Microsoft

Offline Assessment for Active Directory

Prerequisites

All data collection and analysis is done locally on the tools machine.

No data is transported outside your Active Directory environment to help protect your data. Your data is analyzed using our RAP expert system that is part of the Offline Assessment client.

Internet connectivity is needed to:

- *Activate your account*
- *Download the toolset*

How to prepare for your Offline Assessment for Active Directory.

The Tools machine is used to connect to each of your Domain Controllers (DCs) and retrieve information from them, communicating over Remote Procedure Call (RPC), Server Message Block (SMB), Lightweight Directory Access Protocol (LDAP) and Distributed Component Object Model (DCOM).

Once the data is collected and the survey answered, the Offline Assessment tool will analyze the data locally.

At a high level, your steps to success are:

1. Install prerequisites on your Tools machine and configure your environment
2. Run discovery and prerequisites checks
3. Collect data from your DCs
4. Complete the survey

A checklist of prerequisite actions follows. Each item links to any additional software required for the Tools machine, and detailed steps included later in this document.

Checklist

Please ensure the following items have been completed before starting your engagement.

1. General Use

☐ A Microsoft Account is required to activate and sign in to the portal to download the toolset.
If you don't have one already, you can create one at http://login.live.com
To learn more about Microsoft Accounts, see: http://windows.microsoft.com/en-US/windows-live/sign-in-what-is-microsoft-account

☐ Ensure access to https://services.premier.microsoft.com

This document was last updated October 04, 2016. To ensure you have the latest version of this document , check here: http://download.microsoft.com/download/1/C/1/1C15BA51-840E-498D-86C6-4BD35D33C79E/Prerequisites_Offline_AD.pdf

2. Data Collection

a. Tools machine hardware and Operating System:

- □ Server-class or high-end workstation machine running Windows Vista/Windows7/Windows8/Windows 10, or Windows Server 2008/Windows Server 2008 R2/Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016.
- □ Minimum: 8GB RAM, 2Ghz dual-core processor, 10 GB of free disk space + an additional 2 GB of free disk space per one million users in the forest.
- □ Joined to one of the domains of the forest to be assessed.
- □ Using English (United States) locale setting for date and time formats.

b. Software for Tools machine:

- □ Microsoft .NET Framework 4.0 installed.
- □ Windows PowerShell 2.0 or later installed.

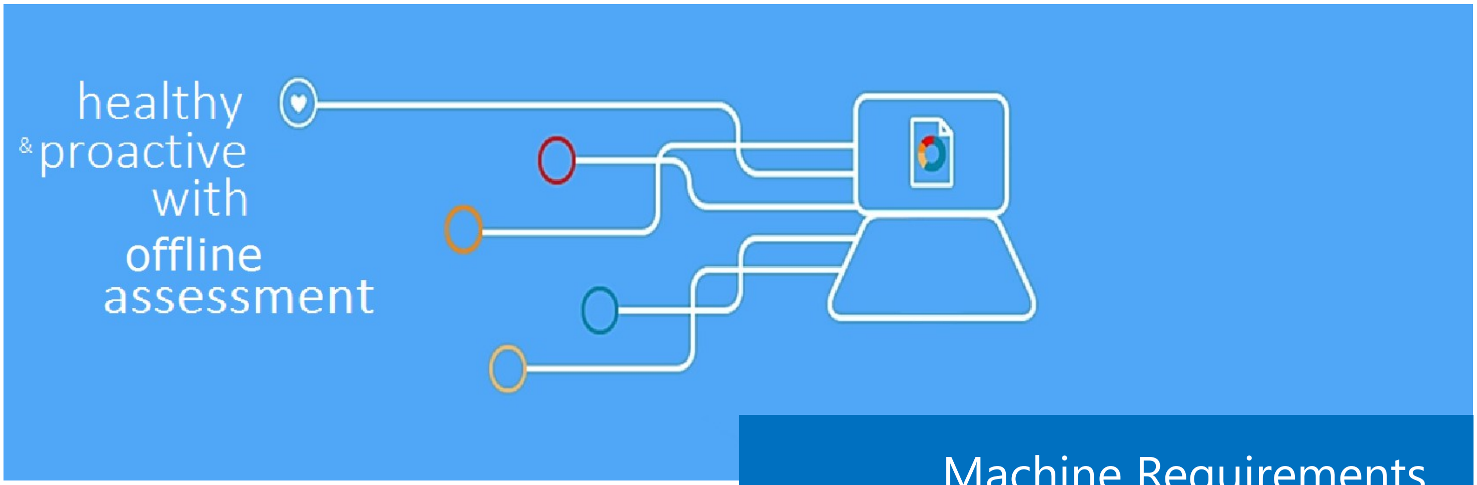
c. Account Rights:

- □ Enterprise Administrator account with Admin access to every DC in the forest.
- □ Unrestricted network access to every DC in the forest.

The Appendix Data Collection Methods details the methods used to collect data.

The rest of this document contains detailed information on the steps discussed above.

Once you have completed these prerequisites, you are ready to start the Offline Assessment.



Machine Requirements and Account Rights

1. Hardware and Software

Server-class or high-end workstation computer equipped with the following:

- Minimum single 2Ghz processor — Recommended dual-core/multi-core 2Ghz or higher processors.
- Minimum 8 GB RAM.
- Minimum 10 GB of free disk space + an additional 2 GB of free disk space per one million users in the forest.
- Windows Vista, Windows 7, Windows 8, Windows 10, Windows Server 2016, Windows Server 2012, Windows Server 2012 R2, Windows Server 2008, or Windows Server 2008 R2.
- Running 64-bit operating system.
- Using English (United States) locale setting for date and time formats.
- At least a 1024x768 screen resolution (higher preferred).
- Must be a member of the assessed AD Forest (member of the Forest Root Domain is preferred not but required)
- Microsoft® .NET Framework 4.0.— http://www.microsoft.com/en-us/download/details.aspx?id=17851
- Windows PowerShell 2.0 or higher
    - Windows PowerShell 2.0 is part of the Windows Management Framework — http://support.microsoft.com/kb/968929
- Networked “Documents” or redirected “Documents” folders are not supported. Local “Documents” folder on the data collection machine is required.

2. Accounts Rights

- A domain account with the following:
    - Enterprise Administrator
    - Administrative access to every DC in the forest.
    - Administrative access to all Microsoft Domain Name System (DNS) servers that the servers participate with.

  WARNING: Do not use the Run As feature to start OfflineAssessmentClient.exe. Some collectors might fail. The account starting the offline client must logon to the local machine.

- A Microsoft Account is required to activate and sign in to the Premier Proactive Assessment Services portal (https://services.premier.microsoft.com). This is where you where you will activate your access token and download the toolset.

  If you don’t have one already, you can create one at http://login.live.com
- Contact your TAM if the token in your Welcome Email has expired or can no longer be activated. Tokens expire after ten days. Your TAM can provide new activation tokens for additional people.

3. Network and Remote Access

- ♦ Short name resolution must work from the Tools machine. This typically means making sure DNS suffixes for all domains in the forest are added on the Tools machine.

- ♦ Unrestricted network access to every server in the environment
  - ∗ This means access through any firewalls, and router ACLs that might be limiting traffic to any DCs. This includes remote access to DCOM, Remote Registry service, Windows Management Instrumentation (WMI) services, and default administrative shares (C$, D$, IPC$).
  - ∗ Ensure that the machine you use to collect data has complete TCP/UDP access, including RPC access to all DCs. For a complete list of protocols, services and ports required by AD, see http://support.microsoft.com/kb/179442.

4. Garbage Collection Diagnostics (White Space) Logging (Optional but Recommended)

Diagnostic logging can be enabled for the garbage collection process so Active Directory IT staff knows how much white space exists in each DC's database. Although not mandatory, this information can be very useful in these scenarios:

∗ If the environment was upgraded from Windows Server 2000 to Windows Server 2003.

or

∗ If many objects have been deleted.

or

∗ If the DCs have existed for many years.

For more information on the Garbage Collection Process, see: http://support.microsoft.com/kb/198793

- ♦ To enable garbage collection diagnostics logging:
  - ∗ Change the following Registry value manually from **0** to **1**: **HKLM\System\CurrentControlSet\Services\NTDS\Diagnostics\6 Garbage Collection**
  - ∗ After the diagnostic logging has been enabled on a DC, it will generate an Event ID 1646 the next time garbage collection runs. By default, this occurs every 12 hours. No reboot or service restart is required for the change to take effect.
  - ∗ This option can be disable easily by resetting the Registry value to **0**. The Database Information test of the toolset will detect the existence of the Event ID 1646, read and parse the text, and then display the information in the portal
  - ∗ Sample Visual Basic (VB) code to enable Garbage Collection Diagnostics (White Space) Logging is mentioned in the next Section.

♦ Script to Enable Garbage Collection (White Space) logging on all DCs

- ∗ Copy the code on the next pages into a file called **EnableWhiteSpace.VBS.**

  Be aware to only copy the code and not page numbers.
- ∗ Run it using the following command: **cscript EnableWhiteSpace.VBS**

```
--  START COPY HERE ——
'************
'*** Init ***
'************
on error resume next

Set objRootDSE = GetObject("LDAP://RootDSE")
ConfigNC = objRootDSE.Get("configurationNamingContext")
RootNC = Replace(lcase(ConfigNC),"cn=configuration,","")

ObjCatDN = "CN=NTDS-DSA,CN=Schema," & ConfigNC
ObjCatDN2 = "CN=NTDS-DSA-RO,CN=Schema," & ConfigNC

const HKEY_LOCAL_MACHINE = &H80000002
const HKEY_CURRENT_USER = &H80000001

'************
'*** Main ***
'************
GetDCs
GetRODCs
'****************************
'*** Write Registry Value ***
'****************************
Function WriteRegistryValue(Hive,KeyPath,ValueName,RegValue,DNSHostName)

       Set oReg=GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & DNSHostName & "\root\default:StdRegProv")

       WriteRegistryValue=""
       oReg.SetDwordValue Hive,KeyPath,ValueName,RegValue
       WriteRegistryValue = err.number
       wscript.echo "rc: " & err.number
       wscript.echo ""

       Set oReg = Nothing

End Function

'***************
'*** Get DCs ***
'***************
Sub GetDCs

       LDAPWhereClause = " WHERE ObjectCategory='" & ObjCatDN & "'"
       LDAPAttributes = "DistinguishedName"
       FromClause = "GC://" & RootNC

       ProcessLDAPQuery FromClause,LDAPWhereClause,LDAPAttributes

End Sub
'*****************
'*** Get RODCs ***
'*****************
Sub GetRODCs

       LDAPWhereClause = " WHERE ObjectCategory='" & ObjCatDN2 & "'"
       LDAPAttributes = "DistinguishedName"
       FromClause = "GC://" & RootNC

       ProcessLDAPQuery FromClause,LDAPWhereClause,LDAPAttributes

End Sub

'**************************
'*** Process LDAP Query ***
```

```
'**************************
Sub ProcessLDAPQuery(FromClause,LDAPWhereClause,LDAPAttributes)

       ADS_SCOPE_SUBTREE = 2
       QueryString = "SELECT " & LDAPAttributes & " FROM '" & FromClause & "' " &
Trim(LDAPWhereClause )

       Dim oConnection, oCommand, oRecordset
       Set oConnection = CreateObject("ADODB.Connection")
       Set oCommand = CreateObject("ADODB.Command")

       oConnection.Provider = "ADsDSOObject"
       oConnection.Open "Active Directory Provider"

       Set oCommand.ActiveConnection = oConnection
       oCommand.CommandText = Trim(QueryString)
       oCommand.Properties("Page Size") = 1000
       oCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE

       Set oRecordset = oCommand.Execute
       'wscript.echo "QueryString: " & QueryString

       While (NOT oRecordset.EOF)

              ObjectDN=oRecordSet.Fields("DistinguishedName").Value

              'wscript.echo "DN: " & ObjectDN

              set objDC = GetObject("LDAP://" & Replace(ucase(ObjectDN),"CN=NTDS SETTINGS,",""))
              DNSHostname = objDC.DnsHostname

              wscript.echo "DC: " & DNSHostname

              RC = WriteRegistryValue
(HKEY_LOCAL_MACHINE,"System\CurrentControlSet\Services\NTDS\Diagnostics","6 Garbage
Collection",1,DNSHostName)

              oRecordset.moveNext

       wend
       set oConnection = Nothing
       set oCommand = Nothing
       set oRecordset = Nothing
       set objRootDSE = Nothing

End Sub

—-  END COPY HERE ——
```

Appendix: Data Collection Methods

Offline Assessment for Active Directory uses multiple data collection methods to collect information. This section describes the methods used to collect data from an Active Directory environment. No VB scripts are used to collect data. Data collection uses workflows and collectors. The collectors are:

1. Registry Collectors
2. LDAP Collectors
3. .NET Framework
4. EventLogCollector
5. Active Directory Service Interfaces (ADSI)
6. Windows PowerShell
7. FileDataCollector
8. WMI
9. DCDIAGAPI
10. NTFRSAPI
11. Custom C# Code

1. Registry Collectors

Registry keys and values are read from the data collection machine and all Domain Controllers. They include items such as:

- Service information from HKLM\SYSTEM\CurrentControlSet\Services.

  This allows to determine where the AD Database and log files are located on each DC and get detailed information on each service relevant to the proper function of AD. We do not collect all services, only the ones relevant to AD.
- Operating System information from HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion

  This allows to determine Operation System information such as Windows Server 2003, Windows Server 2008, Windows Server 2012 or Windows Server 2016.

2. LDAP Collectors

LDAP queries are used to collect data for the Domain, DCs, nTDSSiteSettings objects, Partitions and other components from AD itself. For a complete list of ports required by AD, see: http://support.microsoft.com/kb/179442.

3. .NET Framework

The Offline client leverages the System.DirectoryServices.ActiveDirectory .NET Framework Namespace and uses the following methods:

- GetReplicationNeighbors is called to retrieve the replication status details.
- Domain.GetAllTrustRelationships— to get a collection of the trust relationships in each domain.
- Forest.GetAllTrustRelationships— collection of the trust relationships of the forest.

4. EventLogCollector

Collects event logs from Domain Controllers. We collect the last 7 days of Warnings and Errors from the Application, Distributed File System Replication (DFSR), DNS, File Replication Service (FRS), and System event logs. Only for the Directory Services event log, we also collect informational events to detect the amount of white space in the database if whitespace logging has been enabled.

5. ADSI

Using the Domain ObjectClass, we use Active Directory Service Interfaces (ADSI) to get the domain password information for each domain in the forest. The domain password information consists of the domain's minimum password age, maximum password age, minimum password length, and other settings stored in the Default Domain Policy.

6. Windows PowerShell

Collects various information, such as:

- SYSVOL details which is looking for the content of the SYSVOL folder, determining file sizes and morphed folders (if they exist).

7. FileDataCollector

Enumerates files in a folder on a remote machine, and optionally retrieves those files.

8. Windows Management Instrumentation (WMI)

WMI is used to collect various information such as:

- WIN32_Volume
  Collects information on Volume Settings for each DC in the forest. The information is used for instance to determine the system volume and drive letter which allows the client to collect information on files located on the system drive.
- Win32_Process
  Collect information on the processes running on each DC in the forest. The information provides insight in processes that consume a large amount of threads, memory or have a large page file usage.
- Win32_LogicalDisk
  Used to collect information on the logical disks. We use the information to determine the amount of free space on the disk where the database or log files are located.

9. DCDIAGAPI

Collects diagnostics information from DCs. DCDIAG analyzes the state for all DCs in the forest and reports any problems it detects.

10. NTFRSAPI

File Replication Service (FRS) can be used to replicate the SYSVOL and Netlogon folder contents. The NTFRSapi is used to dump the internal tables, thread and memory information for the NT File Replication Service (NTFRS) for DCs. It provides insight in the health of the FRS.

11. Custom C# Code

Collects information not captured using other collectors.