for that issue loads up in the tool.

B. Lingering object symptoms on an individual DC

Perform this task on **DC2**.

Tip:

For ease of command entry: There is a file on **Win8Client** in the D:\files directory, called **fix_lab.txt** that contains all necessary commands needed for this lab. There is a mixture of both CMD-line and PowerShell commands in the file. To execute the commands:

1. Open an elevated PowerShell prompt on **Win8Client**.
2. Copy the commands for the step you are working on, and paste them into the PowerShell window.
3. It is best to copy the **Files** directory to the root of the C:\ drive before executing any commands. Some commands attempt to output files to the current working directory (which will fail for D:\Files because it is a read-only ISO file attached to the VM guest.

Alternately, you can copy them from the lab manual.

1. Initiate replication between **DC1** and **DC2** (have DC1 pull from DC2)

```
Repadmin /replicate dc1 dc2 "dc=root,dc=contoso,dc=com"
```

   Replication completes successfully:

```
Sync from dc2 to dc1 completed successfully
```

2. Check replication the other way (have **DC2** pull from **DC1**)

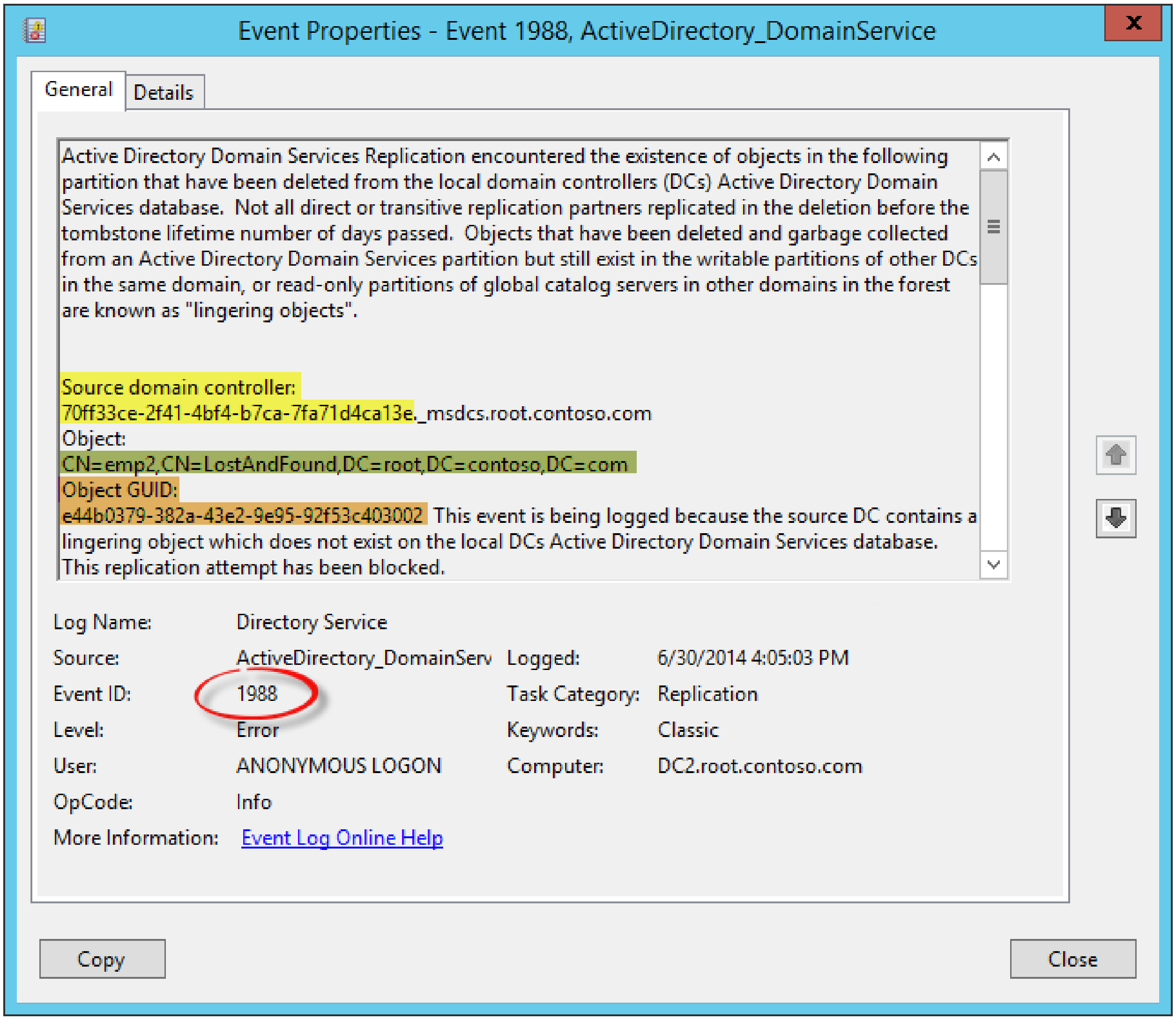
```
Repadmin /replicate dc2 dc1 "dc=root,dc=contoso,dc=com"
```

   Replication fails with the following error:

```
DsReplicaSync() failed with status 8606 (0x219e):
Insufficient attributes were given to create an object. This object may not exist because it may have been deleted and already garbage collected.
```

   Event 1988 is logged in the Directory Service event log on **DC2**.

3. Review the Directory Services event log on **DC2** for event 1988 using event viewer (eventvwr.msc) or PowerShell


Event Properties - Event 1988, ActiveDirectory_DomainService
General
Details
Copy
Close

Active Directory Domain Services Replication encountered the existence of objects in the following partition that have been deleted from the local domain controllers (DCs) Active Directory Domain Services database. Not all direct or transitive replication partners replicated in the deletion before the tombstone lifetime number of days passed. Objects that have been deleted and garbage collected from an Active Directory Domain Services partition but still exist in the writable partitions of other DCs in the same domain, or read-only partitions of global catalog servers in other domains in the forest are known as "lingering objects".

Source domain controller:
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com
Object:
CN=emp2,CN=LostAndFound,DC=root,DC=contoso,DC=com
Object GUID:
e44b0379-382a-43e2-9e95-92f53c403002 This event is being logged because the source DC contains a lingering object which does not exist on the local DCs Active Directory Domain Services database.
This replication attempt has been blocked.

Log Name: Directory Service
Source: ActiveDirectory_DomainServ Logged: 6/30/2014 4:05:03 PM
Event ID: 1988 Task Category: Replication
Level: Error Keywords: Classic
User: ANONYMOUS LOGON Computer: DC2.root.contoso.com
OpCode: Info
More Information: Event Log Online Help

Figure 5 Event 1988 - take note of Source DC, Object name and object GUID

```
Get-WinEvent -LogName "Directory Service" -MaxEvents 5 | fl
```

To just return the last 10 event ID 1988s from DC2's Directory Service event log:

```
get-winevent -LogName "Directory Service" -ComputerName dc2 -MaxEvents 10 | Where-Object {$_.ID -eq
"1988"} | fl
```

Note:

Event 1988 only reports the first lingering object encountered during the replication attempt. There are usually many more lingering objects present on the source DC. Use repadmin /removelingeringobjects with the **/advisory_mode** switch to have all lingering objects reported for that partition.

4. Identify the following from event 1988 (they are needed later in the exercise):

- Object GUID: **e44b0379-382a-43e2-9e95-92f53c403002**
- Source DC: **DC1.root.contoso.com**
- Partition DN: **DC=root,DC=contoso,DC=com**

How can you translate the DNS alias provided in the event to the host name of the source DC?

See the answer in this tasks section in the appendix.

Is DC2 configured for Strict or Loose Replication Consistency?

What event is logged on the destination DC when there is an attempt to send changes for a lingering object when strict replication consistency is enabled?

What event is logged on the destination DC when there is an attempt to send changes for a lingering object when loose replication consistency is enabled?

Task 2 - Lingering object analysis

In this task, you will use repadmin to return replication metadata for the lingering object identified in event ID 1988. The repadmin output will allow you to identify DCs containing the lingering object reported in the event.

Perform this task **DC2** and **DC1**.

1. Obtain the ObjectGUID reported in the event on **DC2**. (see Figure 5 for location of ObjectGUID)
2. Identify all DCs that have a copy of this object using repadmin /showobjmeta

```
Repadmin /showobjmeta * "<GUID=e44b0379-382a-43e2-9e95-92f53c403002>" >emp2.txt
```

3. Open **emp2.txt**. Any DC that returns replication metadata for this object are DCs containing one or more lingering objects. DCs that do not have a copy of the object report status 8439, "The distinguished name specified for this replication operation is invalid".

Which DCs return replication metadata for the object?

See the ***Answers*** section in the Appendix if needed.

Important: This is a good method to conduct a quick spot check of DCs containing the lingering object reported in the event. It is NOT a good method to discover all lingering objects. For more information, see the **Lingering Object discovery** section of the appendix.

Is the EMP2 user account the only lingering object present on DC1?
It is likely there are many more. We will use repadmin in the next step to check for more objects in the Root partition on DC1.

4. Obtain DC2's DSA ObjectGUID and use repadmin /removelingeringobjects with the /advisory_mode parameter to identify all lingering objects in the **ROOT** partition on **DC1**.

Note:

In order to use the /removelingeringobjects command you need to know three things:

1. You need to know which DCs contain lingering objects
2. Which partition the lingering object resides in
3. The DSA Object GUID of a good reference DC that hosts that partition that does not contain lingering objects

a. Since DC2 is the only other DC in the **ROOT** partition, we will have to use it as the reference DC. Obtain the DSA object GUID on DC2:

```
Repadmin /showrepl DC2 >DC2_showrepl.txt
```

The DSA object GUID is at the top of the output and will look like this:

```
DSA object GUID: 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
```

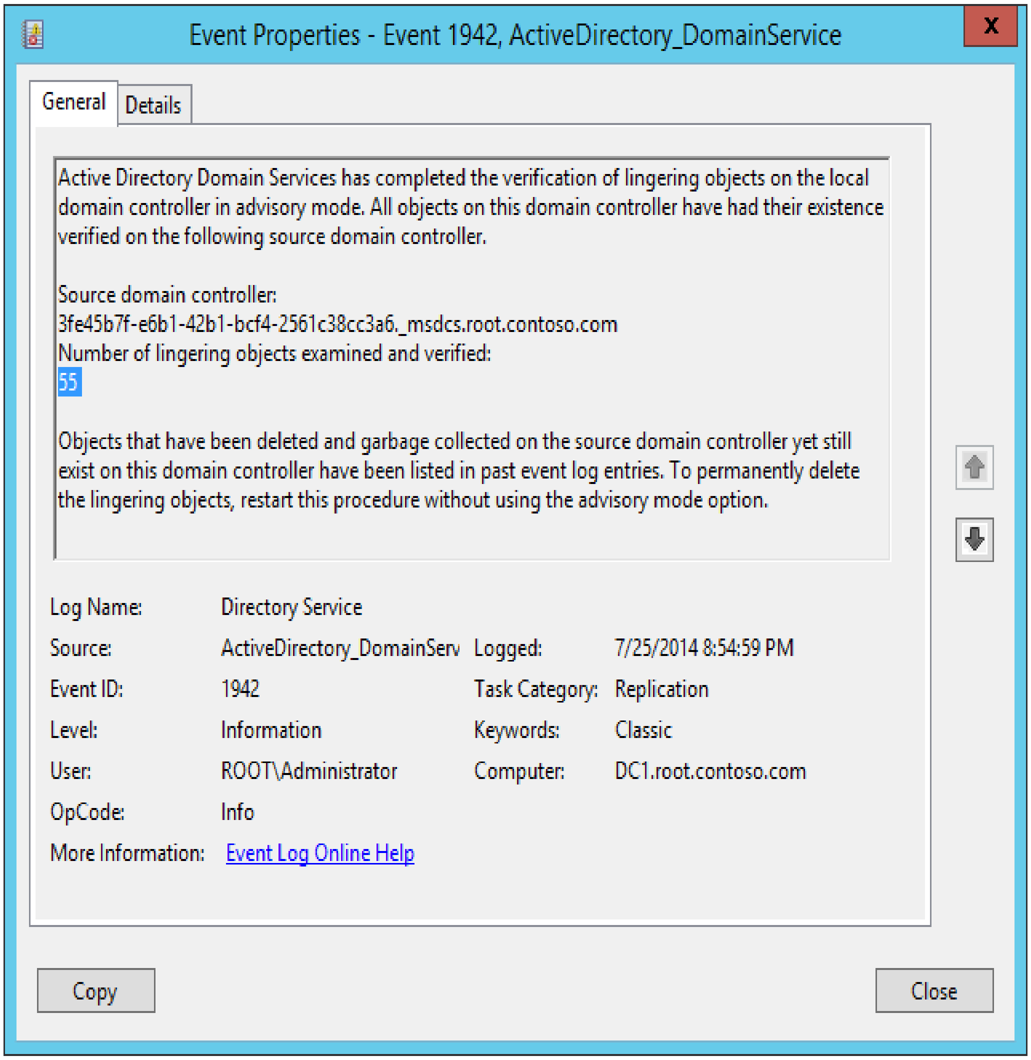
b. In the following command, you will verify the existence of lingering objects on DC1 by comparing its copy of the ROOT partition with DC2.

Run the following repadmin command (ensure you use the /advisory_mode parameter)

```
Repadmin /removelingeringobjects DC1 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
"dc=root,dc=contoso,dc=com" /Advisory_Mode
```

```
RemoveLingeringObjects successful on dc1.
```

c. Connect to **DC1**. Review the Directory Service event log on **DC1**. If there are any lingering objects present, each one will be reported in its own event ID 1946. The total count of lingering objects for the partition is reported in event 1942.



Event Properties - Event 1942, ActiveDirectory_DomainService

Active Directory Domain Services has completed the verification of lingering objects on the local domain controller in advisory mode. All objects on this domain controller have had their existence verified on the following source domain controller.

Source domain controller:
3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6._msdcs.root.contoso.com
Number of lingering objects examined and verified:
55

Objects that have been deleted and garbage collected on the source domain controller yet still exist on this domain controller have been listed in past event log entries. To permanently delete the lingering objects, restart this procedure without using the advisory mode option.

Log Name: Directory Service
Source: ActiveDirectory_DomainServ
Logged: 7/25/2014 8:54:59 PM
Event ID: 1942
Task Category: Replication
Level: Information
Keywords: Classic
User: ROOT\Administrator
Computer: DC1.root.contoso.com
OpCode: Info
More Information: Event Log Online Help

Figure 6 Event 1942 indicating the presence of 55 lingering objects

```
get-winevent -LogName "Directory Service" -ComputerName dc1 -MaxEvents 10 | Where-Object {$_.ID -eq
"1942"} | fl >DC1_DSevents1942.txt
```

We compared **DC1** against **DC2** for the **root** partition. How do we know that DC2 is clean? Earlier we saw that AD replication completes successfully from DC2. As mentioned in the exercise introduction, you cannot assume a DC is clean just because replication completes without error; we will now check **DC2** against **DC1**.

d. Use repadmin to check for the existence of lingering objects in the **root** partition on **DC2**

- Obtain DSA object GUID from the only other DC in the domain DC1

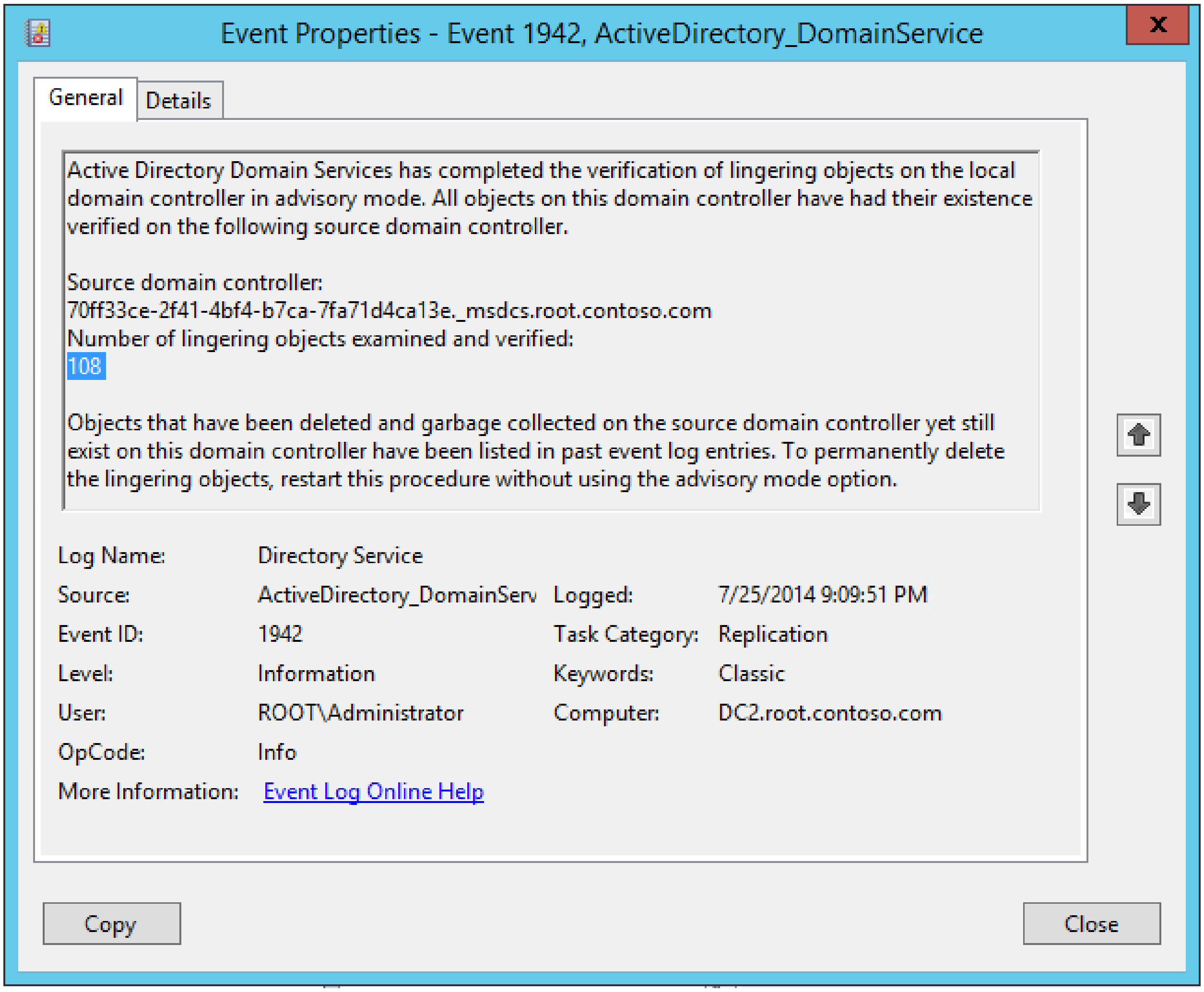
```
Repadmin /showrepl DC1
```

```
DSA object GUID: 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
```

- Use repadmin /removelingeringobjects with the **/Advisory_Mode** parameter

```
repadmin /removelingeringobjects dc2 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
dc=root,dc=contoso,dc=com /Advisory_Mode
```

On **DC2**, review event ID 1942 logged in the Directory Service event log


Event Properties - Event 1942, ActiveDirectory_DomainService
General
Details
Copy
Close

Active Directory Domain Services has completed the verification of lingering objects on the local domain controller in advisory mode. All objects on this domain controller have had their existence verified on the following source domain controller.

Source domain controller:
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com
Number of lingering objects examined and verified:
108

Objects that have been deleted and garbage collected on the source domain controller yet still exist on this domain controller have been listed in past event log entries. To permanently delete the lingering objects, restart this procedure without using the advisory mode option.

Log Name: Directory Service
Source: ActiveDirectory_DomainServ
Logged: 7/25/2014 9:09:51 PM
Event ID: 1942
Task Category: Replication
Level: Information
Keywords: Classic
User: ROOT\Administrator
Computer: DC2.root.contoso.com
OpCode: Info
More Information: Event Log Online Help

Figure 7 Event 1942 indicating 108 lingering objects on DC2

Event 1942 indicates that DC2 also contains lingering objects in the **root** partition. This is notable because we saw no indication of a lingering object problem for this partition from the AD replication status report.

PowerShell method to view forest-wide replication results (Optional steps)

Perform this step on **DC2** if desired.

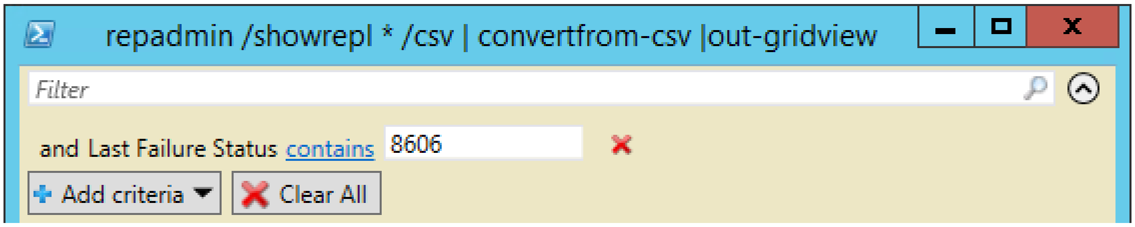
1. Open a PowerShell prompt and type the following commands, and then press ENTER:

PowerShell:

```
Repadmin /showrepl * /csv | convertfrom-csv | out-gridview
```

   a. Select **Add criteria** and check **Last Failure Status**. Select **Add**.

b. Type **8606** in the text box.



repadmin /showrepl * /csv | convertfrom-csv |out-gridview

and Last Failure Status contains 8606

Exercise Review

We reviewed lingering object fundamentals: core concepts and terminology

Lingering object symptoms:

a. For strict replication consistency:

- AD replication status 8606 and event 1988

b. For loose replication consistency:

- Event 1388

In this exercise:

1. We began by getting a forest-wide AD replication status report. In the report, we found that replication was failing on all DCs in almost all partitions with error 8606, "Insufficient attributes were given to create an object…"
2. We then went to one DC and found a single lingering object reported in event 1988. We dug into the details of the event and identified all DCs with the lingering object.
3. We then used repadmin to discover that there were actually many more lingering objects than just the one reported.
4. Finally, we checked for lingering objects on the DC that was not displaying any symptoms, and discovered that it actually had more lingering objects than the DC with the symptoms.

Exercise 2: Lingering Object Diagnosis and Documentation

In this Exercise:

Use several tools to identify the full scope of a lingering object problem.

Documenting all lingering objects has traditionally been a challenging problem. The new Lingering Object tool makes this a simple task, as you will discover in this exercise.

Lingering Object discovery

- Introducing the Lingering Object Liquidator tool.
- Repldiag
- Replfix

AD replication status 8606 and event ID 1988 are good indicators of lingering objects (when the DCs are configured for Strict Replication Consistency).

As you saw in the prior lesson, however, lingering objects can be present on a DC without any noticeable symptoms. AD replication is based on change notifications; if there are no changes to an object that is lingering, it is not replicated, and therefore there are no symptoms of the condition. For this reason, when cleaning up lingering objects, do not just clean up the DCs logging the errors; instead, assume that all DCs may contain them, and clean them up as well.

Important:

When lingering objects are discovered, assume they are present on all DCs in all partitions. Do not just clean up the DCs reporting the errors. Repldiag automates the majority of the cleanup work. See the **Lingering Object discovery and cleanup** section for more information.

Lingering Object discovery and cleanup

Repadmin /removelingeringobjects /advisory_mode is a good method to conduct a spot check of lingering objects on an individual DC, per partition basis.

However, lingering objects may exist on DCs without any noticeable symptoms. For that reason, checking and cleaning up just the DCs that report errors is not a good method to ensure all lingering objects are removed from the environment.

To remove lingering objects

1. Determine the root cause of the lingering object issue and prevent it from occurring again
2. Assume all DCs contain lingering objects in all partitions and clean up everyone

Those that clean up just the source DCs reported with AD replication status 8606 usually find they have more objects to clean up later.

To accomplish the above using repadmin, you need to do the following:

1. Identify one DC per partition to use as a reference DC
2. Clean up each DC identified in step 1 against all other DCs that host a writeable copy of the same partition. This DC is now considered "clean" and suitable to use as a reference DC.
3. Clean up all other DCs against the reference DCs

In the simple, five DC, three-domain lab environment, this requires 30 separate executions of the repadmin command. In a real-word production environment, the count of repadmin executions is usually in the hundreds or thousands.

ⓘ **More:**

For more information, see:

Clean that Active Directory Forest of Lingering Objects
http://blogs.technet.com/b/glennl/archive/2007/07/26/clean-that-active-directory-forest-of-lingering-objects.aspx

The good news is that Lingering Object Liquidator and repldiag /removelingeringobjects automates the above for you.

- Repldiag requires just one execution: **Repldiag /removelingeringobjects**
- With Lingering Object Liquidator, you just click the **RemoveAll** button

Blindly removing all objects is fine for most; however, some people like to know what is going to be removed from their Active Directory ---especially if the problem is widespread like it is in the lab environment. The domain controllers in this environment have not replicated with each other in over a year. It is usually wise to forcefully demote a DC that has not replicated in that length of time. However, this is not an option here since all DCs fall into this category. For that reason, we need to identify and then remove all objects before replication is enabled. It is generally a good idea to document what you are going to delete before you delete it.

Task 1 - Lingering Object Discovery

In this task, we will use several additional tools that help with lingering object discovery. Featured here is the new GUI-based Lingering Object Removal tool. We will explore a beta version of this tool. We will touch on repldiag and use another tool called Replfix to round out our lingering object discovery options.

Introducing Lingering Object Liquidator

More:

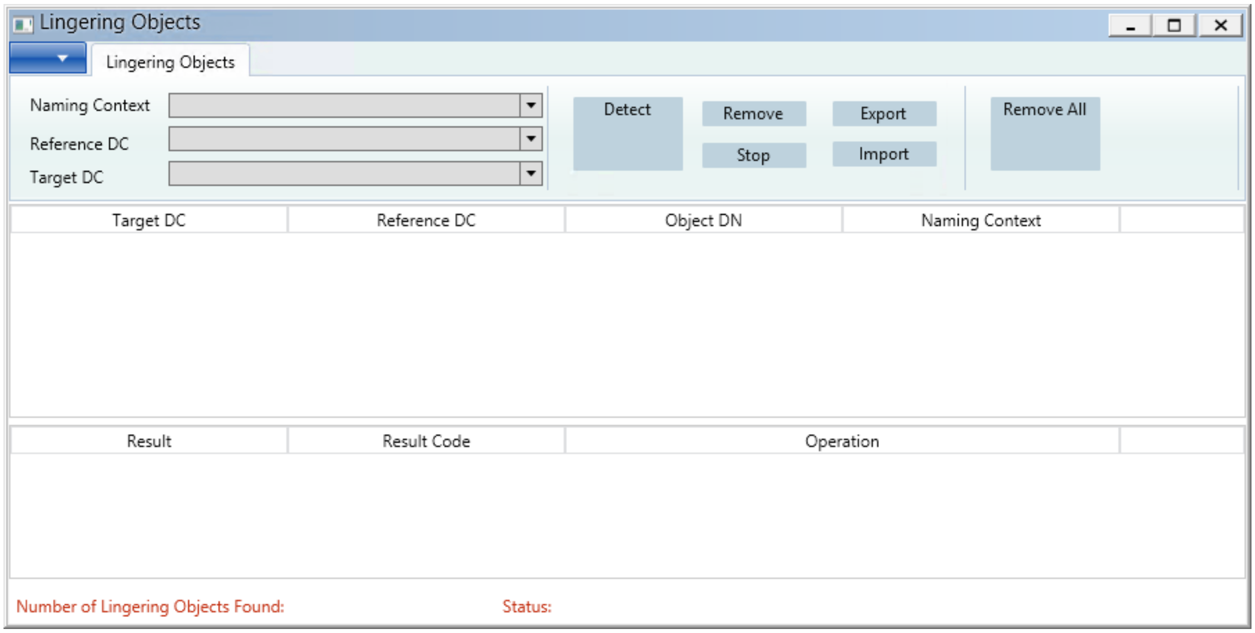
Lingering Object Liquidator automates the discovery and removal of lingering objects by using the ***DRSReplicaVerifyObjects*** method used by repadmin /removelingeringobjects and repldiag combined with the ***removeLingeringObject*** rootDSE primitive used by LDP.EXE. Tool features include:

- Combines both discovery and removal of lingering objects in one interface
- Is available via the Microsoft Connect site
    - See post on *blogs.technet.com/askds* for instructions
- The version of the tool at the Microsoft Connect site is an early beta build and does not have the fit and finish of a finished product
- Feature improvements beyond what you see in this version are under consideration

Connect to **DC1** for this task.

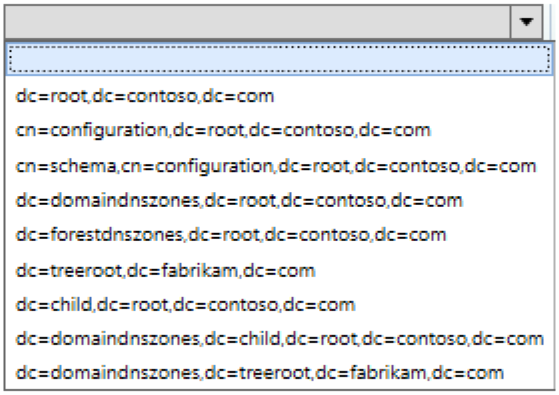
1. Launch **Lingering Objects.exe** from the shortcut on the desktop of **DC1**.

   If you get a **Windows protected your PC** SmartScreen prompt, click **More info** and then click **Run anyway**.



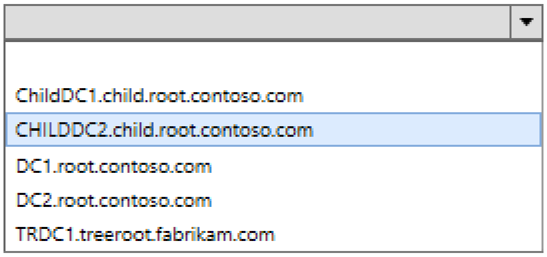
2. Explore the UI:

   Naming Context



dc=root,dc=contoso,dc=com
cn=configuration,dc=root,dc=contoso,dc=com
cn=schema,cn=configuration,dc=root,dc=contoso,dc=com
dc=domaindnszones,dc=root,dc=contoso,dc=com
dc=forestdnszones,dc=root,dc=contoso,dc=com
dc=treeroot,dc=fabrikam,dc=com
dc=child,dc=root,dc=contoso,dc=com
dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com
dc=domaindnszones,dc=treeroot,dc=fabrikam,dc=com

Reference DC: the DC you will compare to the target DC. The reference DC hosts a writeable copy of the partition



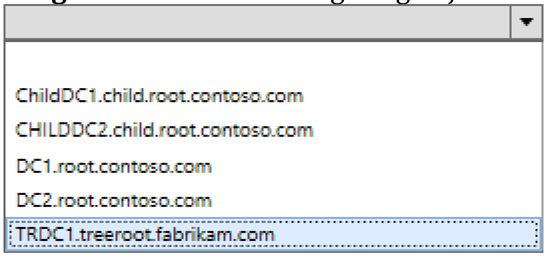
ChildDC1.child.root.contoso.com
CHILDDC2.child.root.contoso.com
DC1.root.contoso.com
DC2.root.contoso.com
TRDC1.treeroot.fabrikam.com

Note: ChildDC2 should not be listed here since it is an RODC.

More:

The version of the tool in this lab is still in development and does not represent the finished product. In other words, expect crashes, quirks and everything else normally encountered with beta software.

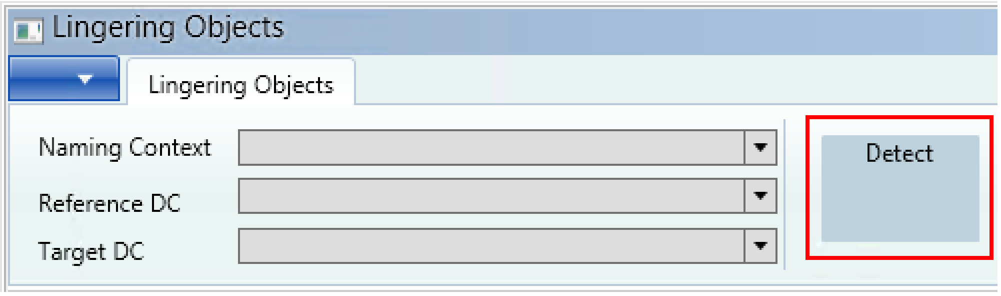
Target DC: the DC that lingering objects are to be removed from



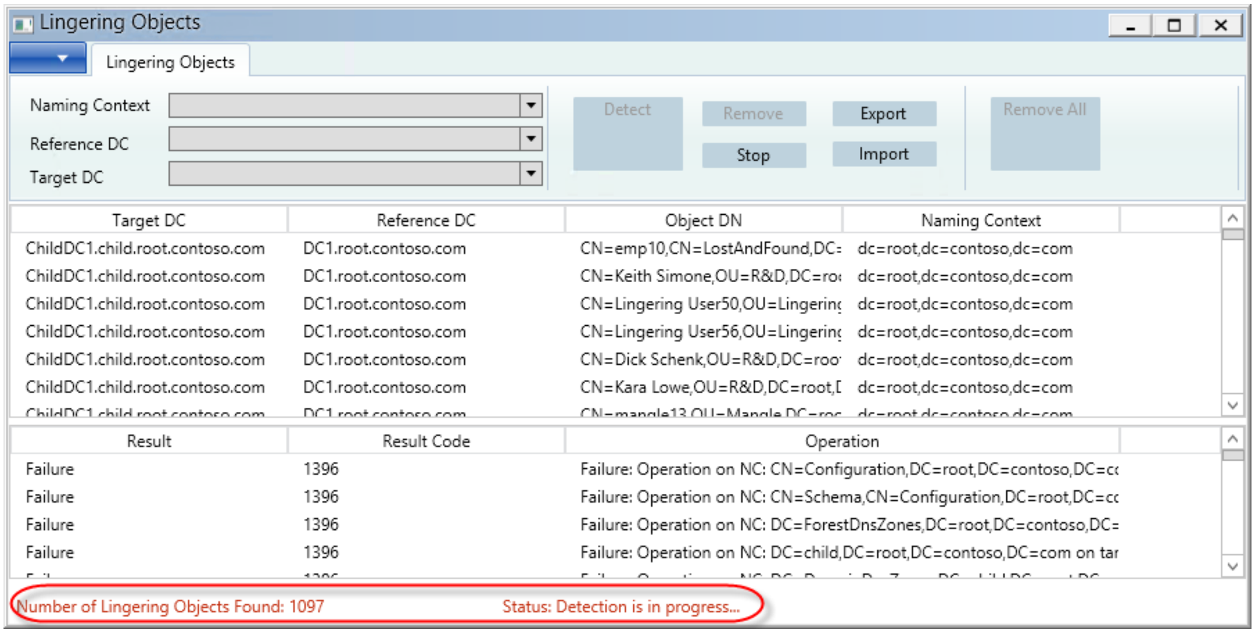
ChildDC1.child.root.contoso.com
CHILDDC2.child.root.contoso.com
DC1.root.contoso.com
DC2.root.contoso.com
TRDC1.treeroot.fabrikam.com

3. Leave all fields blank to have the entire environment scanned, and then click **Detect.**

   The tool does a comparison amongst all DCs for all partitions in a pairwise fashion when all fields are left blank. In a large environment, this comparison will take a great deal of time as the operation targets (n * (n-1)) number of DCs in the forest for all locally held partitions. For shorter, targeted operations, select a naming context, reference DC and target DC. The reference DC must hold a writable copy of the selected naming context.



During the scan, several buttons are disabled, and the current count of lingering objects is displayed in the status bar at the bottom of the screen along with the current tool status. During this execution phase, the tool is running in an advisory mode and reading the event log data reported on each target DC.



Target DC	Reference DC	Object DN	Naming Context
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=emp10,CN=LostAndFound,DC=	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Keith Simone,OU=R&D,DC=ro	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User50,OU=Lingerin	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User56,OU=Lingerin	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Dick Schenk,OU=R&D,DC=roo	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Kara Lowe,OU=R&D,DC=root,[	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=mangle13,OU=Mangle,DC=roc	dc=root,dc=contoso,dc=com

Result	Result Code	Operation
Failure	1396	Failure: Operation on NC: CN=Configuration,DC=root,DC=contoso,DC=c
Failure	1396	Failure: Operation on NC: CN=Schema,CN=Configuration,DC=root,DC=c
Failure	1396	Failure: Operation on NC: DC=ForestDnsZones,DC=root,DC=contoso,DC=
Failure	1396	Failure: Operation on NC: DC=child,DC=root,DC=contoso,DC=com on tar

Number of Lingering Objects Found: 1097 Status: Detection is in progress...

When the scan is complete, the status bar updates, buttons are re-enabled and total count of lingering objects is displayed. The log pane at the bottom of the window updates with any errors encountered during the scan.

Error 1396 is logged if the tool incorrectly uses an RODC as a reference DC.

Error 8440 is logged when the targeted reference DC doesn't host a writable copy of the partition.

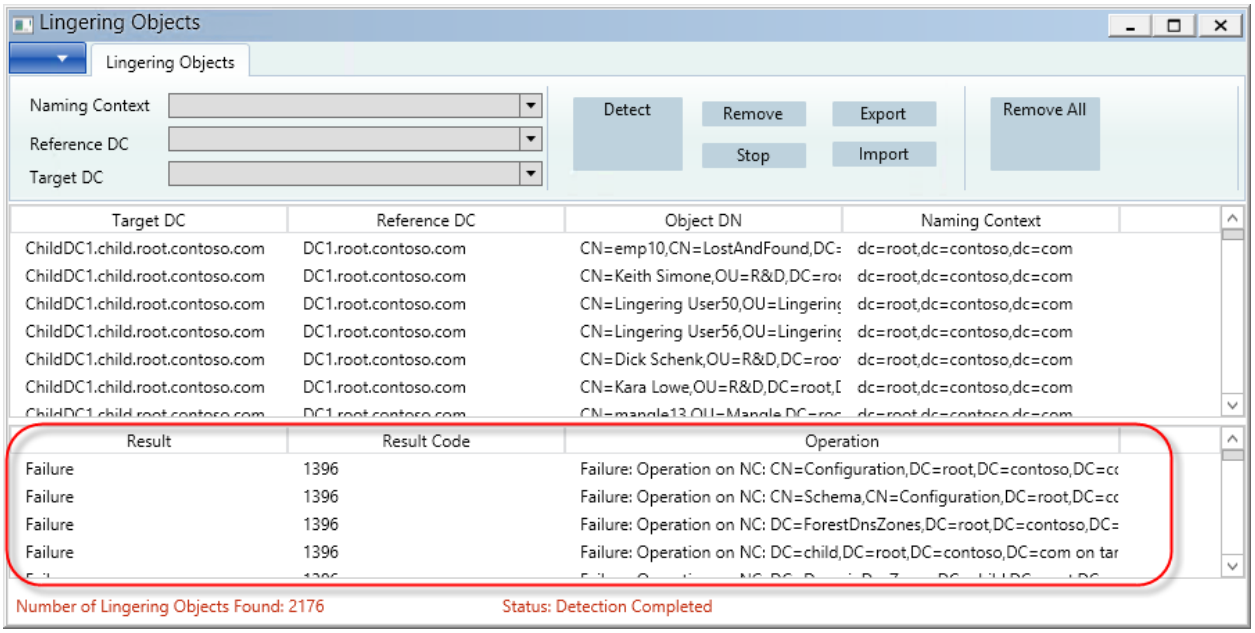
Note:

Lingering Object Liquidator discovery method

- Leverages **DRSReplicaVerifyObjects** method in Advisory Mode
- Runs for all DCs and all Partitions

- Collects lingering object event ID 1946s and displays objects in main content pane
- List can be exported to CSV for offline analysis (or modification for import)
- Supports import and removal of objects from CSV import (leverage for objects not discoverable using DRSReplicaVerifyObjects)
- Supports removal of objects by DRSReplicaVerifyObjects and LDAP rootDSE removeLingeringobjects modification

The tool leverages the Advisory Mode method exposed by *DRSReplicaVerifyObjects* that both repadmin /removelingeringobjects /Advisory_Mode and repldiag /removelingeringobjects use. In addition to the normal *Advisory Mode* related events logged on each DC, it displays each of the lingering objects within the main content pane.



Target DC	Reference DC	Object DN	Naming Context
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=emp10,CN=LostAndFound,DC=	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Keith Simone,OU=R&D,DC=ro	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User50,OU=Lingerin	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User56,OU=Lingerin	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Dick Schenk,OU=R&D,DC=roo	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Kara Lowe,OU=R&D,DC=root,D	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=mangle13,OU=Mangle,DC=roc	dc=root,dc=contoso,dc=com

Result	Result Code	Operation
Failure	1396	Failure: Operation on NC: CN=Configuration,DC=root,DC=contoso,DC=cc
Failure	1396	Failure: Operation on NC: CN=Schema,CN=Configuration,DC=root,DC=cc
Failure	1396	Failure: Operation on NC: DC=ForestDnsZones,DC=root,DC=contoso,DC=
Failure	1396	Failure: Operation on NC: DC=child,DC=root,DC=contoso,DC=com on tar

Number of Lingering Objects Found: 2176

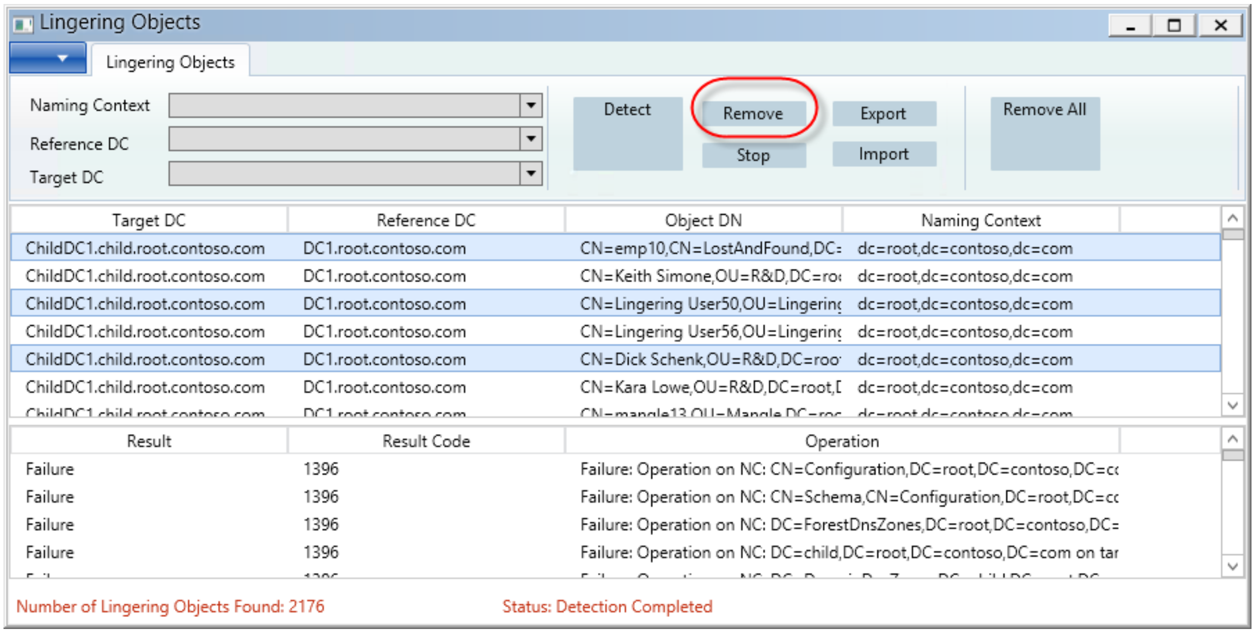
Status: Detection Completed

Details of the scan operation are logged in the **linger.log.txt** file in the same directory as the tool's executable.

The **Export** button allows you to export a list of all lingering objects listed in the main pane into a CSV file. View the file in Excel, modify if necessary and use the **Import** button later to view the objects without having to do a new scan. The Import feature is also useful if you discover abandoned objects (not discoverable with DRSReplicaVerifyObjects) that you need to remove.

Removal of individual objects

4. Select two or three objects (hold down the **Ctrl** key to select multiple objects, or the **SHIFT** key to select a range of objects) and then select **Remove**.

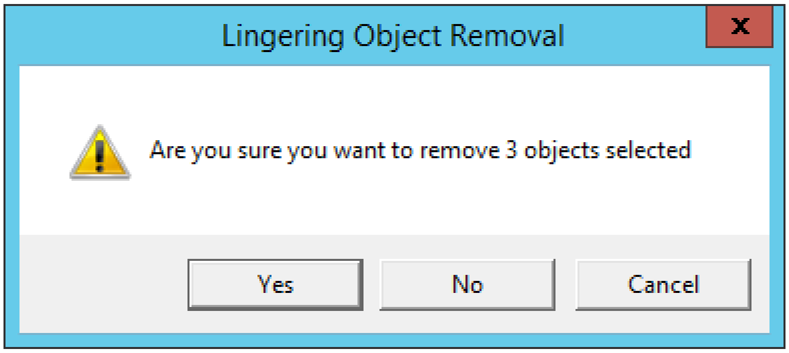


Target DC	Reference DC	Object DN	Naming Context
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=emp10,CN=LostAndFound,DC=	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Keith Simone,OU=R&D,DC=ro	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User50,OU=Lingering	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Lingering User56,OU=Lingering	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Dick Schenk,OU=R&D,DC=roo	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=Kara Lowe,OU=R&D,DC=root,D	dc=root,dc=contoso,dc=com
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=mangle13,OU=Mangle,DC=roo	dc=root,dc=contoso,dc=com

Result	Result Code	Operation
Failure	1396	Failure: Operation on NC: CN=Configuration,DC=root,DC=contoso,DC=cc
Failure	1396	Failure: Operation on NC: CN=Schema,CN=Configuration,DC=root,DC=cc
Failure	1396	Failure: Operation on NC: DC=ForestDnsZones,DC=root,DC=contoso,DC=
Failure	1396	Failure: Operation on NC: DC=child,DC=root,DC=contoso,DC=com on tar
Failure	1396	

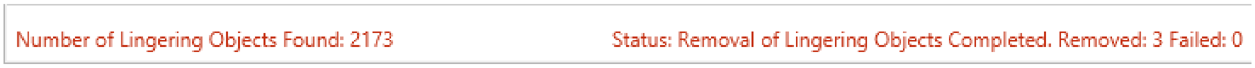
Number of Lingering Objects Found: 2176

Status: Detection Completed



Are you sure you want to remove 3 objects selected

The status bar is updated with the new count of lingering objects and the status of the removal operation:



Number of Lingering Objects Found: 2173

Status: Removal of Lingering Objects Completed. Removed: 3 Failed: 0

The tool dumps a list of attributes for each object before removal and logs this along with the results of the object removal in the **removedLingeringObjects.log.txt** log file. This log file is in the same location as the tool's executable.

C:\tools\LingeringObjects\removedLingeringObjects.log.txt

```
the obj DN:
<GUID=0bb376aa1c82a348997e5187ff012f4a>;<SID=010500000000000515000000609701d7b0ce8f6a3e529d669f040000>;CN=
Dick Schenk,OU=R&D,DC=root,DC=contoso,DC=com

objectClass:top, person, organizationalPerson, user;

sn:Schenk ;

whenCreated:20121126224220.0Z;
```

```
name:Dick Schenk;

objectSid:S-1-5-21-3607205728-1787809456-1721586238-1183;primaryGroupID:513;

sAMAccountType:805306368;

uSNChanged:32958;

objectCategory:<GUID=11ba1167b1b0af429187547c7d089c61>;CN=Person,CN=Schema,CN=Configuration,DC=root,DC=contoso,
DC=com;

whenChanged:20121126224322.0Z;

cn:Dick Schenk;

uSNCreated:32958;

l:Boulder;

distinguishedName:<GUID=0bb376aa1c82a348997e5187ff012f4a>;<SID=010500000000000515000000609701d7b0ce8f6a3e52
9d669f040000>;CN=Dick Schenk,OU=R&D,DC=root,DC=contoso,DC=com;

displayName:Dick Schenk ;

st:Colorado;

dSCorePropagationData:16010101000000.0Z;

userPrincipalName:Dick@root.contoso.com;

givenName:Dick;

instanceType:0;

sAMAccountName:Dick;

userAccountControl:650;

objectGUID:aa76b30b-821c-48a3-997e-5187ff012f4a;

value is :<GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e>:<GUID=aa76b30b-821c-48a3-997e-5187ff012f4a>

Lingering Obj CN=Dick Schenk,OU=R&D,DC=root,DC=contoso,DC=com is removed from the directory, mod response result code
= Success

----------------------------------------------

RemoveLingeringObject returned Success
```

Repldiag discovery

An alternate methods of discovery is to use repldiag.exe with the /AdvisoryMode switch.

```
Repldiag /removelingeringobjects /AdvisoryMode
```

- Leverages DRSReplicaVerifyObjects method in Advisory Mode (Like the LingeringObjects.exe tool)
- Run against almost all DCs (does not support RODCs), all partitions sans Schema
- Event ID 1946s are logged on each DC in the forest
- Need separate method to collect event message text from each DC for lingering object identification (can leverage PowerShell)

Replfix discovery

The Lingering Objects tool and repldiag do an excellent job of lingering object discovery. However, they do not identify one class of lingering objects, "Abandoned delete / Live lingering objects ".

Tip:

Abandoned delete / Live lingering objects

An object deleted on one DC that was never replicated to other DCs hosting a writable copy of the NC for that object. The deletion replicates to DCs/GCs hosting a read-only copy of the NC. The DC that originated the object deletion goes offline prior to replicating the change to other DCs hosting a writable copy of the partition.

Replfix.exe does a good job of discovery of this lingering object type.

Replfix is an unsupported tool that can be leveraged for lingering object discovery and removal. In order to use it, you must first get LDIFDE dumps of the partition from DCs you want replfix to analyze, then you use the tool to compare the two ldifde files. The tool leverages the LDAP rootDSE removeLingeringObject modification for lingering object removal.

Perform the following task on **Win8Client**.

1. LIDFDE dumps of the **root** partition from each DC

   Copy the following LDIFDE commands and paste into a command prompt on **Win8Client**.

Tip:

For ease of command entry: There is a file on **Win8Client** in the D:\files directory, called **fix_lab.txt** that contains all necessary commands needed for this lab. There is a mixture of both CMD-line and PowerShell commands in the file. To execute the commands:

1. Open an elevated PowerShell prompt on **Win8Client**.
2. Copy the commands for the step you are working on, and paste them into the PowerShell window.
3. It is best to copy the **Files** directory to the root of the C:\ drive before executing any commands. Some commands attempt to output files to the current working directory (which will fail for D:\Files because it is a read-only ISO file attached to the VM guest.

Alternately, you can copy them from the lab manual.

```
Ldifde -f dc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com
```

```
Ldifde -f dc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com
```

```
Ldifde -f trdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com -t 3268
```

```
Ldifde -f childdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com -t
3268
```

```
Ldifde -f childdc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com -t
3268
```

2. Compare the LDIFDE files with replfix.exe
    a. Copy replfix.exe from the **D:\files** on the **win8client** to the same directory as the files created in the prior step. (replfix is also located on **DC1**)

Replfix syntax

```
replfix <dc1.ldf> <dc2.ldf> -lingering <lingering1.ldf lingering2.ldf> [-log <log.txt>] [-debug] -domaindn "domaindn" [-rootdn
"rootdn"]-bloom <id>
```

    b. Copy the replfix commands below to perform the comparison

```
replfix dc1_root.ldf dc2_root.ldf -lingering dc1_root_lingering.ldf dc2_root_lingering.ldf -log
root_dc1_dc2.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
```

```
Checking dc2_root.ldf against dc1_root.ldf

Number of lingering objects detected on this server are: 108


Checking dc1_root.ldf against dc2_root.ldf

Number of lingering objects detected on this server are: 55

The operation was successful.
```

```
replfix dc1_root.ldf childdc1_root.ldf -lingering dc1_root_lingering_childdc1.ldf
childdc1_root_lingering.ldf -log root_dc1_childdc1.log -domaindn "dc=root,dc=contoso,dc=com" -
rootdn "dc=root,dc=contoso,dc=com"
```

```
Checking childdc1_root.ldf against dc1_root.ldf

........

Number of lingering objects detected on this server are: 142


Checking dc1_root.ldf against childdc1_root.ldf

........

Number of lingering objects detected on this server are: 9

The operation was successful.
```

```
replfix dc1_root.ldf childdc2_root.ldf -lingering dc1_root_lingering_childdc2.ldf
childdc2_root_lingering_dc1.ldf -log root_dc1_childdc2.log -domaindn "dc=root,dc=contoso,dc=com" -
rootdn "dc=root,dc=contoso,dc=com"
```

```
Checking childdc2_root.ldf against dc1_root.ldf
```

```
........
Number of lingering objects detected on this server are: 145


Checking dc1_root.ldf against childdc2_root.ldf
........
Number of lingering objects detected on this server are: 9
The operation was successful.
```

```
replfix dc1_root.ldf trdc1_root.ldf -lingering dc1_root_lingering_trdc1.ldf
trdc1_root_lingering_dc1.ldf -log root_dc1_trdc1.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn
"dc=root,dc=contoso,dc=com"
```

```
Checking trdc1_root.ldf against dc1_root.ldf
........
Number of lingering objects detected on this server are: 145


Checking dc1_root.ldf against trdc1_root.ldf
........
Number of lingering objects detected on this server are: 9
The operation was successful.
```

3. Review one of the .log files created by the various replfix commands to see a list of lingering objects present
   a. You can also view the screen output from the replfix commands for a quick overview of the level of divergence between the two DCs.

Important: Pay attention to the scenarios where DCs hosting a writeable copy of the NC are compared against GCs - the second check in the examples above. Replfix.exe is currently the only tool that supports this reverse comparison. However, the objects discovered could simply be flagged due to AD replication latency. For that reason, investigate the replication metadata for each object to determine if it is truly a lingering object.

One example:

```
Checking trdc1_root.ldf against dc1_root.ldf
........
Number of lingering objects detected on this server are: 145


Checking dc1_root.ldf against trdc1_root.ldf
........
Number of lingering objects detected on this server are: 9
```

In this example, DC1 (which hosts a writeable copy of the root partition) has **nine** lingering objects according to TRDC1 (which hosts a read-only copy of the partition).

Note: In this task, we used replfix.exe for discovery of lingering objects only (not removal. The tool created importable LDIFDE files that could be leveraged for object cleanup. We will not be using this removal method. We will look at various removal methods in the next exercise.

Exercise Review

In this exercise, we explored alternate lingering object discovery methods. Using a tool that does a forest-wide discovery of lingering objects is preferred over picking individual DCs and individual partitions.

Lingering Object Liquidator, repldiag and repadmin /removelingeringobjects all leverage the same function for lingering object discovery. Replfix.exe uses a different mechanism for lingering object discovery and is useful for discovery of abandoned deleted objects because it compares two DCs against each other both ways; the tool's usage is complicated and should be leveraged only when there is a need to do a reverse comparison.

Exercise 3 Lingering object removal methods

Methods to Remove Lingering Objects

In this Exercise:

In this exercise, you will use LDP, Repadmin, Repldiag and Lingering Objects.exe to remove lingering objects.

You will see the benefits of each method in order to help you understand which cleanup method to use

More:

There are many methods to remove lingering objects. This lab presents:

- Lingering Object Liquidator
- Repldiag /removelingeringobjects
- Repadmin /removelingeringobjects
- RemoveLingeringObject rootDSE modification variants

Other removal options include Repadmin /rehost | Repadmin /unhost with Repadmin /add (for GC read-only partitions)

Common methods to remove lingering objects include:

- **DRSReplicaVerifyObjects** methods
  - Repadmin /Removelingeringobjects
  - Repldiag /RemoveLingeringObjects
  - The new Lingering Object GUI-based discovery and removal tool (Lingering Object Liquidator)
- **RemoveLingeringObject** rootDSE modification variants
  - Manually through LDP or using script
  - Replfix compares LDIFDE files and then creates LDIFDE script
  - The new Lingering Object GUI-based discovery and removal tool
- **Rehost** the partition:
  - Repadmin /rehost (or /unhost and /add) (only if the partition is not-writable on the DC containing lingering objects)
  - Ugly options
    - Un-GC (but you don't really have control over who the DCs sources the partition from)
    - Demote and Promote (DCPromo)

Table 2: Lingering object removal methods

Removal method	Object / Partition & and Removal Capabilities	Details
Lingering Object Liquidator	Per-object and per-partition removal Leverages: • RemoveLingeringObjects LDAP rootDSE modification • DRSReplicaVerifyObjects method	• GUI-based. • Quickly displays all lingering objects in the forest to which the executing computer is joined. • Built-in discovery via DRSReplicaVerifyObjects method • Automated method to remove lingering objects from all partitions • Removes lingering objects from all DCs (including RODCs) but not lingering links.
Repldiag /removelingeringobjects	Per-partition removal Leverages: • DRSReplicaVerifyObjects method	• Command line only • Automated method to remove lingering objects from all partitions • Built-in discovery via DRSReplicaVerifyObjects • Displays discovered objects in events on DCs • Does not remove lingering links. Does not remove lingering objects from RODCs (yet)
LDAP RemoveLingeringObjects rootDSE primative (most commonly executed using LDP.EXE or an LDIFDE import script)	Per-object removal	• Requires a separate discovery method • Removes a single object per execution unless scripted.

Removal method	Object / Partition & and Removal Capabilities	Details
Repadmin /removelingeringobjects	Per-partition removal Leverages: • DRSReplicaVerifyObjects method	• Command line only • Built-in discovery via DRSReplicaVerifyObjects • Displays discovered objects in events on DCs • Requires many executions if a comprehensive (n * n-1 pairwise cleanup is required. Note: repldiag and the Lingering Object Liquidator tool automate this task.

Task 1 - Remove lingering objects using LDP

In this task, you will discover lingering objects using the Lingering Object Liquidator, but you will remove one using LDP. LDP leverages the LDAP RemoveLingeringObject rootDSE modification. You could also use another LDAP tool to perform the same object removal procedure (such as LDIFDE). The task is covered here so that a thorough review of lingering object removal methods are demonstrated in this exercise.

Perform this task on **Win8Client** and **ChildDC1**.

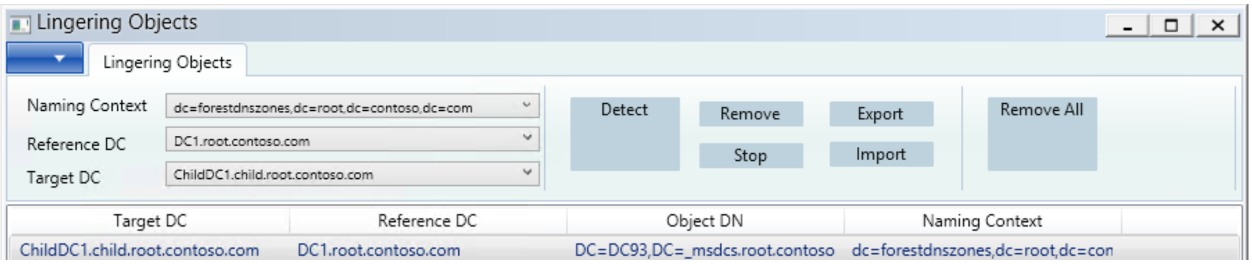
In this task, you will remove a DNS record in the **ForestDnsZones** partition from **ChildDC1** using LDP.

Per partition Lingering Object Discovery using the Lingering Object Liquidator

1. Connect to **Win8Client**.
2. Copy the **d:\Files** directory to the root of the **c:\** drive (if you have not already in a prior exercise)
3. Open the Lingering Objects tool: "C:\files\LingeringObjects\LingeringObjects.exe"

   If the tool is open from a prior step, close it and reopen it
4. Choose **Naming Context** and select **dc=forestdnszones,dc=root,dc=contoso,dc=com**
5. Choose **Reference DC** and then select **DC1.root.contoso.com**
6. Choose **Target DC** and then select **ChildDC1.child.root.contoso.com**
7. Select **Detect**

Results: Two lingering objects are discovered



Naming Context: dc=forestdnszones,dc=root,dc=contoso,dc=com

Reference DC: DC1.root.contoso.com

Target DC: ChildDC1.child.root.contoso.com

Target DC	Reference DC	Object DN	Naming Context
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	DC=DC93,DC=_msdcs.root.contoso	dc=forestdnszones,dc=root,dc=con

Two DNS records: **DC93** and **DC91**

Get Object and Reference DC details for lingering object removal

We just used the Lingering Objects tool to discover a lingering object on **ChildDC1** that does not exist on **DC1**.

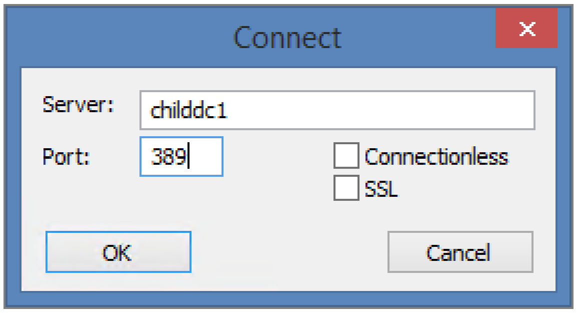
In order to remove an object using LDP, you need:

- The **objectGUID** for the object (We will use LDP to get this, but there are certainly many other methods)
- The **DSA object GUID** for a DC that hosts a writeable copy of the partition that does not have the object in the partition (DC1 for this example).

Next we will use **LDP** to view the DC93 object in order to get the objectGUID of the object

Perform these steps from **Win8Client**

8. Open LDP, connect and bind to the DC that has the lingering object
    a. From the **Connection** menu, choose **Connect**
    b. In the **Server** name field, type **childdc1**, ensure the port used is **389** and then choose **OK**



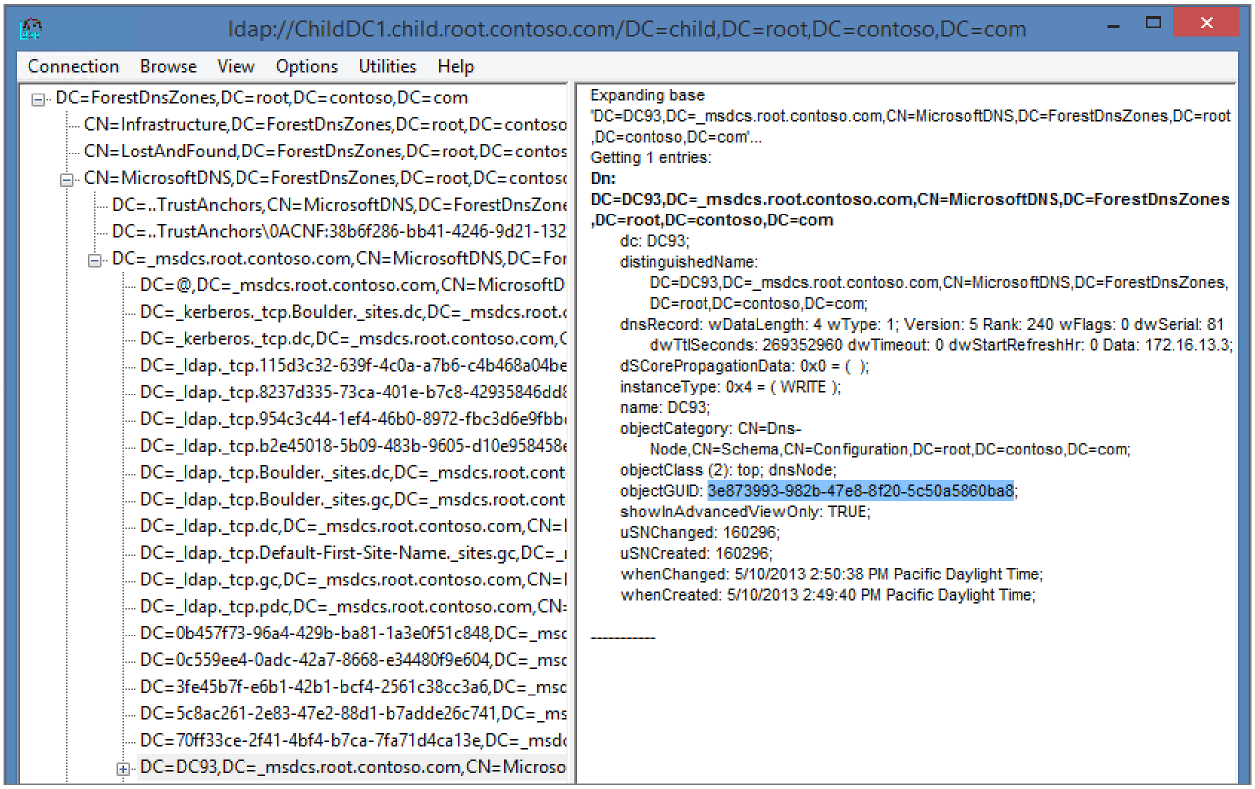
Server: childdc1

Port: 389

☐ Connectionless

☐ SSL

9. Select the **Connection** menu again, select **Bind** and then **OK**. (**Ctrl** + **B** is the keyboard shortcut)
10. From the **View** menu, select **Tree**, from the **BaseDN** menu, select **DC=ForestDnsZones,DC=root,DC=contoso,DC=com** and then select **OK**.
11. Expand **DC=ForestDNSZones…**, expand **CN=MicrosoftDNS…**, expand **DC=_msdcs.root.contoso.com…**


ldap://ChildDC1.child.root.contoso.com/DC=child,DC=root,DC=contoso,DC=com
Connection
Browse
View
Options
Utilities
Help
DC=ForestDnsZones,DC=root,DC=contoso,DC=com
CN=Infrastructure,DC=ForestDnsZones,DC=root,DC=contoso
CN=LostAndFound,DC=ForestDnsZones,DC=root,DC=contos
CN=MicrosoftDNS,DC=ForestDnsZones,DC=root,DC=contos
DC=..TrustAnchors,CN=MicrosoftDNS,DC=ForestDnsZone
DC=..TrustAnchors\0ACNF:38b6f286-bb41-4246-9d21-132
DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=For
DC=@,DC=_msdcs.root.contoso.com,CN=MicrosoftD
DC=_kerberos._tcp.Boulder._sites.dc,DC=_msdcs.root.
DC=_kerberos._tcp.dc,DC=_msdcs.root.contoso.com,C
DC=_ldap._tcp.115d3c32-639f-4c0a-a7b6-c4b468a04be
DC=_ldap._tcp.8237d335-73ca-401e-b7c8-42935846dd
DC=_ldap._tcp.954c3c44-1ef4-46b0-8972-fbc3d6e9fbb
DC=_ldap._tcp.b2e45018-5b09-483b-9605-d10e958458
DC=_ldap._tcp.Boulder._sites.dc,DC=_msdcs.root.cont
DC=_ldap._tcp.Boulder._sites.gc,DC=_msdcs.root.cont
DC=_ldap._tcp.dc,DC=_msdcs.root.contoso.com,CN=
DC=_ldap._tcp.Default-First-Site-Name._sites.gc,DC=_
DC=_ldap._tcp.gc,DC=_msdcs.root.contoso.com,CN=
DC=_ldap._tcp.pdc,DC=_msdcs.root.contoso.com,CN=
DC=0b457f73-96a4-429b-ba81-1a3e0f51c848,DC=_msc
DC=0c559ee4-0adc-42a7-8668-e34480f9e604,DC=_msc
DC=3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6,DC=_msc
DC=5c8ac261-2e83-47e2-88d1-b7adde26c741,DC=_ms
DC=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e,DC=_msd
DC=DC93,DC=_msdcs.root.contoso.com,CN=Microso

```
Expanding base
'DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones,DC=root
,DC=contoso,DC=com'...
Getting 1 entries:
Dn:
DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones
,DC=root,DC=contoso,DC=com
      dc: DC93;
      distinguishedName:
            DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones,
            DC=root,DC=contoso,DC=com;
      dnsRecord: wDataLength: 4 wType: 1; Version: 5 Rank: 240 wFlags: 0 dwSerial: 81
            dwTtlSeconds: 269352960 dwTimeout: 0 dwStartRefreshHr: 0 Data: 172.16.13.3;
      dSCorePropagationData: 0x0 = (  );
      instanceType: 0x4 = ( WRITE );
      name: DC93;
      objectCategory: CN=Dns-
            Node,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com;
      objectClass (2): top; dnsNode;
      objectGUID: 3e873993-982b-47e8-8f20-5c50a5860ba8;
      showInAdvancedViewOnly: TRUE;
      uSNChanged: 160296;
      uSNCreated: 160296;
      whenChanged: 5/10/2013 2:50:38 PM Pacific Daylight Time;
      whenCreated: 5/10/2013 2:49:40 PM Pacific Daylight Time;

-----------
```

12. Double click **DC=93** and copy the objectGUID for the DC93 object

```
Dn: DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones,DC=root,DC=contoso,DC=com
dc: DC93;
distinguishedName:
DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones,DC=root,DC=contoso,DC=com;
dnsRecord: wDataLength: 4 wType: 1; Version: 5 Rank: 240 wFlags: 0 dwSerial: 81 dwTtlSeconds: 269352960
dwTimeout: 0 dwStartRefreshHr: 0 Data: 172.16.13.3;
dSCorePropagationData: 0x0 = (  );
instanceType: 0x4 = ( WRITE );
name: DC93;
objectCategory: CN=Dns-Node,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com;
objectClass (2): top; dnsNode;
objectGUID: 3e873993-982b-47e8-8f20-5c50a5860ba8;
showInAdvancedViewOnly: TRUE;
uSNChanged: 160296;
uSNCreated: 160296;
whenChanged: 5/10/2013 2:50:38 PM Pacific Daylight Time;
whenCreated: 5/10/2013 2:49:40 PM Pacific Daylight Time;
```

Leave LDP open as it is needed after the following step

13. Get the DSA object GUID for **DC1**

    a. repadmin /showrepl DC1
    (one of many ways to get the DSA object GUID for DC1 is via repadmin /showrepl)

```
C:\ >repadmin /showrepl dc1
Boulder\DC1
```

```
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
DSA invocationID: 088cb927-32b3-4a1b-8084-e679d0cc146d

==== INBOUND NEIGHBORS ======================================

DC=root,DC=contoso,DC=com
   Boulder\DC2 via RPC
      DSA object GUID: 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
      Last attempt @ 2014-07-22 08:35:56 was successful.
```

We now have everything we need to remove this object:

1. The objectGUID of the lingering object:

   3e873993-982b-47e8-8f20-5c50a5860ba8

2. The DSA object GUID from a DC that is a good reference DC (hosts a writable copy of the partition and does not contain the object)

   70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e

Remove the Object

1. Switch back to LDP, from the **Browse** menu, select **Modify**
2. In the **Attribute** box, type **RemoveLingeringObject**.
3. Type **<GUID=** as the value.
4. Append the **DSA object GUID** of the reference domain controller

```
<GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
```

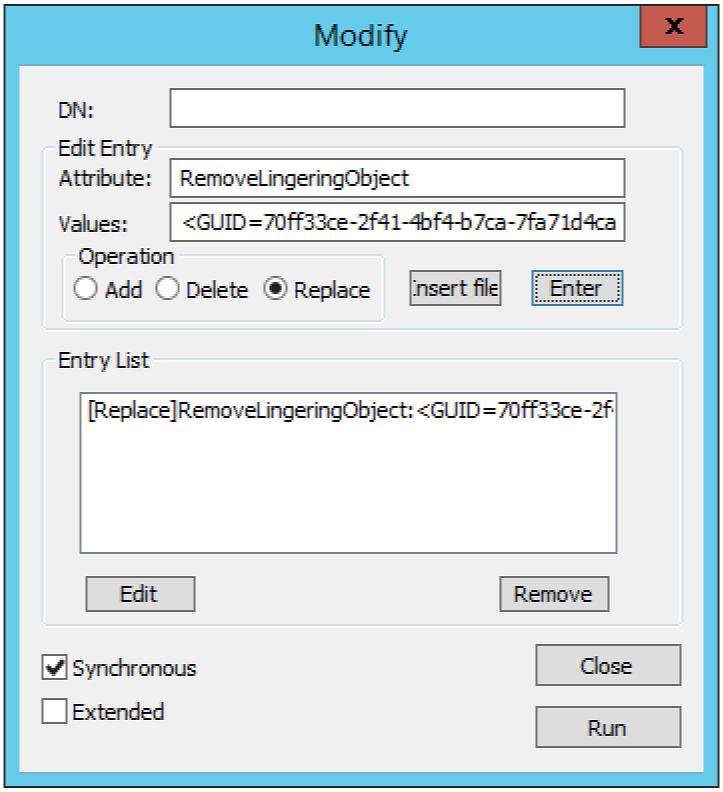
5. Append **> : <GUID=**. Do not omit the spaces.

```
<GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e> : <GUID=
```

6. Append the ObjectGUID of the lingering object.
7. Append **>**.
8. The complete value should look similar to:

```
<GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e> : <GUID=3e873993-982b-47e8-8f20-5c50a5860ba8>
```

9. Click the **Replace** operation, and then click **Enter** on the interface. Now the command appears in the **Entry** list.



Modify

DN:

Edit Entry

Attribute: RemoveLingeringObject

Values: <GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca

Operation

○ Add ○ Delete ● Replace

Entry List

[Replace]RemoveLingeringObject:<GUID=70ff33ce-2f

☑ Synchronous

□ Extended

10. Click **Run** to have the object removed. The main content pane of LDP contains the result of the request. It will look like this if the operation was successful.

```
***Call Modify...
ldap_modify_s(ld, '(null)',[1] attrs);
Modified "".
```

Task 2 - Remove lingering objects using repadmin

In this task, you will remove lingering objects using **repadmin /removelingeringobjects**.

- Repadmin /removelingeringobjects

  Remove objects from one partition on one DC per command line execution

- Note that may also cleanup lingering objects in GC read-only partitions by rehosting the partition using repadmin
  - Repadmin /rehost
  - Repadmin /unhost followed up with repadmin /add

In the last task, we removed one object from the ForestDNSZones partition from ChildDC1. However, one or more lingering objects still remain, so replication is still blocked. In this task, we will use repadmin /removelingeringobjects to remove the remaining objects from this partition (as compared with DC1).

Note:

Lingering Object removal using repadmin /removelingeringobjects

The command's syntax is:

repadmin /removelingeringobjects *LingeringDC* *ReferenceDC_DSA_GUID* PartitionDN

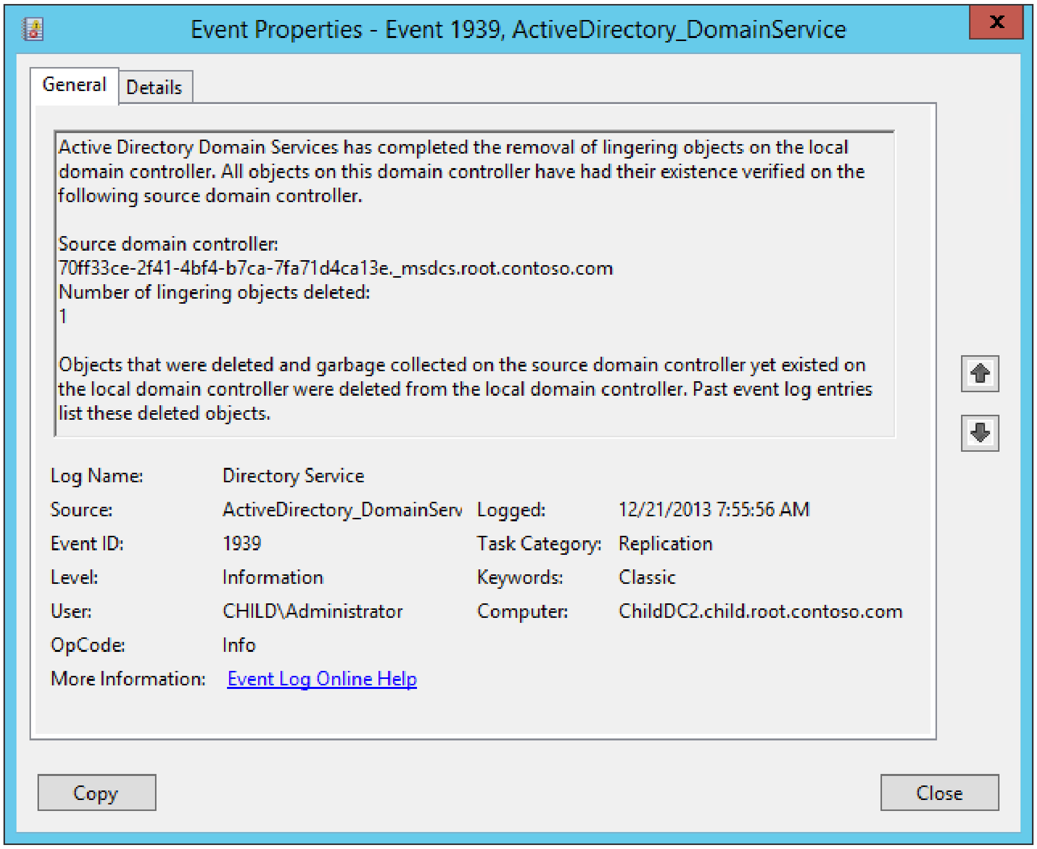
Where:

- LingeringDC: FQDN of DC that has the lingering objects
- ReferenceDC_DSA_GUID: The DSA GUID of a domain controller that hosts a writeable copy of the partition
- PartitionDN: The distinguished name of the directory partition where the lingering objects exist

1. Run the following command to clean up the remaining object(s) in the ForestDNSZones partition on **childdc1**

```
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-
7fa71d4ca13e "dc=forestdnszones,dc=root,dc=contoso,dc=com"
```

2. Review the Directory Service event log on **ChildDC1** for the results of the lingering object removal request. Review the details of event ID **1939**, which reports the status of the lingering object removal process.


Event Properties - Event 1939, ActiveDirectory_DomainService
General
Details
Copy
Close

Active Directory Domain Services has completed the removal of lingering objects on the local domain controller. All objects on this domain controller have had their existence verified on the following source domain controller.

Source domain controller:
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com
Number of lingering objects deleted:
1

Objects that were deleted and garbage collected on the source domain controller yet existed on the local domain controller were deleted from the local domain controller. Past event log entries list these deleted objects.

Log Name: Directory Service
Source: ActiveDirectory_DomainServ
Logged: 12/21/2013 7:55:56 AM
Event ID: 1939
Task Category: Replication
Level: Information
Keywords: Classic
User: CHILD\Administrator
Computer: ChildDC2.child.root.contoso.com
OpCode: Info
More Information: Event Log Online Help

```
get-winevent -LogName "Directory Service" -ComputerName childdc1 -MaxEvents 10 | Where-Object {$_.ID -eq
"1939"} | fl
```

Note:

At this point, the ForestDNSZone partition is clean on childdc1 as compared to DC1. Thoroughly cleaning this partition requires that you compare childdc1 against everyone else and then compare all of them against childdc1. Also, keep in mind, that if there are lingering objects in one partition, there are usually lingering objects in the other partitions.

Task 3 - Remove lingering objects using Repldiag

In the last task, you cleaned up one partition on one DC. There is still a lot of work to do if you want to do a thorough job of lingering object removal though. In this task, you will leverage a tool that automates the majority of the lingering object removal work needed for most environments.

Note:

- Repldiag requires a well-connected topology. It will fail to run in environments that suffer from poor network connectivity *.
- Always check for the latest version on CodePlex: *http://activedirectoryutils.codeplex.com/*

  * There is a hidden parameter that allows the tool to continue in spite of topology issues, but do not use it without recognizing the ramifications: Use of the /BypassStabilityCheck parameter will likely result in a failure to fully clean up the environment.

Repldiag will run commands to remove lingering objects from all partitions.

Important:

When lingering objects are discovered, assume they are present on all DCs in all partitions. Do not just clean up the DCs reporting the errors. Repldiag automates the majority of the cleanup work. See the **Lingering Object discovery and cleanup** section more information.

Perform this task on **Win8Client**.

The following command will check for and remove lingering objects from most DCs (RODCs are not checked) for all partitions (except Schema)

1. From **Win8Client**, run the following from an elevated command prompt

```
Repldiag /removelingeringobjects
```

2. Close and Reopen the Lingering Object tool (if already opened) and select **Detect**

   Are all objects removed from the environment?

Notice the RODC in the child domain still contains lingering objects.

Note:

At the time of this writing, Replidag (v 2.0.4947.18978) does not remove lingering objects from RODCs. (It was developed prior to the existence of RODCs.) This functionality will be implemented eventually.

If you used repldiag to remove the lingering objects, you are done with this task, and do not need to perform the alternate task steps.

Repadmin /removelingeringobjects equivalent steps

Important:

Do not perform the following steps. Just review the commands, and move onto Task 4. These commands are provided here to show you how much time you save with tools like **repldiag** and the **Lingering Objects tool.**

Clean up the reference DCs first

Configuration partition

```
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-
7fa71d4ca13e "cn=configuration,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc1.child.root.contoso.com 3fe45b7f-e6b1-42b1-bcf4-
2561c38cc3a6 "cn=configuration,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc1.child.root.contoso.com 0b457f73-96a4-429b-ba81-
1a3e0f51c848 "cn=configuration,dc=root,dc=contoso,dc=com"
```

ForestDNSZones partition

```
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-
7fa71d4ca13e "dc=forestdnszones,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc1.child.root.contoso.com 3fe45b7f-e6b1-42b1-bcf4-
2561c38cc3a6 "dc=forestdnszones,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc1.child.root.contoso.com 0b457f73-96a4-429b-ba81-
1a3e0f51c848 "dc=forestdnszones,dc=root,dc=contoso,dc=com"
```

Root domain partition

```
repadmin /removelingeringobjects dc1.root.contoso.com 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
"dc=root,dc=contoso,dc=com"
```

DomainDNSZones application partition for the **root** domain

```
repadmin /removelingeringobjects dc1.root.contoso.com 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
"dc=domaindnszones,dc=root,dc=contoso,dc=com"
```

Note:

You do not need to clean up reference DCs for the **Child**, **TreeRoot** or their **DomainDNSZones** partitions. This is because there is only one DC in each domain that hosts a writable copy of the partition. The schema partition is not checked or cleaned up because you cannot delete objects from the schema.

Now that the reference DCs are cleaned up. Clean up all remaining DCs against the reference DCs

Configuration

```
Repadmin /removelingeringobjects dc1.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"cn=configuration,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects dc2.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"cn=configuration,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc2.child.root.contoso.com 0c559ee4-0adc-42a7-8668-
e34480f9e604 "cn=configuration,dc=root,dc=contoso,dc=com"



Repadmin /removelingeringobjects trdc1.treeroot.fabrikam.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"cn=configuration,dc=root,dc=contoso,dc=com"
```

ForestDNSZones

```
Repadmin /removelingeringobjects dc1.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=forestdnszones,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects dc2.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=forestdnszones,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc2.child.root.contoso.com 0c559ee4-0adc-42a7-8668-
e34480f9e604 "dc=forestdnszones,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects trdc1.treeroot.fabrikam.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=forestdnszones,dc=root,dc=contoso,dc=com"
```

Root domain partition

```
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-
7fa71d4ca13e "dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc2.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-
7fa71d4ca13e "dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects dc2.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
"dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects trdc1.treeroot.fabrikam.com 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
"dc=root,dc=contoso,dc=com"
```

DomainDNSZones - Root

```
Repadmin /removelingeringobjects dc2.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
"dc=domaindnszones,dc=root,dc=contoso,dc=com"
```

Child domain partition

```
Repadmin /removelingeringobjects dc1.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=child,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects dc2.root.contoso.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=child,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects childdc2.child.root.contoso.com 0c559ee4-0adc-42a7-8668-
e34480f9e604 "dc=child,dc=root,dc=contoso,dc=com"

Repadmin /removelingeringobjects trdc1.treeroot.fabrikam.com 0c559ee4-0adc-42a7-8668-e34480f9e604
"dc=child,dc=root,dc=contoso,dc=com"
```

DomainDNSZones - Child

```
Repadmin /removelingeringobjects childdc2.child.root.contoso.com 0c559ee4-0adc-42a7-8668-
e34480f9e604 "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com"
```

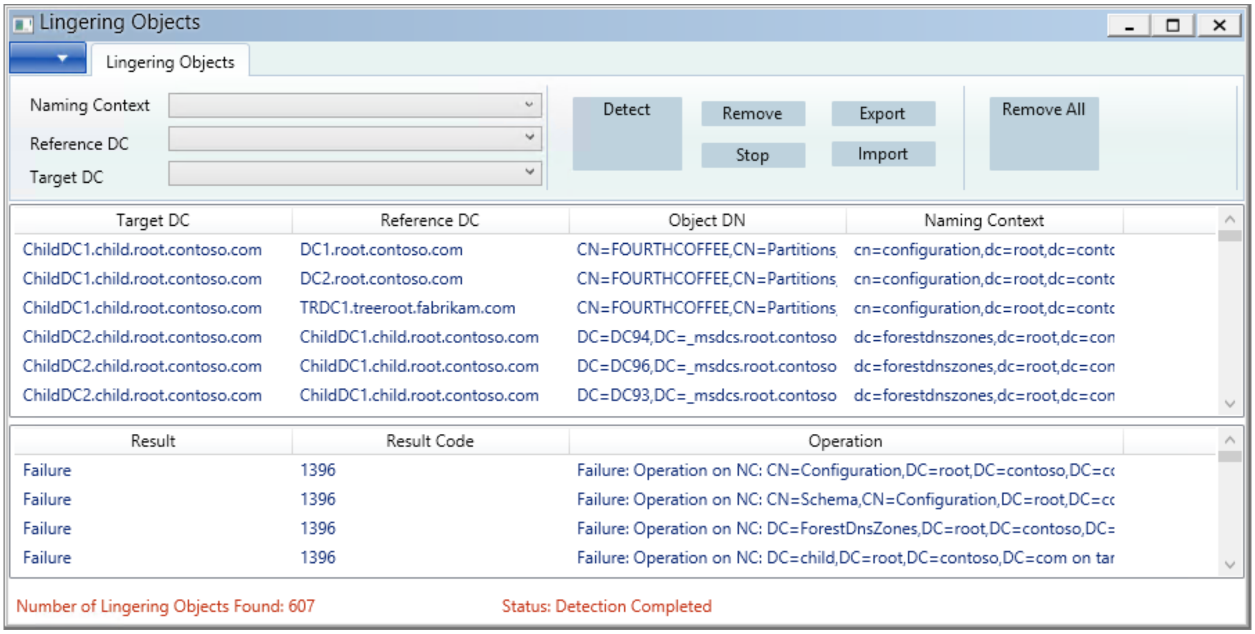
TreeRoot domain partition

```
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 0b457f73-96a4-429b-ba81-1a3e0f51c848 "dc=treeroot,dc=fabrikam,dc=com"

Repadmin /removelingeringobjects childdc2.child.root.contoso.com 0b457f73-96a4-429b-ba81-1a3e0f51c848 "dc=treeroot,dc=fabrikam,dc=com"

Repadmin /removelingeringobjects dc1.root.contoso.com 0b457f73-96a4-429b-ba81-1a3e0f51c848 "dc=treeroot,dc=fabrikam,dc=com"

Repadmin /removelingeringobjects dc2.root.contoso.com 0b457f73-96a4-429b-ba81-1a3e0f51c848 "dc=treeroot,dc=fabrikam,dc=com"
```

Task 4 - Remove Lingering Objects using the Lingering Object Liquidator tool

In Exercise 2, you leveraged LOL to remove individual objects. You also have a one-button option to remove all objects present in the environment. In this task, you will remove the remaining lingering objects visible in the tool.

1. On **Win8client**, open the Lingering Objects tool.
2. Click the **Detect** button to see if repldiag already removed all objects from the environment.



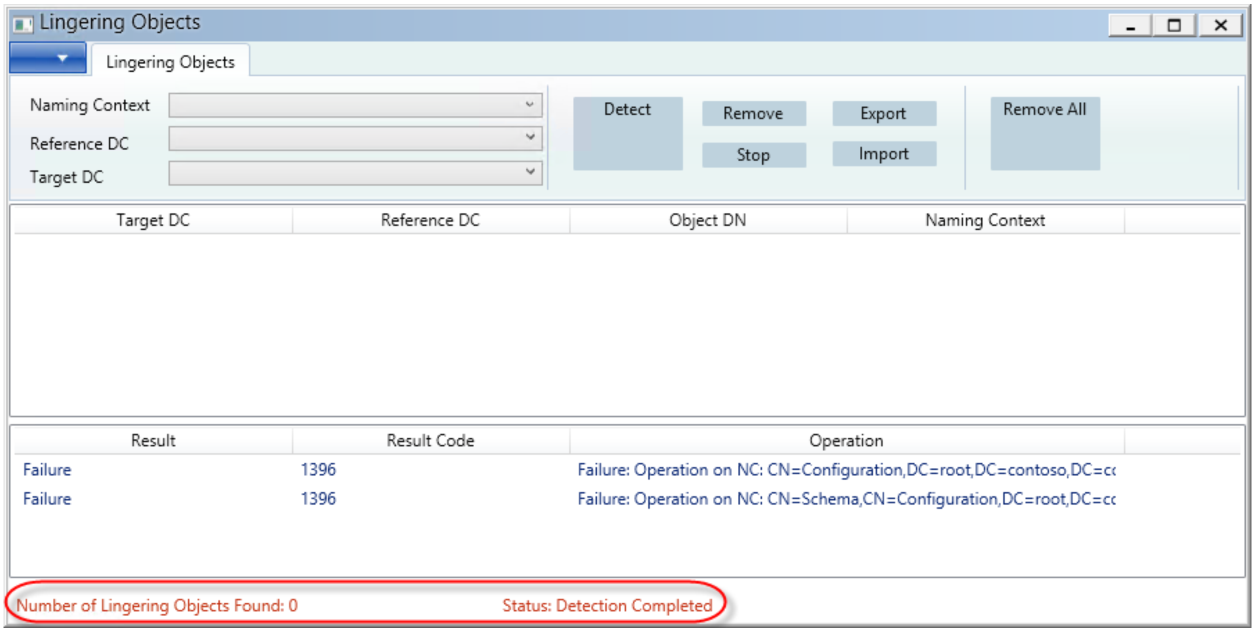
Target DC	Reference DC	Object DN	Naming Context
ChildDC1.child.root.contoso.com	DC1.root.contoso.com	CN=FOURTHCOFFEE,CN=Partitions,	cn=configuration,dc=root,dc=contc
ChildDC1.child.root.contoso.com	DC2.root.contoso.com	CN=FOURTHCOFFEE,CN=Partitions,	cn=configuration,dc=root,dc=contc
ChildDC1.child.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=FOURTHCOFFEE,CN=Partitions,	cn=configuration,dc=root,dc=contc
ChildDC2.child.root.contoso.com	ChildDC1.child.root.contoso.com	DC=DC94,DC=_msdcs.root.contoso	dc=forestdnszones,dc=root,dc=con
ChildDC2.child.root.contoso.com	ChildDC1.child.root.contoso.com	DC=DC96,DC=_msdcs.root.contoso	dc=forestdnszones,dc=root,dc=con
ChildDC2.child.root.contoso.com	ChildDC1.child.root.contoso.com	DC=DC93,DC=_msdcs.root.contoso	dc=forestdnszones,dc=root,dc=con

Result	Result Code	Operation
Failure	1396	Failure: Operation on NC: CN=Configuration,DC=root,DC=contoso,DC=cc
Failure	1396	Failure: Operation on NC: CN=Schema,CN=Configuration,DC=root,DC=cc
Failure	1396	Failure: Operation on NC: DC=ForestDnsZones,DC=root,DC=contoso,DC=
Failure	1396	Failure: Operation on NC: DC=child,DC=root,DC=contoso,DC=com on tar

Number of Lingering Objects Found: 607

Status: Detection Completed

3. Click the **Removal All** button. The status bar is updated with the count of lingering objects removed. (the count may differ to the discovered amount due to a bug in the tool-this is a display issue only and the objects are actually removed)
4. Close the tool and reopen it so that the main content pane is cleared.
5. Click the **Detect** button and verify no lingering objects are found.



Target DC	Reference DC	Object DN	Naming Context

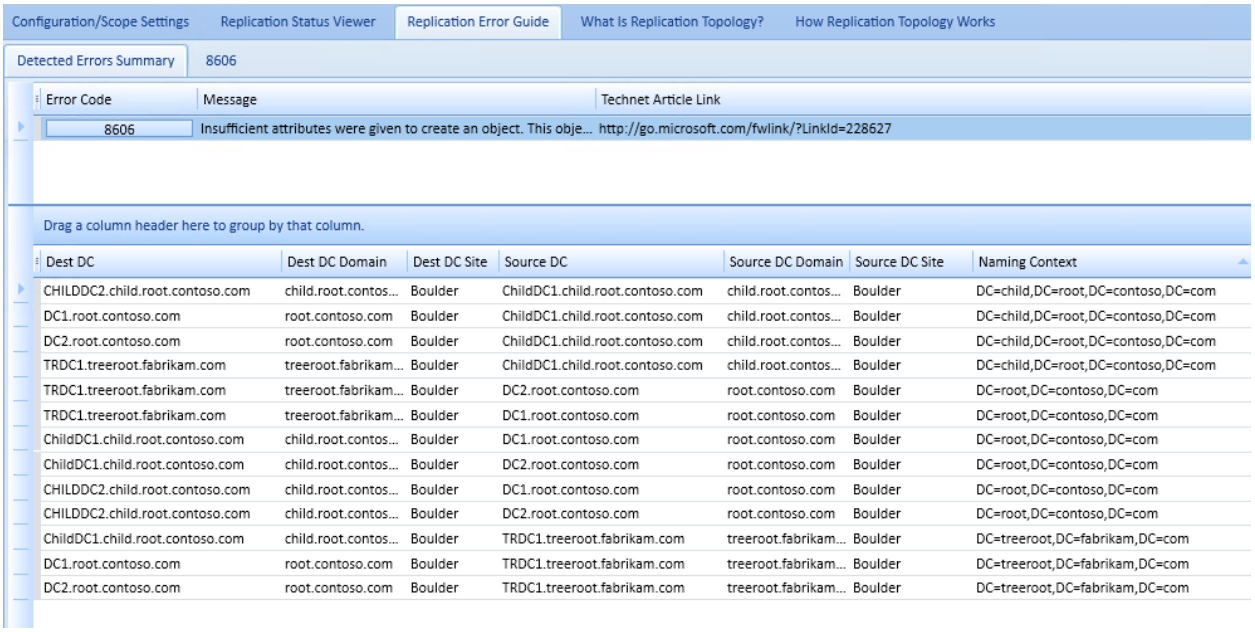
Result	Result Code	Operation
Failure	1396	Failure: Operation on NC: CN=Configuration,DC=root,DC=contoso,DC=cc
Failure	1396	Failure: Operation on NC: CN=Schema,CN=Configuration,DC=root,DC=cc

Number of Lingering Objects Found: 0 Status: Detection Completed

6. Initiate replication on all DCs.

```
Repadmin /syncall dc1 /Aed
Repadmin /syncall dc2 /Aed
Repadmin /syncall childdc1 /Aed
Repadmin /syncall childdc2 /Aed
Repadmin /syncall trdc1 /Aed
```

7. Check forest-wide AD replication using ADReplstatus or repadmin /showrepl * /csv


Configuration/Scope Settings
Replication Status Viewer
Replication Error Guide
What Is Replication Topology?
How Replication Topology Works
Detected Errors Summary
8606

Error Code	Message	Technet Article Link
8606	Insufficient attributes were given to create an object. This obje...	http://go.microsoft.com/fwlink/?LinkId=228627

Drag a column header here to group by that column.

Dest DC	Dest DC Domain	Dest DC Site	Source DC	Source DC Domain	Source DC Site	Naming Context
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC=child,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC1.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
CHILDDC2.child.root.contoso.com	child.root.contos...	Boulder	DC2.root.contoso.com	root.contoso.com	Boulder	DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	child.root.contos...	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com
DC1.root.contoso.com	root.contoso.com	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com
DC2.root.contoso.com	root.contoso.com	Boulder	TRDC1.treeroot.fabrikam.com	treeroot.fabrikam...	Boulder	DC=treeroot,DC=fabrikam,DC=com

```
repadmin /showrepl * /csv | convertfrom-csv | out-gridview
```

The only replication error that remains is error 8606 for the **Child**, **Root** and **TreeRoot** partitions.

Why are there still symptoms of lingering objects in the environment?

In the next lesson, we will explore a special class of lingering objects not detected via the DRSReplicaVerifyObjects method: **abandoned objects**.

Task 5 - "Live" lingering object (abandoned deleted object) remediation

After a thorough removal of lingering objects in the last task, we discovered there are still symptoms of lingering objects in the environment. In this exercise, we explore a special class of lingering object, called a "live" lingering object.

More:

"Live" lingering object / Abandoned deleted object

An object deleted on one DC that never replicated to other DCs hosting a writable copy of the NC for that object. The deletion replicates to DCs/GCs hosting a read-only copy of the NC. The DC that originated the object deletion goes offline prior to replicating the change to other DCs hosting a writable copy of the partition. The lingering object remains "live" on the remaining DCs due to the abandoned delete.

Scenario:

Destination DC/GCs report that source DCs have lingering objects in source DC partition:

- Root.contoso.com: DC1 and DC2
- Child.root.contoso.com: ChildDC1 and ChildDC2
  - **ChildDC1** replicates **Root** partition from **DC1** and replication fails with error 8606

Perform this task on **win8client**.

Event 1988 can identify one object for us, but as discussed earlier, event 1988 only reports the first object encountered and there are usually many more. We will use **replfix.exe** to identify the rest.

1. From **win8client**, switch to the **C:\Files** directory (folder copied from the D drive in an earlier exercise)
2. Execute **ldifde_replfixCMDs.bat**
   - The contents of the *ldifde_replfixCMDs*.bat batch file are also included in the Appendix.
   - This batch file initiates all of the ldifde exports that replfix.exe needs for its analysis.
3. Execute the *Replfix_cmds.bat* file (also included in the Appendix).

- This runs replfix against each of the LDIF files in a pairwise fashion so that all DCs are checked for their respective partitions.
- There are two LDIF files and one log generate for each commands execution.
- The summary output for all command execution is in the file, run.log.

4. Open the **run.log** file and examine the output to help determine the scope of the problem

```
Checking childdc1_root.ldf against dc1_root.ldf

........

Number of lingering objects detected on this server are: 0


Checking dc1_root.ldf against childdc1_root.ldf

........

Number of lingering objects detected on this server are: 9

The operation was successful.
```

We can see from the example that the lingering objects in the **root** partition on **ChildDC1** have been removed (from our repldiag and LingeringObjects.exe removal cleanup steps in Tasks 3 and 4), but DC1 (which hosts a writeable copy of the root partition)still has 9 lingering objects according to ChildDC1 (which hosts a read-only copy of the partition).

5. Review each of the DC to DC comparison logs. (eg. Root_dc1_childdc1.log)

A review of the logs for the **root** partition reveals that DC1 and DC2 have two user objects, "Carl Woodbury" and "Cassie McKenzie" and child objects associated with the same two user objects while the GCs do not.

The next step is to determine if these objects are not present on the GCs due to AD replication latency or if they are perhaps live lingering objects.

6. Collect replication metadata for each object

a. Obtain the **Object GUID** for the **Carl Woodbury** user object. This is displayed in the **root_dc1_childdc1.log** file.

Repadmin /showobjmeta * "<GUID=***ObjectGUID***>" >carl_obj.txt

Analyze the data and determine if this is an AD replication latency issue, or if these are live lingering objects.

b. Look at the showobjmeta output for the object in order to get the following information:

Look at attribute metadata for an attribute that was stamped at time of object creation (such as objectClass) and is still version 1. In other words, identify the following metadata for the earliest timestamp so you can determine when the object was created.

   a. Originating DSA GUID

   b. Originating USN

c. Look at the /ShowUTDVec output for the object's partition

   c. Open **utdvec_root.txt**

   d. For the objects that replfix discovered: correlate the replication metadata with the /showutdvec output from each GC to determine if that GC should have seen the object creation USN.

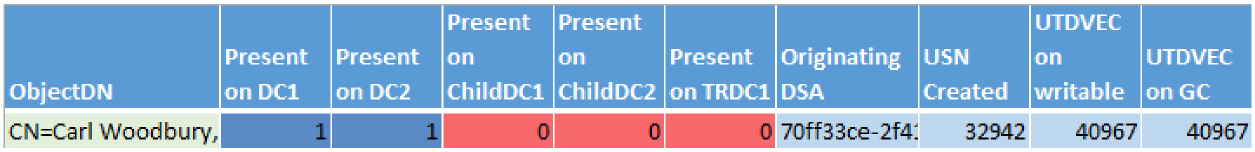
   e. If the GC has a USN value for the DSA GUID higher than the one used for object creation by the same DSA, then this GC should have seen this object creation. This data indicates a live lingering object / abandoned delete scenario. If the USN in the showutdvec output is lower than the value used for object creation then this is AD replication latency.

Replication metadata for the Carl Woodbury user object:

```
 Repadmin: running command /showobjmeta against full DC DC1.root.contoso.com
31 entries.
Loc.USN                    Originating DSA  Org.USN  Org.Time/Date       Ver Attribute
=======       =============== ========= =============       === =========
 32942     70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e     32942 2012-11-26 14:42:20    1 objectClass
  32942     70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e     32942 2012-11-26 14:42:20    1 cn
  32942     70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e     32942 2012-11-26 14:42:20    1 sn
 233681     fef36435-b9b7-4ab9-afa2-c788ed12354c    233681 2013-11-26 14:30:48    2 l
  32942     70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e     32942 2012-11-26 14:42:20    1 st
  32942     70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e     32942 2012-11-26 14:42:20    1 postalCode
```

Repadmin /showutdvec output of the Root partition

```
Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com

9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    221009 @ Time 2013-05-10 04:09:59
606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152692 @ Time 2013-05-10 05:04:52
9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    188781 @ Time 2013-05-10 05:55:26
336d313f-cce1-4c52-a57e-1135d54985fa @ USN     70324 @ Time 2013-05-10 05:56:41
fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    233259 @ Time 2013-05-20 13:00:28
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38
```



ObjectDN	Present on DC1	Present on DC2	Present on ChildDC1	Present on ChildDC2	Present on TRDC1	Originating DSA	USN Created	UTDVEC on writable	UTDVEC on GC
CN=Carl Woodbury,	1	1	0	0	0	70ff33ce-2f41	32942	40967	40967

A review of the data reveals that the rest of the DCs also have "live" lingering objects in their partitions according to the GCs.

Scenario Details

- Objects deleted on DC1 (root partition)
- Originating delete is only seen by GCs
- DC1 that originated deletion goes away for good before replicating knowledge of the deletion to other R/W DCs for Root partition
- No DCs hosting a R/W copy of the partition ever receive the knowledge of the deletion before TSL # of days
- GCs remove the object after TSL # of days go by via garbage collection

Effective status:

- Objects are still present on remaining R/W DCs
- GCs have garbage collected these objects so they are no longer present on GCs
- When GCs attempt to replicate the Root partition from R/W DCs; replication fails with error 8606 – since we are configured for Strict Replication Consistency
- GCs report DCs hosting a R/W copy of the partition have lingering object(s) for the same partition via event ID 1988
- Repadmin /RemoveLingeringObjects and other tools that leverage DRSReplicaVerifyObjects fails to identify objects
- Replfix is used for discovery of objects in this state

Scenario example

Domain Controllers	Cn=joe,cn=users,dc=root,dc=contoso,dc=com	Sample user (doesn't actually exist in this lab) in Root partition
Dc1.root.contoso.com	Object present	Full object visible with LDAP tools (use repadmin /showobj to observe the object only exists on the R/W DCs)
Dc2.root.contoso.com	Object present	
Childdc1.root.contoso.com	Object tombstoned and garbage collected	• Showutdvec reports higher USN seen by DC that originated delete than remaining R/W DCs

		• Originating DC no longer present in the environment
Childdc2.root.contoso.com	Object tombstoned and garbage collected	Same
Trdc1.treeroot.fabrikam.com	Object tombstoned and garbage collected	same

Live Lingering object Cleanup options

Cleanup options:	Result	Pros	Cons
Repadmin /rehost Root partition on each GC	Objects now present on GCs	• Easy to implement • Resolves problem without having to first discover all objects • Can be used in place of removelingeringobjects • Cleans up other classes of lingering objects present on the target DC	• Could be a lengthy recovery – partition size, network connections speed • Replication of all objects, not just the ones impacted • GC still advertises as a GC while partition may not be present on DC
Repadmin /replicate with the /full switch to each GC from a R/W DC	Objects now present on GCs	• Easy to implement • Resolves the problem without first having to discover all objects	• Full partition sync • Must have cleaned up partition with removelingeringobjects first
Authoritatively restore each object	Objects now present on GCs	• Touches just the objects restored • Poses the least risk	• Harder to implement • Discovery of all objects required before implementation
Replfix	Discovery only – doesn't fix	• Useful for discovery only	• Can't be used to remove objects for this specific scenario • ldifde file cannot be used for cleanup since all R/W DCs still have the object present – Replfix leverages the LDAP RemoveLingeringObjects rootDSE modification

Note:

To save lab time, we go with the easiest / fastest method. However, weigh the Pros and Cons of each scenario for your customer's environment. I prefer the **authoritative restore of each object** method since that option poses the least amount of risk to the environment.

7. Use repadmin /replicate with the /full parameter to have the GCs get a copy of the live lingering object(s), then update replication status.

```
repadmin /replicate dc2 dc1 dc=root,dc=contoso,dc=com /full

repadmin /replicate dc1 dc2 dc=root,dc=contoso,dc=com /full

repadmin /replicate * dc1 dc=root,dc=contoso,dc=com /full

repadmin /replicate * childdc1 dc=child,dc=root,dc=contoso,dc=com /full

repadmin /replicate * trdc1 dc=treeroot,dc=fabrikam,dc=com /full

repadmin /syncall dc1 /Aed

repadmin /syncall dc2 /Aed

repadmin /syncall childdc1 /Aed

repadmin /syncall childdc2 /Aed

repadmin /syncall trdc1 /Aed
```

8. Check forest-wide replication status

AD Replication now completes successfully for each partition. However, there are still data divergence issues in this Active Directory environment. In the next optional Exercise, we will leverage a tool called Oabvalidate to aid in the discovery of the data divergence.

Note:

Basic Data collection to identify abandoned objects

- Sample object is present in the Engineering OU
- Repadmin /showattr * "<GUID=ObjectGuid>" /gc >show.txt
- Repadmin /showobjmeta * "<GUID=ObjectGUID>" >>show.txt
- Identify Originating DSA for object creation from showobjmeta output
- Use Repadmin /showutdvec to determine highest USN received by RW replicas from originator of this object

Exercise Review

In this exercise, we removed lingering objects using LDP, repadmin, repldiag and Lingering Object Liquidator. We also identified and re-animated live lingering objects.

(Optional) Exercise 4: Lingering Link identification and cleanup

Time permitting. This exercise is not fully documented due to time constraints. Try this exercise if there is still time remaining.

During this exercise, you will identify all lingering-linked values in the environment. You will leverage a tool called Oabvalidate.exe that was originally written for Microsoft Exchange Offline Address Book generation failure troubleshooting. Further development went into the tool recently to help in the discovery of other AD data inconsistency issues. It is not a requirement to have Exchange in the environment (if you execute the tool from a command-line and pass an LDAP filter as an argument). The tool scans for a variety of AD data inconsistencies and logs the data to the user's Documents directory.

Scenario: Group membership consistency issues.

Perform this task on **Win8client**

1. Open an elevated command prompt and run oabvalidate.exe against **DC1**

```
Oabvalidate dc1 "(Objectclass=*)"
```

   - Ignore the Oabvalidate window that opens and closes
   - Output is logged in the Documents directory in a folder named *data_timestamp-<DC Name>*

2. Next check **DC2**

```
Oabvalidate dc2 "(Objectclass=*)"
```

   If the command appears to hang without returning to a command prompt, open a new command prompt window and run the remaining commands one at a time

3. Check **ChildDC1**

```
Oabvalidate childdc1 "(Objectclass=*)"
```

4. Next up: **ChildDC2**

```
Oabvalidate childdc2 "(Objectclass=*)"
```

5. Finally, check **TRDC1**

```
Oabvalidate trdc1 "(Objectclass=*)"
```

6. Open problemattributes.txt in Excel (tab and semicolon delimited). Steps for this are in the Appendix in the *Open problemattributes.txt in Excel* section.

Note:

A consolidated copy of this data is present in the **c:\files\ALL_DCs_ProblemAttributes.xlsx** file to speed up data analysis for this lab.

Problem attributes.txt from each DC reveals the following scenario:

- There are many lingering links in the member attribute of several group objects.
- The group membership inconsistencies are all for read-only copies of the group.

7. Identify one object on DC1: **LLGroup1** is listed with two member attributes listed as lingeringLink

   Oabvalidate reports:

CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

GUID=8a6efacc-bc38-4431-b577-2b3207f90155>

- member
- LingeringLink
- CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com
  - GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7

8. Collect repadmin /showattr and repadmin /showobjmeta data for this (and all other objects) reported in the problemattributes.txt files

   Sample data collection for this object and attribute value:

```
repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)"
/deleted /atts:member /long /allvalues /gc >obj_8a6efacc-bc38-4431-b577-2b3207f90155.txt

repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked  >>obj_8a6efacc-
bc38-4431-b577-2b3207f90155.txt

repadmin /showattr * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /filter:"(objectclass=*)"
/deleted /long /allvalues /gc >0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt

repadmin /showobjmeta * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /linked >>0974a6d0-8a75-
4f9b-bb83-be236c1e43f7.txt
```

   A batch file that collects this data is located in the **c:\Files** directory.

   repadmin_cmds.bat

9. Review group membership differences for object **LLGroup1**. This data is collected in the repadmin_cmds.bat file: **obj_8a6efacc-bc38-4431-b577-2b3207f90155.txt**

   DCs in the child domain host a writable copy of this object. ChildDC1 is the authoritative source for this object since the only other DC in the Child domain is an RODC.

   DC1, DC2 and **TRDC1** list four users in the member attribute in **LLGroup1**. **ChildDC1** only reports two users in this group.

DC1.root.contoso.com	ChildDC1.child.root.contoso.com

CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com
CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com;
CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com;
CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com;
CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com
	CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com
	CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com;
	CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com;

DC2 and TRDC1 list the same membership as DC1	ChildDC2 reports a lot more members than ChildDC1

10. Review the replication metadata for these objects.

a. Look at the object's attribute values on DCs containing a writable copy of the object and compare them to GCs.

Several of the members for these group objects do not exist on the DCs that hosts a writable copy of the partition.

Brackish Waters - 0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt

```
Repadmin: running command /showattr against full DC ChildDC1.child.root.contoso.com

Can not locate the object for this DN: <GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>

Error: An LDAP lookup operation failed with the following error:

   LDAP Error 32(0x20): No Such Object
   Server Win32 Error 8333(0x208d): Directory object not found.
   Extended Information: 0000208D: NameErr: DSID-03100213, problem 2001 (NO_OBJECT), data 0, best match of:
```

The object is not present on ChildDC1. However, it is present on DC1:

```
Repadmin: running command /showattr against full DC DC1.root.contoso.com

DN: CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com

   4> objectClass: top;
        person;
        organizationalPerson;
        user
   1> cn: Brackish Waters
   1> sn: Waters
   1> givenName: Brackish
   1> distinguishedName: CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com
   1> instanceType: 0x0 = (  )
   1> whenCreated: 5/10/2013 4:36:04 AM Pacific Daylight Time
   1> whenChanged: 5/10/2013 4:36:04 AM Pacific Daylight Time
   1> displayName: Brackish Waters
   1> uSNCreated: 221388
   2> memberOf: CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com;
        CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com
   1> uSNChanged: 221388
   1> name: Brackish Waters
   1> objectGUID: 0974a6d0-8a75-4f9b-bb83-be236c1e43f7
   1> userAccountControl: 0x200 = ( NORMAL_ACCOUNT )
   1> primaryGroupID: 513 = ( GROUP_RID_USERS )
```

```
1> objectSid: S-1-5-21-2499208487-3024782808-1675863513-1810
1> sAMAccountName: brackw
1> sAMAccountType: 805306368 = ( NORMAL_USER_ACCOUNT )
1> userPrincipalName: brackw@child.root.contoso.com
1> objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
1> dSCorePropagationData: 0x0 = (  )
```

Table 3 Repadmin /showobjmeta output for Brackish Waters

Loc. USN	Originating DSA	Org. USN	Org. Date	Org. Time	Version	Attribute
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	objectClass
221388	fef36435-b9b7-4ab9-afa2-c788ed12354c	221388	2013-05-10	4:36:04	1	cn
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	sn
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	givenName
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	instanceType
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	whenCreated
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	displayName
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	nTSecurityDescriptor
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	name
221388	606f5d34-7202-4073-83fb-aac8bb109868	152549	2013-05-10	4:36:04	4	userAccountControl
221388	606f5d34-7202-4073-83fb-aac8bb109868	152544	2013-05-10	4:36:04	1	primaryGroupID
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	objectSid
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	sAMAccountName
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	sAMAccountType
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	userPrincipalName
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	objectCategory

From repadmin /showobjmeta output in Table 3, we can see that the user Object was created on a DC with this DSAGUID **606f5d34-7202-4073-83fb-aac8bb109868** at 2013-05-10 04:36:04. We will use this information in the next step when we look at the up-to-dateness vector on each DC.

11. Review the repadmin /showutdvec data for each of these partitions and compare with the replication metadata for each of the objects found in the prior step.

    - Next, we use repadmin /showutdvec.

```
repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_child.txt
```

```
Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com


606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152523 @ Time 2013-05-10 04:30:16

a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07

2c96270d-88c7-4a3a-9fa1-46fc01e0605d @ USN    188431 @ Time 2014-06-30 12:44:19

5575185f-49be-4f3c-ba1e-5f5f8c36e9f4 @ USN    208912 @ Time 2014-08-04 06:33:32

505a541f-301d-4497-9fd8-21b111aaaf24 @ USN    212983 @ Time 2014-08-05 14:31:20
```

From this output, we can see that highest change that ChildDC1 received from the replication partner that created the Brackish Waters account is **152523.** However, the USN used by the DC that created the object is higher than the one in the up-to-dateness vector for ChildDC1.

221388	606f5d34-7202-4073-83fb-aac8bb109868	**152543**	2013-05-10	4:36:04	1	whenCreated

Therefore ChildDC1 never received the originating create for this object.

- Showutdvec from other DCs does show that they received this and other changes: 152695 @ Time 2013-05-10 05:05:19

```
Repadmin: running command /showutdvec against full DC DC1.root.contoso.com


9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    220910 @ Time 2013-05-10 03:56:55

606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152695 @ Time 2013-05-10 05:05:19

9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    188760 @ Time 2013-05-10 05:49:14

336d313f-cce1-4c52-a57e-1135d54985fa @ USN    77953 @ Time 2013-05-10 16:04:57

70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN    40967 @ Time 2014-05-09 08:08:38

80afd2de-4153-433a-90ad-995564a80cf0 @ USN    45063 @ Time 2014-05-09 08:12:01

a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN    40966 @ Time 2014-05-09 08:12:07

9a9e8c55-d7d2-4c31-bc04-25abec3765ca @ USN    32774 @ Time 2014-05-09 08:12:13

08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN    45718 @ Time 2014-05-09 08:45:24

fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    258063 @ Time 2014-06-30 10:40:19

c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    180238 @ Time 2014-06-30 11:22:58

f829c787-ca31-415b-97d6-cdc30406a87a @ USN    213004 @ Time 2014-06-30 12:44:11

2c96270d-88c7-4a3a-9fa1-46fc01e0605d @ USN    188431 @ Time 2014-06-30 12:44:19

faf6ee99-63cf-4180-97dd-baf6b901558a @ USN    237581 @ Time 2014-08-04 06:33:17

8c448b6e-949f-441f-999c-36344f52187e @ USN    282640 @ Time 2014-08-04 06:33:18

5575185f-49be-4f3c-ba1e-5f5f8c36e9f4 @ USN    208912 @ Time 2014-08-04 06:33:32

eefc6141-3458-484d-8951-5c392edd7ace @ USN    204815 @ Time 2014-08-04 06:34:04

7c14f75d-73c8-4e24-9657-4bd594cd8f84 @ USN    242278 @ Time 2014-08-05 12:49:18

3e6cc17b-7404-48b5-a717-b754c854edcd @ USN    208915 @ Time 2014-08-05 13:49:17

505a541f-301d-4497-9fd8-21b111aaaf24 @ USN    212981 @ Time 2014-08-05 14:20:23

e9025546-218c-4ccc-bfc6-e2d76364d838 @ USN    287406 @ Time 2014-08-05 14:31:20
```

This is an abandoned object.

Abandoned object	An object created on one DC that never got replicated to other DCs hosting a writable copy of the NC but does get replicated to DCs/GCs hosting a read-only copy of the NC. The originating DC goes offline prior to replicating the	Discovery of this object type is challenging. An easy indicator is destination GCs in strict mode that log 1988s for objects that are R/W in the source DCs partition.

	originating write to other DCs that contain a writable copy of the partition.	• Look at all objects in partition (or to make it not so complicated – just pick a single object) • Look at USN in object's replmetadata for originating create • Look at the Up-to-dateness-Vector in /showutdvec output for object partition on all R/W DCs for Originating DSA GUID reported in #2 • Alert on object where #2 is higher than #3

- Identify the abandoned objects based on the Oabvalidate and replication metadata output.
  - Leverage the consolidated Problem Attributes Excel file.

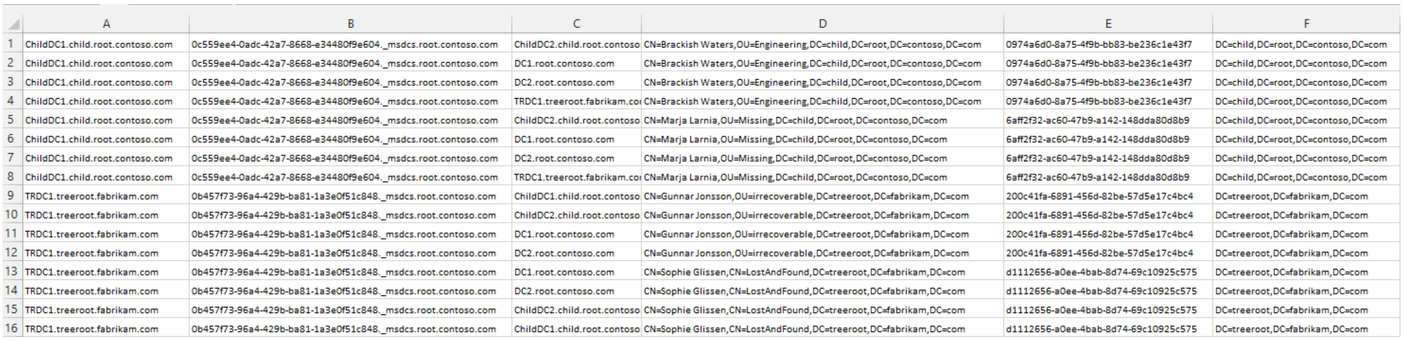
    Abandoned objects can be removed with the LDAP RemoveLingeringObject rootDSE modify procedure. Perhaps the easiest way to do all these objects in bulk is to remove them all from all GCs.

Tip:

To save lab time, the full analysis is done for you. It is documented in the **Data Analysis** tab in the **All_DCs_ProblemAttributes.xlsx** file. The process used for the analysis is detailed in the Abandoned object identification using conditional formatting section in the Appendix.

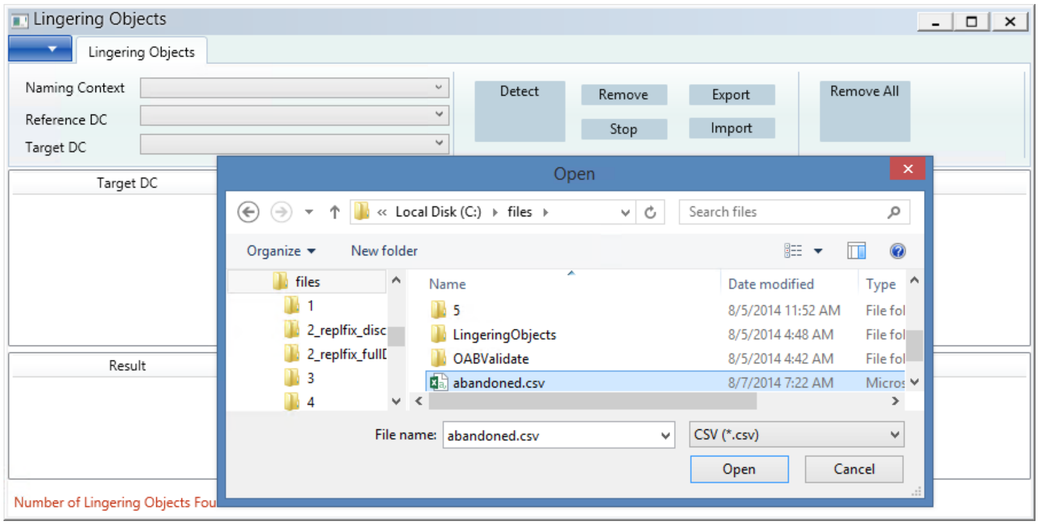
- Create a Lingering Objects tool importable CSV file to make light work of the abandoned object removal. The following format is required for the CSV file:

FQDN of RWDC,CNAME of RWDC,FQDN of DC to remove object from, DN of the object, Object GUID of the object, DN of the object's partition



	A	B	C	D	E	F
1	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	ChildDC2.child.root.contoso	CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	0974a6d0-8a75-4f9b-bb83-be236c1e43f7	DC=child,DC=root,DC=contoso,DC=com
2	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	DC1.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	0974a6d0-8a75-4f9b-bb83-be236c1e43f7	DC=child,DC=root,DC=contoso,DC=com
3	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	DC2.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	0974a6d0-8a75-4f9b-bb83-be236c1e43f7	DC=child,DC=root,DC=contoso,DC=com
4	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	TRDC1.treeroot.fabrikam.co	CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	0974a6d0-8a75-4f9b-bb83-be236c1e43f7	DC=child,DC=root,DC=contoso,DC=com
5	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	ChildDC2.child.root.contoso	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=contoso,DC=com	6aff2f32-ac60-47b9-a142-148dda80d8b9	DC=child,DC=root,DC=contoso,DC=com
6	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	DC1.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=contoso,DC=com	6aff2f32-ac60-47b9-a142-148dda80d8b9	DC=child,DC=root,DC=contoso,DC=com
7	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	DC2.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=contoso,DC=com	6aff2f32-ac60-47b9-a142-148dda80d8b9	DC=child,DC=root,DC=contoso,DC=com
8	ChildDC1.child.root.contoso.com	0c559ee4-0adc-42a7-8668-e34480f9e604._msdcs.root.contoso.com	TRDC1.treeroot.fabrikam.co	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=contoso,DC=com	6aff2f32-ac60-47b9-a142-148dda80d8b9	DC=child,DC=root,DC=contoso,DC=com
9	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	ChildDC1.child.root.contoso	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,DC=fabrikam,DC=com	200c41fa-6891-456d-82be-57d5e17c4bc4	DC=treeroot,DC=fabrikam,DC=com
10	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	ChildDC2.child.root.contoso	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,DC=fabrikam,DC=com	200c41fa-6891-456d-82be-57d5e17c4bc4	DC=treeroot,DC=fabrikam,DC=com
11	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	DC1.root.contoso.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,DC=fabrikam,DC=com	200c41fa-6891-456d-82be-57d5e17c4bc4	DC=treeroot,DC=fabrikam,DC=com
12	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	DC2.root.contoso.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,DC=fabrikam,DC=com	200c41fa-6891-456d-82be-57d5e17c4bc4	DC=treeroot,DC=fabrikam,DC=com
13	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	DC1.root.contoso.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,DC=fabrikam,DC=com	d1112656-a0ee-4bab-8d74-69c10925c575	DC=treeroot,DC=fabrikam,DC=com
14	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	DC2.root.contoso.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,DC=fabrikam,DC=com	d1112656-a0ee-4bab-8d74-69c10925c575	DC=treeroot,DC=fabrikam,DC=com
15	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	ChildDC2.child.root.contoso	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,DC=fabrikam,DC=com	d1112656-a0ee-4bab-8d74-69c10925c575	DC=treeroot,DC=fabrikam,DC=com
16	TRDC1.treeroot.fabrikam.com	0b457f73-96a4-429b-ba81-1a3e0f51c848._msdcs.root.contoso.com	ChildDC1.child.root.contoso	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,DC=fabrikam,DC=com	d1112656-a0ee-4bab-8d74-69c10925c575	DC=treeroot,DC=fabrikam,DC=com

  - You can also leverage one that has been created for you in the C:\files directory: **abandoned.csv**
  - Once you have the file, open the **Lingering Objects** tool and select the **Import** button, browse to the file and choose Open.



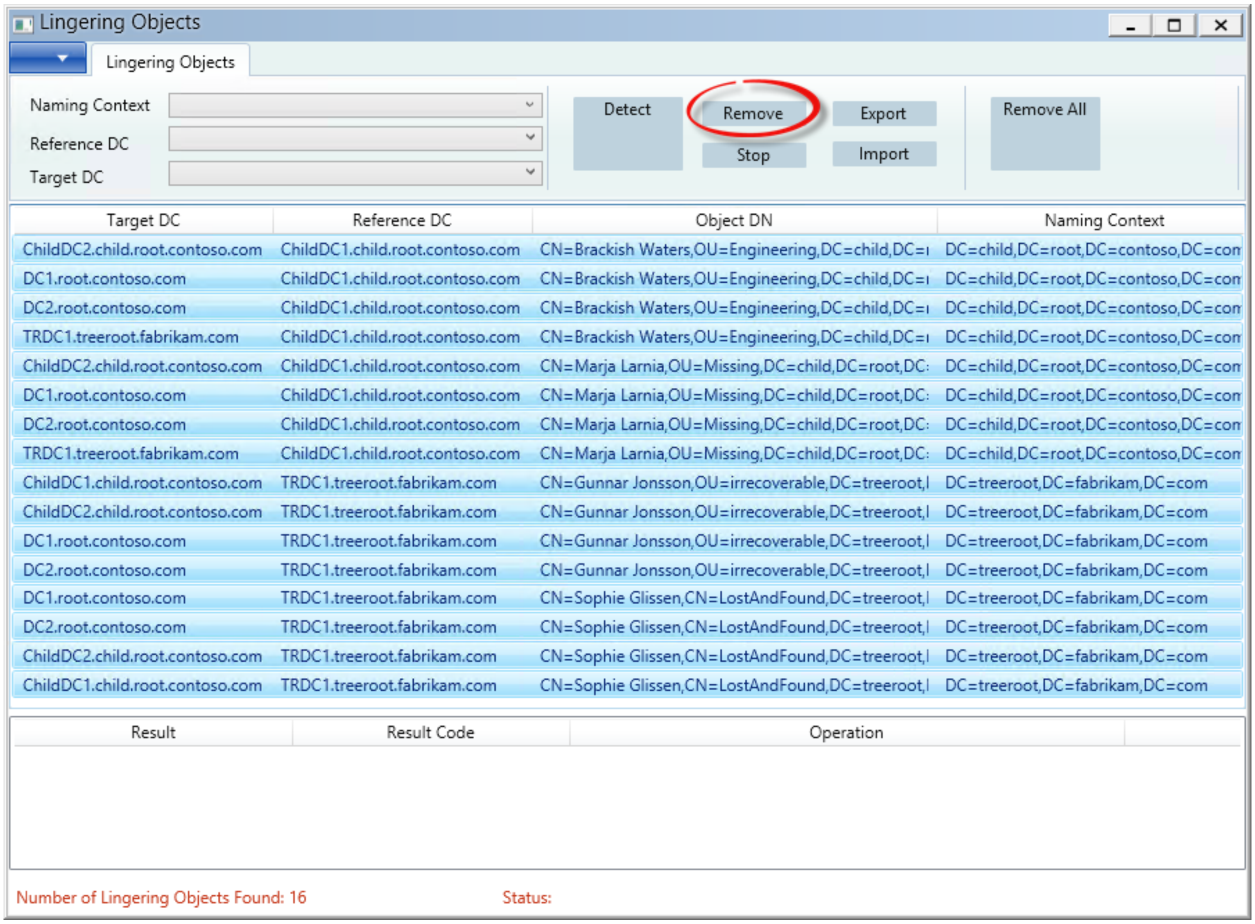
Name	Date modified	Type
5	8/5/2014 11:52 AM	File fol
LingeringObjects	8/5/2014 4:48 AM	File fol
OABValidate	8/5/2014 4:42 AM	File fol
abandoned.csv	8/7/2014 7:22 AM	Micros

File name: abandoned.csv

CSV (*.csv)

Number of Lingering Objects Fou

- Select all objects and then choose **Remove**.



Target DC	Reference DC	Object DN	Naming Context
ChildDC2.child.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=r	DC=child,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=r	DC=child,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=r	DC=child,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	ChildDC1.child.root.contoso.com	CN=Brackish Waters,OU=Engineering,DC=child,DC=r	DC=child,DC=root,DC=contoso,DC=com
ChildDC2.child.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=	DC=child,DC=root,DC=contoso,DC=com
DC1.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=	DC=child,DC=root,DC=contoso,DC=com
DC2.root.contoso.com	ChildDC1.child.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=	DC=child,DC=root,DC=contoso,DC=com
TRDC1.treeroot.fabrikam.com	ChildDC1.child.root.contoso.com	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=	DC=child,DC=root,DC=contoso,DC=com
ChildDC1.child.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
ChildDC2.child.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
DC1.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
DC2.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
DC1.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
DC2.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
ChildDC2.child.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com
ChildDC1.child.root.contoso.com	TRDC1.treeroot.fabrikam.com	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,I	DC=treeroot,DC=fabrikam,DC=com

Result	Result Code	Operation

Number of Lingering Objects Found: 16

Status:

Review replication metadata to verify the objects were removed.

What impact does this have on the group membership issues for the same objects?

All issues related to these four objects are cleared up except one: Brackish Waters is still listed as a member on Oabvalidate output from childdc1.

Next we will deal with the membership issues for the objects that are still present in AD.

This is one of the easier scenarios to correct because you can simply add the user back to the group and then remove them again.

CN=Tabatha Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com

CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com

Are lingering links in this group:

CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

```
Set-ADGroup -Add:@{'Member'="CN=Tabatha Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com", "CN=Juliette
Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com", "CN=Ulysses
Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -
Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -
Remove:@{'Member'="CN=Tabatha Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com", "CN=Juliette
Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com", "CN=Ulysses
Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com"} -Server:"TRDC1.treeroot.fabrikam.com"
```

CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com

Chase is a lingering link in the following two groups:

CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com;

CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com

```
Set-ADGroup -Add:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -Identity:"
CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com
```

CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com

Becker is a lingering link in the following groups:

CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com is a lingering link in LingeringLinkGroup2, LingeringLinkGroup3, LingeringLinkGroup4 and LingeringLinkGroup5 but not listed in Oabvalidate output

Built in groups show as lingering objects on DC1 and DC2's Oabvalidate output

To correct group membership for objects that no longer exist you need to create new objects with the same extended DN as the phantom members and then delete them again

1. Review membership enable extended DN control
2. Copy GUID out of the member attribute (from extended DN in member attribute) for all missing members
3. Convert the objectGUID portion to base64
4. Create LDIFDE importable files to add users
5. Change DSHeuristics to allow ObjectGUID specify on add
6. Import ldifde
7. Remove objects

Note:

The files needed to fix group membership for the absent users have already been created for you to save lab time.

1. Import FixAbsent.ldf

   ldifde -i -f c:\files\FixAbsent.ldf -s trdc1.treeroot.fabrikam.com

2. Import DeletaAbsentUsers.ldf

   ldifde -i -f c:\files\deleteAbsentusers.ldf -s trdc1.treeroot.fabrikam.com

(Not required) Exercise 5: Troubleshoot and resolve AD replication error 8614

8614 | The directory service cannot replicate with this server because the time since the last replication with this server has exceeded the tombstone lifetime.

Important: This exercise is needed only if error 8614 is logged in showrepl or adreplstatus output.

Error 8614 is logged when a destination DC has not replicated with a source DC over an existing replication connection for longer than tombstone lifetime.

Warning:

- This quarantine is put in place on a per-replica, per-partition basis so that replication with an out of date DC does not introduce lingering objects into the environment.
- If this issue occurs in a production environment, careful consideration should be made prior to removing the replication safeguard.
- In some cases, forceful demotion of the source DC makes more sense. See the content linked in the appendix for more information.
- Large jumps in system time (forward or backward) are common causes of this issue

In this exercise, you will use repadmin to resolve AD replication error 8614 in a supported manner.

Perform this exercise from **Win8Client**.

1. Run the AD Replication Status tool or repadmin /showrepl * /csv. Review the output. **If AD replication error 8614 is not present, then do not do this exercise.**

2. Ensure Strict Replication consistency is set on all DCs

   ```
   Repadmin /regkey * +strict
   ```

   In the output of the above command, verify status for all DCs: registry key set

   ```
   "Strict Replication Consistency" REG_DWORD 0x0000001 (1)
   ```

3. Remove lingering objects if present using repldiag (skip if already performed in exercise 4).

   ```
   Repldiag /removelingeringobjects
   ```

4. Run the following command on destination DCs that fail to replicate from source DCs with error 8614: (replace *DestinationDCName* with the actual DC name)

🚫 **Do Not:**

Do not run the following command without first verifying that Strict replication consistency is enabled.

```
Repadmin /regkey DestinationDCName +AllowDivergent
```

In this lab environment, it is safe to just temporarily set the registry value on all DCs

```
Repadmin /regkey * +AllowDivergent
```

Verify status from all DCs:

```
"Allow Replication With Divergent and Corrupt Partner" REG_DWORD 0x0000001 (1)
```

5. Initiate replication to all destination DCs from all source DCs where replication failed with status 8614
6. Use repadmin /showrepl * /csv or the AD Replication Status tool to verify error 8614 is no longer logged in the environment
7. Delete the registry value so that the replication quarantine safeguards are back in place

```
Repadmin /regkey * -AllowDivergent
```

Appendix

Exercise 1

Answers

How can you translate the alias provided in the event to the host name of the DC?

1. Copy the alias out of the event (highlight and Ctrl + C)
2. Ping 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6._msdcs.root.contoso.com

Other options include:

- Look at the SRV record in the forest root MSDCS DNS zone (_msdcs.root.contoso.com) in the DNS Management snap-in
- Output repadmin /showrepl * to a text file and match up the GUID reported in the event to the DSA object GUID.
- Use an LDAP query tool (such as Repadmin or PowerShell) to dump the ObjectGUID of the NTDS Settings object:

Command Prompt:

```
Repadmin /showattr DC1 "<GUID=3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6>" /atts:DN
```

Return all DSA objectGUIDs

```
Repadmin /showattr DC1 NCOBJ:Config: /filter:"(Objectclass=NTDSDSA)" /atts:objectGUID /subtree
```

PowerShell:

```
PS C:\>Get-ADObject -Identity 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
```

Return all DSA objectGUIDs

```
PS C:\>Get-ADObject -LDAPFilter "(Objectclass=ntdsdsa)" -SearchBase
"cn=configuration,dc=root,dc=contoso,dc=com" | Out-GridView
```

Is DC2 configured for Strict or Loose Replication Consistency?

Strict replication consistency

What event is logged on the destination DC when there is an attempt to send changes for a lingering object when strict replication consistency is enabled?

Event ID 1988 is logged in the Directory Service event log

What event is logged on the destination DC when there is an attempt to send changes for a lingering object when loose replication consistency is enabled?

Event ID 1388 is logged in the Directory Service event log

Which DCs return replication metadata for the object?

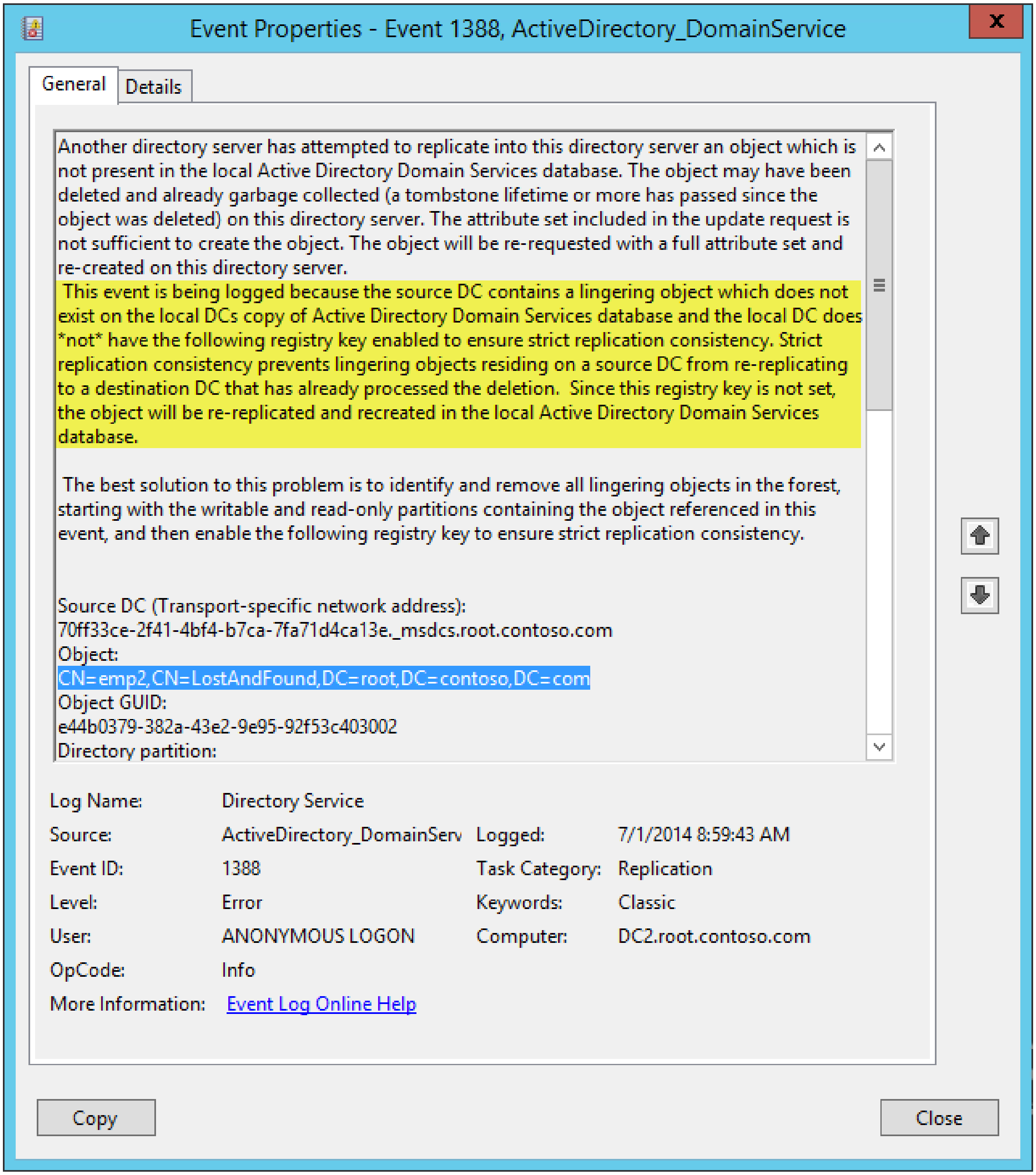
All DCs except for **DC2** return replication metadata for the object. **DC1, ChildDC1, ChildDC2** and **TRDC1** have this lingering object.

Lingering Object symptoms

Event or Error status	Event or error text	Implication
AD Replication status **8606**	"Insufficient attributes were given to create an object. This object may not exist because it may have been deleted."	Lingering objects are present on the source DC (destination DC is operating in Strict Replication Consistency mode)
AD Replication status **8614**	The directory service cannot replicate with this server because the time since the last replication with this server has exceeded the tombstone lifetime.	Lingering objects likely exist in the environment
AD Replication status 8240	There is no such object on the server	Lingering object may exist on the source DC
Directory Service **event ID 1988**	Active Directory Domain Services Replication encountered the existence of objects in the following partition that have been deleted from the local domain controllers (DCs) Active Directory Domain Services database.	Lingering objects exist on the source DC specified in the event (Destination DC is running with Strict Replication Consistency)
Directory Service **event ID 1388**	This destination system received an update for an object that should have been present locally but was not.	Lingering objects were reanimated on the DC logging the event Destination DC is running with Loose Replication Consistency
Directory Service **event ID 2042**	It has been too long since this server last replicated with the named source server.	Lingering object may exist on the source DC

Loose replication consistency

Event 1388



Event Properties - Event 1388, ActiveDirectory_DomainService
General
Details

Another directory server has attempted to replicate into this directory server an object which is not present in the local Active Directory Domain Services database. The object may have been deleted and already garbage collected (a tombstone lifetime or more has passed since the object was deleted) on this directory server. The attribute set included in the update request is not sufficient to create the object. The object will be re-requested with a full attribute set and re-created on this directory server.

This event is being logged because the source DC contains a lingering object which does not exist on the local DCs copy of Active Directory Domain Services database and the local DC does *not* have the following registry key enabled to ensure strict replication consistency. Strict replication consistency prevents lingering objects residing on a source DC from re-replicating to a destination DC that has already processed the deletion. Since this registry key is not set, the object will be re-replicated and recreated in the local Active Directory Domain Services database.

The best solution to this problem is to identify and remove all lingering objects in the forest, starting with the writable and read-only partitions containing the object referenced in this event, and then enable the following registry key to ensure strict replication consistency.

Source DC (Transport-specific network address):
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com
Object:
CN=emp2,CN=LostAndFound,DC=root,DC=contoso,DC=com
Object GUID:
e44b0379-382a-43e2-9e95-92f53c403002
Directory partition:

Log Name:	Directory Service		
Source:	ActiveDirectory_DomainServ	Logged:	7/1/2014 8:59:43 AM
Event ID:	1388	Task Category:	Replication
Level:	Error	Keywords:	Classic
User:	ANONYMOUS LOGON	Computer:	DC2.root.contoso.com
OpCode:	Info		
More Information:	Event Log Online Help		

Advisory Mode

The DRSReplicaVerifyObjects method allows for a parameter to be passed that reports each lingering object in the event log (event 1946) without actually removing it. Event ID 1942 is logged as a summary event containing the count of lingering objects on the server.

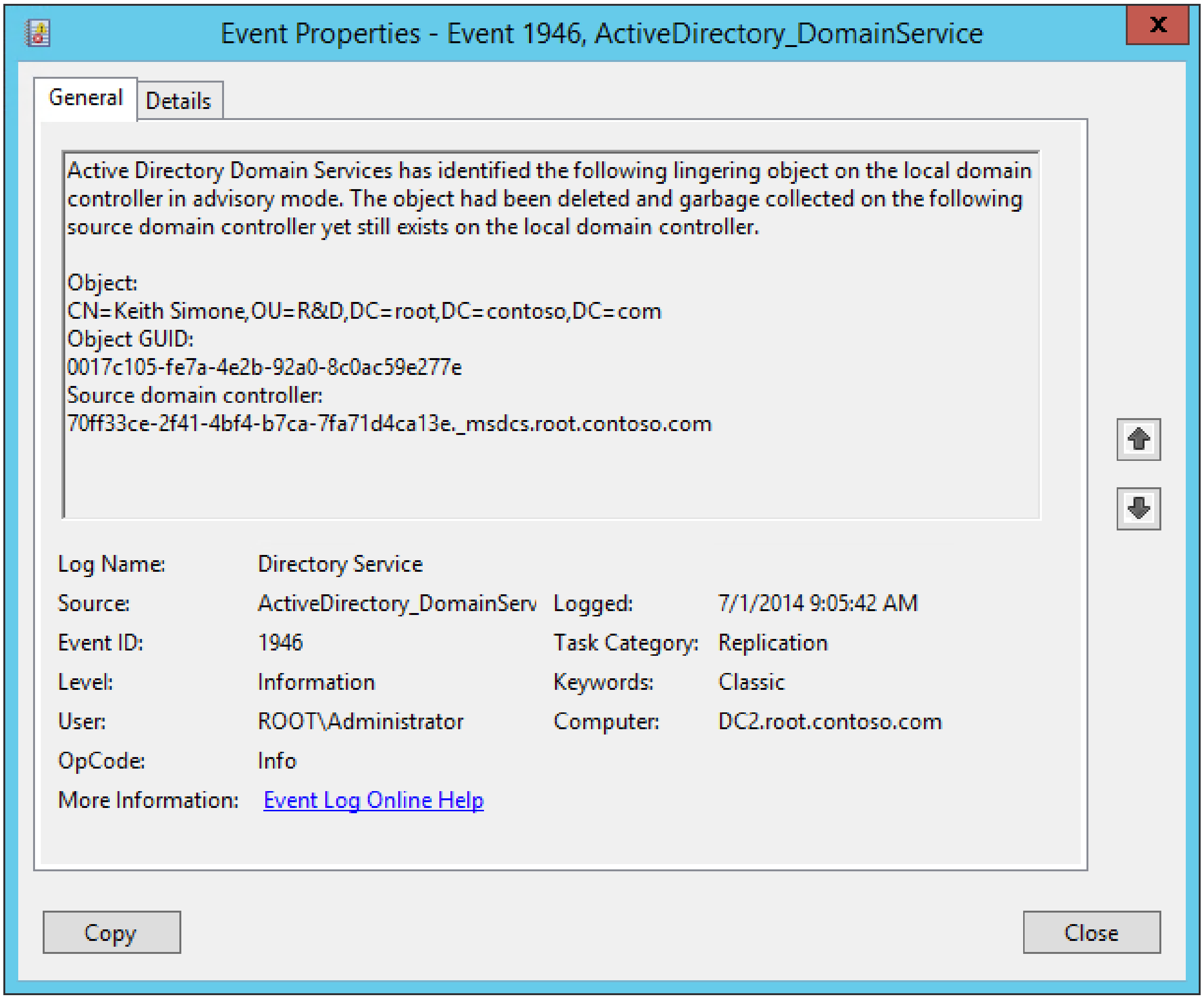
Tool	Parameter to run in Advisory Mode
Repadmin /removelingeringobjects	/Advisory_Mode
Repldiag /removelingeringobjects	/AdvisoryMode
Lingering Objects.exe	Click the **Discover** button

Event 1946

One event ID 1946 per lingering object is logged in the Directory Service event log on the checked DC. This event indicates the presence of a lingering object on the local DC where the event is logged.

In the message text:

- Object DN and Object GUID of the lingering object
- Source DC DNS CNAME that was used as a reference DC (This DC does not have the lingering object)


Event Properties - Event 1946, ActiveDirectory_DomainService
General
Details
Copy
Close

Active Directory Domain Services has identified the following lingering object on the local domain controller in advisory mode. The object had been deleted and garbage collected on the following source domain controller yet still exists on the local domain controller.

Object:
CN=Keith Simone,OU=R&D,DC=root,DC=contoso,DC=com
Object GUID:
0017c105-fe7a-4e2b-92a0-8c0ac59e277e
Source domain controller:
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com

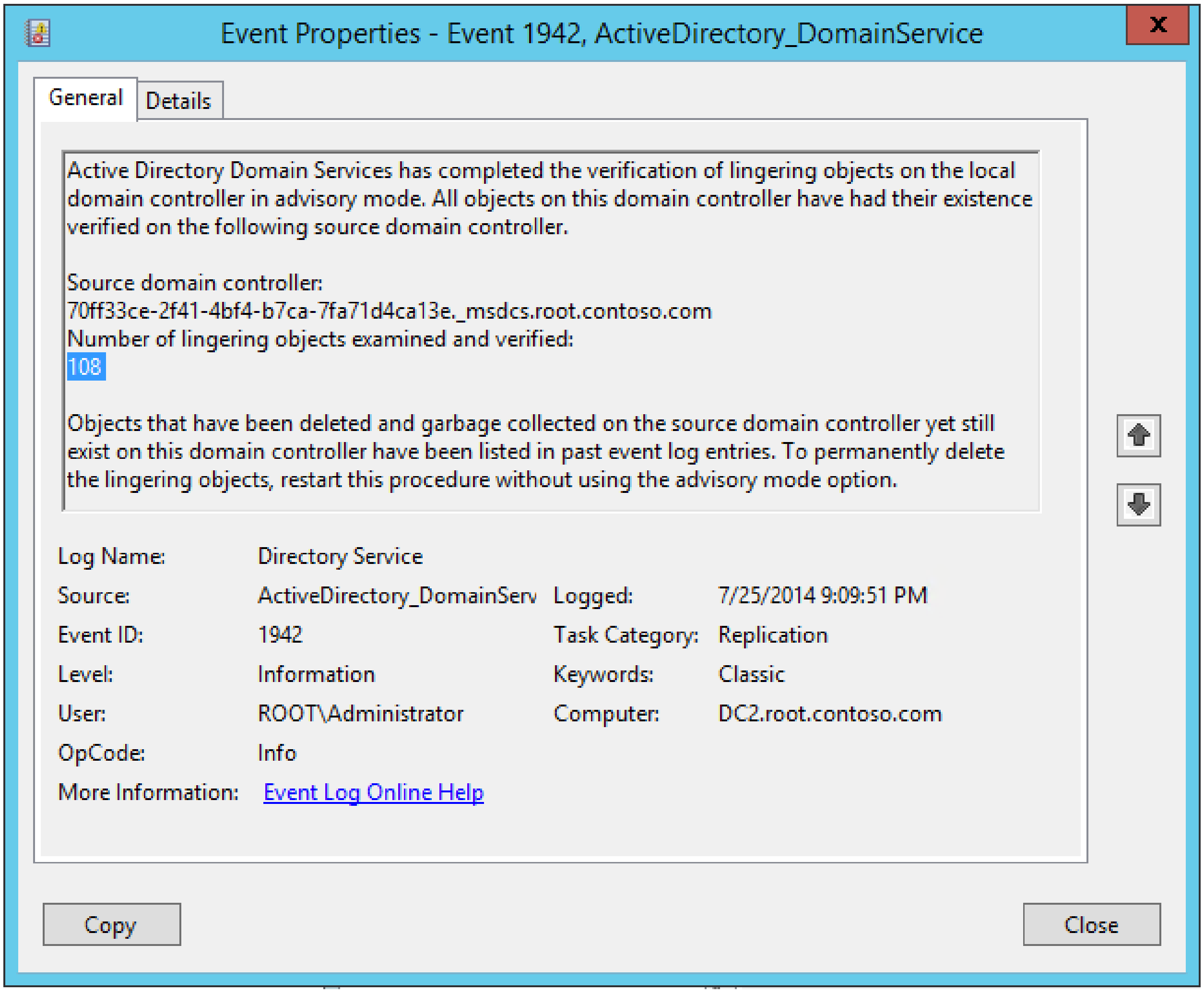
Log Name:	Directory Service		
Source:	ActiveDirectory_DomainServ	Logged:	7/1/2014 9:05:42 AM
Event ID:	1946	Task Category:	Replication
Level:	Information	Keywords:	Classic
User:	ROOT\Administrator	Computer:	DC2.root.contoso.com
OpCode:	Info		
More Information:	Event Log Online Help		

Event 1942

One event ID 1942 per Advisory Mode run is logged in the Directory Service event log on the DC where Advisory Mode was targeted. This summary event gives the total count of lingering objects present on the local DC where the event is logged.

In the message text:

- Number of lingering objects present on the local DC
- Source DC DNS CNAME that was used as a reference DC (This DC does not have the lingering object)


Event Properties - Event 1942, ActiveDirectory_DomainService
General
Details
Copy
Close

Active Directory Domain Services has completed the verification of lingering objects on the local domain controller in advisory mode. All objects on this domain controller have had their existence verified on the following source domain controller.

Source domain controller:
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e._msdcs.root.contoso.com
Number of lingering objects examined and verified:
108

Objects that have been deleted and garbage collected on the source domain controller yet still exist on this domain controller have been listed in past event log entries. To permanently delete the lingering objects, restart this procedure without using the advisory mode option.

Log Name:	Directory Service		
Source:	ActiveDirectory_DomainServ	Logged:	7/25/2014 9:09:51 PM
Event ID:	1942	Task Category:	Replication
Level:	Information	Keywords:	Classic
User:	ROOT\Administrator	Computer:	DC2.root.contoso.com
OpCode:	Info		
More Information:	Event Log Online Help		

Lingering Object Job Aid

Lingering Object Glossary

Table 4 Lingering Object glossary

Term	Description	Notes
Abandoned delete	An object deleted on one DC that was never replicated to other DCs hosting a writable copy of the NC for that object. The deletion replicates to DCs/GCs hosting a read-only copy of the NC. The DC that originated the object deletion goes offline prior to replicating the change to other DCs hosting a writable copy of the partition.	Symptoms: GCs report source DCs have lingering objects in source DC partition: Root.contoso.com: DC1 and DC2 Child.root.contoso.com: ChildDC1 ChildDC1 replicates Root partition from DC1 and replication fails with error 8606
Abandoned object	An object created on one DC that never got replicated to other DCs hosting a writable copy	

	of the NC but does get replicated to DCs/GCs hosting a read-only copy of the NC. The originating DC goes offline prior to replicating the originating write to other DCs that contain a writable copy of the partition.	
Lingering link	A linked attribute contains the DN of an object that no longer exists in Active Directory. These stale references are referred to as lingering links.	
Lingering Object	An object that is present on one replica, but has been deleted and garbage collected on another replica.	

Loose Replication Consistency	With this behavior enabled, if a destination DC receives a change to an attribute for an object that it does not have, the entire object is replicated to the target for the sake of replication consistency. This undesirable behavior causes a lingering object to be "reanimated."	Warning: This setting will cause the undesirable behavior of reanimation of lingering objects.
		Event 1388 is logged in the DS event log of the destination DC when a source DC replicates changes for a lingering object
		For all domain controllers, type:
		`repadmin /regkey * -strict`
		For all global catalog servers, type:
		`repadmin /regkey gc: -strict`

Strict Replication Consistency	With this behavior enabled, if a destination DC receives a change to an attribute for an object that it does not have, replication is blocked with the source DC for the partition where the lingering object was detected. Event 1988 is logged in the Director Services event log on the destination DC and AD replication error status 8606 is logged for the last replication failure status message (visible in repadmin /showrepl output).	• Defines how a destination DC behaves if a source DC sends updates to an object that does not exist in the destination DC's local copy of Active Directory. • Destination DCs should see USN for creates before object is modified • Only modifies for lingering objects arrive for object not on destination DC • Only destination DC's enforce strict replication and log events • Destination DCs stop replicating from source DC's partitions containing LO's • Lingering objects are quarantined on source DCs where they can be detected • End-to-end replication may be impacted for partitions containing lingering objects • Administrators must remove lingering objects to restore replication

		- For all domain controllers, type:
		`repadmin /regkey * +strict`
		For all global catalog servers, type:
		`repadmin /regkey gc: +strict`

Tombstone	An object that has been deleted but not yet garbage collected This object is retained in the database for the tombstone lifetime so that other DCs have an opportunity to learn of the object's deletion	
Tombstone Lifetime (TSL)	The amount of time tombstones are retained in Active Directory before being garbage collected and permanently purged from the database.	

Deleted object	When AD recycle bin is enabled, an object that is deleted (deleted object) is recoverable with a full set of attributes using a PowerShell command (2008 R2) or via PowerShell and a GUI- based tool (ADAC) in Windows Server 2012). The object remains in this state until the **deleted object lifetime** expires and then it becomes a **recycled object**.	IsDeleted = True
		IsRecycled = <not set>
		Stored in the **Deleted Objects** container in most instances (some objects do not get moved on deletion).

Deleted object lifetime	The deleted object lifetime is determined by the value of the **msDS-deletedObjectLifetime** attribute. - By default, tombstoneLifetime is set to null. When tombstoneLifetime is set to null, the tombstone lifetime defaults to 60 days (hard-coded in the system). - By default, msDS-deletedObjectLifetime is also set to null. When msDS-deletedObjectLifetime is set to null, the deleted object lifetime is set to the value of the tombstone lifetime. If msDS-deletedObjectLifetime is manually set, it becomes the effective lifetime of a system state backup.	CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=<mydomain>,DC=<com> Attribute: **msDS-deletedObjectLifetime**
Garbage Collection	A process that permanently deletes tombstone objects or recycled objects - runs on DCs every 12 hours by default / 15 minutes after restart	`Repadmin /setattr "" "" doGarbageCollection add 1"`

	Can be manually initiated with LDP, LDIFDE or other LDAP tool	

Recycled object	After a deleted object lifetime expires, the logically deleted object is turned into a recycled object and most of its attributes are stripped away.	IsDeleted = True
		IsRecycled = True
		Can only be recovered if *toggle recycled objects* flag is used during the authoritative restore process.

Tombstone	Generically, this is an object that has been deleted but not garbage collected. Prior to the introduction of the AD recycle bin, this is the term for a deleted object.
	If AD recycle bin is enabled:
	An object that is deleted retains all of its attribute values and does not become a recycled object until the **deleted object lifetime** expires.
	If AD recycle bin is not enabled:
	A deleted object immediately becomes a tombstone and is stripped of most attribute values.
	To recover a tombstone with a full set of attributes, you must perform an authoritative restore.
		If AD recycle bin is not enabled:
		IsDeleted = True
		IsRecycled = True
		If AD recycle bin is enabled and the object is within the deleted object lifetime:
		IsDeleted=True
		IsRecycled=not set
		If AD recycle bin is enabled and the object is now a recycled object:
		IsDeleted=True
		IsRecycled=True

Tombstone Lifetime (TSL)	The number of days before tombstones or recycled objects are eligible for garbage collection. By default, tombstoneLifetime is set to null. When tombstoneLifetime is set to null, the tombstone lifetime defaults to 60 days (hard-coded in the system). This is also the effective lifetime of a system state backup. If msDS-deletedObjectLifetime is manually set, it becomes the effective lifetime of a system state backup.	CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC =<mydomain>,DC=<com> Attribute: tombstoneLifetime

Replication Consistency Settings

Section by - Jasmin Hashmani

Strict Replication Consistency

- Defines how a destination DC behaves if a source DC sends updates to an object that does not exist in the destination DC's local copy of Active Directory.
    - Destination DCs should see USN for creates before object is modified
    - Only modifies for lingering objects arrive for object not on destination DC
    - Only destination DC's enforce strict replication and log events
- Destination DCs stop replicating from source DC's partitions containing LO's
- Lingering objects are quarantined on source DCs where they can be detected
- End-to-end replication may be impacted for partitions containing lingering objects
- Administrators must remove lingering objects to restore replication

Enabling Strict Replication

Use Repadmin from Window Server 2003 SP1 or later to set strict replication via command prompt:

- For all domain controllers, type:
  `repadmin /regkey * +strict`
- For all global catalog servers, type:
  `repadmin /regkey gc: +strict`

You can also enable strict replication by manually setting the **Strict Replication Consistency** registry value to **1**.

```
Key: HKLM\System\CurrentControlSet\Services\NTDS\Parameter

Value: Strict Replication Consistency

Type: (Reg_DWORD)

Value Data: 1


1(enabled): Inbound replication of the specified directory partition from the source is stopped on
the destination.
```

Warning:

Ensure you are prepared to deal with replication failures after enabling strict replication consistency due to the existence of lingering objects.

Loose Replication Consistency

If you enable Loose Replication Consistency, if a destination receives a change to an object that it does not have, the entire object is replicated to the target for the sake of replication consistency. This behavior causes a lingering object to be reapplied to all domain controllers in the replication topology.

Enable Loose Replication

Use Repadmin (from Window Server 2003 SP1 or later) to set strict replication via command prompt:

- For all domain controllers, type:
  `repadmin /regkey * -strict`
- For all global catalog servers, type:
  `repadmin /regkey gc: -strict`

You can also enable strict replication by manually setting the **Strict Replication Consistency** registry value to **0**.

```
Key: HKLM\System\CurrentControlSet\Services\NTDS\Parameters
Value: Strict Replication Consistency
Type: (Reg_DWORD)
Value Data: 0

0 (disabled): The destination requests the full object from the source domain
controller, and the lingering object is revived in the directory.
```

Critical: The Loose Replication Consistency setting will cause the undesirable behavior of reanimation of lingering objects.

Default Settings for Strict Replication Consistency

The default value for the strict replication consistency registry entry is determined by the conditions under which the domain controller was installed into the forest.

Note: Raising the domain or forest functional level does not change the replication consistency setting on any domain controller.

Upgrade Path	Default	Notes
Windows NT 4.0	Loose	
Windows 2000 RTM Root	Loose	A post-SP2 NTDSA.DLL defaulted to strict replication consistency but was quickly recalled. Windows 2000 Services 1 through 4 all default to loose replication consistency.
Windows NT 4.0 to Windows 2000 Root	Loose	
Windows 2000 to Windows Server 2003 SP1	Loose	Upgrading a Windows 2000 forest to Windows Server

		2003 slipstreamed with SP1 does not enabled strict replication consistency.
Windows Server 2003 RTM Root	Strict	DCPROMO creates an operational GUID that causes Windows Server 2003 domain controllers to inherit strict replication mode but is ignored by Windows 2000 domain controllers.
Windows Server 2003 SP1 root and later: **Windows Server 2003 R2** **Windows Server 2008** **Windows Server 2008 R2** **Windows Server 2012** **Windows Server 2012 R2**	Strict	Same as above.
Windows NT 4.0 to Windows Server 2003 root	Strict	DCPROMO creates an operational GUID that causes Windows Server 2003 domain controllers to inherit strict replication mode but is ignored by Windows 2000 domain controllers.

More Information:

For more information about this topic, see:

http://blogs.technet.com/b/askds/archive/2010/02/15/strict-replication-consistency-myth-versus-reality.aspx

Repadmin RLO example usage

The command's syntax is:

repadmin /removelingeringobjects *LingeringDC ReferenceDC_DSA_GUID Partition*

Where:
LingeringDC: FQDN of DC that has the lingering objects
ReferenceDC_DSA_GUID: The DSA GUID of a domain controller that hosts a writeable copy of the partition
Partition: The distinguished name of the directory partition where the lingering objects exist

So for example:

We have a server named **DC1.contoso.com** that contains lingering objects. We know that the lingering object is in the **childdomain.contoso.com** partition. We know that **DC3.childdomain.contoso.com** hosts a writeable copy of the partition and doesn't contain any lingering objects.

We need to find the DSA GUID of DC3 is, so we run: repadmin /showrepl DC3.childdomain.contoso.com
At the top of the output, locate the DC Object GUID entry. This is the GUID you need to enter in the command for the reference DC.

The command would be

```
repadmin /removelingeringobjects DC1.contoso.com  5ed02b33-a6ab-4576-b109-
bb688221e6e3  dc=childdomain,dc=contoso,dc=com
```

Repldiag quick reference

Removing lingering objects from a forest with repldiag is as simple as running repldiag /removelingeringobjects. However, it is usually best to exercise some control over the process in larger environments. The option /OverRideReferenceDC allows you to select which DC to use for cleanup. The option /outputrepadmincommandlinesyntax allows you to see what a forest-wide cleanup looks like using repadmin.

Repldiag /removelingeringobjects /outputrepadmincommandlinesyntax

This will give you output of corresponding repadmin /removelingeringobjects syntax. View the output to get an understanding of the steps repldiag uses holistically remove lingering objects

1. It first selects one DC per partition to use as a reference DC.

From the developer:

> **Reference DC selection:**
> "It is based on the DC with the highest number of link objects on a per partition basis. The assumption is that this is a hub/well connected system. This may also select a multiple "reference" DCs according to each partition." - Ken Brumfield

2. It then cleans the reference DCs up against all other DCs for the partition(s) they were selected as a reference for.
3. Finally, it cleans up all other DCs in the forest with the new "cleaned up" reference DCs as sources.

The /outputrepadmincommandlinesyntax option does not actually attempt object cleanup. You would need to leave this option off if you want to execute lingering object cleanup.

Sample Repldiag /removelingeringobjects /outputrepadmincommandlinesyntax output

```
Number Complete,Status,Server Name,Naming Context,Reference DC,Duration,Error Code,Error Message
repadmin /removelingeringobjects loncontosodc.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com 87ccb4f8-1057-4cfa-aed6-79b5626db9fd
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com 4009aef6-b279-43d2-82f6-4298f02505e8
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com b3ff6e2e-6025-4782-9d7b-54b0431a374a
dc=forestdnszones,dc=contoso,dc=com
```

```
repadmin /removelingeringobjects loncontosodc.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com 87ccb4f8-1057-4cfa-aed6-79b5626db9fd
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com 4009aef6-b279-43d2-82f6-4298f02505e8
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com b3ff6e2e-6025-4782-9d7b-54b0431a374a
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com 87ccb4f8-1057-4cfa-aed6-79b5626db9fd
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com 4009aef6-b279-43d2-82f6-4298f02505e8
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com b3ff6e2e-6025-4782-9d7b-54b0431a374a
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com 87ccb4f8-1057-4cfa-aed6-79b5626db9fd
dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com 4009aef6-b279-43d2-82f6-4298f02505e8
dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com b3ff6e2e-6025-4782-9d7b-54b0431a374a
dc=corp,dc=contoso,dc=com
Reference NCs cleaned in 0h:0m:0s.  Cleaning everything else against reference NCs.
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects dalcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects nycorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects seacorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=forestdnszones,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects dalcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects nycorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects seacorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
cn=configuration,dc=contoso,dc=com
repadmin /removelingeringobjects 5thwardcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=contoso,dc=com
repadmin /removelingeringobjects dalcorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=contoso,dc=com
repadmin /removelingeringobjects nycorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=contoso,dc=com
repadmin /removelingeringobjects seacorpdc.corp.contoso.com a29bbfda-8425-4cb9-9c66-8e07d505a5c6
dc=contoso,dc=com
repadmin /removelingeringobjects dalcorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects nycorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects seacorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=domaindnszones,dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects loncontosodc.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects dalcorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=corp,dc=contoso,dc=com
repadmin /removelingeringobjects nycorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=corp,dc=contoso,dc=com
```

```
repadmin /removelingeringobjects seacorpdc.corp.contoso.com 9653cb84-7aa2-4a59-ab46-382e5dc1d3a8
dc=corp,dc=contoso,dc=com
All NCs cleaned in 0h:0m:0s.
```

This output can also be viewed in Excel: Copy commands to a text file. Modify the text file to include only the command portion of the output. Then open up the text file in Excel. (space delimited)

From the developer:

Does the /outputrepadmincommandlinesyntax exactly mirror the internal operation of repldiag when it performs the lingering object removals?

"Short answer = yes.

Long answer:

The key is that the read/write authoritative reference must be cleaned by comparing to all the other r/w references. Then everything can be done in parallel against the authoritative reference.

Repldiag is multi-threaded and runs one management thread per NC to create the clean authoritative reference, and then spawns multiple threads to clean against the authoritative reference. So different NCs may complete at different rates depending on number of r/w partitions (in addition to normal factors such as network latency and bandwidth).

As such, both they syntax and native functionality respect the need to serially clean the authoritative reference and then everything else after. In terms of actual order beyond that, there is none of significance to worry about.

In summary, yes the output order is the same as the syntax. Excluding the multi-threading considerations.

The code logic is essentially:

```
f (!isOutputSyntax)
     DsVerifyReplica(...)
Else
     Console.Write line(...)
```

W/console.write line handling the thread synchronization for the output." - Ken Brumfield

More control: /OverRideReferenceDC

This option allows you to specify a DC that you want to be used as a reference DC for the partition specified. In a large distributed environment, take careful consideration when choosing the reference DC. Things to consider when choosing a suitable reference DC:

- Well connected: Fast WAN link.
- Performance: Excellent server class hardware: Disk, RAM, CPU and NIC
- Critical Network Applications / Services do not depend on this DC: Such as an Exchange facing DC

- Other DCs don't report replication failures with reference DC as the source: filter repadmin /showrepl * /csv ouput, or use the topology report created by repldiag /save.

```
repldiag /removelingeringobjects
/overridedefaultreferencedc:"cn=configuration,dc=contoso,dc=com":nycorpdc.corp.contoso.com
/overridedefaultreferencedc:"dc=corp,dc=contoso,dc=com":seacorpdc.corp.contoso.com
/overridedefaultreferencedc:"dc=forestdnszones,dc=contoso,dc=com":5thwardcorpdc.corp.contoso.com
/outputrepadmincommandlinesyntax

Replication topology analyzer.  Written by kenbrumf@microsoft.com
Version:  2.0.3397.24022
Command Line Switch:  /removelingeringobjects
Command Line Switch:  /overridedefaultreferencedc:cn=configuration,dc=contoso,dc=com:nycorpdc.corp.contoso.com
Command Line Switch:  /overridedefaultreferencedc:dc=corp,dc=contoso,dc=com:seacorpdc.corp.contoso.com
Command Line Switch:
/overridedefaultreferencedc:dc=forestdnszones,dc=contoso,dc=com:5thwardcorpdc.corp.contoso.com
Command Line Switch:  /outputrepadmincommandlinesyntax

Attempting to override NC cn=configuration,dc=contoso,dc=com with DC nycorpdc.corp.contoso.com...      Overriden
Attempting to override NC dc=corp,dc=contoso,dc=com with DC seacorpdc.corp.contoso.com...      Overriden
Attempting to override NC dc=forestdnszones,dc=contoso,dc=com with DC 5thwardcorpdc.corp.contoso.com...
Overriden
```

/UseRobustDCLocation

Query every DC for a list of DCs in the forest. This ensures replication instability does not cause any DCs to be missed. We have had cases where we clean up lingering objects in the forest but due to an AD topology problem, some DCs were not cleaned up. This option is usually recommended if you want it to do a thorough job.

Lingering Links

Attributes on user or group objects contain references to the following items:

- Unresolvable Distinguished Names (DN): The DN in the attribute points to an object that is not present in the directory.

  For example:

  - Attribute values contain DNs that have been DEL mangled.
  - Attribute values contain DNs that point to an object that was removed from AD DS. But references to that object were never cleaned up.

The scenario in which objects are removed from AD DS but not cleaned up is also known as one of the following:

- Lingering Links
- Lingering Linked Values

More specifically, Single- and Multi-valued linked attributes, such as **Manager** on a user account or **Member** on a group object, contain stale references to objects that are no longer present in AD DS. Such stale references can occur on many attributes and object classes. As of today, this problem most commonly occurs on the following objects and attributes.

Object Class	**Attributes**
Group	Member
User	Manager

Complete Attribute list that may contain stale references for an Exchange OABGen failure scenario

altRecipient
altRecipientBL
assistant
authOrigBL
bridgeheadServerListBL
defaultClassStore
directReports
distinguishedName
dLMemRejectPermsBL
dLMemSubmitPermsBL
dynamicLDAPServer
homeMDB
homeMTA
isPrivilegeHolder
kMServer
lastKnownParent
managedObjects
manager
masteredBy
member
memberOf
msExchConferenceMailboxBL
msExchControllingZone
msExchIMVirtualServer
msExchQueryBaseDN
msExchUseOAB
netbootSCPBL
nonSecurityMemberBL
ownerBL
preferredOU
publicDelegates
publicDelegatesBL
queryPolicyBL
secretary
seeAlso
serverReferenceBL
showInAddressBook
siteObjectBL
unAuthOrigBL

The lack of end-to-end replication of directory partitions defined in the forest within a rolling tombstone lifetime number of days or time jumps which prematurely purge knowledge of deletes before end-to-end replication can result in AD database divergence amongst DCs. Such long term conditions can cause Lingering Objects. Lingering objects are very common and can cause this problem. However, there are other potential causes of “bad data” in Active Directory that are often confused with Lingering Objects. These are lesser-known and do not show up in a check for lingering objects (when running repadmin /removelingeringobjects).

Other potential causes of invalid data in AD:

Root Cause	**Description**
Lingering link	A linked attribute contains the DN of an object that no longer exists in Active Directory. These stale references are referred to as lingering links.
Abandoned object	An object created on one DC that never got replicated to other DCs hosting a writable copy of the NC but does get replicated to DCs/GCs hosting a read-only copy of the NC. The originating DC goes offline prior to replicating the originating write to other DCs that contain a writable copy of the partition.
Abandoned delete	An object deleted on one DC that never got replicated to other DCs hosting a writable copy of the NC for that object. The deletion replicates to DCs/GCs hosting a read-only copy of the NC. The DC that originated the object deletion goes offline prior to replicating the change to other DCs hosting a writable copy of the partition.

Resolution

High-level overview:

There are two major problems to contend with that can lead to considerable time to resolution:

Problem 1: Identify all objects and/or attributes containing bad data that would cause oabgen to fail.

Problem 2: If lingering objects were identified, then proceed with lingering object removal. However, if the identification phase reveals lingering links, proceed with Attribute cleanup.

This stale data may exist on objects residing in read-only Global Catalogs, on DCs with writable copies of a directory partition or both.

Once the attributes causing Oabgen to fail have been identified, your first goal should be to vet the validity and consistency of attribute values on forward link across all replicas hosting writable copies of the objects home directory partition. Then you focus on DCs hosting a read-only copy of the NC.

Workflow

1. Identify all attributes on all objects that contain stale references causing oabgen to fail
2. Determine whether any DC hosting a writable copy of the NC for the object also contains attributes with invalid references
    - If they do, then delete the bad reference (DN) from the attribute
    - If the DCs that are writable for this object do not contain the invalid references and they only exist on DCs hosting a read-only copy of the partition, then additional steps are required
3. Verify that your infrastructure master is not a global catalog server (unless all DCs are GCs).
4. Verify that DCs containing the invalid references are able to successfully replicate from a DC hosting a writable copy of the NC.
5. If replication is successful then move on to one of the proposed workarounds in the Attribute Cleanup section

Identification

If Exchange is installed in the environment, MSExchange event 9339 reports one object leading to the problem. However, the problem is usually much more wide-spread than this. The challenge here is to identify all users/groups containing invalid references that will lead to the errors.

Potential identification mechanisms:

OABValidateThis is the best tool to use when the problem is wide-spread. This tool was enhanced to address this specific problem.

CSVDE or **LDIFDE** export of the group and then look for DEL mangled references (DEL mangled references are only one example of bad data, so this is usually not a good method of identification).

LDP *dumpdatabase* (Microsoft support assistance may be required).

In some cases **oabvalidate** will fail to identify a problematic attribute. You may be able to identify the attribute with an LDP database dump of ntds.dit:

Use LDP to dump the database with the dumpdatabase command. Find the Distinguished Name Tag (DNT) of the object reported in the event. Look at the BDNTs for this object. Go to

the DNT entry for each BDNT and identify any that have a value of False.
A script that parses the text from the database dump would make this an easier task.

Script Logic:

1. Look for Object value of False (Object is a phantom and not present in the DB)
2. CNT = Reference count CNT > 0 (means someone still references this phantom)
3. Look at BDNT (Backlink DNT) -ignore Deleted Objects container
4. Create object hierarchy using DNT and PDNT stopping at DNT 2 (root object)
5. List all objects that meet these conditions. List all objects that reference these objects.
6. Report Name and ObjectGUID of both in CSV importable format.
7. Use repadmin /showattr * and / or repadmin /showobjmeta * to report data for the object. Compare differences.

Attribute Cleanup

Workaround until cleanup can be performed:

- Continue to use Exchange 2003 or Exchange 2007 mailbox server for OAL generation.

Determine whether any writable DCs contain objects with attributes containing invalid references.
Search all DCs by object DN or objectGUID. Repadmin /showobjmeta can be used for issues with group membership, otherwise use repadmin /showattr:

- Repadmin /showattr * "<GUID=*ObjectGUID*>" /atts /allvalues /gc /long >attr.txt
- Repadmin /showobj * "<GUID=*ObjectGUID*>" >objmeta.txt

If there is a single DC hosting a writable copy of the partition where the object exists with improper attribute references, then cleanup may be as simple as:

- Delete or clear the invalid reference on this DC and outbound replicate the changes.

However, if the problem only exists on the GCs hosting a read-only copy of the partition where the groups exist, then there is quite a bit of work to do:
There is no easy resolution to this problem. The following are viable workarounds and each has its own pros and cons. Review the following four methods and the table below to help you choose the best solution for your environment.

Method 1: Delete and recreate

Delete the object. Verify that the object no longer exists on all DCs. Recreate the object and repopulate attribute values. If the objects are security principals, then the object will have a new SID with this method. If objects or files are permissioned with the old SID then this method is not desirable.

Method 2: Delete and restore with an Authoritative Restore

Delete the objects. Verify that the objects no longer exist on all DCs. Perform an authoritative restore of the objects on a DC that hasn't processed the deletion.
Objects are completely restored to the state that exists on the recovery DC. This method also restores backlinks (i.e. where a group was a member of another group).

Note If the DCs are running Windows Server 2003, then they will all most likely need to be patched with a QFE version of ntdsa.dll before implementing recovery procedures. The recovery DC will need an updated version of ntdsutil.exe.

1. Use LDP to obtain the following for each affected object: ObjectGUID and Distinguished Name
2. Use repadmin to generate replication metadata for an object on all DCs
   Repadmin /showobjmeta * "DNofObject" >c:\ALLDCsmetab4deletion.txt
3. Identify and prepare a recovery DC

   Verify object and valid attribute values exist on a DC hosting a writable copy of the partition.

   Use repadmin to disable inbound replication and then boot this DC into DC Restore Mode. (or stop the Active Directory Domain Services service on Server 2008 or later)
4. Delete the object on another DC hosting a writable copy of the NC
5. Allow end-to-end replication of the deletion to take place
6. Verify object's removal with repadmin /showobjmeta *

   To verify the objects no longer exist on the GCs:
   repadmin /showobjmeta * "DNofObject" >c:\ALLDCsmetaAfterdeletion.txt
   * All DCs that host the partition the object was in should report status 8333 "Directory Object Not Found"
   * All DCs that don't host the partition will report status 8439 "The distinguished name specified for this replication operation is invalid"
   * If metadata is returned you must wait until all DCs process the deletion
   * If a different status code is returned you will need to investigate on a per DC basis

7. Perform an authoritative restore of the object(s) on the DC that is booted into DS Restore mode
8. Boot the recovery DC into normal mode and allow replication of the changes to occur
9. Import any ldifde files that were created as part of the authoritative restore process
10. Re-enable inbound replication on the recovery DC

Method 3: Delete and restore with adrestore.exe
SID is retained but most attributes will have to be repopulated. If backlinks are present and need to be restored then a Microsoft internal utility may need to be used prior to object deletion. (Microsoft Commercial Technical Support assistance may be required)
Method 4: Global Re-host
Un-host the partition from all GCs in the forest simultaneously. Re-host from DCs hosting a writable copy of the partition where the objects exist.
The following un-host and re-host procedures will need to be performed on all DCs that contain a read-only copy of the partition in the forest. Failing to cleanup even one GC in the environment can cause the problem to recur in the environment after the cleanup steps have been performed

1. Verify that all DCs that host a writable copy of the NC have valid attribute values for the affected objects
2. Repadmin /unhost DSA <Naming Context>

3. Verify that no other GCs host the partition prior to re-hosting the partition. There should be an event ID 1660 logged in the Directory Services event log on every DC where the partition was un-hosted.

   Event ID 1658 is the status event logged in the Directory Services event log to indicate how many objects still need to be removed before the partition is completely removed. Event ID 1660 is logged in the Directory Services event log when the partition has been successfully removed from the database.

4. Repadmin /options <DSA> +disable_ntdsconn_xlate
5. Repadmin /add <Naming Context> <Dest DSA> <Source DSA> /readonly
6. Repadmin /replicate <Dest DSA> <Source DSA> <Naming Context>
7. Repadmin /options <DSA> -disable_ntdsconn_xlate

Alternatively you could do the following:

1. Verify that all DCs that host a writable copy of the NC have valid attribute values for the affected objects
2. Disable outbound replication on all DCs that host a read-only copy of the partition
3. Run the following on each of these DCs
4. Repadmin /rehost DSA <Naming Context> <Good Source DSA Address>
5. Verify the issue has been resolved on each DC using repadmin /showobjmeta or repadmin /showattr
6. Re-enable outbound replication on all DCs that host a read-only copy of the partition.

There are multiple ways to resolve this problem. The following table lists both valid and invalid ways to resolve the issue. Invalid methods are displayed so that time is not wasted performing them.

Invalid attribute value exists on a writable copy of the NC	Action	Pro	Con
	Remove just the invalid attribute values from the attribute in question from a DC hosting a writable copy of the NC	This is the preferred solution. If this is an option, then performing this step should also resolve the issue on DCs hosting a read-only copy of the partition.	This will only work if the bad data exist on an attribute for an object contained on a DC hosting a writable copy of the partition. (not in a GC's read-only copy of the partition)
Invalid attribute value exists only on a read-only copy of the NC	**Action**	**Pro**	**Con**
	Check for and remove lingering objects	Easy step to implement if the problem is caused by lingering objects	Will not clean up all conditions including abandoned objects and lingering linked values.

		(check with /advisory_mode first)	Requires you to be in strict mode. If a GC considers an abandoned object, strict mode does not block inbound replication of abandoned objects.
	Initiate a full replication cycle using repadmin with a known good source (you will need to create a replication connection using repadmin /add if one doesn't already exist then run: repadmin /replicate destinationDC sourceDCFQDN PartitionDN /readonly /full)	Easy step to implement. If this does not correct the attribute data then a rehost or object deletion may be required)	This command may take a very long time to complete if the partition in question contains a large amount of objects
	Unhost and rehost the partition from a known good source	Ensures GC hosts a valid copy of the partition. Good solution to the problem in small environments or where data divergence is limited to a few DCs.	Is challenging and time-consuming in a large environment with this method as it may require all GCs to be cleaned up at the same time. (and it may be necessary to disable outbound replication on the same GCs during the duration of the cleanup procedure as it may be possible for a "clean" GC to re-replicate bad data from a "dirty" GC.
	Delete the object from a DC containing a writable copy of the NC	Easy solution where the problem is isolated to attribute values on a single object	Depending on the object type, this solution many additional problems
	Delete and then authoritatively restore the object on a DC containing a writable copy of the NC. 1. prior to object deletion: Verify object and valid attribute values exist on a secondary DC and then boot this DC into DS Restore Mode.	This will resolve the problem as long as you correctly identified all objects containing attributes with invalid data. LDIFDE files will be created automatically during the authoritative restore that will aid in complete recovery of forward-link / back-link pairs.	There is down-time associated with this while the objects are in their deleted state. This may require you to install several QFEs on the recovery DC and replica DCs to update ntdsa.dll and ntdsutil.exe

	2. Delete the object on another DC hosting a writable copy of the NC. 3. Allow end-to-end replication of the deletion to take place. 4. Verify object's removal with repadmin /showobjmeta * 5. Perform an authoritative restore of the object(s) on the DC that is booted into DS Restore mode)		
	Delete the object and then use adrestore.exe to un-delete the object from a DC containing a writable copy of the NC. Then re-populate attribute values using ldifde.	This will resolve the problem as long as you correctly identified all objects containing attributes with invalid data.	There is down-time associated with this while the objects are in their deleted state. This action requires a good export of the object. In the case where groups are nested, you would also need an export of that groups membership to correct backlinks. (groupadd.exe can help with this part)
	Replfix solution documented in KB 914024	The solution provided in 914024 does not resolve this issue.	This solution was created for one specific customer and this fails to resolve the problem
	repadmin /replsingleobj		Only works if both source and destination DC host a writable copy of the partition
	NULL out the attribute values on the object from a DC hosting a writable copy of the NC		This will not remove lingering link values if the Forest Functional Level is 2003 or later (as Link -value replication (LVR) will be enabled)

More Information

Sample experience with issue caused by Lingering-linked values:

An Active Directory forest consists of root domain Contoso.com with child domain corp.Contoso.com, grandchild domain na.corp.contoso.com and tree domain fabrikam.com. A universal group (which could also be a distribution or security enabled group) is created in the contoso.com domain and the membership consists of

```
contoso.com\adam
corp.contoso.com\john
na.corp.contoso.com\kim
fabrikam.com\gary
```

Viewing the member attribute for the universal group shows 4 members. The fabrikam.com domain gets force demoted and the user object na.corp.contoso.com\kim is deleted from the na.corp.contoso.com domain, at a time when end-to-end replication does not take place for TSL number of days. On GCs hosting a read-only copy of the NC, the member attribute of the universal group continues to show 4 members in the group when only two of the 4 listed members, contoso.com\adam and corp.contoso.com\john are valid.

Note the sample problem above involves users added to groups in the domain partition but the problem themselves exists for both single and mult-valued attributes on objects in any writable domain partition.
Group object DN: CN=FailBoatDL,OU=Groups,DC=contoso,DC=com
Attribute:member
DNs referenced in Attribute: (Group membership)
Object exist in this NC (naming context / domain): contoso.com

```
cn=adam,cn=users,dc=contoso,dc=com
cn=john,cn=users,dc=corp,dc=contoso,dc=com
cn=kim,cn=users,dc=na,dc=corp,dc=contoso,dc=com
cn=gary,cn=users,dc=fabrikam,dc=com
```

After domain deletion and the deletion of another user object:

```
Group membership on DCs hosting a writable copy of the NC:
cn=adam,cn=users,dc=contoso,dc=com
cn=john,cn=users,dc=corp,dc=contoso,dc=com

Group membership on DCs hosting a read-only copy of the NC:
cn=adam,cn=users,dc=contoso,dc=com
cn=john,cn=users,dc=corp,dc=contoso,dc=com
cn=kim,cn=users,dc=na,dc=corp,dc=contoso,dc=com
cn=gary,cn=users,dc=fabrikam,dc=com
```

Repadmin /removelingeringobjects

Removing Lingering Objects with Repadmin

Repadmin includes an advanced switch (view using /experthelp) to remove lingering objects from a specific server.

To remove outdated (lingering) objects from a directory partition on a domain controller that has not replicated for a tombstone lifetime, perform the following.

1. Using Repadmin, type the following at the command line:

```
Repadmin /RemoveLingeringObjects DestinationDC SourceDC_Guid
DirectoryPartition (Optional switch /advisory_mode)
```

Where:

- **DestinationDC** is the DNS name of the DC to remove lingering objects from
- **SourceDC_Guid** is the DSA objectGUID of the DC to use as a reference

  To obtain the Source DC's DSA objectGUID, do one of the following.

  - Use Repadmin /showrepl *SourceDCName*. The domain controller’s object GUID is listed as “domain controller object GUID.”

OR

  - In Active Directory Sites and Services, find the Source domain controller under Sites\<the domain controller's Site>\ Servers\ DCname\ NTDS Settings\ Properties. Look in the DNS Alias box. The GUID prior to _msdcs.forestrootname.com is the domain controllers Object GUID. Repadmin only needs the GUID. Omit _msdcs.forestrootname.com from the Repadmin syntax.

- **DirectoryPartition** is the distinguished name of the directory partition from which to remove outdated objects.

2. Repeat the procedure for the following partitions, as needed.
   - Domain directory partition
     dc=DomainName,dc=ForestRootDomainName
     DC=root,DC=Contoso,DC=com
   - Configuration directory partition
     cn=configuration,dc=DomainName,dc=ForestRootDomainName
     DC=root,DC=Contoso,DC=com
   - Application directory partition or partitions
     cn=ApplicationDirectoryPartitionName,dc=DomainName,dc=ForestRootDomainName

   The following is an example of the command syntax.

```
C:\>repadmin /removelingeringobjects 5thwarddc.child.contoso.com B0AE6093-
15F5-4DB8-836B-4495F3B19493 dc=contoso,dc=com /advisory_mode

RemoveLingeringObjects successful on 5thwarddc.child.contoso.com
```

Events Associated with Lingering Object Removal

When removing lingering objects, the target domain controller (the domain controller with the lingering objects) will record all removal information, including source domain controller, objects removed, and a total count of all objects removed.

- **Event ID 1937**: NTDS Replication. Lingering Object Removal has been initiated on this domain controller. All objects on this DC will have their existence verified on the following source domain controller. Objects that have been deleted and garbage collected from the source domain controller will be DELETED from this domain controller if they still exist. Subsequent event logs will list all deleted objects.

```
Source DC: <source DC guid ._msdcs.<forest root>
```

- **Event ID 1945**: NTDS Replication. Lingering Object Removal will DELETE the following object. Its deletion and garbage collection was detected on the source domain controller without replicating the deletion to this domain controller.

```
Object:DC= <dn of lingering object>
Object GUID:<objectGUID>
Source DC: <dc guid> ._msdcs.<forest root>
```

- **Event ID 1939**: NTDS Replication. Lingering Object Removal has executed successfully on this domain controller. All objects on this domain controller have had their existence verified on the source domain controller. Objects that had been deleted and garbage collected from the source domain controller were DELETED from this domain controller. Previous event logs list all such objects.

```
Source DC: <source DC guid> ._msdcs.<forest root>
Lingering Objects Deleted 23
```

RemoveLingeringObjects: How it Works

From *How the Active Directory Replication Model Works*

When you run **repadmin /removelingeringobjects** , the tool performs the following steps to compare the directories of the source and destination domain controllers and log (or remove) any found lingering objects:

1. Check to ensure that the directory partition and the source domain controller are valid.

2. Verify that the user has the DS-Replication-Manage-Topology extended right on the directory partition container object specified in <NC>. This extended right is required to verify object state between two domain controllers. Members of the Domain Admins group have this right by default.

3. Ensure that both source and destination use the same objects for comparison by merging the up-to-dateness vectors to filter out any objects that have not replicated from the source to the destination or from the destination to the source. This check rules out a lingering object on the destination if the destination has not received the tombstone from the source, and vice versa. Any such nonreplicated objects are removed from the comparison.

4. Create the list of object GUIDs for each domain controller to be compared. Examine the metadata of each object and use the merged up-to-dateness vector to determine whether the object should be present on both source and destination.

5. For each GUID that is in the list for the destination, determine if it is in the list of GUIDs for the source.

6. If a GUID is not found on the source, the object is identified to be outdated on the destination and is either displayed or deleted on the destination server. If advisory mode has been specified, the GUID is displayed only."

Lingering objects in the deleted objects container

There are two classic scenarios here; one of which requires no action and the other definitely need to be dealt with.

Some background: (Assuming the AD recycle bin is NOT enabled)

- Objects that are deleted become tombstones
- They still contain attributes that can be modified

- If there are attribute value changes for a tombstone, then that change still needs to replicate
- These objects eventually get removed via an independent task that runs on each DC, known as Garbage Collection
  - An object becomes eligible for garbage collection once its deleted time meets the tombstone lifetime
  - The DC will remove the objects eligible for garbage collection once the next time the garbage collection task runs (every 12 hours by default, or 15 minutes after reboot - can also be triggered manually via an LDAP rootDSE modification
  - Since garbage collection is an independent task that runs on each DC: not all DCs remove these tombstones at the same time
    - you have up to a 12 hour window where some DCs have the tombstones and some that have already removed them

Here are two scenarios:

Transient lingering objects - See MSKB *2002034* for an example

1. A change is made to an attribute on a deleted object that is at the cusp of being eligible for garbage collection
   a. This change replicates to other DCs that have already garbage collected the object
   b. The destination DCs are operating in Strict replication consistency mode
      i. Replication is blocked with the source DC until the destination DC garbage collects the object

Standard lingering objects in the deleted objects container

1. A change is made to an attribute on a deleted object
   a. This change replicates to other DCs
   b. For a number of different reasons, some DCs do not have the object. The destination DCs are operating in Strict replication consistency mode
      i. Replication is blocked with AD replication status 8606 from the source DC until the destination DC garbage collects the object (up to TSL # of days) - (Event 1988 is logged on the destination DC)
      ii. If this condition is allowed to persist beyond TSL # of days, replication is blocked with AD replication status 8614 (event ID 2042 is logged on the destination DC)

Exercise 3

Replfix

ldifde_replfixCMDs.bat

```
rem "Root partition"

Ldifde -f dc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com

Ldifde -f dc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com

Ldifde -f trdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com -t 3268

Ldifde -f childdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com -t 3268

Ldifde -f childdc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com -t 3268

rem "Child partition"

Ldifde -f childdc1_child.ldf -d "dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com

Ldifde -f childdc2_child.ldf -d "dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com

Ldifde -f dc1_child.ldf -d "dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com -t 3268

Ldifde -f dc2_child.ldf -d "dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com -t 3268

Ldifde -f trdc1_child.ldf -d "dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com -t 3268

rem "TreeRoot partition"

Ldifde -f trdc1_treeroot.ldf -d "dc=treeroot,dc=fabrikam,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com

Ldifde -f dc1_treeroot.ldf -d "dc=treeroot,dc=fabrikam,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com -t 3268

Ldifde -f dc2_treeroot.ldf -d "dc=treeroot,dc=fabrikam,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com -t 3268

Ldifde -f childdc1_treeroot.ldf -d "dc=treeroot,dc=fabrikam,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com -t 3268

Ldifde -f childdc2_treeroot.ldf -d "dc=treeroot,dc=fabrikam,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com -t 3268

rem "Config partition"

Ldifde -f dc1_Config.ldf -d "cn=configuration,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com

Ldifde -f dc2_Config.ldf -d "cn=configuration,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com

Ldifde -f trdc1_Config.ldf -d "cn=configuration,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com
```

```
Ldifde -f childdc1_Config.ldf -d "cn=configuration,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com

Ldifde -f childdc2_Config.ldf -d "cn=configuration,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com

rem "ForestDNSZones partition"

Ldifde -f dc1_ForestDNSZones.ldf -d "dc=forestdnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com

Ldifde -f dc2_ForestDNSZones.ldf -d "dc=forestdnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com

Ldifde -f trdc1_ForestDNSZones.ldf -d dc=forestdnszones,"dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com

Ldifde -f childdc1_ForestDNSZones.ldf -d "dc=forestdnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com

Ldifde -f childdc2_ForestDNSZones.ldf -d "dc=forestdnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com

rem "rootDNSZones partition"

Ldifde -f dc1_rootDNSZones.ldf -d "dc=domaindnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com

Ldifde -f dc2_rootDNSZoness.ldf -d "dc=domaindnszones,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com

rem "childDNSZones partition"

Ldifde -f childdc1_childDNSZones.ldf -d "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com

Ldifde -f childdc2_childDNSZones.ldf -d "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l "replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com
```

Replfix_cmds.bat

```
time /t >run.log

echo . >>run.log

echo ######################################## >>run.log

echo "Root Partition" >>run.log

echo "DC1" >>run.log

echo ######################################## >>run.log

replfix dc1_root.ldf dc2_root.ldf -lingering dc1_root_lingering_dc2.ldf dc2_root_lingering_dc1.ldf -log root_dc1_dc2.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_root.ldf childdc1_root.ldf -lingering dc1_root_lingering_childdc1.ldf childdc1_root_lingering_dc1.ldf -log root_dc1_childdc1.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_root.ldf childdc2_root.ldf -lingering dc1_root_lingering_childdc2.ldf childdc2_root_lingering_dc1.ldf -log root_dc1_childdc2.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_root.ldf trdc1_root.ldf -lingering dc1_root_lingering_trdc1.ldf trdc1_root_lingering_dc1.ldf -log root_dc1_trdc1.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ######################################## >>run.log
echo "Root Partition" >>run.log

echo "DC2" >>run.log

echo ####################################### >>run.log

replfix dc2_root.ldf childdc1_root.ldf -lingering dc2_root_lingering_childdc1.ldf childdc1_root_lingering_dc2.ldf -log
root_dc2_childdc1.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_root.ldf childdc2_root.ldf -lingering dc2_root_lingering_childdc2.ldf childdc2_root_lingering_dc2.ldf -log
root_dc2_childdc2.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_root.ldf trdc1_root.ldf -lingering dc2_root_lingering_trdc1.ldf trdc1_root_lingering_dc2.ldf -log root_dc2_trdc1.log -
domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ####################################### >>run.log

echo "Child Partition" >>run.log

echo "ChildDC1" >>run.log

echo ####################################### >>run.log

replfix childdc1_child.ldf dc2_child.ldf -lingering childdc1_child_lingering_dc2.ldf dc2_child_lingering_childdc1.ldf -log
child_childdc1_dc2.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix childdc1_child.ldf dc1_child.ldf -lingering childdc1_child_lingering_dc1.ldf dc1_child_lingering_childdc1.ldf -log
child_childdc1_dc1.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix childdc1_child.ldf childdc2_child.ldf -lingering childdc1_child_lingering_childdc2.ldf childdc2_child_lingering_childdc1.ldf -
log child_childdc1_childdc2.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
>>run.log

replfix childdc1_child.ldf trdc1_child.ldf -lingering childdc1_child_lingering_trdc1.ldf trdc1_child_lingering_childdc1.ldf -log
child_childdc1_trdc1.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ####################################### >>run.log

echo "Child Partition" >>run.log

echo "ChildDC2" >>run.log

echo ####################################### >>run.log

replfix childdc2_child.ldf dc2_child.ldf -lingering childdc2_child_lingering_dc2.ldf dc2_child_lingering_childdc2.ldf -log
child_childdc2_dc2.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix childdc2_child.ldf dc1_child.ldf -lingering childdc2_child_lingering_dc1.ldf dc1_child_lingering_childdc2.ldf -log
child_childdc2_dc1.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix childdc2_child.ldf trdc1_child.ldf -lingering childdc2_child_lingering_trdc1.ldf trdc1_child_lingering_childdc2.ldf -log
child_childdc2_trdc1.log -domaindn "dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ####################################### >>run.log

echo "TreeRoot Partition" >>run.log

echo "TRDC1" >>run.log

echo ####################################### >>run.log

replfix trdc1_treeroot.ldf dc1_treeroot.ldf -lingering trdc1_treeroot_lingering_dc1.ldf dc1_treeroot_lingering.ldf -log
treeroot_trdc1_dc1.log -domaindn "dc=treeroot,dc=fabrikam,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix trdc1_treeroot.ldf dc2_treeroot.ldf -lingering trdc1_treeroot_lingering_dc2.ldf dc2_treeroot_lingering.ldf -log
treeroot_trdc1_dc2.log -domaindn "dc=treeroot,dc=fabrikam,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log
```

```
replfix trdc1_treeroot.ldf childdc1_treeroot.ldf -lingering trdc1_treeroot_lingering_childdc1.ldf childdc1_treeroot_lingering.ldf -log treeroot_trdc1_childdc1.log -domaindn "dc=treeroot,dc=fabrikam,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix trdc1_treeroot.ldf childdc2_treeroot.ldf -lingering trdc1_treeroot_lingering_childdc2.ldf childdc2_treeroot_lingering.ldf -log treeroot_trdc1_childdc2.log -domaindn "dc=treeroot,dc=fabrikam,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ######################################## >>run.log

echo "Configuration Partition" >>run.log

echo ######################################## >>run.log

replfix dc1_config.ldf dc2_config.ldf -lingering dc1_config_lingering_dc2.ldf dc2_config_lingering_dc1.ldf -log config_dc1_dc2.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_config.ldf childdc1_config.ldf -lingering dc1_config_lingering_childdc1.ldf childdc1_config_lingering_dc1.ldf -log config_dc1_childdc1.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_config.ldf trdc1_config.ldf -lingering dc1_config_lingering_trdc1.ldf trdc1_config_lingering_dc1.ldf -log config_dc1_trdc1.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_config.ldf childdc1_config.ldf -lingering dc2_config_lingering_childdc1.ldf childdc1_config_lingering_dc2.ldf -log config_dc2_childdc1.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_config.ldf trdc1_config.ldf -lingering dc2_config_lingering_trdc1.ldf trdc1_config_lingering_dc2.ldf -log config_dc2_trdc1.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix trdc1_config.ldf childdc1_config.ldf -lingering trdc1_config_lingering_childdc1.ldf childdc1_config_lingering_trdc1.ldf -log config_trdc1_childdc1.log -domaindn "cn=configuration,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ######################################## >>run.log

echo "ForestDNSZones Partition" >>run.log

echo ######################################## >>run.log

replfix dc1_forestdnszones.ldf dc2_forestdnszones.ldf -lingering dc1_forestdnszones_lingering_dc2.ldf dc2_forestdnszones_lingering_dc1.ldf -log forestdnszones_dc1_dc2.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_forestdnszones.ldf childdc1_forestdnszones.ldf -lingering dc1_forestdnszones_lingering_childdc1.ldf childdc1_forestdnszones_lingering_dc1.ldf -log forestdnszones_dc1_childdc1.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc1_forestdnszones.ldf trdc1_forestdnszones.ldf -lingering dc1_forestdnszones_lingering_trdc1.ldf trdc1_forestdnszones_lingering_dc1.ldf -log forestdnszones_dc1_trdc1.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_forestdnszones.ldf childdc1_forestdnszones.ldf -lingering dc2_forestdnszones_lingering_childdc1.ldf childdc1_forestdnszones_lingering_dc2.ldf -log forestdnszones_dc2_childdc1.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix dc2_forestdnszones.ldf trdc1_forestdnszones.ldf -lingering dc2_forestdnszones_lingering_trdc1.ldf trdc1_forestdnszones_lingering_dc2.ldf -log forestdnszones_dc2_trdc1.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

replfix trdc1_forestdnszones.ldf childdc1_forestdnszones.ldf -lingering trdc1_forestdnszones_lingering_childdc1.ldf childdc1_forestdnszones_lingering_trdc1.ldf -log forestdnszones_trdc1_childdc1.log -domaindn "dc=forestdnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log
```

```
echo . >>run.log

echo ######################################### >>run.log

echo "Root DomainDNSZones partition" >>run.log

echo ######################################### >>run.log

replfix dc1_rootdnszones.ldf dc2_rootdnszones.ldf -lingering dc1_rootdnszones_lingering_dc2.ldf
dc2_rootdnszones_lingering_dc1.ldf -log rootdnszones_dc1_dc2.log -domaindn
"dc=domaindnszones,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo . >>run.log

echo ######################################### >>run.log

echo "Child DomainDNSZones partition" >>run.log

echo ######################################### >>run.log

replfix childdc1_childdnszones.ldf childdc2_childdnszones.ldf -lingering childdc1_childdnszones_lingering_childdc2.ldf
childdc2_childdnszones_lingering_childdc1.ldf -log childdnszones_childdc1_childdc2.log -domaindn
"dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com" >>run.log

echo ############### FINISHED ############## >>run.log

time /t >>run.log
```

Exercise 4

Use Excel's concatenate function to create repadmin commands for large data collection

Data collection for Lingering Links issue

Use Excel to create all required repadmin commands for data collection when there are a large number of objects.

The commands in this lab are built using Excel's concatenate function leveraging data within the Problem Attributes file created by the cmd: oabvalidate.exe DCNAME "(Objectclass=*)"

Command to create in Excel:

```
repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)"
/deleted /atts:member /long /allvalues /gc >8a6efacc-bc38-4431-b577-2b3207f90155.txt
```

Where B2 contents are objectGUID in this format: <GUID=8a6efacc-bc38-4431-b577-2b3207f90155>

Where C2 contents are objectGUID in this format: 8a6efacc-bc38-4431-b577-2b3207f90155

1. In a new column, in the first cell under the heading, type in the following

```
=concatenate("repadmin /showattr * ",CHAR(34),B2,CHAR(34),"
/filter:",CHAR(34),"(objectclass=*)",CHAR(34)," /deleted /atts:member /long /allvalues
/gc >",C2,".txt")
```

2. Press Enter

Results from Excel function:

```
repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)" /deleted
/atts:member /long /allvalues /gc >8a6efacc-bc38-4431-b577-2b3207f90155.txt
```

Command to create in Excel:

```
repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked >>8a6efacc-bc38-
4431-b577-2b3207f90155.txt
```

Excel

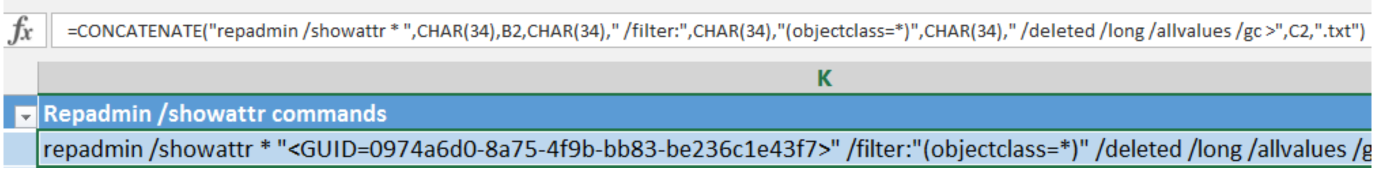
```
=concatenate("repadmin /showobjmeta * ",CHAR(34),B2,CHAR(34)," /linked
>>",C2,".txt")
```

Results from Excel function:

```
repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked >>8a6efacc-bc38-4431-b577-
2b3207f90155.txt
```

For an example where this process is used:

1. Open the c:\files\ALL_DCs_ProblemAttributes.xlsx spreadsheet in Excel
2. Select the **All Attributes with Commands** sheet.
3. Click on any cell in column K or L



```
=CONCATENATE("repadmin /showattr * ",CHAR(34),B2,CHAR(34)," /filter:",CHAR(34),"(objectclass=*)",CHAR(34)," /deleted /long /allvalues /gc >",C2,".txt")
```

K
Repadmin /showattr commands
repadmin /showattr * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /filter:"(objectclass=*)" /deleted /long /allvalues /g

Repadmin_cmds.bat

```
REM Data collection for Lingering Links issue

REM Commands built using Excel's concatenate function leveraging data within the Problem Attributes file created by the cmd:
oabvalidate.exe DCNAME "(Objectclass=*)"

REM Command to create in Excel:

REM repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)" /deleted /atts:member
/long /allvalues /gc >8a6efacc-bc38-4431-b577-2b3207f90155.txt

REM

REM where B2 contents are objectGUID in this format: <GUID=8a6efacc-bc38-4431-b577-2b3207f90155>

REM where C2 contents are objectGUID in this format: 8a6efacc-bc38-4431-b577-2b3207f90155

REM =concatenate("repadmin /showattr * ",CHAR(34),B2,CHAR(34)," /filter:",CHAR(34),"(objectclass=*)",CHAR(34)," /deleted
/atts:member /long /allvalues /gc >",C2,".txt")

REM

REM Results from Excel function:
```

```
REM repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >8a6efacc-bc38-4431-b577-2b3207f90155.txt

REM

REM Command to create in Excel:

REM repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked >>8a6efacc-bc38-4431-b577-2b3207f90155.txt

REM

REM =concatenate("repadmin /showobjmeta * ",CHAR(34),B2,CHAR(34)," /linked >>",C2,".txt")

REM Results from Excel function:

REM repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked >>8a6efacc-bc38-4431-b577-2b3207f90155.txt

REM ###############################################

REM Collect Repadmin /Showattr for each object

REM ###############################################

repadmin /showattr * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >8a6efacc-bc38-4431-b577-2b3207f90155.txt

repadmin /showattr * "<GUID=c6cce68d-5637-4035-8809-92d96f816e12>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >c6cce68d-5637-4035-8809-92d96f816e12.txt

repadmin /showattr * "<GUID=5c7bf2ac-fa70-484f-be1f-f059687d6721>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >5c7bf2ac-fa70-484f-be1f-f059687d6721.txt

repadmin /showattr * "<GUID=27ebf0f6-d853-40c3-876e-8b3a249fc8f7>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >27ebf0f6-d853-40c3-876e-8b3a249fc8f7.txt

repadmin /showattr * "<GUID=73a83289-f468-4435-88c5-f53d33711e28>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >73a83289-f468-4435-88c5-f53d33711e28.txt

repadmin /showattr * "<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >89ec9417-6e71-48e9-9655-e1efa48cfe3c.txt

repadmin /showattr * "<GUID=661f7d8d-20de-4f82-bf91-dc6470a1f451>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >661f7d8d-20de-4f82-bf91-dc6470a1f451.txt

repadmin /showattr * "<GUID=02b750d0-8dd2-4674-ab2e-6a024aeab1fe>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >02b750d0-8dd2-4674-ab2e-6a024aeab1fe.txt

repadmin /showattr * "<GUID=df11f042-e2a1-464a-8862-567098e226b0>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >df11f042-e2a1-464a-8862-567098e226b0.txt

repadmin /showattr * "<GUID=bfe317b4-4486-475c-9421-096205a43b26>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >bfe317b4-4486-475c-9421-096205a43b26.txt

repadmin /showattr * "<GUID=bee6a6d7-4eb6-4efa-b9f5-148f3e3fb06c>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >bee6a6d7-4eb6-4efa-b9f5-148f3e3fb06c.txt

repadmin /showattr * "<GUID=dde8a6f6-7e2b-497a-b002-b1949306b79e>" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >dde8a6f6-7e2b-497a-b002-b1949306b79e.txt

REM ##################################################

REM Collect Repadmin /Showobjmeta for each object

REM ##################################################

repadmin /showobjmeta * "<GUID=8a6efacc-bc38-4431-b577-2b3207f90155>" /linked  >>8a6efacc-bc38-4431-b577-2b3207f90155.txt
repadmin /showobjmeta * "<GUID=c6cce68d-5637-4035-8809-92d96f816e12>" /linked  >>c6cce68d-5637-4035-8809-92d96f816e12.txt

repadmin /showobjmeta * "<GUID=5c7bf2ac-fa70-484f-be1f-f059687d6721>" /linked  >>5c7bf2ac-fa70-484f-be1f-f059687d6721.txt

repadmin /showobjmeta * "<GUID=27ebf0f6-d853-40c3-876e-8b3a249fc8f7>" /linked  >>27ebf0f6-d853-40c3-876e-8b3a249fc8f7.txt

repadmin /showobjmeta * "<GUID=73a83289-f468-4435-88c5-f53d33711e28>" /linked  >>73a83289-f468-4435-88c5-f53d33711e28.txt

repadmin /showobjmeta * "<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c>" /linked  >>89ec9417-6e71-48e9-9655-e1efa48cfe3c.txt

repadmin /showobjmeta * "<GUID=661f7d8d-20de-4f82-bf91-dc6470a1f451>" /linked  >>661f7d8d-20de-4f82-bf91-dc6470a1f451.txt

repadmin /showobjmeta * "<GUID=02b750d0-8dd2-4674-ab2e-6a024aeab1fe>" /linked  >>02b750d0-8dd2-4674-ab2e-6a024aeab1fe.txt

repadmin /showobjmeta * "<GUID=df11f042-e2a1-464a-8862-567098e226b0>" /linked  >>df11f042-e2a1-464a-8862-567098e226b0.txt

repadmin /showobjmeta * "<GUID=bfe317b4-4486-475c-9421-096205a43b26>" /linked  >>bfe317b4-4486-475c-9421-096205a43b26.txt

repadmin /showobjmeta * "<GUID=bee6a6d7-4eb6-4efa-b9f5-148f3e3fb06c>" /linked  >>bee6a6d7-4eb6-4efa-b9f5-148f3e3fb06c.txt

repadmin /showobjmeta * "<GUID=dde8a6f6-7e2b-497a-b002-b1949306b79e>" /linked  >>dde8a6f6-7e2b-497a-b002-b1949306b79e.txt

REM ##############################################################################

REM Collect Repadmin /Showattr for each object referenced in the attribute

REM ##############################################################################

repadmin /showattr * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt

repadmin /showattr * "<GUID=6aff2f32-ac60-47b9-a142-148dda80d8b9>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >6aff2f32-ac60-47b9-a142-148dda80d8b9.txt

repadmin /showattr * "<GUID=200c41fa-6891-456d-82be-57d5e17c4bc4>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >200c41fa-6891-456d-82be-57d5e17c4bc4.txt

repadmin /showattr * "<GUID=d1112656-a0ee-4bab-8d74-69c10925c575>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >d1112656-a0ee-4bab-8d74-69c10925c575.txt

repadmin /showattr * "<GUID=c1fe8cd3-e623-4f51-b748-9467a65b86ad>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >c1fe8cd3-e623-4f51-b748-9467a65b86ad.txt

repadmin /showattr * "<GUID=c76cd855-909b-424f-bdc7-3ac3269ea0e0>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >c76cd855-909b-424f-bdc7-3ac3269ea0e0.txt

repadmin /showattr * "<GUID=be0fef43-0410-4620-8ff9-5e913296223b>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >be0fef43-0410-4620-8ff9-5e913296223b.txt

repadmin /showattr * "<GUID=0a0904dd-aa68-41e6-991c-46053aab98f8>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >0a0904dd-aa68-41e6-991c-46053aab98f8.txt

repadmin /showattr * "<GUID=858868d4-dada-4ea0-955a-248b85228a99>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >858868d4-dada-4ea0-955a-248b85228a99.txt

repadmin /showattr * "<GUID=d54db29a-8f1f-4ac3-af48-c3d2d07ec3bd>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >d54db29a-8f1f-4ac3-af48-c3d2d07ec3bd.txt
repadmin /showattr * "<GUID=4f50e768-bdf7-4ec8-908f-70b185baf463>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >4f50e768-bdf7-4ec8-908f-70b185baf463.txt

repadmin /showattr * "<GUID=17582af0-933f-499b-b781-11a205203eba>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >17582af0-933f-499b-b781-11a205203eba.txt

repadmin /showattr * "<GUID=c1a312d2-5fcc-4f6c-9f3d-fc87aa0fbcb0>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >c1a312d2-5fcc-4f6c-9f3d-fc87aa0fbcb0.txt

repadmin /showattr * "<GUID=90598ab8-78f9-4d22-bccb-1c74eca33aa2>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >90598ab8-78f9-4d22-bccb-1c74eca33aa2.txt

repadmin /showattr * "<GUID=250efeb5-1fcc-4768-913b-4b7f7c6a5c29>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >250efeb5-1fcc-4768-913b-4b7f7c6a5c29.txt

repadmin /showattr * "<GUID=3a460ea5-ed40-48f1-bfa0-99ade611e696>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >3a460ea5-ed40-48f1-bfa0-99ade611e696.txt

repadmin /showattr * "<GUID=606407a5-0c1e-4a7f-b383-820ea426e8c8>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >606407a5-0c1e-4a7f-b383-820ea426e8c8.txt

repadmin /showattr * "<GUID=3b70489f-6329-4fe5-b16b-6faa44391903>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >3b70489f-6329-4fe5-b16b-6faa44391903.txt

repadmin /showattr * "<GUID=d60b7347-12a5-4ec1-b9c2-0bd0a783b8c0>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >d60b7347-12a5-4ec1-b9c2-0bd0a783b8c0.txt

repadmin /showattr * "<GUID=b2eb5c44-c428-4612-a0b4-b0c2a1b345ea>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >b2eb5c44-c428-4612-a0b4-b0c2a1b345ea.txt

repadmin /showattr * "<GUID=ea04d741-d60a-4afc-922a-ac77b70a50f7>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >ea04d741-d60a-4afc-922a-ac77b70a50f7.txt

repadmin /showattr * "<GUID=f2197040-6d98-40da-abf9-f2fab0403d8e>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >f2197040-6d98-40da-abf9-f2fab0403d8e.txt

repadmin /showattr * "<GUID=4f5d57ed-e8ee-4cd9-8dff-ab738794d32d>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >4f5d57ed-e8ee-4cd9-8dff-ab738794d32d.txt

repadmin /showattr * "<GUID=207e16c4-268a-4fa8-95a9-220dc3d3e6b0>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >207e16c4-268a-4fa8-95a9-220dc3d3e6b0.txt

repadmin /showattr * "<GUID=9c83496a-8f80-4c71-81fe-693a3faf3991>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >9c83496a-8f80-4c71-81fe-693a3faf3991.txt

repadmin /showattr * "<GUID=56f77f3e-eba4-4e42-8c50-c7a60ec87bb5>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >56f77f3e-eba4-4e42-8c50-c7a60ec87bb5.txt

repadmin /showattr * "<GUID=d4929c0a-0e5e-47d9-a9e9-b6917cd19cd1>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >d4929c0a-0e5e-47d9-a9e9-b6917cd19cd1.txt

REM ##################################################################

REM Collect Repadmin /Showobjmeta for each object in the attribute

REM ##################################################################

repadmin /showobjmeta * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /linked >>0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt

repadmin /showobjmeta * "<GUID=6aff2f32-ac60-47b9-a142-148dda80d8b9>" /linked >>6aff2f32-ac60-47b9-a142-148dda80d8b9.txt

repadmin /showobjmeta * "<GUID=200c41fa-6891-456d-82be-57d5e17c4bc4>" /linked >>200c41fa-6891-456d-82be-57d5e17c4bc4.txt

repadmin /showobjmeta * "<GUID=d1112656-a0ee-4bab-8d74-69c10925c575>" /linked >>d1112656-a0ee-4bab-8d74-69c10925c575.txt
repadmin /showobjmeta * "<GUID=c1fe8cd3-e623-4f51-b748-9467a65b86ad>" /linked >>c1fe8cd3-e623-4f51-b748-9467a65b86ad.txt

repadmin /showobjmeta * "<GUID=c76cd855-909b-424f-bdc7-3ac3269ea0e0>" /linked >>c76cd855-909b-424f-bdc7-3ac3269ea0e0.txt

repadmin /showobjmeta * "<GUID=be0fef43-0410-4620-8ff9-5e913296223b>" /linked >>be0fef43-0410-4620-8ff9-5e913296223b.txt

repadmin /showobjmeta * "<GUID=0a0904dd-aa68-41e6-991c-46053aab98f8>" /linked >>0a0904dd-aa68-41e6-991c-46053aab98f8.txt

repadmin /showobjmeta * "<GUID=858868d4-dada-4ea0-955a-248b85228a99>" /linked >>858868d4-dada-4ea0-955a-248b85228a99.txt

repadmin /showobjmeta * "<GUID=d54db29a-8f1f-4ac3-af48-c3d2d07ec3bd>" /linked >>d54db29a-8f1f-4ac3-af48-c3d2d07ec3bd.txt

repadmin /showobjmeta * "<GUID=4f50e768-bdf7-4ec8-908f-70b185baf463>" /linked >>4f50e768-bdf7-4ec8-908f-70b185baf463.txt

repadmin /showobjmeta * "<GUID=17582af0-933f-499b-b781-11a205203eba>" /linked >>17582af0-933f-499b-b781-11a205203eba.txt

repadmin /showobjmeta * "<GUID=c1a312d2-5fcc-4f6c-9f3d-fc87aa0fbcb0>" /linked >>c1a312d2-5fcc-4f6c-9f3d-fc87aa0fbcb0.txt

repadmin /showobjmeta * "<GUID=90598ab8-78f9-4d22-bccb-1c74eca33aa2>" /linked >>90598ab8-78f9-4d22-bccb-1c74eca33aa2.txt

repadmin /showobjmeta * "<GUID=250efeb5-1fcc-4768-913b-4b7f7c6a5c29>" /linked >>250efeb5-1fcc-4768-913b-4b7f7c6a5c29.txt

repadmin /showobjmeta * "<GUID=3a460ea5-ed40-48f1-bfa0-99ade611e696>" /linked >>3a460ea5-ed40-48f1-bfa0-99ade611e696.txt

repadmin /showobjmeta * "<GUID=606407a5-0c1e-4a7f-b383-820ea426e8c8>" /linked >>606407a5-0c1e-4a7f-b383-820ea426e8c8.txt

repadmin /showobjmeta * "<GUID=3b70489f-6329-4fe5-b16b-6faa44391903>" /linked >>3b70489f-6329-4fe5-b16b-6faa44391903.txt

repadmin /showobjmeta * "<GUID=d60b7347-12a5-4ec1-b9c2-0bd0a783b8c0>" /linked >>d60b7347-12a5-4ec1-b9c2-0bd0a783b8c0.txt

repadmin /showobjmeta * "<GUID=b2eb5c44-c428-4612-a0b4-b0c2a1b345ea>" /linked >>b2eb5c44-c428-4612-a0b4-b0c2a1b345ea.txt

repadmin /showobjmeta * "<GUID=ea04d741-d60a-4afc-922a-ac77b70a50f7>" /linked >>ea04d741-d60a-4afc-922a-ac77b70a50f7.txt

repadmin /showobjmeta * "<GUID=f2197040-6d98-40da-abf9-f2fab0403d8e>" /linked >>f2197040-6d98-40da-abf9-f2fab0403d8e.txt

repadmin /showobjmeta * "<GUID=4f5d57ed-e8ee-4cd9-8dff-ab738794d32d>" /linked >>4f5d57ed-e8ee-4cd9-8dff-ab738794d32d.txt

repadmin /showobjmeta * "<GUID=207e16c4-268a-4fa8-95a9-220dc3d3e6b0>" /linked >>207e16c4-268a-4fa8-95a9-220dc3d3e6b0.txt

repadmin /showobjmeta * "<GUID=9c83496a-8f80-4c71-81fe-693a3faf3991>" /linked >>9c83496a-8f80-4c71-81fe-693a3faf3991.txt

repadmin /showobjmeta * "<GUID=56f77f3e-eba4-4e42-8c50-c7a60ec87bb5>" /linked >>56f77f3e-eba4-4e42-8c50-c7a60ec87bb5.txt

repadmin /showobjmeta * "<GUID=d4929c0a-0e5e-47d9-a9e9-b6917cd19cd1>" /linked >>d4929c0a-0e5e-47d9-a9e9-b6917cd19cd1.txt
```

```
REM "Done"
```

fix_lab.bat

```
# Copy and paste commands into an elevated PowerShell prompt

# Do not run as a batch job

#############################

#Exercise 1 Task 1

#############################

mkdir c:\files

cd c:\files

copy d:\files\*.*

cd c:\files

mkdir 1

cd 1

repadmin /syncall dc1 /Aed

repadmin /syncall dc2 /Aed

repadmin /syncall childdc1 /Aed

repadmin /syncall childdc2 /Aed

repadmin /syncall trdc1 /Aed

repadmin /showrepl * /csv >showrepl1.csv

repadmin /replicate dc2 dc1 "dc=root,dc=contoso,dc=com"

PING 127.0.0.1 -n 6

get-winevent -LogName "Directory Service" -ComputerName dc2 -MaxEvents 10 | Where-Object {$_.ID -eq "1988"} | fl
>DC2_DSevents.txt

#############################

#Exercise 1 Task 2

#############################

Repadmin /showobjmeta * "<GUID=e44b0379-382a-43e2-9e95-92f53c403002>" >emp2.txt

Repadmin /showrepl DC2 >DC2_showrepl.txt

Repadmin /removelingeringobjects DC1 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6 "dc=root,dc=contoso,dc=com"
/Advisory_Mode

PING 127.0.0.1 -n 6

get-winevent -LogName "Directory Service" -ComputerName dc1 -MaxEvents 10 | Where-Object {$_.ID -eq "1942"} | fl
>DC1_DSevents1942.txt

# use repadmin to get the DSA object GUID from DC1 to use with in the /removelingeringobjects command

repadmin /showrepl DC1 >DC1_showrepl.txt

# DC1 DSA object GUID = 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e

repadmin /removelingeringobjects dc2 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e "dc=root,dc=contoso,dc=com" /Advisory_Mode

PING 127.0.0.1 -n 6
```

```
get-winevent -LogName "Directory Service" -ComputerName dc2 -MaxEvents 10 | Where-Object {$_.ID -eq "1942"} | fl
>DC2_DSevents1942.txt
repadmin /showrepl * /csv | convertfrom-csv | out-gridview
#############################
#End of Exercise 1
#############################
#Review collected logs and Exercise summary
pause
#############################
#Exercise 2 Task 1
#############################
#Excercise 2: Open up lingering object tool on DC1 and click "Detect" and walk through those steps before continuing
start-sleep -s 20
cd c:\files
.\lingeringobjects\LingeringObjects.exe
#Replfix discovery
#Copy replfix.exe to the current working directory
cd c:\files
mkdir 2_replfix_discovery
cd 2_replfix_discovery
copy "c:\files\replfix.exe"
Ldifde -f dc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc1.root.contoso.com
Ldifde -f dc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s dc2.root.contoso.com
Ldifde -f trdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s trdc1.treeroot.fabrikam.com -t 3268
Ldifde -f childdc1_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc1.child.root.contoso.com -t 3268
Ldifde -f childdc2_root.ldf -d "dc=root,dc=contoso,dc=com" -p subtree -r "(objectclass=*)" -l
"replPropertyMetadata,objectGUID,replUptodateVector" -x -1 -s childdc2.child.root.contoso.com -t 3268
.\replfix dc1_root.ldf dc2_root.ldf -lingering dc1_root_lingering.ldf dc2_root_lingering.ldf -log root_dc1_dc2.log -domaindn
"dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
.\replfix dc1_root.ldf childdc1_root.ldf -lingering dc1_root_lingering_childdc1.ldf childdc1_root_lingering.ldf -log
root_dc1_childdc1.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
.\replfix dc1_root.ldf childdc2_root.ldf -lingering dc1_root_lingering_childdc2.ldf childdc2_root_lingering_dc1.ldf -log
root_dc1_childdc2.log -domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
.\replfix dc1_root.ldf trdc1_root.ldf -lingering dc1_root_lingering_trdc1.ldf trdc1_root_lingering_dc1.ldf -log root_dc1_trdc1.log -
domaindn "dc=root,dc=contoso,dc=com" -rootdn "dc=root,dc=contoso,dc=com"
cd c:\files
mkdir 2_replfix_fullDiscovery
cd 2_replfix_fullDiscovery
```

```
copy "c:\files\replfix.exe"

copy c:\files\ldifde_replfixCMDs.bat

copy c:\files\replfix_cmds.bat

start ldifde_replfixCMDs.bat

PING 127.0.0.1 -n 16

start replfix_cmds.bat

#replfix was used as a discovery mechanism to discover additional lingering objects that DRSReplicaVerifyObjects advisory mode are unable to show (because it will compare writable DC against a GC for its own parition) Discovery of Abandoned deleted object "live lingering Objects"

#############################

#End of Exercise 2

#############################

#Review collected logs and Exercise summary

pause

#############################

#Exercise 3 Task 1

#############################

#perform object removal using LDP method in lab manual

#Remove a lingering DNS object on ChildDC1 from the ForestDNSZones partition, using DC1 as a reference DC

# DC1 DSA object GUID = 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e

# Lingering Object (DC=DC93,DC=_msdcs.root.contoso.com,CN=MicrosoftDNS,DC=ForestDnsZones,DC=root,DC=contoso,DC=com)

# Lingering Object objectGUID = 3e873993-982b-47e8-8f20-5c50a5860ba8

# LDP, connect and bind to childdc1 port 389

# Browse / Modify / Attribute: RemoveLingeringOjbect

# Values: <GUID=70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e> : <GUID=3e873993-982b-47e8-8f20-5c50a5860ba8>

#

#############################

#Exercise 3 Task 2

#############################

pause

cd c:\files

mkdir 3

cd 3

Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e "dc=forestdnszones,dc=root,dc=contoso,dc=com"

PING 127.0.0.1 -n 6

get-winevent -LogName "Directory Service" -ComputerName childdc1 -MaxEvents 10 | Where-Object {$_.ID -eq "1939"} | fl >ChildDC1_DSevents1939.txt

#review childdc1_dsevents1939.txt file
```

```
pause
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e
"dc=forestdnszones,dc=root,dc=contoso,dc=com" /advisory_mode
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 3fe45b7f-e6b1-42b1-bcf4-2561c38cc3a6
"dc=forestdnszones,dc=root,dc=contoso,dc=com" /advisory_mode
Repadmin /removelingeringobjects childdc1.child.root.contoso.com 0b457f73-96a4-429b-ba81-1a3e0f51c848
"dc=forestdnszones,dc=root,dc=contoso,dc=com" /advisory_mode
PING 127.0.0.1 -n 6
get-winevent -LogName "Directory Service" -ComputerName childdc1 -MaxEvents 10 | Where-Object {$_.ID -eq "1942"} | fl
>ChildDC1_DSevents1942.txt
##############################
#Exercise 3 Task 3
##############################
#Next step is cleanup via repldiag
pause
#to view corresponding repadmin syntax, output to a file for review later
repldiag /removelingeringobjects /OutputRepadminCommandLineSyntax >repadminCMDLineSyntax.txt
repldiag /removelingeringobjects
#removes most lingering objects from the environment
##############################
#Exercise 3 Task 4
##############################
#take note, that many of the lingering objects are removed, but there are still a few that remain
#Next step is to remove lingering objects via the new Lingering Objects.exe tool
#Click Detect to discover lingering objects that still exist in the environment and then click RemovalAll, finally click Discover again
pause
#open Lingering Objects tool and select Detect
cd c:\files
.\lingeringobjects\LingeringObjects.exe
pause
#Click RemoveAll button
pause
# Reopen lingering objects tool and click Detect again to ensure all objects are removed
.\lingeringobjects\LingeringObjects.exe
pause
repadmin /syncall dc1 /Aed
repadmin /syncall dc2 /Aed
repadmin /syncall childdc1 /Aed
repadmin /syncall childdc2 /Aed
repadmin /syncall trdc1 /Aed

repadmin /showrepl * /csv >showrepl2.csv

repadmin /showrepl * /csv | convertfrom-csv | out-gridview

############################

#Exercise 3 Task 5

############################

#take note, that many of the AD replication errors are now cleared up, but there are still a few that remain

#Next we will use replfix to discover remaining objects

#copy ldifde_replfixCMDs.bat and replfix_cmds.bat file to DC1 - switch to DC1 then execute ldifde bat file followed by replfix bat file

cd c:\files

mkdir 4

cd 4

mkdir fullreplfixdiscovery

cd fullreplfixdiscovery

copy "c:\files\replfix.exe"

copy c:\files\ldifde_replfixCMDs.bat

copy c:\files\replfix_cmds.bat

start ldifde_replfixCMDs.bat

PING 127.0.0.1 -n 16

start replfix_cmds.bat

repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_child.txt

repadmin /showutdvec * "dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_root.txt

repadmin /showutdvec * "dc=treeroot,dc=fabrikam,dc=com" /latency /nocache >utdvec_treeroot.txt

repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency >>utdvec_child.txt

repadmin /showutdvec * "dc=root,dc=contoso,dc=com" /latency >>utdvec_root.txt

repadmin /showutdvec * "dc=treeroot,dc=fabrikam,dc=com" /latency >>utdvec_treeroot.txt

repadmin /showutdvec * "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_child.txt

repadmin /showutdvec * "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" /latency >>utdvec_childdnszones.txt

repadmin /showutdvec * "dc=domaindnszones,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_rootdnszones.txt

repadmin /showutdvec * "dc=domaindnszones,dc=root,dc=contoso,dc=com" /latency /nocache >>utdvec_rootdnszones.txt

repadmin /showutdvec * "dc=forestdnszones,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_forestrootdnszones.txt

repadmin /showutdvec * "dc=forestdnszones,dc=root,dc=contoso,dc=com" /latency >>utdvec_forestrootdnszones.txt

repadmin /showutdvec * "cn=configuration,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_config.txt

repadmin /showutdvec * "cn=configuration,dc=root,dc=contoso,dc=com" /latency /nocache >>utdvec_config.txt

#if the above bat files don't produce output, just execute them manually

# Review the run.log to get a quick overview of the count of abandoned deleted / live lingering objects on DCs hosting a read-only copy of the NCs

#next step is to initiate full replica sync to have GCs suck in live lingering objects
```

```
pause
repadmin /replicate dc2 dc1 "dc=root,dc=contoso,dc=com" /full
repadmin /replicate dc1 dc2 "dc=root,dc=contoso,dc=com" /full
repadmin /replicate * dc1 "dc=root,dc=contoso,dc=com" /full
repadmin /replicate * childdc1 "dc=child,dc=root,dc=contoso,dc=com" /full
repadmin /replicate * trdc1 "dc=treeroot,dc=fabrikam,dc=com" /full
repadmin /syncall dc1 /Aed
repadmin /syncall dc2 /Aed
repadmin /syncall childdc1 /Aed
repadmin /syncall childdc2 /Aed
repadmin /syncall trdc1 /Aed
repadmin /showrepl * /csv >showrepl4.csv
repadmin /showrepl * /csv | convertfrom-csv | out-gridview
#
#############################
#End of Exercise 3
#############################
#no more replication issues reported but there are still data inconsistencies in AD, we will us oabvalidate in the next exercise to find inconsistent group membership issues
pause
cd c:\files
mkdir 5
cd 5
mkdir full_replfixDiscovery
cd full_replfixDiscovery
copy "c:\files\replfix.exe"
copy c:\files\ldifde_replfixCMDs.bat
copy c:\files\replfix_cmds.bat
start ldifde_replfixCMDs.bat
PING 127.0.0.1 -n 16
start replfix_cmds.bat
cd c:\files\5
mkdir repadmin
cd repadmin
c:\files\oabvalidate\Oabvalidate.exe dc1 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe dc2 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe childdc1 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe childdc2 "(Objectclass=*)"
```

```
c:\files\oabvalidate\Oabvalidate.exe trdc1 "(Objectclass=*)"
start "C:\files\repadmin_cmds.bat"
pause
#Review problem attributes.txt file - import into Excel, tab delimited
#Note: To save a lot of time for data analysis: All data is consolidated into d:\ALL_DCs_ProblemAttributes.xlsx
#after reviewing objects, collect replication metadata for each group object and the lingering values using repadmin
#All repadmin commands needed for this step are in lab document and in repadmin_cmds.bat on the D drive of win8client
#commands are also present in the ALL_DCs_ProblemAttributes.xlsx
#oabvalidate reveals the following groups have inconsistent group membership
repadmin /showattr * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup1.txt
repadmin /showattr * "CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup2.txt
repadmin /showattr * "CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup3.txt
repadmin /showattr * "CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup4.txt
repadmin /showattr * "CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup5.txt
repadmin /showattr * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup1.txt
repadmin /showattr * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup2.txt
repadmin /showattr * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup3.txt
repadmin /showattr * "CN=LingeringLinkgroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup4.txt
repadmin /showattr * "CN=LingeringLinkgroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup5.txt
repadmin /showattr * "CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup1.txt
repadmin /showattr * "CN=LLGroup2,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup2.txt
repadmin /showattr * "CN=LLGroup3,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup3.txt
repadmin /showattr * "CN=LLGroup4,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup4.txt
repadmin /showattr * "CN=LLGroup5,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup5.txt
pause
#
#Review problem attributes in excel to see the issues
#Review repadmin output to determine scope of problem
```

```
#Look at user Brackish Waters on all DCs except childdc1
#review replication metadata for this object to determine what happened
repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_child.txt
repadmin /showutdvec * "dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_root.txt
repadmin /showutdvec * "dc=treeroot,dc=fabrikam,dc=com" /latency /nocache >utdvec_treeroot.txt
repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency >>utdvec_child.txt
repadmin /showutdvec * "dc=root,dc=contoso,dc=com" /latency >>utdvec_root.txt
repadmin /showutdvec * "dc=treeroot,dc=fabrikam,dc=com" /latency >>utdvec_treeroot.txt
repadmin /showutdvec * "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_childdnszones.txt
repadmin /showutdvec * "dc=domaindnszones,dc=child,dc=root,dc=contoso,dc=com" /latency >>utdvec_childdnszones.txt
repadmin /showutdvec * "dc=domaindnszones,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_rootdnszones.txt
repadmin /showutdvec * "dc=domaindnszones,dc=root,dc=contoso,dc=com" /latency /nocache >>utdvec_rootdnszones.txt
repadmin /showutdvec * "dc=forestdnszones,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_forestrootdnszones.txt
repadmin /showutdvec * "dc=forestdnszones,dc=root,dc=contoso,dc=com" /latency >>utdvec_forestrootdnszones.txt
repadmin /showutdvec * "cn=configuration,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_config.txt
repadmin /showutdvec * "cn=configuration,dc=root,dc=contoso,dc=com" /latency /nocache >>utdvec_config.txt
pause
# Review abandoned objects in the spreadsheet (Data analysis tab) Abandoned objects have yellow fill with bold text
#Open the Lingering Objects tool and import the abandoned.csv file. Highlight all objects and choose remove
.\lingeringobjects\LingeringObjects.exe
pause
cd c:\files\5
mkdir after_abandoned_removed
cd after_abandoned_removed
c:\files\oabvalidate\Oabvalidate.exe dc1 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe dc2 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe childdc1 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe childdc2 "(Objectclass=*)"
c:\files\oabvalidate\Oabvalidate.exe trdc1 "(Objectclass=*)"
repadmin /showattr * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt
repadmin /showattr * "<GUID=6aff2f32-ac60-47b9-a142-148dda80d8b9>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >6aff2f32-ac60-47b9-a142-148dda80d8b9.txt
repadmin /showattr * "<GUID=200c41fa-6891-456d-82be-57d5e17c4bc4>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >200c41fa-6891-456d-82be-57d5e17c4bc4.txt
repadmin /showattr * "<GUID=d1112656-a0ee-4bab-8d74-69c10925c575>" /filter:"(objectclass=*)" /deleted /long /allvalues /gc >d1112656-a0ee-4bab-8d74-69c10925c575.txt
repadmin /showobjmeta * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" /linked >>0974a6d0-8a75-4f9b-bb83-be236c1e43f7.txt
repadmin /showobjmeta * "<GUID=6aff2f32-ac60-47b9-a142-148dda80d8b9>" /linked >>6aff2f32-ac60-47b9-a142-148dda80d8b9.txt
repadmin /showobjmeta * "<GUID=200c41fa-6891-456d-82be-57d5e17c4bc4>" /linked >>200c41fa-6891-456d-82be-57d5e17c4bc4.txt
repadmin /showobjmeta * "<GUID=d1112656-a0ee-4bab-8d74-69c10925c575>" /linked >>d1112656-a0ee-4bab-8d74-69c10925c575.txt
repadmin /syncall dc1 /Aed
repadmin /syncall dc2 /Aed
repadmin /syncall childdc1 /Aed
repadmin /syncall childdc2 /Aed
repadmin /syncall trdc1 /Aed
PING 127.0.0.1 -n 4
repadmin /showrepl * /csv >showrepl5.csv
repadmin /showrepl * /csv | convertfrom-csv | out-gridview
start .
pause
# Review replication metadata output to confirm removal
#
# Membership issues are mostly corrected for the abandoned objects
# take care of the membership issues for the user objects that still exist by adding them back to the group and then remove them again
cd c:\files\5
mkdir lingeringlinks
cd lingeringlinks
repadmin /showobjmeta * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" >beforeLLinkGroup1.txt
repadmin /showobjmeta * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" >beforeLingeringLinkgroup1.txt
repadmin /showobjmeta * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" >beforeLingeringLinkgroup2.txt
repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" >beforeLingeringLinkgroup3.txt
repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" >beforeLingeringLinkgroup4.txt
repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" >beforeLingeringLinkgroup5.txt
repadmin /showattr * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >LLinkGroup1_members.txt
repadmin /showattr * "CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >LLinkGroup2_members.txt
repadmin /showattr * "CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >LLinkGroup3_members.txt
repadmin /showattr * "CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >LLinkGroup4_members.txt
repadmin /showattr * "CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc
>LLinkGroup5_members.txt

repadmin /showattr * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues
/long /gc >LingeringLinkgroup1_members.txt

repadmin /showattr * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues
/long /gc >LingeringLinkgroup2_members.txt

repadmin /showattr * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues
/long /gc >LingeringLinkgroup3_members.txt

repadmin /showattr * "CN=LingeringLinkgroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues
/long /gc >LingeringLinkgroup4_members.txt

repadmin /showattr * "CN=LingeringLinkgroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues
/long /gc >LingeringLinkgroup5_members.txt

######################

#Fix groups by adding and removing members -only valid for users that still exist in AD ---not a good option for large groups

######################

cd c:\files\5\lingeringlinks

mkdir addremove

cd addremove

Set-ADGroup -Add:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Identity:"CN=LingeringLinkGroup2,OU=Lingering
Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Identity:"CN=LingeringLinkGroup3,OU=Lingering
Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Identity:"CN=LingeringLinkGroup4,OU=Lingering
Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Identity:"CN=LingeringLinkGroup5,OU=Lingering
Links,DC=root,DC=contoso,DC=com" -Server:"DC1.root.contoso.com"

#LLinkGroup1

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com","CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Tabatha
Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com", "CN=Juliette
Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com", "CN=Ulysses
Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Remove:@{'Member'="CN=Art
Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com","CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Tabatha
Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com", "CN=Juliette
Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com", "CN=Ulysses
Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com"} -Server:"TRDC1.treeroot.fabrikam.com"

#LLinkGroup2

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com","CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Chase
Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Remove:@{'Member'="CN=Art
Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com","CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Chase
Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -Server:"TRDC1.treeroot.fabrikam.com"

#LLinkGroup3

Set-ADGroup -Add:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -
Remove:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"TRDC1.treeroot.fabrikam.com"

#LLinkGroup4

Set-ADGroup -Add:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Chase
Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -
Remove:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com","CN=Chase
Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -Server:"TRDC1.treeroot.fabrikam.com"

#LLinkGroup5

Set-ADGroup -Add:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -Server:"TRDC1.treeroot.fabrikam.com"

Set-ADGroup -Identity:"CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -
Remove:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"TRDC1.treeroot.fabrikam.com"

#Child domain groups

Set-ADGroup -Add:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Server:"ChildDC1.child.root.contoso.com"

Set-ADGroup -Identity:"CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"ChildDC1.child.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLGroup2,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Server:"ChildDC1.child.root.contoso.com"

Set-ADGroup -Identity:"CN=LLGroup2,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"ChildDC1.child.root.contoso.com"

Set-ADGroup -Add:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Identity:"CN=LLGroup4,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Server:"ChildDC1.child.root.contoso.com"
```

```
Set-ADGroup -Identity:"CN=LLGroup4,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"ChildDC1.child.root.contoso.com"

start-sleep -s 5

repadmin /syncall childdc1 "DC=child,DC=root,DC=contoso,DC=com" /edP

repadmin /syncall trdc1 "dc=treeroot,dc=fabrikam,dc=com" /edP

repadmin /syncall dc1 "DC=root,DC=contoso,DC=com" /edP

PING 127.0.0.1 -n 6

Set-ADGroup -Identity:"CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -
Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkGroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkGroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

Set-ADGroup -Identity:"CN=LingeringLinkGroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com" -
Remove:@{'Member'="CN=Art Cowles,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com", "CN=Becker
Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com"} -Server:"DC1.root.contoso.com"

repadmin /syncall trdc1 "dc=treeroot,dc=fabrikam,dc=com" /edP

repadmin /syncall dc1 "DC=root,DC=contoso,DC=com" /edP

PING 127.0.0.1 -n 15

repadmin /showobjmeta * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" >afterLLinkGroup1.txt

repadmin /showobjmeta * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com"
>afterLingeringLinkgroup1.txt

repadmin /showobjmeta * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com"
>afterLingeringLinkgroup2.txt

repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com"
>afterLingeringLinkgroup3.txt

repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com"
>afterLingeringLinkgroup4.txt

repadmin /showobjmeta * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com"
>afterLingeringLinkgroup5.txt

repadmin /showattr * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc
>>LLinkGroup1_members.txt

repadmin /showattr * "CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc
>>LLinkGroup2_members.txt
```

```
repadmin /showattr * "CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >>LLinkGroup3_members.txt
repadmin /showattr * "CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >>LLinkGroup4_members.txt
repadmin /showattr * "CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /atts:member /allvalues /long /gc >>LLinkGroup5_members.txt
repadmin /showattr * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc >>LingeringLinkgroup1_members.txt
repadmin /showattr * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc >>LingeringLinkgroup2_members.txt
repadmin /showattr * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc >>LingeringLinkgroup3_members.txt
repadmin /showattr * "CN=LingeringLinkgroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc >>LingeringLinkgroup4_members.txt
repadmin /showattr * "CN=LingeringLinkgroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc >>LingeringLinkgroup5_members.txt
# We were able to fix some of the group membership issues by adding and then removing the users from the groups.
# However, some of the links point to users that no longer exist in AD
# Next we will create new user objects using the same objectGUID as the old member and then simply delete them again
start .
cd c:\files\5
mkdir lingeringlinks_createuser
cd lingeringlinks_createuser
ldifde -i -f c:\files\FixAbsent.ldf -s trdc1.treeroot.fabrikam.com
repadmin /syncall trdc1 "dc=treeroot,dc=fabrikam,dc=com" /edP
PING 127.0.0.1 -n 15
ldifde -i -f c:\files\deleteAbsentusers.ldf -s trdc1.treeroot.fabrikam.com
PING 127.0.0.1 -n 5
repadmin /syncall trdc1 "dc=treeroot,dc=fabrikam,dc=com" /edP
PING 127.0.0.1 -n 5
repadmin /syncall dc1 "DC=root,DC=contoso,DC=com" /edP
repadmin /syncall childdc2 /Aed
PING 127.0.0.1 -n 5
#End of lab check
cd c:\files
mkdir finished
cd finished
repadmin /showattr * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" /atts:member /allvalues /long /gc /extended >LingeringLinkgroup1_membersext.txt
repadmin /showattr * "CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup1.txt
repadmin /showattr * "CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup2.txt

repadmin /showattr * "CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup3.txt

repadmin /showattr * "CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup4.txt

repadmin /showattr * "CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >TR_LLinkGroup5.txt

repadmin /showattr * "CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup1.txt

repadmin /showattr * "CN=LingeringLinkgroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup2.txt

repadmin /showattr * "CN=LingeringLinkgroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup3.txt

repadmin /showattr * "CN=LingeringLinkgroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup4.txt

repadmin /showattr * "CN=LingeringLinkgroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Root_LingeringLinkgroup5.txt

repadmin /showattr * "CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup1.txt

repadmin /showattr * "CN=LLGroup2,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup2.txt

repadmin /showattr * "CN=LLGroup3,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup3.txt

repadmin /showattr * "CN=LLGroup4,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup4.txt

repadmin /showattr * "CN=LLGroup5,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" /filter:"(objectclass=*)" /deleted /atts:member /long /allvalues /gc >Child_LLGroup5.txt

repadmin /showrepl * /csv >repl.csv

repadmin /showrepl * /csv | convertfrom-csv | out-gridview

pause
```

Open problemattributes.txt in Excel

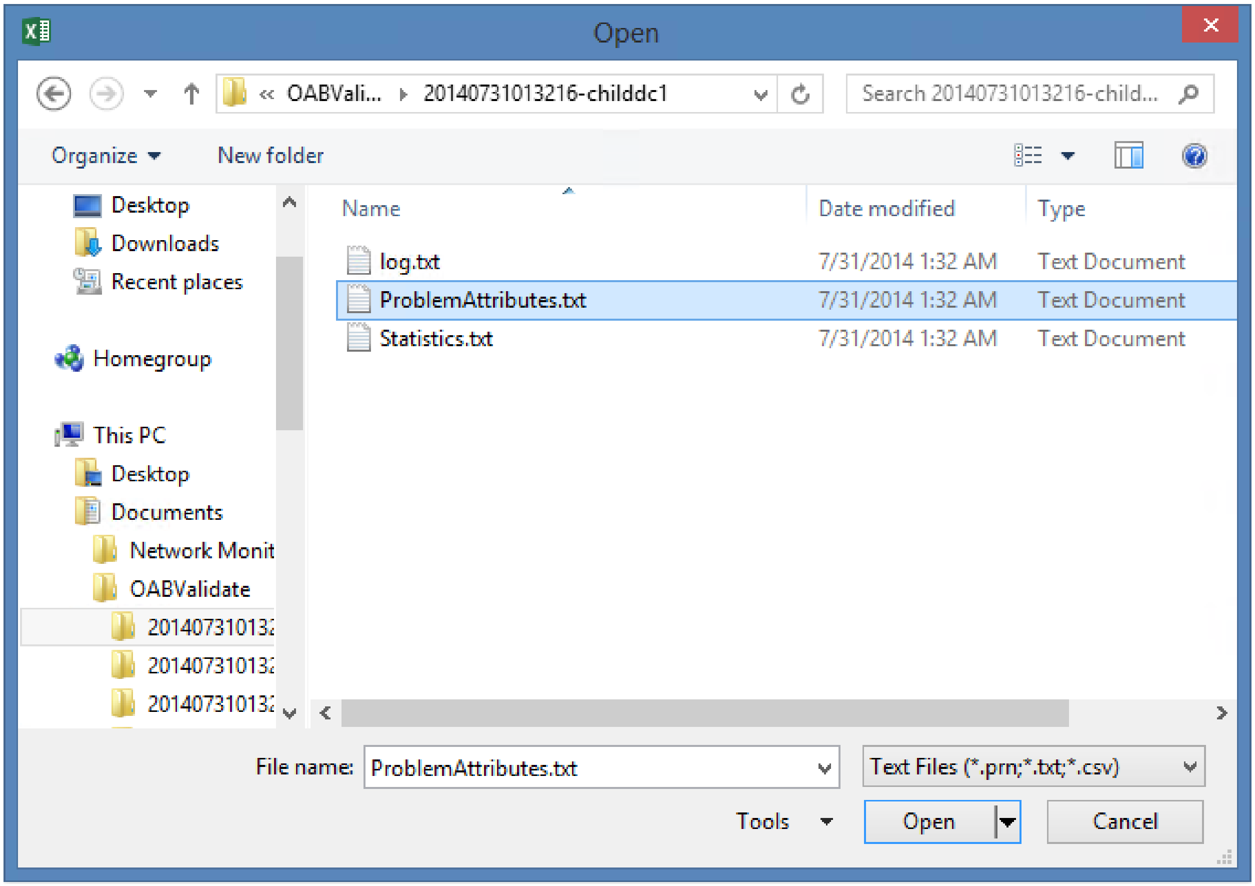
1. Open Microsoft Excel
2. From the **File** menu, choose **Open**

    b. Browse to Documents directory

```
C:\users\administrator.ROOT\Documents\OABValidate\Date_TimeStamp-DCName\
```

    c. In the files of type box, **All Excel Files** drop down, select **Text Files (*.prn;*.txt;*.csv)**

    d. Select **ProblemAttributes.txt** and then select **Open**

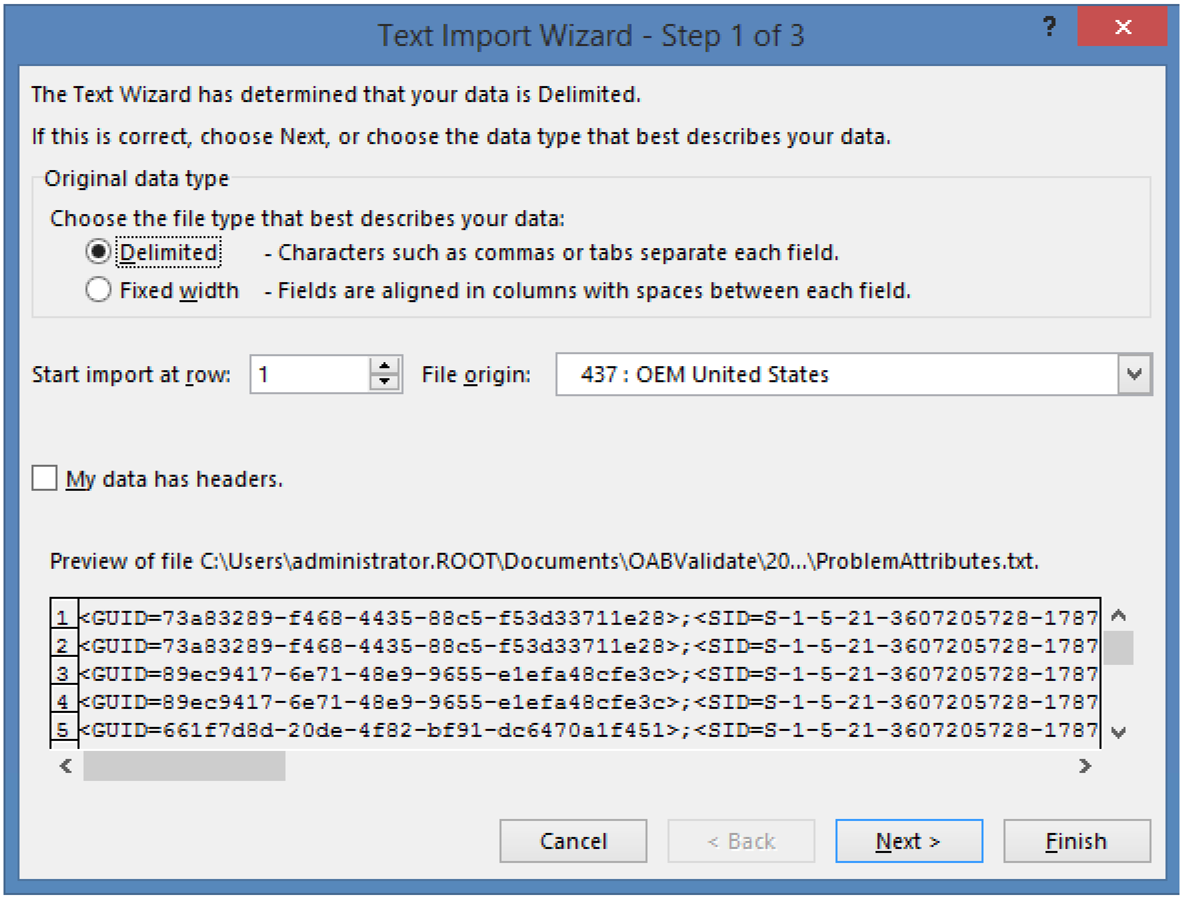


Name	Date modified	Type
log.txt	7/31/2014 1:32 AM	Text Document
ProblemAttributes.txt	7/31/2014 1:32 AM	Text Document
Statistics.txt	7/31/2014 1:32 AM	Text Document

File name: ProblemAttributes.txt

Text Files (*.prn;*.txt;*.csv)

3. On the **Text Import Wizard - Step 1 of 3** screen, ensure **Delimited** and is selected, and then choose **Next**



The Text Wizard has determined that your data is Delimited.

If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

● Delimited - Characters such as commas or tabs separate each field.

○ Fixed width - Fields are aligned in columns with spaces between each field.

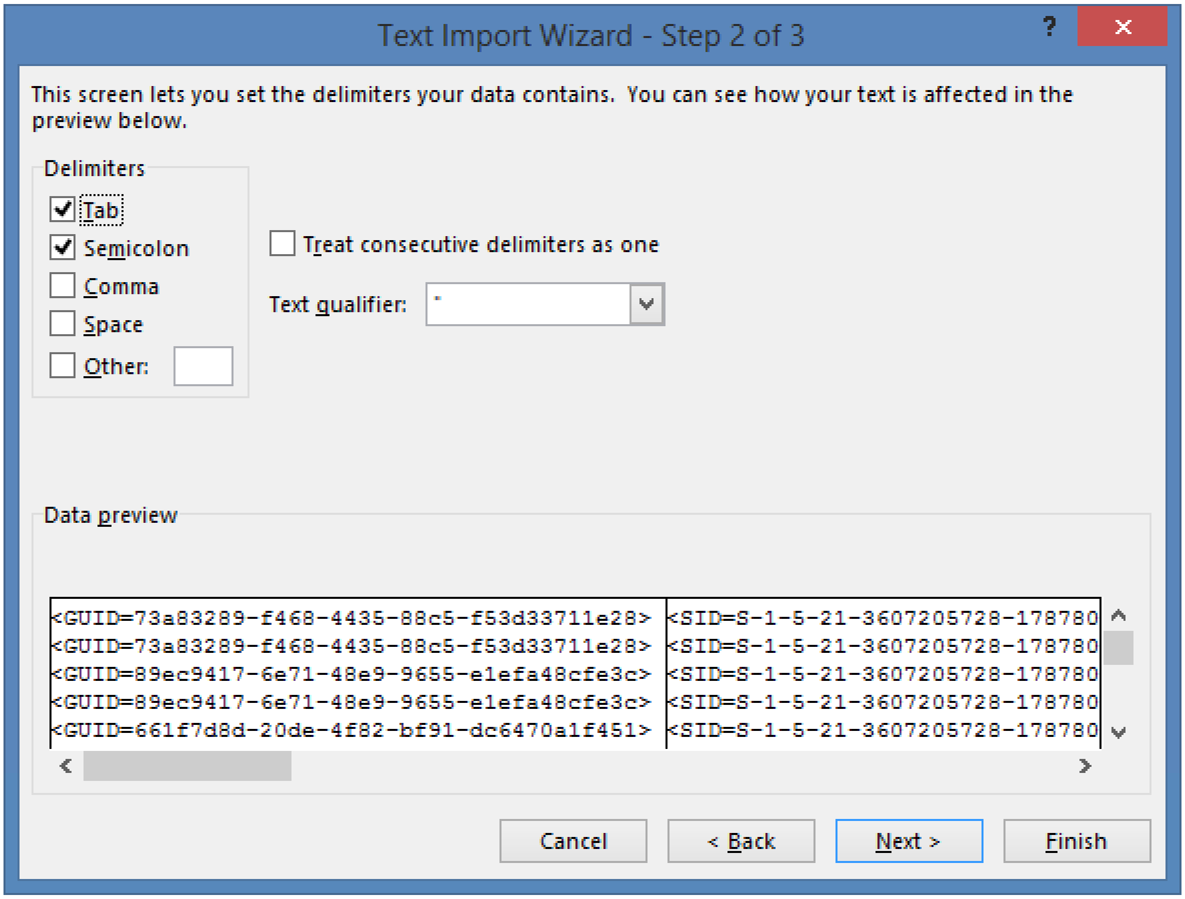
Start import at row: 1 File origin: 437 : OEM United States

□ My data has headers.

Preview of file C:\Users\administrator.ROOT\Documents\OABValidate\20...\ProblemAttributes.txt.

```
1 <GUID=73a83289-f468-4435-88c5-f53d33711e28>;<SID=S-1-5-21-3607205728-1787
2 <GUID=73a83289-f468-4435-88c5-f53d33711e28>;<SID=S-1-5-21-3607205728-1787
3 <GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c>;<SID=S-1-5-21-3607205728-1787
4 <GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c>;<SID=S-1-5-21-3607205728-1787
5 <GUID=661f7d8d-20de-4f82-bf91-dc6470a1f451>;<SID=S-1-5-21-3607205728-1787
```

4. On the **Text Import Wizard - Step 2 of 3** screen, check the **Tab** and **Semicolon** boxes (ensure both are selected) in the **Delimiters** section



This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

Delimiters

☑ Tab

☑ Semicolon

□ Comma

□ Space

□ Other:

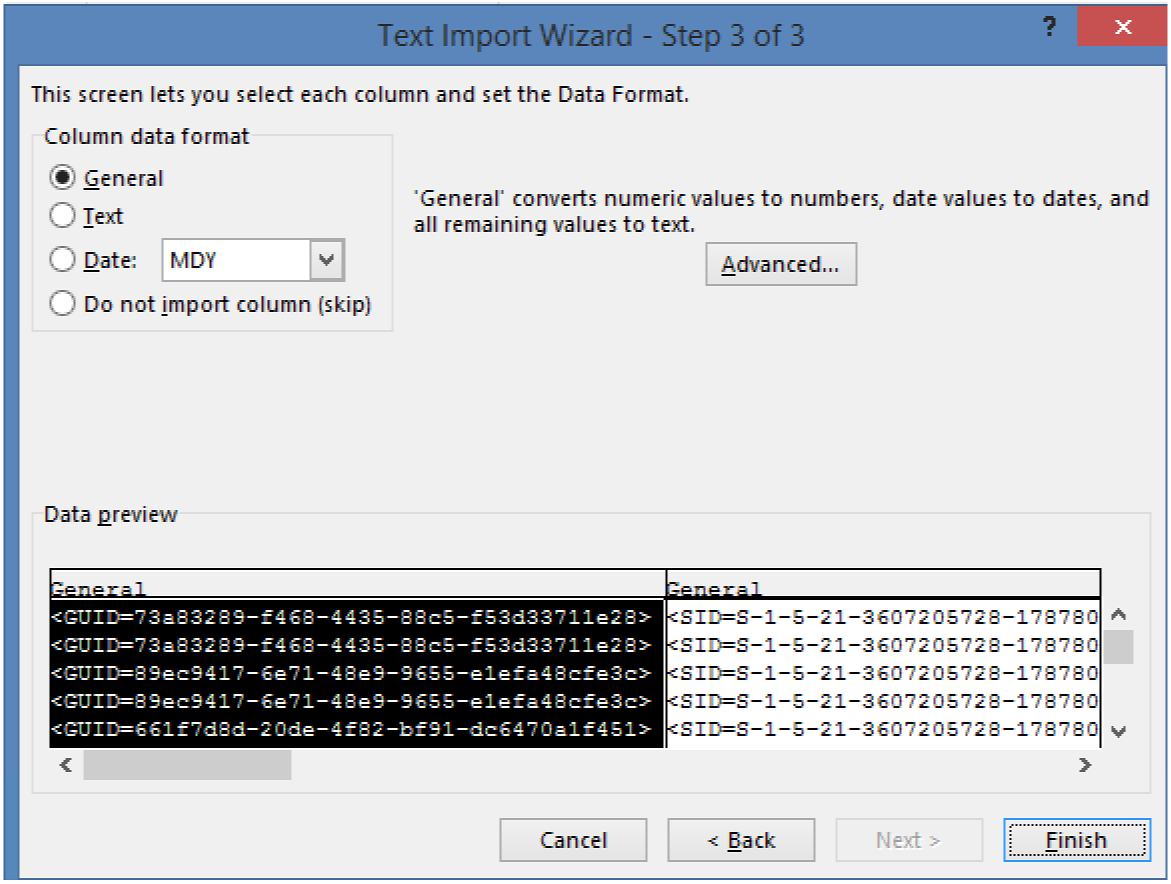
□ Treat consecutive delimiters as one

Text qualifier: "

Data preview

```
<GUID=73a83289-f468-4435-88c5-f53d33711e28> <SID=S-1-5-21-3607205728-178780
<GUID=73a83289-f468-4435-88c5-f53d33711e28> <SID=S-1-5-21-3607205728-178780
<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c> <SID=S-1-5-21-3607205728-178780
<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c> <SID=S-1-5-21-3607205728-178780
<GUID=661f7d8d-20de-4f82-bf91-dc6470a1f451> <SID=S-1-5-21-3607205728-178780
```

5. On the **Text Import Wizard - Step 3 of 3**, leave the default option selected and then choose **Finish**.



This screen lets you select each column and set the Data Format.

Column data format

● General
○ Text
○ Date: MDY
○ Do not import column (skip)

'General' converts numeric values to numbers, date values to dates, and all remaining values to text.

Data preview

```
General                                            General
<GUID=73a83289-f468-4435-88c5-f53d33711e28> <SID=S-1-5-21-3607205728-178780
<GUID=73a83289-f468-4435-88c5-f53d33711e28> <SID=S-1-5-21-3607205728-178780
<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c> <SID=S-1-5-21-3607205728-178780
<GUID=89ec9417-6e71-48e9-9655-e1efa48cfe3c> <SID=S-1-5-21-3607205728-178780
<GUID=661f7d8d-20de-4f82-bf91-dc6470a1f451> <SID=S-1-5-21-3607205728-178780
```

From the **Home** menu, select **Format as Table,** select a desired Table style and then choose **OK**.

Column A: Object - ObjectGUID

Column B: Object - ObjectSID

Column C: Object - Object DN

Column D: Attribute name

Column E: Problem type (lingering object, lingering link)

Attribute data - Object GUID

Column F: Attribute data - Object SID

Column G Attribute data - Object DN

Populate the sheet with per DC object presence status

Obtain replication metadata for each object referenced in the attribute data column.

1. Create new columns to mark if object exists on a given DC

   "Present on DC1", "Present on DC2", "Present on ChildDC1", "Present on ChildDC2", and "Present on TRDC1"

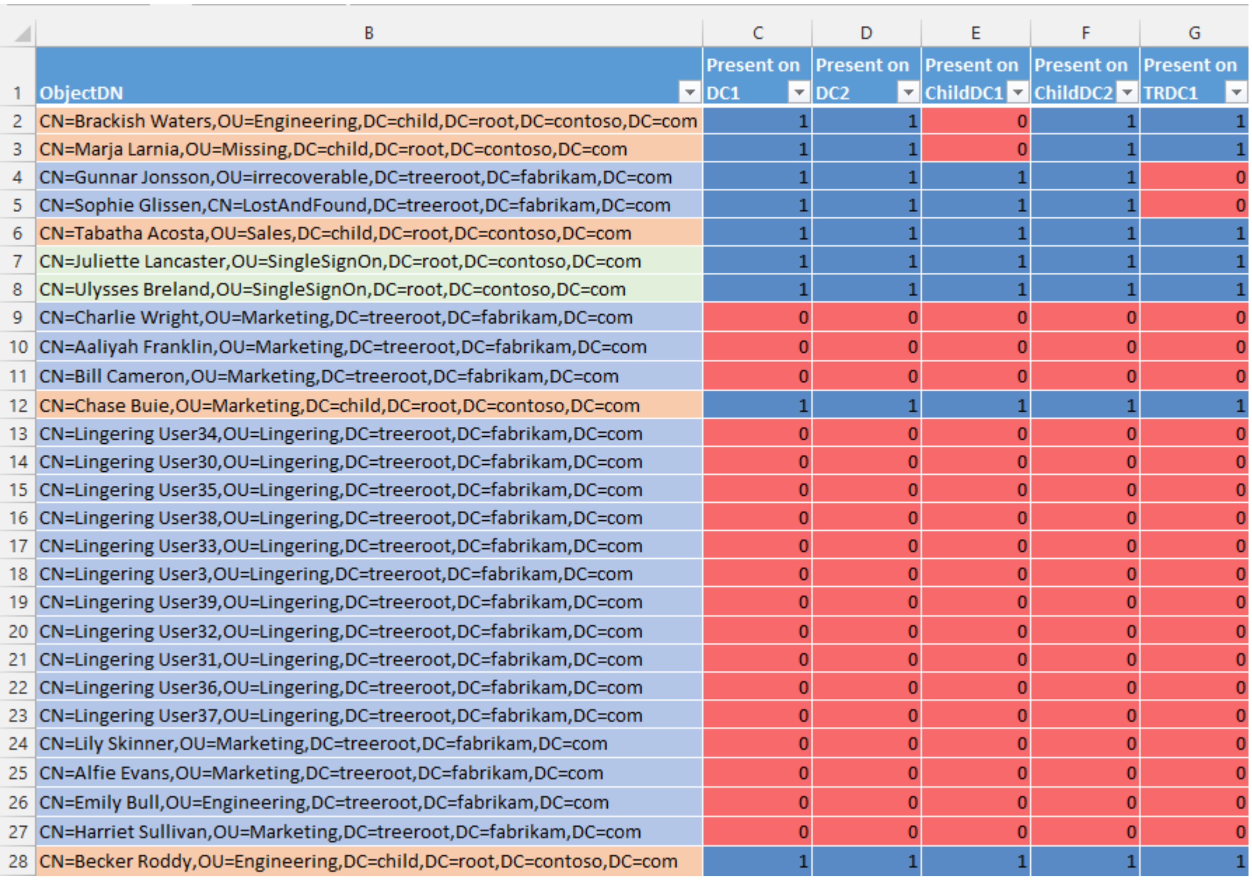
2. Review the repadmin /showobjmeta output for each object and populate the spreadsheet.

For each object:

a. Place a 1 in the cell if the object exists on the DC

b. Place a 0 in the cell if the object does not exist on the DC

See Figure 8 for an example.



	B	C	D	E	F	G
1	ObjectDN	Present on DC1	Present on DC2	Present on ChildDC1	Present on ChildDC2	Present on TRDC1
2	CN=Brackish Waters,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	1	1	0	1	1
3	CN=Marja Larnia,OU=Missing,DC=child,DC=root,DC=contoso,DC=com	1	1	0	1	1
4	CN=Gunnar Jonsson,OU=irrecoverable,DC=treeroot,DC=fabrikam,DC=com	1	1	1	1	0
5	CN=Sophie Glissen,CN=LostAndFound,DC=treeroot,DC=fabrikam,DC=com	1	1	1	1	0
6	CN=Tabatha Acosta,OU=Sales,DC=child,DC=root,DC=contoso,DC=com	1	1	1	1	1
7	CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com	1	1	1	1	1
8	CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com	1	1	1	1	1
9	CN=Charlie Wright,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
10	CN=Aaliyah Franklin,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
11	CN=Bill Cameron,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
12	CN=Chase Buie,OU=Marketing,DC=child,DC=root,DC=contoso,DC=com	1	1	1	1	1
13	CN=Lingering User34,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
14	CN=Lingering User30,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
15	CN=Lingering User35,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
16	CN=Lingering User38,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
17	CN=Lingering User33,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
18	CN=Lingering User3,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
19	CN=Lingering User39,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
20	CN=Lingering User32,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
21	CN=Lingering User31,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
22	CN=Lingering User36,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
23	CN=Lingering User37,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
24	CN=Lily Skinner,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
25	CN=Alfie Evans,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
26	CN=Emily Bull,OU=Engineering,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
27	CN=Harriet Sullivan,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com	0	0	0	0	0
28	CN=Becker Roddy,OU=Engineering,DC=child,DC=root,DC=contoso,DC=com	1	1	1	1	1

Figure 8 Object existence tally

Populate the sheet with replication metadata

Obtain replication metadata for each object referenced in the attribute value (member attribute in this example):

Originating DSA GUID

Obtain the Originating DSA GUID from repadmin /showobjmeta output.

```
repadmin /showobjmeta * "<GUID=0974a6d0-8a75-4f9b-bb83-be236c1e43f7>" >0974a6d0-8a75-4f9b-bb83-
be236c1e43f7.txt
```

Table 5 Repadmin /showobjmeta output for user object with Originating DSA highlighted

Loc. USN	Originating DSA	Org. USN	Org. Date	Org. Time	Version	Attribute
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	whenCreated

Originating USN value on whenCreated attribute

Obtain the Org. USN value from the whenCreated attribute in the repadmin /showobjmeta output.

Table 6 Repadmin /showobjmeta output for user object with Originating USN value highlighted

Loc. USN	Originating DSA	Org. USN	Org. Date	Org. Time	Version	Attribute
221388	606f5d34-7202-4073-83fb-aac8bb109868	152543	2013-05-10	4:36:04	1	whenCreated

USN in Up-to-dateness vector for Originating (Org.) DSA for the referenced object's partition from a DC writable for the object and GCs

Obtain the USN in Up-to-dateness vector for Originating DSA for the referenced object's partition from a DC writable for the object and GCs from repadmin /showutdvec output.

```
repadmin /showutdvec * "dc=child,dc=root,dc=contoso,dc=com" /latency /nocache >utdvec_child.txt
```

```
Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com
606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152523 @ Time 2013-05-10 04:30:16
a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07
```

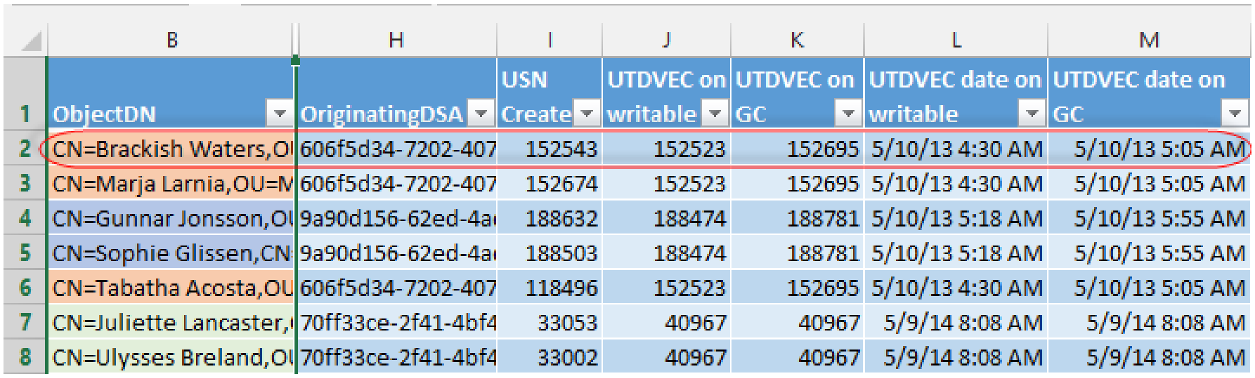
Date and Timestamp in the UTDVEC output for the Org. DSA

Obtain the Date and timestamp in the UTDVEC output for the Originating DSA from repadmin /showutdvec output.

```
Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com
606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152523 @ Time 2013-05-10 04:30:16
a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07
```

Add this data to the spreadsheet

Add the replication metadata to the spreadsheet for each object. Review Figure 8. In this example, the data obtained from repadmin has been added to columns I through M. Note that columns C through G have been hidden to make data entry easier.



	B	H	I	J	K	L	M
1	ObjectDN	OriginatingDSA	USN Create	UTDVEC on writable	UTDVEC on GC	UTDVEC date on writable	UTDVEC date on GC
2	CN=Brackish Waters,OI	606f5d34-7202-407	152543	152523	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
3	CN=Marja Larnia,OU=M	606f5d34-7202-407	152674	152523	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
4	CN=Gunnar Jonsson,OI	9a90d156-62ed-4a	188632	188474	188781	5/10/13 5:18 AM	5/10/13 5:55 AM
5	CN=Sophie Glissen,CN	9a90d156-62ed-4a	188503	188474	188781	5/10/13 5:18 AM	5/10/13 5:55 AM
6	CN=Tabatha Acosta,OU	606f5d34-7202-407	118496	152523	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
7	CN=Juliette Lancaster,	70ff33ce-2f41-4bf4	33053	40967	40967	5/9/14 8:08 AM	5/9/14 8:08 AM
8	CN=Ulysses Breland,OI	70ff33ce-2f41-4bf4	33002	40967	40967	5/9/14 8:08 AM	5/9/14 8:08 AM

Figure 9 replication metadata and UTDVEC correlation table data

Abandoned object identification using conditional formatting

Use conditional formatting rules to make analysis easier for a large amount of objects reported by Oabvalidate. In the following steps, you will apply a conditional formatting rule to focus attention on potential abandoned objects.

More:

Abandoned object

An object created on one DC that never got replicated to other DCs hosting a writable copy of the NC but does get replicated to DCs/GCs hosting a read-only copy of the NC. The originating DC goes offline prior to replicating the originating write to other DCs that contain a writable copy of the partition. Discovery of this object type is challenging. An easy indicator is destination GCs in strict mode that log 1988s for objects that are R/W in the source DCs partition.

For each object:

1. Look at USN in object's replmetadata for originating create
2. Look at UpToDatenessVector in /showutdvec output for object partition on all R/W DCs for Originating DSA GUID reported in #1
3. Alert on object where #1 is higher than #2

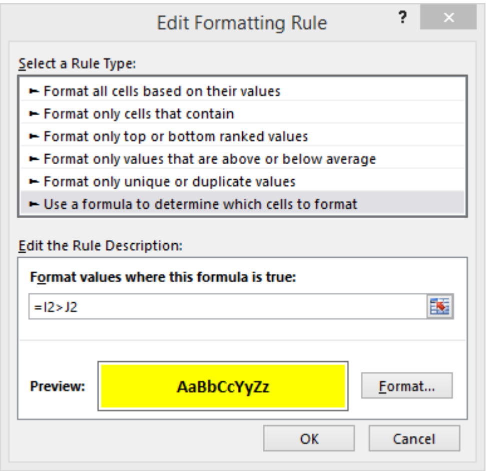
Using *Figure 9* as an example:

1. Highlight **column J**
2. Then, click **Home** > **Conditional Formatting** > **New Rule**.
3. In the **New Formatting Rule** dialog box, click **Use a formula to determine which cells to format**.
4. Under **Format values where this formula is true**, type the formula: **=I2>J2**

   The value in **I2** is the originating USN used to create the object.

   The value in **J2** is the highest USN received by writable DCs from the object's originating DSA.
5. Click **Format**.

6. On the **Font** tab, in the **Font style** area, select **Bold**. On the **Fill** tab, in the **Background Color** area, select **Yellow**.

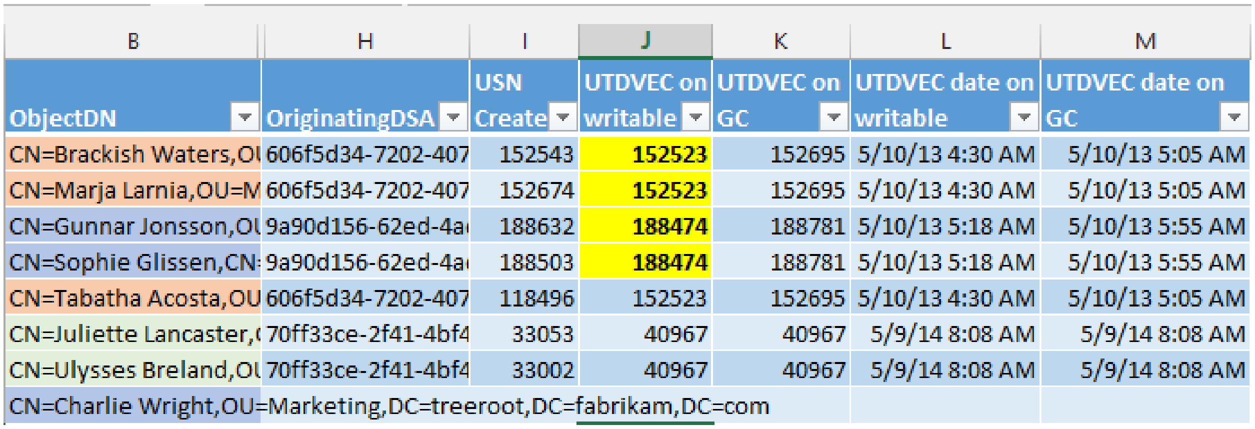

Edit Formatting Rule
?
×
Select a Rule Type:
► Format all cells based on their values
► Format only cells that contain
► Format only top or bottom ranked values
► Format only values that are above or below average
► Format only unique or duplicate values
► Use a formula to determine which cells to format
Edit the Rule Description:
Preview:
AaBbCcYyZz
Format...
OK
Cancel

Format values where this formula is true:

```
=I2>J2
```

7. Click **OK** until the dialog boxes are closed.

8. The formatting is applied to column J



B	H	I	J	K	L	M
ObjectDN	OriginatingDSA	USN Create	UTDVEC on writable	UTDVEC on GC	UTDVEC date on writable	UTDVEC date on GC
CN=Brackish Waters,OI	606f5d34-7202-407	152543	**152523**	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
CN=Marja Larnia,OU=M	606f5d34-7202-407	152674	**152523**	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
CN=Gunnar Jonsson,OI	9a90d156-62ed-4a	188632	**188474**	188781	5/10/13 5:18 AM	5/10/13 5:55 AM
CN=Sophie Glissen,CN:	9a90d156-62ed-4a	188503	**188474**	188781	5/10/13 5:18 AM	5/10/13 5:55 AM
CN=Tabatha Acosta,OU	606f5d34-7202-407	118496	152523	152695	5/10/13 4:30 AM	5/10/13 5:05 AM
CN=Juliette Lancaster,(	70ff33ce-2f41-4bf4	33053	40967	40967	5/9/14 8:08 AM	5/9/14 8:08 AM
CN=Ulysses Breland,OI	70ff33ce-2f41-4bf4	33002	40967	40967	5/9/14 8:08 AM	5/9/14 8:08 AM
CN=Charlie Wright,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com						

Figure 10 Conditional formatting applied to column J

Any cells that have a yellow fill with bold font are potential abandoned objects.

Brackish Waters, Marja Larnia, Gunnar Jonsson and Sophie Glissen all appear to be abandoned objects.

These objects can be removed using the RemoveLingeringObjects rootDSE modification.

References

4.1.24.3 Server Behavior of the IDL_DRSReplicaVerifyObjects Method

Tracking Updates

http://technet.microsoft.com/en-us/library/cc961798.aspx

DS Heuristics

Constraints -DSHeuristics

The state of AD replication in the lab environment

Repadmin /showrepl * /csv >showrepl.csv

Format as table

Filter column K

Destination DSA Site	Destination DSA	Naming Context	Source DSA Site	Source DSA	Number of Failures	Last Failure Time	Last Success Time	Last Failure Status
Boulder	CHILDDC1	DC=fourthcoffee,DC=com	Boulder	DC1	277	7/8/2014 3:59	5/10/2013 14:51	8440
Boulder	TRDC1	DC=fourthcoffee,DC=com	Boulder	DC2	282	7/8/2014 3:46	5/10/2013 14:54	8440

Destination DSA Site	Destination DSA	Naming Context	Source DSA Site	Source DSA	Number of Failures	Last Failure Time	Last Success Time	Last Failure Status
Boulder	TRDC1	CN=Configuration,DC=root,DC=contoso,DC=com	Boulder	FOURTHDC1	283	7/8/2014 3:45	5/10/2013 5:19	8524
Boulder	TRDC1	CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com	Boulder	FOURTHDC1	282	7/8/2014 3:46	5/10/2013 5:19	8524
Boulder	CHILDDC2	CN=Configuration,DC=root,DC=contoso,DC=com	Boulder	FOURTHDC1	276	7/8/2014 4:00	5/10/2013 5:48	8524
Boulder	CHILDDC2	CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com	Boulder	FOURTHDC1	276	7/8/2014 4:00	5/10/2013 5:48	8524

Destination DSA Site	Destination DSA	Naming Context	Source DSA Site	Source DSA	Number of Failures	Last Failure Time	Last Success Time	Last Failure Status
Boulder	DC1	DC=DomainDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC2	291	7/8/2014 3:45	5/11/2013 11:54	8606
Boulder	DC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC2	265	7/8/2014 3:45	5/11/2013 11:54	8606
Boulder	DC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	289	7/8/2014 3:45	5/10/2013 14:54	8606
Boulder	DC1	DC=child,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	3738	7/8/2014 4:23	5/10/2013 15:40	8606
Boulder	DC1	DC=treeroot,DC=fabrikam,DC=com	Boulder	TRDC1	380	7/8/2014 4:00	(never)	8606
Boulder	CHILDDC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC1	278	7/8/2014 3:59	5/10/2013 14:51	8606
Boulder	CHILDDC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	TRDC1	278	7/8/2014 3:59	5/10/2013 14:51	8606
Boulder	CHILDDC1	DC=root,DC=contoso,DC=com	Boulder	DC1	294	7/8/2014 3:59	5/10/2013 14:51	8606
Boulder	CHILDDC1	DC=root,DC=contoso,DC=com	Boulder	DC2	1146	7/8/2014 4:20	(never)	8606
Boulder	CHILDDC1	DC=treeroot,DC=fabrikam,DC=com	Boulder	TRDC1	487	7/8/2014 4:00	5/10/2013 15:07	8606
Boulder	DC2	DC=root,DC=contoso,DC=com	Boulder	DC1	271	7/8/2014 3:53	5/20/2013 13:00	8606
Boulder	DC2	DC=DomainDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC1	289	7/8/2014 3:53	5/20/2013 13:00	8606
Boulder	DC2	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC1	261	7/8/2014 3:53	5/20/2013 13:00	8606
Boulder	DC2	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	TRDC1	283	7/8/2014 3:53	5/10/2013 15:00	8606
Boulder	DC2	DC=child,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	3363	7/8/2014 4:23	(never)	8606

Boulder	DC2	DC=treeroot,DC=fabrikam,DC=com	Boulder	TRDC1	491	7/8/2014 4:00	5/10/2013 15:08	8606
Boulder	TRDC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	DC2	282	7/8/2014 3:46	5/10/2013 14:54	8606
Boulder	TRDC1	DC=ForestDnsZones,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	281	7/8/2014 3:46	5/10/2013 14:54	8606
Boulder	TRDC1	DC=root,DC=contoso,DC=com	Boulder	DC1	189	7/8/2014 3:46	(never)	8606
Boulder	TRDC1	DC=root,DC=contoso,DC=com	Boulder	DC2	1266	7/8/2014 4:20	5/10/2013 15:28	8606
Boulder	TRDC1	DC=child,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	3429	7/8/2014 4:23	5/10/2013 15:40	8606
Boulder	CHILDDC2	DC=root,DC=contoso,DC=com	Boulder	DC1	202	7/8/2014 4:00	(never)	8606
Boulder	CHILDDC2	DC=root,DC=contoso,DC=com	Boulder	DC2	1138	7/8/2014 4:20	(never)	8606
Boulder	CHILDDC2	DC=child,DC=root,DC=contoso,DC=com	Boulder	CHILDDC1	3455	7/8/2014 4:23	5/20/2013 13:18	8606

Lab reproduction steps

Individual steps for each scenario are present in the C:\Files\ Lingering_object_lab_repro_stepsv2.xlsx file on the Win8Client

Scenarios

- Lingering Objects
    - Users
    - Trust account
    - Dns records
    - CNF mangled
    - Lost and Found
    - Abandoned create
    - Abandoned delete
- Lingering objects that have child objects that are lingering
    - Abandoned created

      -a lingering object that has an abandoned created child object
    - Normal Lingering
        - Contoso OID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671
- Lingering Links
- Replication failure after failed intraforest user migration

Abandoned creation / deletion

1. Create normal users and users that will contain child objects and replicate out
2. Take a snapshot of DC1
3. Pause DC2
4. delete user objects
5. create child objects of other user objects
6. create child objects of those child objects
7. create regular user objects

8. Replicate out to GCs
9. Restore snapshot
10. Resume DC2

Lingering Links

Lingering Links

Scenario 1

1. Create new domain
2. Create users in new domain
3. Add users to groups in other domains: root, child and TR
4. Syncall + verify group membership via replication metadata
5. Sever replication and shutoff DC in third domain
6. Disable replication between DCs in group's domain and GCs
7. Metadata cleanup of 3rd domain - verify group membership is removed
8. Advance time beyond TSL
9. Verify member's absent value is removed
10. Re-enable replication

Scenario 2

1. Add users from each domain to universal groups in lingering links OU in each domain
2. Force ad replication of group membership -document membership
3. Sever replication between all GCs
4. Modify group membership on each group by removing users from other domains
5. Advance time and verify absent values are no longer present

CNF

Two scenarios:

Create an OU with "conflicted" in title

- Disable replication between DC1 and DC2
- Create same named objects on both DCs
- Re-enable replication
- Delete objects that are non-CNF mangled, Disable replication, Delete CNF mangled objects from one DC only, advance time, garbage collection

- Create objects and replicate
- Disable replication to GC's
- Delete users, advance time, garbage collection
- Create users with same name in the same OU

Lost and Found

Two scenarios

- Create objects in new OU and replicate to all
- Disable replication to GCs

- Delete users, advance time, garbage collection, re-enable replication
- Delete OU

- Create special OU and replicate
- Disable replication between DC1 and DC2
- Create users in OU on one DC, delete the OU on the other DC
- Re-enable outbound replication on DC where OU was deleted and replicate
- Re-enable replication

Trust account

- Create forest trust
- Disable replication to all
- Remove trust, advance time, garbage collect, reenable replication

DNS

Create records in domaindns zones

For /L %i in (1,1,100) do dnscmd childdc1 /recordadd child.root.contoso.com win7pc%i A 172.16.15.%i

For /L %i in (1,1,100) do dnscmd dc1 /recordadd root.contoso.com win8pc%i A 172.16.14.%i

Disable repl, delete records, advance time and garbage collect

Enable replication

Child objects

Create child objects of users

Create child objects of those child objects

Disable replication, delete objects, advance time / garbage collection

Order of operations

1. **Disable replication with GCs**

Delete forest trust

Delete users for CNF scenario #2

Delete non CNF users for CNF scenario #1

2. **Disable Replication with everyone**
   a. DC1: delete users and DNS records
   b. DC2: delete users and DNS records

Lab setup

FourthCoffee.com

FourthDC1.FourthCoffee.com

```
FOR /L %i in (1,1,100) DO dsadd user "cn=LLUser%i,OU=lingeringlink,DC=fourthcoffee,DC=com" -samid
LingeringLink%i -upn LingeringLink%i@fourthcoffee.com -fn LL -ln User%i -display "Lingering
Link%i" -empid 04100%i -pwd P@ssw0rd -disabled no
```

Root groups

```
New-ADOrganizationalUnit -Name:"SingleSignOn" -Path:"DC=root,DC=contoso,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"dc1.root.contoso.com"

New-ADOrganizationalUnit -Name:"Mangle" -Path:"DC=root,DC=contoso,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"dc1.root.contoso.com"
```

```
New-ADGroup -GroupCategory:"Security" -GroupScope:"Universal" -Name:"LingeringLinkGroup5" -
Path:"OU=Lingering Links,DC=root,DC=contoso,DC=com" -SamAccountName:"LingeringLinkGroup5" -
Server:"DC1.root.contoso.com"
```

Child groups

```
New-ADOrganizationalUnit -Name:"LingeringLinkgroups" -Path:"DC=child,DC=root,DC=contoso,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"ChildDC1.child.root.contoso.com"
```

```
New-ADGroup -GroupCategory:"Security" -GroupScope:"Universal" -Name:"LLGroup1" -
Path:"OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com" -SamAccountName:"LLGroup1" -
Server:"ChildDC1.child.root.contoso.com"
```

```
New-ADGroup -GroupCategory:"Security" -GroupScope:"Universal" -Name:"LLinkGroup1" -Path:"OU=Lingering
Links,DC=child,DC=root,DC=contoso,DC=com" -SamAccountName:"LingeringLinkGroup1" -
Server:"ChildDC1.child.root.contoso.com"
```

TR groups

```
New-ADOrganizationalUnit -Name:"LLinkgroups" -Path:"DC=treeroot,DC=fabrikam,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"trdc1.treeroot.fabrikam.com"
```

```
New-ADGroup -GroupCategory:"Security" -GroupScope:"Universal" -Name:"LLinkGroup1" -
Path:"OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com" -SamAccountName:"LLinkGroup1" -
Server:"trdc1.treeroot.fabrikam.com"
```

FourthCoffee groups

```
New-ADOrganizationalUnit -Name:"LingeringLink" -Path:"DC=fourthcoffee,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"fourthdc1.fourthcoffee.com"

New-ADOrganizationalUnit -Name:"HumanResources" -Path:"DC=fourthcoffee,DC=com" -
ProtectedFromAccidentalDeletion:$true -Server:"fourthdc1.fourthcoffee.com"
```

```
New-ADGroup -GroupCategory:"Security" -GroupScope:"Universal" -Name:"LingeringLinkGroup1" -
Path:"OU=LingeringLink,DC=fourthcoffee,DC=com" -SamAccountName:"LingeringLinkGroup1" -
Server:"fourthdc1.fourthcoffee.com"
```

Import group members using ldifde

```
Ldifde -I -f c:\member.txt
```

Failed migration

- Lingering objects
  - t-2: create **lingering 1 -100**, replicate, sever connection, delete users, advance time, purge objects, reestablish replication

DC1:

```
FOR /L %i in (1,1,100) DO dsadd user "cn=Lingering User%i,OU=lingering,DC=root,DC=contoso,DC=com"
-samid Lingeringuser%i -upn Lingeringuser%i@contoso.com -fn Blue -ln User%i -display "Lingering
User%i" -empid 00100%i -pwd P@ssw0rd -disabled no
```

ChildDC1:

```
FOR /L %i in (1,1,100) DO dsadd user "cn=Lingering
User%i,OU=lingering,DC=child,DC=root,DC=contoso,DC=com" -samid Lingering%i -upn
Lingering%i@contoso.com -fn Lingering -ln User%i -display "Lingering%i" -empid 00200%i -pwd
P@ssw0rd -disabled no
```

TRDC1:

```
FOR /L %i in (1,1,100) DO dsadd user "cn=Lingering
User%i,OU=lingering,DC=treeroot,DC=fabrikam,DC=com" -samid Lingering%i -upn
Lingering%i@fabrikam.com -fn FabLingering -ln User%i -display "FabLingering%i" -empid 00300%i -pwd
P@ssw0rd -disabled no
```

- Abandoned objects - create and delete
  - Abandoned delete:
    Take snapshot of dc1, t-2: create **abandonedDel 1-100**, replicate out, pause DC2, delete objects on DC1 and replicate to GCs, restore snapshot of dc1, resume dc2
  - Abandoned create:
  - Take snapshot of dc1, Pause dc2, on dc1 t-2: create **abandoned 1 - 100**, replicate to GCs, restore snapshot of dc1, resume dc2

- CNF objects that are lingering (t -2: **create objects**, replicate, sever connections, delete objects, advance time to t -1 purge objects, create new with same name, reestablish replication
  - Lingering on one or more DCs
  - Create with same name on writable
  - Replicate to destination lingering
    Alternate:
    disable repl between dc1 and dc2
    Create objects with the same name on both: mangle 1 -10
    Re-establish replication, syncall
    Disable repl to GCs

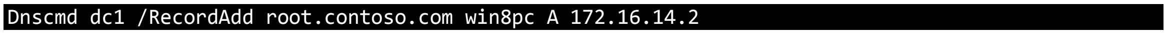
Delete CNF objects on writables, advance time, purge objects
Reestablish replication
Create same named objects

- Lingering objects in RWNC, RONC and DNS records that are lingering
- Lingering objects that are child of other lingering objects
    - t-2: **Add child objects** and replicate
    - Make both objects lingering
- System owned lingering objects - CROSSREF, TDO, NTDS Settings
- Failed migration with DS busy error

LostAndFound lingering objects?
t-2: Create landfuser1 - 100 in ou called X, outbound replicate, sever repl with gcs, delete users but don't delete OU, advance time, purge objects, re-establish connection, delete OU

Lingering Object (8606)
Root:
Lingering objects that only exist on GCs copy of RO partition
Lingering objects that exist on DC2 - that are different from the ones on DC1
Lingering objects that exist on DC1 - that are different from the ones on DC2

domainDNSZones - Root: lingering only on one DC
forestDNSzones - lingering on all but DC1



```
Dnscmd dc1 /RecordAdd root.contoso.com win8pc A 172.16.14.2
```

TreeRoot:

Hyper-v host time changes to a time beyond TSL (in the past) ->result all Hyper-v guests configured for host time synchronization change their clock as well (this is the default configuration for hyper-v) stop and start vmictimesync to force a sync

1. Disable host time synchronization on all VMs
   Disable-VMIntegrationService -Name "Time Synchronization" -vmname adrepl*
2. Fix Hyper-v Host time (all guests are still using old time)

   Create regular user objects that will become lingering later on

   Lingering1 - lingering 100
   Replicate to all DCs

   Abandoned objects
   Stop replication or pause VM DC for one RW replica
   Create user objects: abandoned1 - abandoned100
   Outbound replicate to DCs with RO NC

3. Create user objects on DC1 at this time in the past
4. Move users to Engineering OU

5. Force replication out-> this replicates all new users to DCs in the forest
6. Disable machine account password change on DCs in child and treeroot
7. Pause all VMs other than DC1
8. On DC1, Delete one or more user objects
9. Fix time on DC1, and then force garbage collection
   enable-VMIntegrationService -Name "Time Synchronization" -vmname adrepl*
10. Shutdown DC1, resume other DCs
11. Fix time on remaining DCs and then shutdown
12. Power on all DCs
13. Make changes to one or more user objects (that were deleted from DC1 in step 8) on DC2

Members.txt

```
dn: CN=LLGroup5,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com
changetype: modify
replace: member
member: CN=Lingering User100,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User19,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User18,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User17,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User16,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User15,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User14,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User13,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User12,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User11,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User10,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Lingering User1,OU=Lingering,DC=root,DC=contoso,DC=com
member: CN=Anastasia Delacruz,OU=Engineering,DC=root,DC=contoso,DC=com
member: CN=Bobbie Cazares,OU=Engineering,DC=root,DC=contoso,DC=com
member: CN=Caleb Grider,OU=Engineering,DC=root,DC=contoso,DC=com
member: CN=Lingering User2,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User29,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User28,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User27,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User26,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User25,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User24,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User23,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User22,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User21,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User20,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com
member: CN=Art Cowles,OU=Sales,DC=child,DC=root,DC=contoso,DC=com
member: CN=Becker Roddy,OU=Sales,DC=child,DC=root,DC=contoso,DC=com
member: CN=Chase Buie,OU=Sales,DC=child,DC=root,DC=contoso,DC=com
member: CN=Lingering User39,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User38,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User37,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User36,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User35,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User34,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User33,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
```

```
member: CN=Lingering User32,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User31,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User30,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Lingering User3,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com
member: CN=Aaliyah Franklin,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com
member: CN=Bill Cameron,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com
member: CN=Charlie Wright,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com
member: CN=LLUser49,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser48,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser47,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser46,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser45,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser44,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser43,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser42,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser41,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser40,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=LLUser4,OU=LingeringLink,DC=fourthcoffee,DC=com
member: CN=Chloe Woodcock,OU=HumanResources,DC=fourthcoffee,DC=com
member: CN=Aaron Knaggs,OU=HumanResources,DC=fourthcoffee,DC=com
member: CN=Benjamin Springthorpe,OU=HumanResources,DC=fourthcoffee,DC=com
```

-

#

```
# Windows PowerShell script for AD DS Deployment

#


Import-Module ADDSDeployment

Install-ADDSForest `

-CreateDnsDelegation:$false `

-DatabasePath "C:\Windows\NTDS" `

-DomainMode "Win2012R2" `

-DomainName "root.contoso.com" `

-DomainNetbiosName "ROOT" `

-ForestMode "Win2012R2" `

-InstallDns:$true `

-LogPath "C:\Windows\NTDS" `

-NoRebootOnCompletion:$false `

-SysvolPath "C:\Windows\SYSVOL" `

-Force:$true
```

Lingering Link groups and members

OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup2,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup3,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup4,OU=Lingering Links,DC=root,DC=contoso,DC=com

CN=LingeringLinkGroup5,OU=Lingering Links,DC=root,DC=contoso,DC=com

OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

CN=LLGroup1,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

CN=LLGroup2,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

CN=LLGroup3,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

CN=LLGroup4,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

CN=LLGroup5,OU=LingeringLinkgroups,DC=child,DC=root,DC=contoso,DC=com

OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

CN=LLinkGroup1,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

CN=LLinkGroup2,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

CN=LLinkGroup3,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

CN=LLinkGroup4,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

CN=LLinkGroup5,OU=LLinkgroups,DC=treeroot,DC=fabrikam,DC=com

```
CN=LingeringLinkgroup1,OU=Lingering Links,DC=root,DC=contoso,DC=com

   57> member: CN=Chloe Woodcock,OU=HumanResources,DC=fourthcoffee,DC=com;
       CN=Aaron Knaggs,OU=HumanResources,DC=fourthcoffee,DC=com;
       CN=Benjamin Springthorpe,OU=HumanResources,DC=fourthcoffee,DC=com;
       CN=Aaliyah Franklin,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com;
       CN=Charlie Wright,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com;
       CN=Bill Cameron,OU=Marketing,DC=treeroot,DC=fabrikam,DC=com;
       CN=Art Cowles,OU=Sales,DC=child,DC=root,DC=contoso,DC=com;
       CN=Chase Buie,OU=Sales,DC=child,DC=root,DC=contoso,DC=com;
       CN=Becker Roddy,OU=Sales,DC=child,DC=root,DC=contoso,DC=com;
       CN=Lingering User39,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User38,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User37,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User36,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User35,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User34,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User33,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User32,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User31,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User30,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=Lingering User3,OU=Lingering,DC=treeroot,DC=fabrikam,DC=com;
       CN=LLUser49,OU=LingeringLink,DC=fourthcoffee,DC=com;
       CN=LLUser48,OU=LingeringLink,DC=fourthcoffee,DC=com;
```

```
        CN=LLUser47,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser46,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser45,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser44,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser43,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser42,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser41,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser40,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=LLUser4,OU=LingeringLink,DC=fourthcoffee,DC=com;
        CN=Lingering User29,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User28,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User27,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User26,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User25,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User24,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User23,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User22,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User21,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User20,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User2,OU=Lingering,DC=child,DC=root,DC=contoso,DC=com;
        CN=Lingering User100,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User19,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User18,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User17,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User16,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User15,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User14,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User13,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User12,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User11,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User10,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Lingering User1,OU=Lingering,DC=root,DC=contoso,DC=com;
        CN=Anastasia Delacruz,OU=SingleSignOn,DC=root,DC=contoso,DC=com;
        CN=Bobbie Cazares,OU=Engineering,DC=root,DC=contoso,DC=com;
        CN=Caleb Grider,OU=Engineering,DC=root,DC=contoso,DC=com
```

Child Objects

Contoso Single Sign-On schema extension

Contososinglesignon.ldf

```
dn: CN=contoso-SSOSecretData,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
changetype: add
adminDescription: Contoso Password Manager Secret Data
adminDisplayName: contoso-SSOSecretData
attributeID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671.1.1.0
schemaIDGUID:: s38J1FsV40SrXRCPwNzjKw==
attributeSyntax: 2.5.5.4
cn: contoso-SSOSecretData
instanceType: 4
isSingleValued: TRUE
lDAPDisplayName: contoso-SSOSecretData
distinguishedName: CN=citrix-SSOSecretData,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectClass: attributeSchema
oMSyntax: 20
name: contoso-SSOSecretData
showInAdvancedViewOnly: TRUE
rangeUpper: 256000
```

```
dn: CN=contoso-SSOConfigData,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
changetype: add
adminDescription: Contoso Password Manager Configuration Data
adminDisplayName: contoso-SSOConfigData
attributeID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671.1.2.0
schemaIDGUID:: ah+QbMhOVUeaPnPbXNxe5w==
attributeSyntax: 2.5.5.4
cn: contoso-SSOConfigData
instanceType: 4
isSingleValued: TRUE
lDAPDisplayName: contoso-SSOConfigData
distinguishedName: CN=contoso-SSOConfigData,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectClass: attributeSchema
oMSyntax: 20
name: contoso-SSOConfigData
showInAdvancedViewOnly: TRUE
rangeUpper: 256000

dn: CN=contoso-SSOConfigType,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
changetype: add
adminDescription: Contoso Password Manager Configuration Data Type
adminDisplayName: contoso-SSOConfigType
attributeID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671.1.3.0
schemaIDGUID:: HcLYTj/4dkqlQwLMYPQF3w==
attributeSyntax: 2.5.5.4
cn: contoso-SSOConfigType
instanceType: 4
isSingleValued: TRUE
lDAPDisplayName: contoso-SSOConfigType
distinguishedName: CN=contoso-SSOConfigType,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectClass: attributeSchema
oMSyntax: 20
name: contoso-SSOConfigType
showInAdvancedViewOnly: TRUE
rangeUpper: 256000

DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-

dn: CN=contoso-SSOSecret,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
changetype: add
adminDescription: Contoso Password Manager Secret Object
adminDisplayName: contoso-SSOSecret
cn: contoso-SSOSecret
defaultObjectCategory: CN=contoso-SSOSecret,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
defaultSecurityDescriptor:
D:(A;;RPWPCRCCDCLCLOLORCWOWDSDDTDTSW;;;DA)(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;CO)
governsID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671.2.1.0
schemaIDGUID:: yESC4AFQf0aCU3Su4mDtfQ==
instanceType: 4
lDAPDisplayName: contoso-SSOSecret
mayContain: contoso-SSOSecretData
distinguishedName: CN=contoso-SSOSecret,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
```

```
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: user
name: contoso-SSOSecret
rDNAttID: cn
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE

dn: CN=contoso-SSOConfig,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
changetype: add
adminDescription: Contoso Password Manager Configuration Object
adminDisplayName: contoso-SSOConfig
cn: contoso-SSOConfig
defaultObjectCategory: CN=contoso-SSOConfig,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
defaultSecurityDescriptor:
D:(A;;RPWPCRCCDCLCLOLORCWOWDSDDTDTSW;;;DA)(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;CO)(A;;RPLCLORC;;;A
U)
governsID: 1.2.840.113556.1.8000.2554.4400.22918.14797.18250.39931.11340655.16430671.2.2.0
schemaIDGUID:: ijUz397qkEGJMXFPj7oVmA==
instanceType: 4
lDAPDisplayName: contoso-SSOConfig
mayContain: contoso-SSOConfigData
mayContain: contoso-SSOConfigType
distinguishedName: CN=contoso-SSOConfig,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: organizationalUnit
possSuperiors: container
possSuperiors: user
possSuperiors: domainDNS
name: contoso-SSOConfig
rDNAttID: cn
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE

DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-
```

```
OU=SingleSignOn,DC=root,DC=contoso,DC=com
CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+CN=Jul,CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++CN=SecretData,CN=Jul,CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=JulLan,CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+CN=SecretData,CN=JulLan,CN=Juliette Lancaster,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=Anastasia Delacruz,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+CN=Ana,CN=Anastasia Delacruz,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++CN=SecretData,CN=Ana,CN=Anastasia Delacruz,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=Antonio Boatwright,OU=SingleSignOn,DC=root,DC=contoso,DC=com
```

```
+CN=Ant,CN=Antonio Boatwright,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++CN=SecretData,CN=Ant,CN=Antonio Boatwright,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=Carl Woodbury,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+CN=Car,CN=Carl Woodbury,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++CN=SecretData,CN=Car,CN=Carl Woodbury,OU=SingleSignOn,DC=root,DC=contoso,DC=com

CN=Cassie McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+CN=Cas,CN=Cassie McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++CN=McK,CN=Cas,CN=Cassie McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+++CN=Sie,CN=McK,CN=Cas,CN=Cassie McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
++++CN=Enzie,CN=Sie,CN=McK,CN=Cas,CN=Cassie McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
+++++CN=SecretData,CN=Enzie,CN=Sie,CN=McK,CN=Cas,CN=Cassie
McKenzie,OU=SingleSignOn,DC=root,DC=contoso,DC=com
```

Abandoned child object

```
C:\>repadmin /showattr * "<GUID=433fabf4-dce8-4c66-b70c-ef106ebadb2d>" /GC >ac.txt

C:\>repadmin /showobjmeta * "<GUID=433fabf4-dce8-4c66-b70c-ef106ebadb2d>" >>ac.txt

C:\>repadmin /showutdvec * dc=root,dc=contoso,dc=com >>ac.txt

C:\>repadmin /showutdvec * dc=root,dc=contoso,dc=com >>ac.txt

C:\>repadmin /showsig dc1 >>ac.txt
```

C:\>repadmin /showutdvec * dc=root,dc=contoso,dc=com /nocache >>ac.txt

```
Repadmin: running command /showattr against full DC DC1.root.contoso.com

Can not locate the object for this DN: <GUID=433fabf4-dce8-4c66-b70c-ef106ebadb2d>

Error: An LDAP lookup operation failed with the following error:



   LDAP Error 32(0x20): No Such Object

   Server Win32 Error 8333(0x208d): Directory object not found.

   Extended Information: 0000208D: NameErr: DSID-03100213, problem 2001 (NO_OBJECT), data 0, best match of:
          ''




Repadmin: running command /showattr against full DC ChildDC1.child.root.contoso.com

DN: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com

   2> objectClass: top; classStore
   1> cn: UlyStore
   1> distinguishedName: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com
   1> instanceType: 0x0 = (  )
   1> whenCreated: 5/10/2013 3:51:34 AM Pacific Daylight Time
   1> whenChanged: 5/10/2013 3:51:45 AM Pacific Daylight Time
   1> uSNCreated: 152092
   1> uSNChanged: 152092
   1> name: UlyStore
   1> objectGUID: 433fabf4-dce8-4c66-b70c-ef106ebadb2d
   1> objectCategory: CN=Class-Store,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
   1> dSCorePropagationData: 0x0 = (  )


Repadmin: running command /showattr against full DC DC2.root.contoso.com

Can not locate the object for this DN: <GUID=433fabf4-dce8-4c66-b70c-ef106ebadb2d>

Error: An LDAP lookup operation failed with the following error:



   LDAP Error 32(0x20): No Such Object

   Server Win32 Error 8333(0x208d): Directory object not found.

   Extended Information: 0000208D: NameErr: DSID-03100213, problem 2001 (NO_OBJECT), data 0, best match of:
           ''




Repadmin: running command /showattr against full DC TRDC1.treeroot.fabrikam.com

DN: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com

   2> objectClass: top; classStore
   1> cn: UlyStore
   1> distinguishedName: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com
   1> instanceType: 0x0 = (  )
   1> whenCreated: 5/10/2013 3:51:34 AM Pacific Daylight Time
   1> whenChanged: 5/10/2013 3:51:42 AM Pacific Daylight Time
   1> uSNCreated: 187232
   1> uSNChanged: 187232
   1> name: UlyStore
   1> objectGUID: 433fabf4-dce8-4c66-b70c-ef106ebadb2d
   1> objectCategory: CN=Class-Store,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
   1> dSCorePropagationData: 0x0 = (  )


Repadmin: running command /showattr against read-only DC CHILDDC2.child.root.contoso.com

LDAP error 81 (Server Down) Win32 Err 58.



Repadmin: running command /showattr against full DC FourthDC1.fourthcoffee.com

DN: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com

   2> objectClass: top; classStore
```

```
    1> cn: UlyStore
    1> distinguishedName: CN=UlyStore,CN=Ulysses Breland,OU=SingleSignOn,DC=root,DC=contoso,DC=com
    1> instanceType: 0x0 = (  )
    1> whenCreated: 5/10/2013 3:51:34 AM Pacific Daylight Time
    1> whenChanged: 5/10/2013 3:51:48 AM Pacific Daylight Time
    1> uSNCreated: 69530
    1> uSNChanged: 69530
    1> name: UlyStore
    1> objectGUID: 433fabf4-dce8-4c66-b70c-ef106ebadb2d
    1> objectCategory: CN=Class-Store,CN=Schema,CN=Configuration,DC=root,DC=contoso,DC=com
    1> dSCorePropagationData: 0x0 = (  )


Repadmin: running command /showobjmeta against full DC DC1.root.contoso.com

DsReplicaGetInfo() failed with status 8439 (0x20f7):

    The distinguished name specified for this replication operation is invalid.



Repadmin: running command /showobjmeta against full DC ChildDC1.child.root.contoso.com



7 entries.

Loc.USN                           Originating DSA  Org.USN  Org.Time/Date        Ver Attribute

=======                           =============== ========= =============        === =========

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectClass

 152092                         Boulder\CHILDDC1    152092 2013-05-10 03:51:45    1 cn

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 instanceType

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 whenCreated

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 nTSecurityDescriptor

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 name

 152092     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectCategory

0 entries.

Repadmin: running command /showobjmeta against full DC DC2.root.contoso.com

DsReplicaGetInfo() failed with status 8439 (0x20f7):

    The distinguished name specified for this replication operation is invalid.



Repadmin: running command /showobjmeta against full DC TRDC1.treeroot.fabrikam.com
```

```
7 entries.

Loc.USN                          Originating DSA  Org.USN  Org.Time/Date        Ver Attribute

=======                          =============== ========= =============        === =========

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectClass

 187232                          Boulder\TRDC1    187232 2013-05-10 03:51:42    1 cn

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 instanceType

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 whenCreated

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 nTSecurityDescriptor

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 name

 187232     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectCategory

0 entries.

Repadmin: running command /showobjmeta against read-only DC CHILDDC2.child.root.contoso.com


7 entries.

Loc.USN                          Originating DSA  Org.USN  Org.Time/Date        Ver Attribute

=======                          =============== ========= =============        === =========

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectClass

 193865                        Boulder\CHILDDC2    193865 2013-05-10 03:52:00    1 cn

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 instanceType

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 whenCreated

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 nTSecurityDescriptor

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 name

 193865     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectCategory

0 entries.


Repadmin: running command /showobjmeta against full DC FourthDC1.fourthcoffee.com


7 entries.
```

```
Loc.USN                      Originating DSA  Org.USN  Org.Time/Date       Ver Attribute

=======                      =============== ========= =============       === =========

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectClass

  69530                     Boulder\FOURTHDC1     69530 2013-05-10 03:51:48    1 cn

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 instanceType

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 whenCreated

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 nTSecurityDescriptor

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 name

  69530     9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05    220882 2013-05-10 03:51:34    1 objectCategory

0 entries.

Repadmin: running command /showutdvec against full DC DC1.root.contoso.com

Caching GUIDs.

..

Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24

Boulder\DC1                       @ USN     40967 @ Time 2014-05-09 08:08:38

Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01

Boulder\DC1 (retired)             @ USN    220838 @ Time 2013-05-10 03:33:55

Boulder\DC2                       @ USN    152523 @ Time 2013-05-10 03:58:41

Boulder\DC1                       @ USN    221120 @ Time 2013-05-10 04:06:04


Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com

Caching GUIDs.

..

Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24

Boulder\FOURTHDC1                 @ USN     69616 @ Time 2013-05-10 04:07:37

Boulder\CHILDDC1                  @ USN    152375 @ Time 2013-05-10 04:09:09

Boulder\DC1                       @ USN     40967 @ Time 2014-05-09 08:08:38

Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01
```

```
Boulder\TRDC1                     @ USN    187429 @ Time 2013-05-10 04:08:11

Boulder\TRDC1 (retired)           @ USN     32774 @ Time 2014-05-09 08:12:13

Boulder\DC1 (retired)             @ USN    221009 @ Time 2013-05-10 04:09:59

Boulder\CHILDDC1 (retired)        @ USN     40966 @ Time 2014-05-09 08:12:07

Boulder\DC2                       @ USN    152523 @ Time 2013-05-10 03:58:41

Boulder\DC1                       @ USN    221116 @ Time 2013-05-10 04:05:32


Repadmin: running command /showutdvec against full DC DC2.root.contoso.com

Caching GUIDs.

..

Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24

Boulder\DC1                       @ USN     40967 @ Time 2014-05-09 08:08:38

Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01

Boulder\DC1 (retired)             @ USN    220838 @ Time 2013-05-10 03:33:55

Boulder\DC2                       @ USN    152568 @ Time 2013-05-10 04:06:04

Boulder\DC1                       @ USN    221116 @ Time 2013-05-10 04:05:26


Repadmin: running command /showutdvec against full DC TRDC1.treeroot.fabrikam.com

Caching GUIDs.

..

Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24

Boulder\FOURTHDC1                 @ USN     69726 @ Time 2013-05-10 04:52:14

Boulder\CHILDDC1                  @ USN    152364 @ Time 2013-05-10 04:06:45

Boulder\DC1                       @ USN     40967 @ Time 2014-05-09 08:08:38

Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01

Boulder\TRDC1                     @ USN    187703 @ Time 2013-05-10 04:54:38

Boulder\TRDC1 (retired)           @ USN     32774 @ Time 2014-05-09 08:12:13

Boulder\DC1 (retired)             @ USN    221009 @ Time 2013-05-10 04:09:59

Boulder\CHILDDC1 (retired)        @ USN     40966 @ Time 2014-05-09 08:12:07
```

```
Boulder\DC2                   @ USN    152560 @ Time 2013-05-10 04:03:39
Boulder\DC1                   @ USN    221116 @ Time 2013-05-10 04:05:29


Repadmin: running command /showutdvec against read-only DC CHILDDC2.child.root.contoso.com
Caching GUIDs.
..
Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24
1a9bacc0-e8d0-4659-9634-ec9335d7ec1d @ USN     32773 @ Time 2014-05-09 08:12:05
Boulder\FOURTHDC1                 @ USN     69616 @ Time 2013-05-10 04:07:54
Boulder\CHILDDC1                  @ USN    152364 @ Time 2013-05-10 04:06:35
Boulder\DC1                   @ USN     40967 @ Time 2014-05-09 08:08:38
Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01
Boulder\CHILDDC2                  @ USN    194185 @ Time 2013-05-10 04:54:38
Boulder\TRDC1                   @ USN    187429 @ Time 2013-05-10 04:08:14
Boulder\TRDC1 (retired)           @ USN     32774 @ Time 2014-05-09 08:12:13
Boulder\DC1 (retired)             @ USN    221009 @ Time 2013-05-10 04:09:59
Boulder\CHILDDC1 (retired)          @ USN     40966 @ Time 2014-05-09 08:12:07
Boulder\DC2                   @ USN    152523 @ Time 2013-05-10 03:58:41
Boulder\DC1                   @ USN    221105 @ Time 2013-05-10 04:03:24


Repadmin: running command /showutdvec against full DC FourthDC1.fourthcoffee.com
Caching GUIDs.
..
Boulder\DC2 (retired)             @ USN     45718 @ Time 2014-05-09 08:45:24
Boulder\FOURTHDC1                 @ USN     69734 @ Time 2013-05-10 04:06:04
Boulder\CHILDDC1                  @ USN    150335 @ Time 2013-05-09 15:55:29
Boulder\DC1                   @ USN     40967 @ Time 2014-05-09 08:08:38
Boulder\DC2 (retired)             @ USN     45063 @ Time 2014-05-09 08:12:01
Boulder\TRDC1                   @ USN    184821 @ Time 2013-05-09 15:55:29
```

```
Boulder\TRDC1 (retired)          @ USN     32774 @ Time 2014-05-09 08:12:13
Boulder\DC1 (retired)            @ USN    221009 @ Time 2013-05-10 04:09:59
Boulder\CHILDDC1 (retired)       @ USN     40966 @ Time 2014-05-09 08:12:07
Boulder\DC2                      @ USN    152523 @ Time 2013-05-10 03:58:41
Boulder\DC1                      @ USN    221116 @ Time 2013-05-10 04:05:35



Repadmin: running command /showutdvec against full DC DC1.root.contoso.com
08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38
80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01
9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    220838 @ Time 2013-05-10 03:33:55
c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152523 @ Time 2013-05-10 03:58:41
fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221121 @ Time 2013-05-10 04:07:03


Repadmin: running command /showutdvec against full DC ChildDC1.child.root.contoso.com
08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24
336d313f-cce1-4c52-a57e-1135d54985fa @ USN     69616 @ Time 2013-05-10 04:07:37
606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152379 @ Time 2013-05-10 04:10:09
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38
80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01
9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    187429 @ Time 2013-05-10 04:08:11
9a9e8c55-d7d2-4c31-bc04-25abec3765ca @ USN     32774 @ Time 2014-05-09 08:12:13
9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    221009 @ Time 2013-05-10 04:09:59
a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07
c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152523 @ Time 2013-05-10 03:58:41
fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221121 @ Time 2013-05-10 04:06:36


Repadmin: running command /showutdvec against full DC DC2.root.contoso.com
```

```
08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24

70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38

80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01

9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    220838 @ Time 2013-05-10 03:33:55

c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152572 @ Time 2013-05-10 04:07:03

fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221121 @ Time 2013-05-10 04:06:30


Repadmin: running command /showutdvec against full DC TRDC1.treeroot.fabrikam.com

08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24

336d313f-cce1-4c52-a57e-1135d54985fa @ USN     69726 @ Time 2013-05-10 04:52:14

606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152364 @ Time 2013-05-10 04:06:45

70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38

80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01

9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    187707 @ Time 2013-05-10 04:55:38

9a9e8c55-d7d2-4c31-bc04-25abec3765ca @ USN     32774 @ Time 2014-05-09 08:12:13

9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    221009 @ Time 2013-05-10 04:09:59

a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07

c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152560 @ Time 2013-05-10 04:03:39

fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221121 @ Time 2013-05-10 04:06:33


Repadmin: running command /showutdvec against read-only DC CHILDDC2.child.root.contoso.com

08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24

1a9bacc0-e8d0-4659-9634-ec9335d7ec1d @ USN     32773 @ Time 2014-05-09 08:12:05

336d313f-cce1-4c52-a57e-1135d54985fa @ USN     69616 @ Time 2013-05-10 04:07:54

606f5d34-7202-4073-83fb-aac8bb109868 @ USN    152364 @ Time 2013-05-10 04:06:35

70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38

80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01

849ba98a-a2bd-4d59-80b4-aeec3e8017af @ USN    194185 @ Time 2013-05-10 04:55:38

9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    187429 @ Time 2013-05-10 04:08:14
```

```
9a9e8c55-d7d2-4c31-bc04-25abec3765ca @ USN     32774 @ Time 2014-05-09 08:12:13
9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    221009 @ Time 2013-05-10 04:09:59
a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07
c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152523 @ Time 2013-05-10 03:58:41
fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221105 @ Time 2013-05-10 04:03:24


Repadmin: running command /showutdvec against full DC FourthDC1.fourthcoffee.com
08e1d906-2f72-447b-b4ab-fc24eeda7d21 @ USN     45718 @ Time 2014-05-09 08:45:24
336d313f-cce1-4c52-a57e-1135d54985fa @ USN     69738 @ Time 2013-05-10 04:07:03
606f5d34-7202-4073-83fb-aac8bb109868 @ USN    150335 @ Time 2013-05-09 15:55:29
70ff33ce-2f41-4bf4-b7ca-7fa71d4ca13e @ USN     40967 @ Time 2014-05-09 08:08:38
80afd2de-4153-433a-90ad-995564a80cf0 @ USN     45063 @ Time 2014-05-09 08:12:01
9a90d156-62ed-4ade-ac0a-4fda75e61d22 @ USN    184821 @ Time 2013-05-09 15:55:29
9a9e8c55-d7d2-4c31-bc04-25abec3765ca @ USN     32774 @ Time 2014-05-09 08:12:13
9dd76ca7-cb99-4ce0-a54c-d9e6900d7d05 @ USN    221009 @ Time 2013-05-10 04:09:59
a0c80b91-8247-41ca-a3a3-c40a1094b4a6 @ USN     40966 @ Time 2014-05-09 08:12:07
c82c058e-5aa8-49ba-a312-8e7e6b280df4 @ USN    152523 @ Time 2013-05-10 03:58:41
fef36435-b9b7-4ab9-afa2-c788ed12354c @ USN    221121 @ Time 2013-05-10 04:06:39
```