Global Information Assurance Certification Paper

Securing Windows 20003 with ADAM and MIIS Feature Packs

GIAC Gold Certification

Author: Frederic Dumesle, crapaxon@hotmail.com

Adviser: Charles Hornat

Accepted:

1. Introduction and scope

Windows 2003 has been released almost 5 years ago which is a long time in IT terms. Even though hundreds of article, white papers and case study are broadly available a significant number of scenarios involving Windows 2003 in Perimeter Design are scarce at best. This situation stems from a suspicious feeling towards the operating system in such scenarios. This paper will look at operating system optional components from a security stand point and how they can bring added value by leveraging their functionality. The paper will focus on a specific perimeter scenario and will provide an in depth case study with an end to end approach from design to deployment. The report will often refer to principles learned from SANS GIAC Security essentials track.

2. Windows 2003 feature packs

Summary description

There are 6 add-on components available for window 2003 and the R2 release for both 32bits and 64 bits version. These components are free and don't require a license outside of the core operating system one. The only exception is RMS which requires an RMS client license

Extensive detailed documentation can be found at http://www.microsoft.com/windowsserver2003/techinfo/overview/default.mspx

ADAM

This is an LDAP compliant directory service. This is basically Active Directory without the NOS (network operating system) component. The product is comparable to OpenLDAP. It supports LDAP over SSL, granular authorization model and features the

same management capability as Active Directory (ADSI, MMC, etc). It also supports web services for Directory through DSML[1]

MIIS

Stands for Microsoft Identity Integration Server. MIIS implements what is commonly called a Meta Directory. Its most powerful functionality aggregates data from multiple sources into a virtual space. The product supports business rules that will define data flow from sources to a virtual space and vice versa. The basic version called "Feature Pack" supports Active Directory, Exchange and ADAM while the enterprise edition brings extended sources like SAP, oracle PeopleSoft, Lotus, etc…

RMS

Right management services provide client with pki based authorization. The client is typically a RMS compatible client like outlook which brings the possibility for the end user to control whether outlook or office documents can be printed or forwarded.

ADFS

Active directory Federated Services provide single sign on for web applications relying on NT tokens. It provides an organization with flexibility for authenticating and authorizing external partners without exposing your internal Active Directory.

3. Generic Infrastructure Architecture and Approach

Generic Security Requirements

The matrix below describes generic security requirements that Morpheus would like to address

Strong Authentication	PKI or 2 Factor

[1] http://msdn2.microsoft.com/en-us/library/aa813608.aspx

Audit Failed Log in	Log retention 60 days
Segregation of duties	Different teams with different roles
Strong Password Policy	At least 8 characters and password complexity
Physical Security	Restricted Zone for DMZ Infrastructure

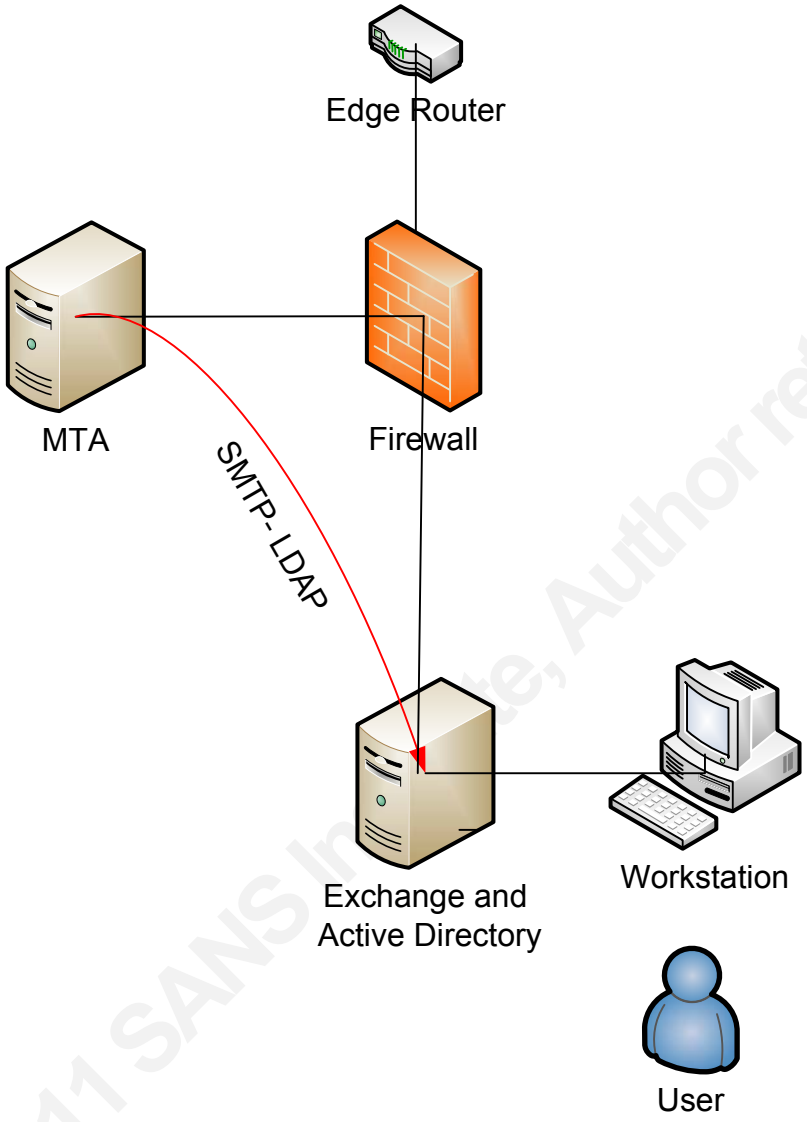
4. Case Study: "Morpheus Inc."

Design

Morpheus Incorporated is the typical medium to large enterprise with One Internal network, one DMZ and internet connectivity. They host an exchange organization and use a single Postfix or Sendmail UNIX based MTA in the DMZ. Additionally the MTA performs email check through LDAP as part of the SMTP transaction. In fact initially Morpheus would accept any mails as long as recipients were *morpheus.com addresses.* However as spammers were sending bulk mails to invalid morpheus.com recipients the company got blacklisted by ISPs as it was sending millions of NDRs. The LDAP check mitigates the issue even though Active Directory is exposed to the internet.

The Visio diagram 1 below describes Morpheus Infrastructure

Morpheus Infrastructure – Diagram 1



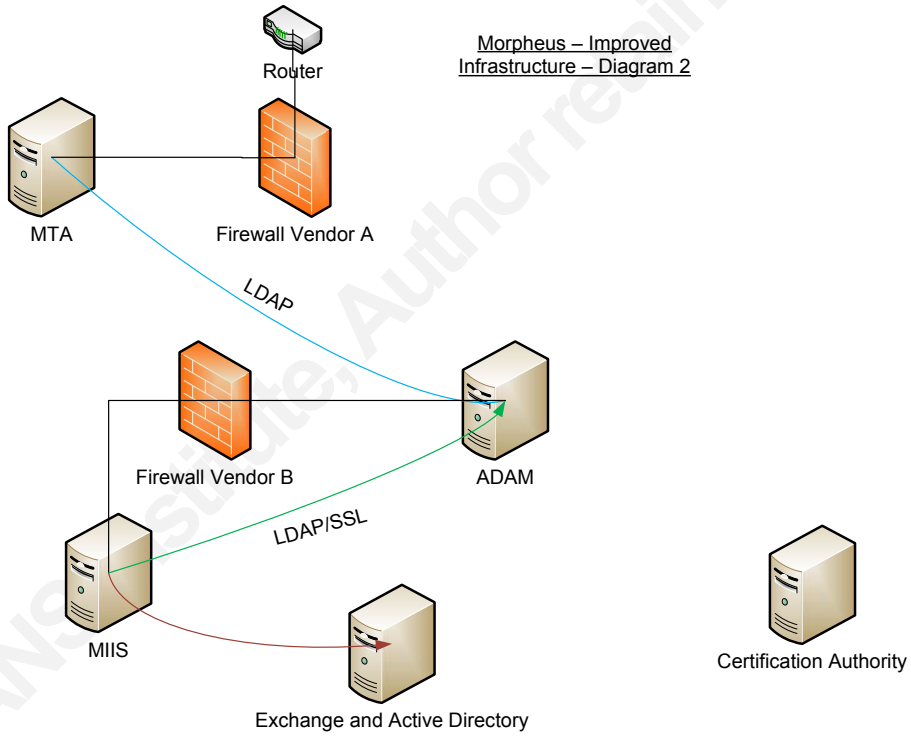
Security Assessment

Morpheus Incorporated asked an external company to produce a security assessment report. The following issues have been identified:

- Single Firewall Vendor

- Single Point of failure for MTA, firewall, router, switches
- Active Directory exposure through LDAP from MTA

New Proposed Design



The above Visio diagram depicts a significantly improved design featuring new components. Although it looks more complex this infrastructure addresses the issues previously described and meets chapter 3 requirements.

Directory Component: ADAM

The ADAM component purpose is processing LDAP requests for email address initiated by the MTA. Instead of exposing Active Directory to the MTA and internet we only expose an ADAM

instance whose schema is customized with a minimum set of attributes and elements. Hence a compromised ADAM would not leak critical data like passwords and other important Active Directory Information.

A number of additional security measures are deployed:

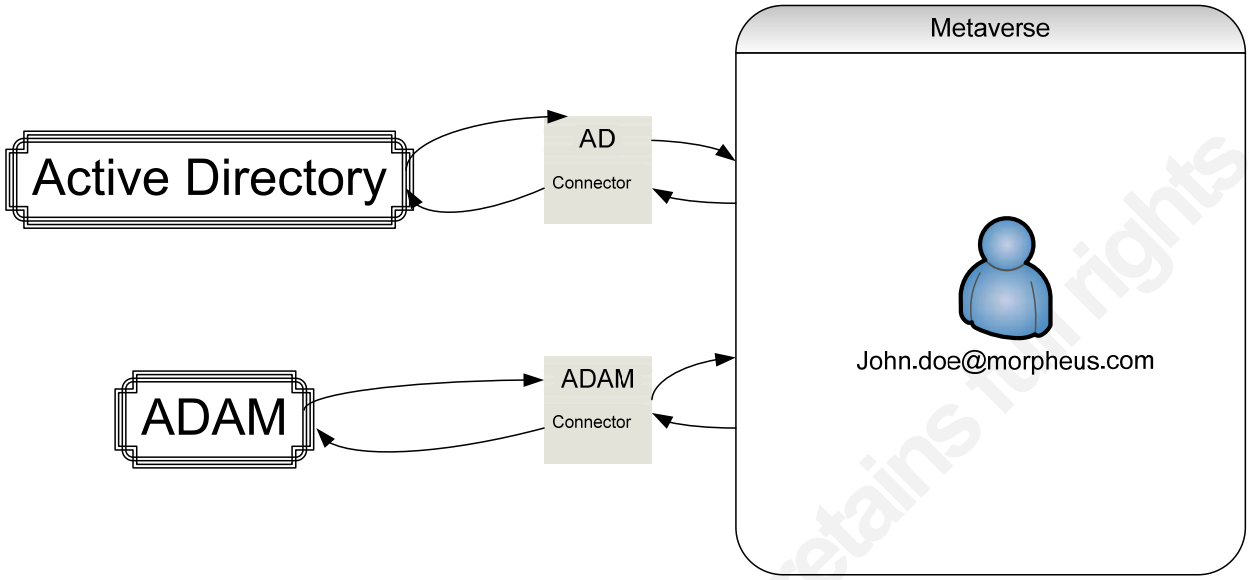
- ADAM is screened by a firewall(see later chapter) in a dedicated DMZ
- LDAP Security mechanism can be added(LDAP over SSL, stronger authentication)

Provisioning Component: MIIS

The Microsoft Identity Feature Pack component purpose is to aggregate data from active directory along with provisioning and de-provisioning of the ADAM Instance.

MIIS has the following core components

- SQL Database
  - All MIIS data live inside a SQL Database (Locally or remote)
- Metaverse
  - A place where aggregated data get ultimately persisted. It has its own schema
- Space connector
  - A buffer space containing data from a specific source



Multiple filters can be defined at the connector space level and also in the metaverse through the usage of specific provisioning rules. The detailed implementation will be provided later in the document. What matters is that the software allows the infrastructure architect to define security mechanisms that address Integrity (ensuring active directory attributes meet specific criteria's), confidentiality (MIIS will use LDAP over SSL when connecting to AD and ADAM), authentication (MIIS uses a PKI certificate to authenticate with the ADAM server, Kerberos when authenticating with Active directory)

Second Firewall Layer

We apply the defense in depth principle by introducing an additional Firewall Layer. This layer will be from a different vendor. Depending on the vendor we could also filter at the application layer and implement specific LDAP filtering. Should the first firewall be compromised Morpheus internal network is still protected by the second layer.

Detailed Design and Implementation

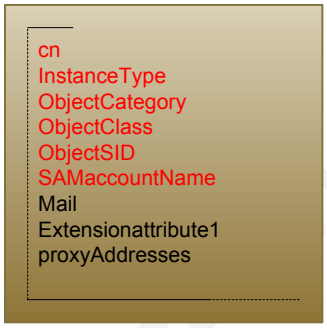
Morpheus.int Domain

The Morpheus.int Active Directory is a regular out of box AD forest. It features an exchange extended schema allowing users to have mailboxes and email addresses.

Mail Design

Morpheus Inc relies on the MTA in the DMZ and exchange for internal mails. Extended exchange attributes are used to implement external contractor and student policies. In fact the "extensionattribute1" is set to "active" to enable internet access. Matching user attributes will be processed and provisioned into the ADAM instance. Any other value will be skipped. Furthermore when a staff gets internet mail access denied the email address is automatically de-provisioned from the ADAM instance. These core processes are under MIIS control.

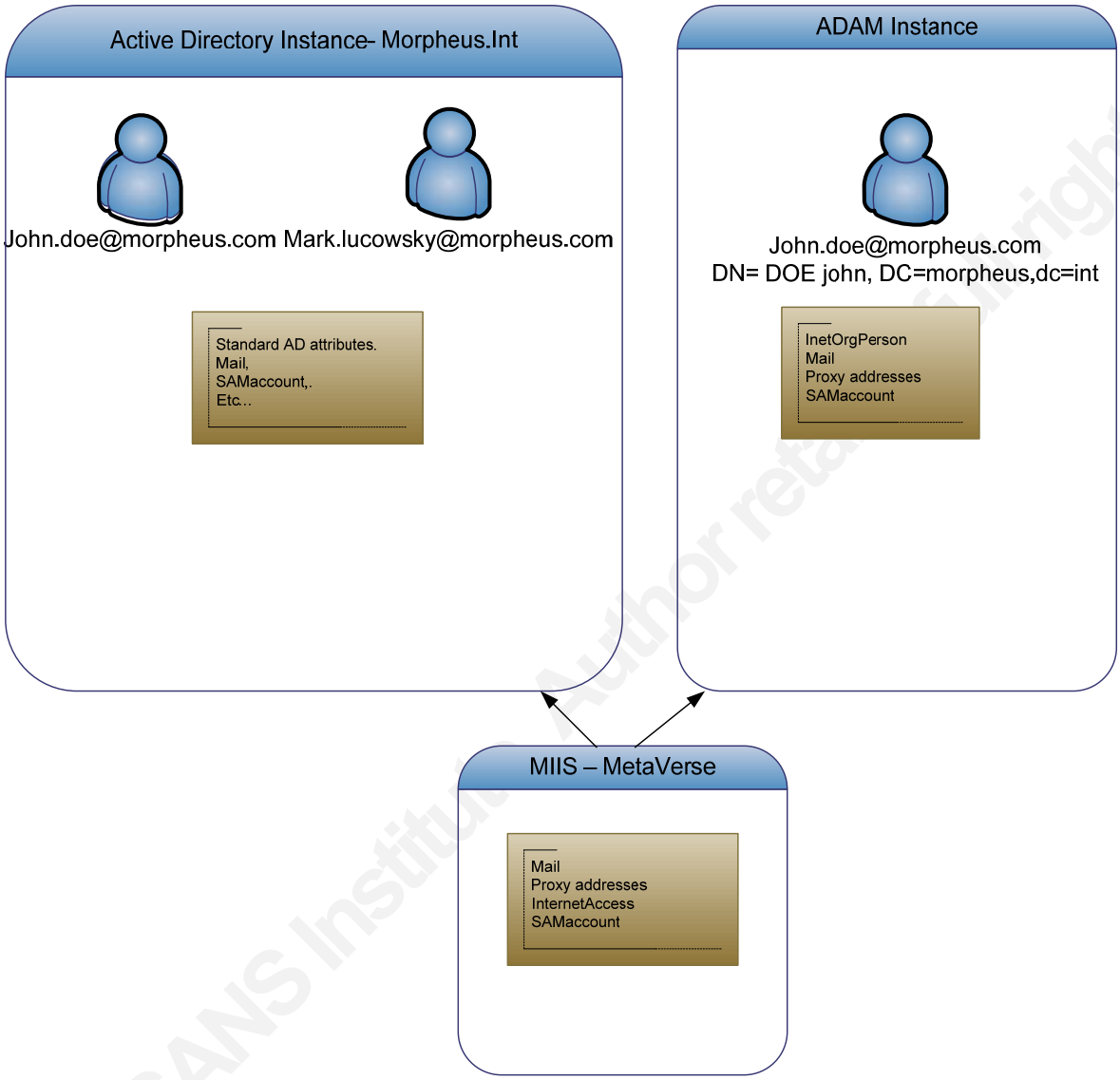
The relevant user attributes are the following for Morpheus.int



The red attributes are mandatory ones. The mail attribute contains the user email address. The ExtensionAttribute1 is a string containing "active" or "inactive". This attribute role defines whether a user has the ability or not to send a mail to the internet.

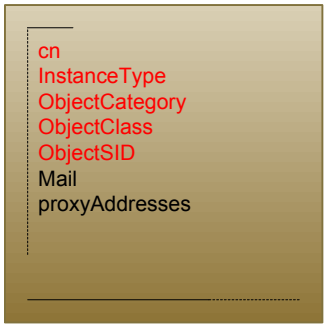
proxyAddresses is used to store multivalue email addresses.

Service View



The diagram above describes the logical view between the three components from a schema perspective.

ADAM Design



Contrary to Active Directory ADAM schema is empty out of the box. The chosen schema will be an X.500 standard inetOrgPerson schema which will be imported through LDIF. Notice the mandatory attributes in RED along with the relevant attributes that are necessary for LDAP search performed by the MTA Client.

Application Partition

The application partition will be very simple and straight forward. We choose for **DC=Morpheus, DC=com.**

Since we need several users defined in the partition a hierarchical structure is the next logical step. We create an organizational unit called ***Service Account.***

Let's now create several users:

Administrator	This account will have full control on ADAM instance. Also used by MIIS to update instance
MTA_Client_xx	The account used by the MTA. Notice we plan for future MTA. One account per MTA server

Monitoring	Support account for external party.read only

The DNs will list as follow:

CN=MTA_Client01, OU=serviceaccount, DC=morpheus, DC=com

CN=Administrator, OU=serviceaccount, DC=morpheus, DC=com

CN=Monitoring, OU=serviceaccount, DC=morpheus, DC=com

Additionally the ADAM instance itself runs under the NETWORK Service account which has very limited privileges

Securing Adam Application

We need to secure ADAM communications with the external world as much as we can. Applying the defense in depth principle we will only allow LDAP over SSL at the firewall level. We will block standard LDAP port too and use packet filtering on the windows server (only allowing LDAP over SSL, ICMP and RDP for remote control)

PKI Deployment

We need to modify a file ACL to allow the ADAM instance service account to read the computer machine Keys. This is part of the LDAP over SSL ADAM implementation. If not setup properly ADAM will not open an SSL socket and will only listen on LDAP 389.

C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\MachineKeys.

Offline Certificate

We must now generate a certificate from a CA then import it in the ADAM server instance. The scenario assumes Morpheus Inc runs a Certification Authority on the internal LAN. The process would be similar Morpheus had chosen for an external trusted CA like VeriSign.

This is a 4 steps process:

- Generate an INF file
- Produce a REQ file
- CA generates a Certificate file using the REQ file as input
- Import the Certificate on the Target Machine

Generating an INF File

The INF file follows a standard for Certification Authority:

Open Notepad or your favorite text editor and paste the following text:

```
[Version]
Signature="$Windows NT$
Here the inf  file


[NewRequest]
Subject = "CN=<DC fqdn>" ; replace with the FQDN of the DC
KeySpec = 1
KeyLength = 1024        ; Can be 1024, 2048, 4096, 8192, or 16384.
                        ;   Larger key sizes are more secure, but have
                        ;   a greater impact on performance.
Exportable = TRUE
MachineKeySet = TRUE
SMIME = False
PrivateKeyArchive = FALSE
UserProtected = FALSE
UseExistingKeySet = FALSE
ProviderName = "Microsoft RSA SChannel Cryptographic Provider"
ProviderType = 12
```

RequestType = PKCS10

KeyUsage = 0xa0

[EnhancedKeyUsageExtension]

OID=1.3.6.1.5.5.7.3.1 ; this is for Server Authentication

Replace the "Subject" with appropriate LDAP path Server name.

Name the file serv.inf and save it somewhere on the hard disk (for instance %USERPROFILE%\)

Producing a REQ File

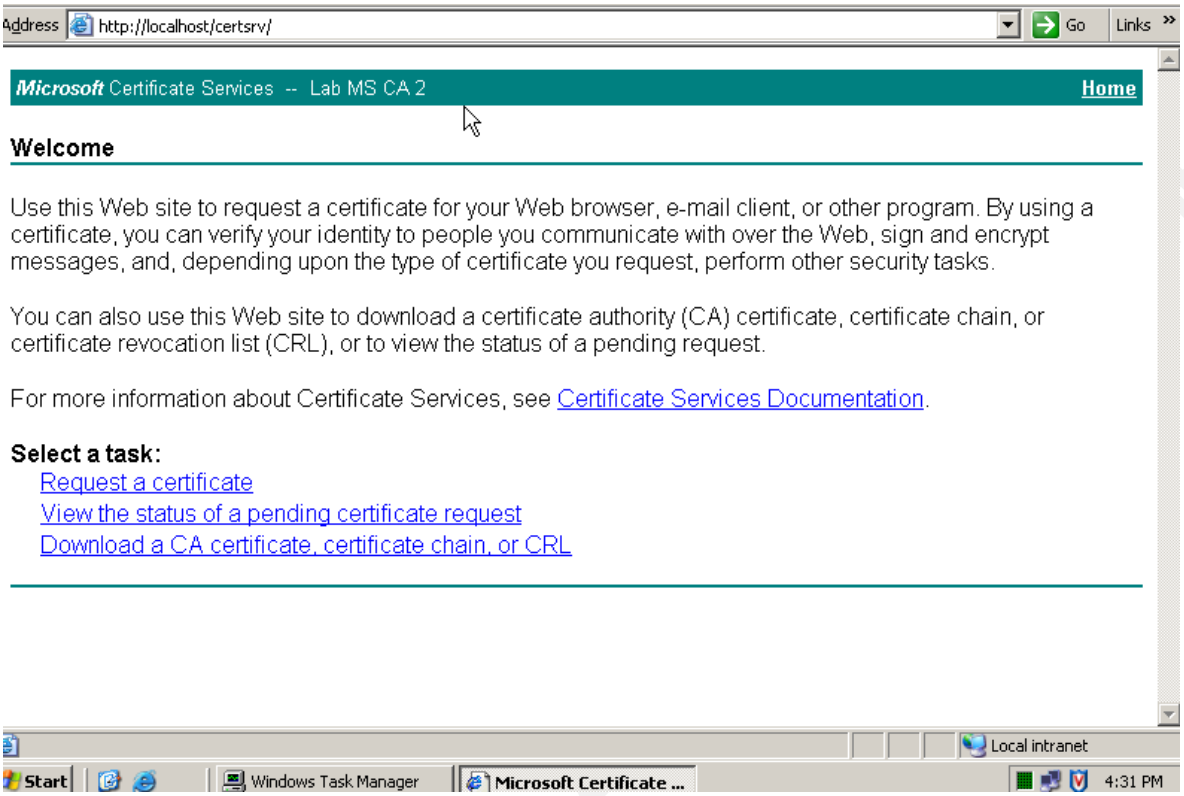
Open a command prompt and make sure you are in the previsously used directory (%USERPROFILE%\).

Run certreq –new. You will be prompted. Select your serv.inf file.

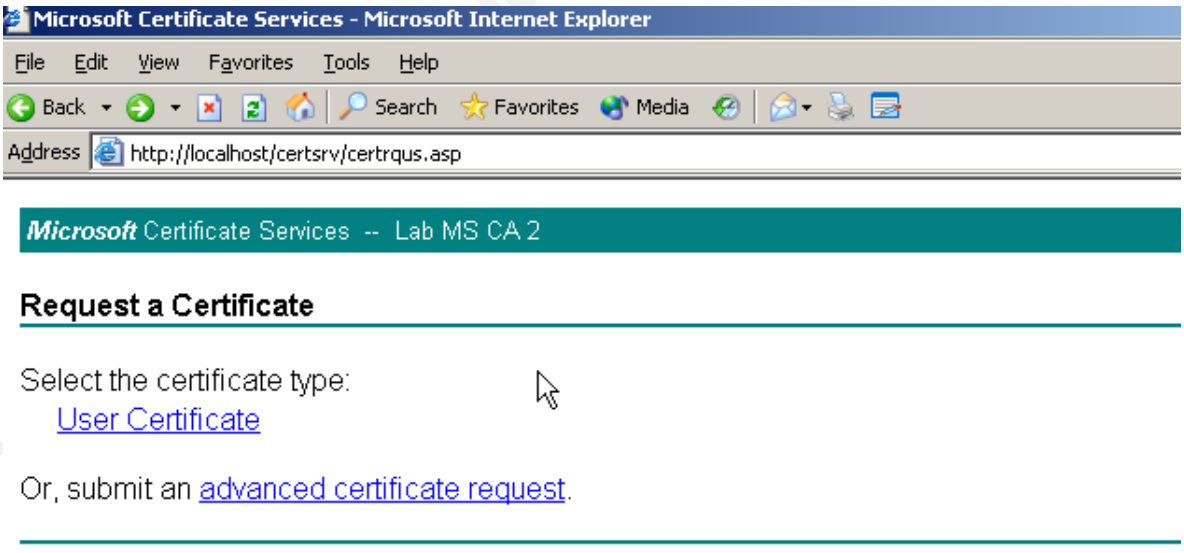
A new file has now been generated and is called serv.req.

CA Cert generation

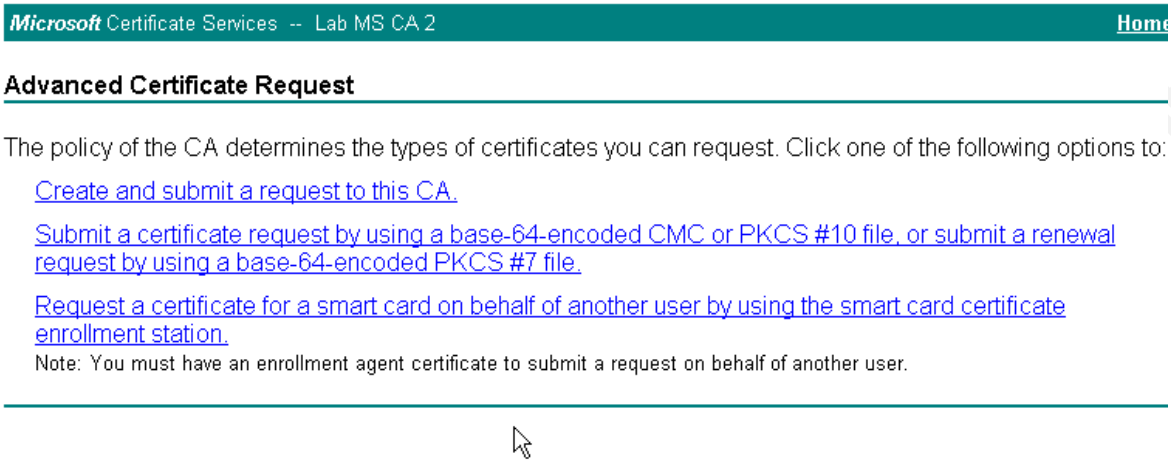
Log in on the Microsoft CA web generation page using the URL: http://localhost/certsrv. You must be logged on the console through terminal services or interactively.



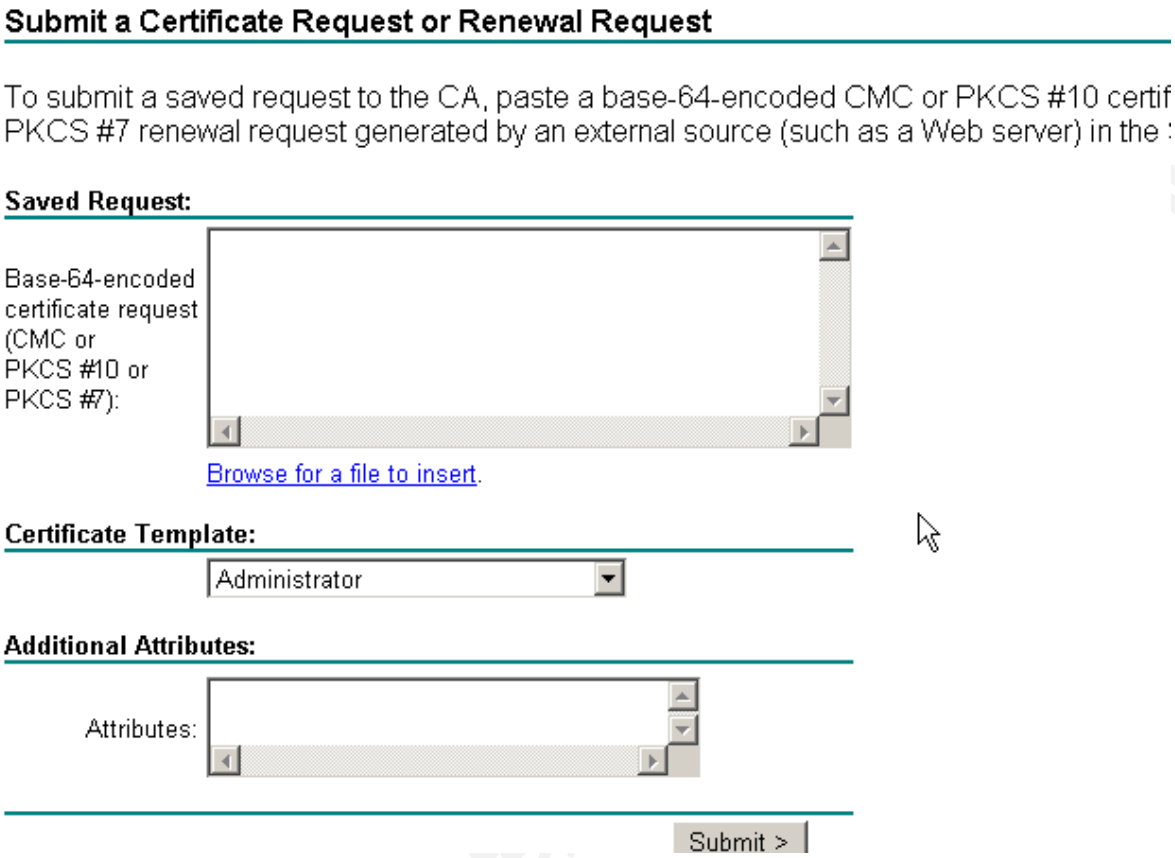
Select request a certificate



Select Advanced certificate request

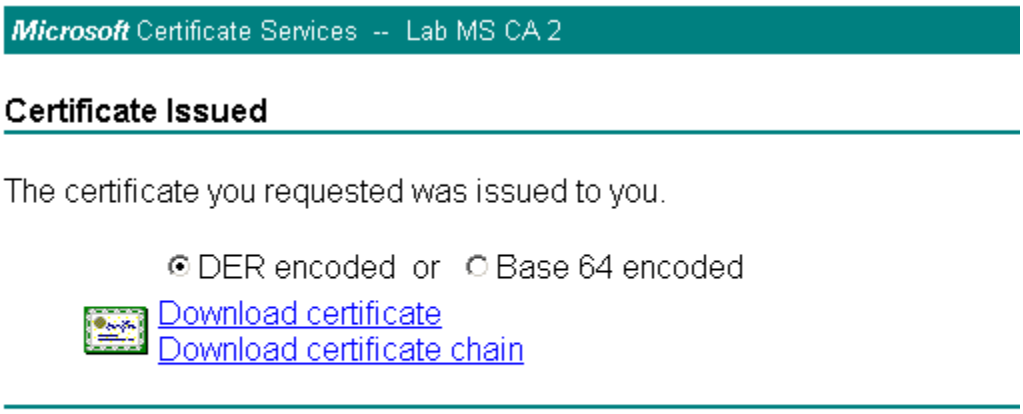


Select "Submit request by using a base64 or PKCS 10 file"



Paste the serv.req file content into "saved request" dialog box

Select **Web Server Template.**



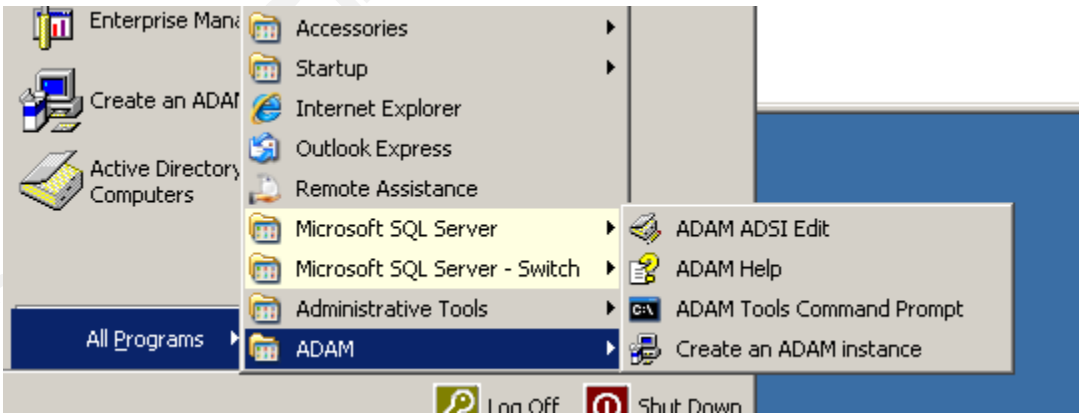
Download and copy the file to the ADAM Server. Use the certificate MMC to import it.

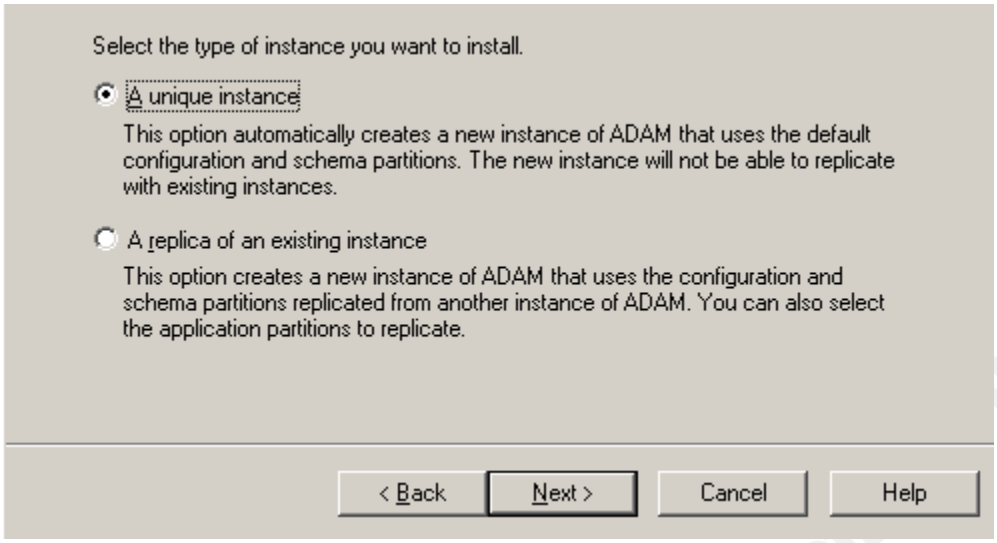
Locking down the server

The Server itself will be secured using a specific template through the Local Policy: we only need a restricted number of services. A specific template should be used for each server (ADAM, MIIS).

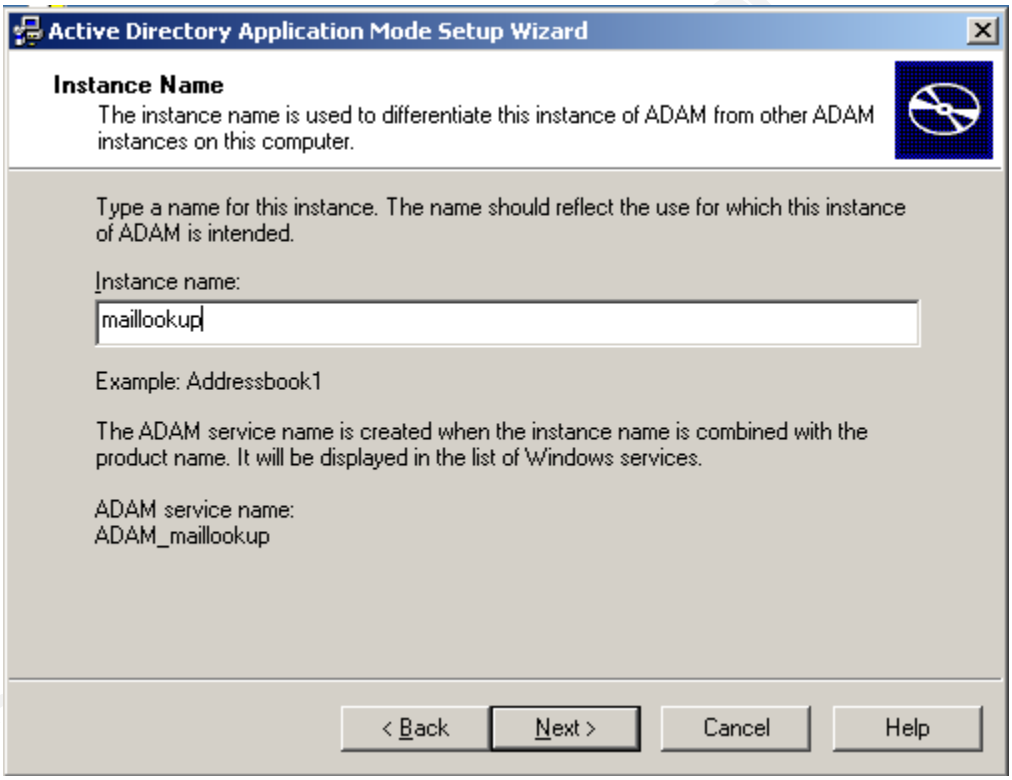
ADAM Installation Guide

Install the default ADAM package then select menu to create a new Instance

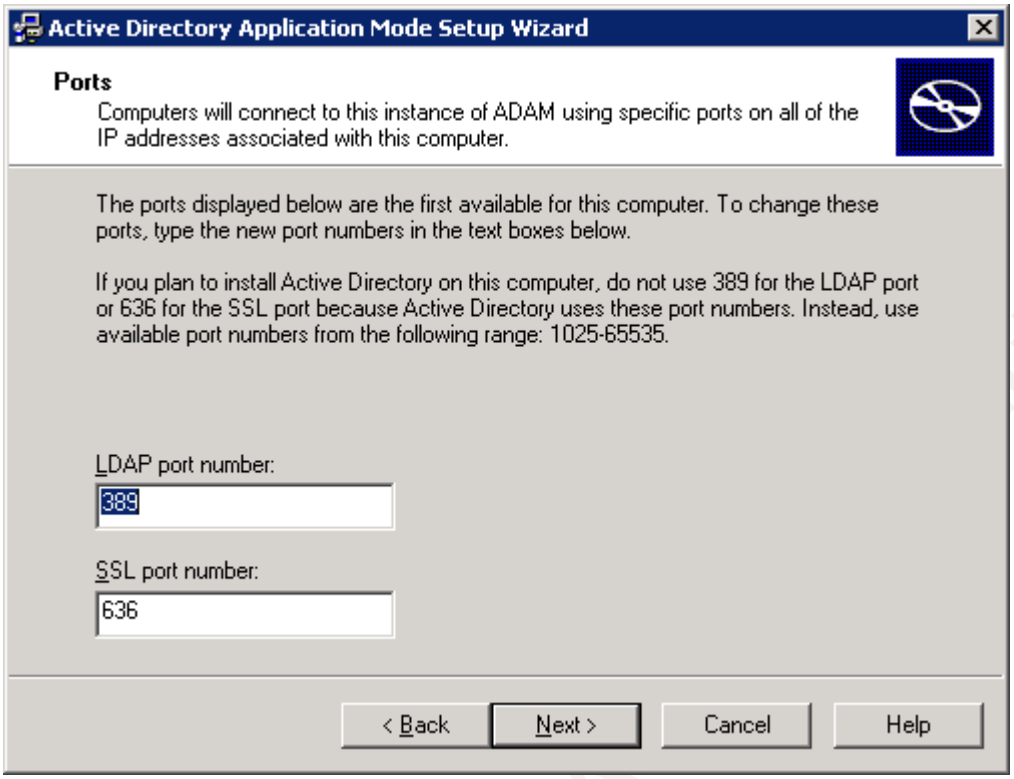


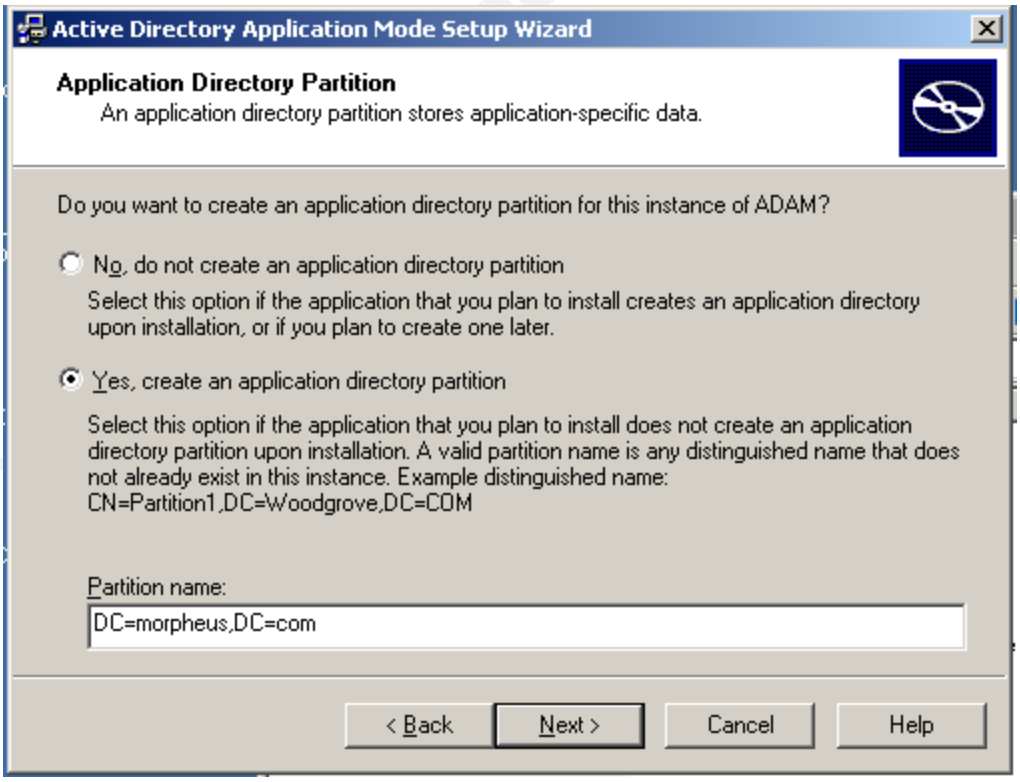


Select "unique instance".

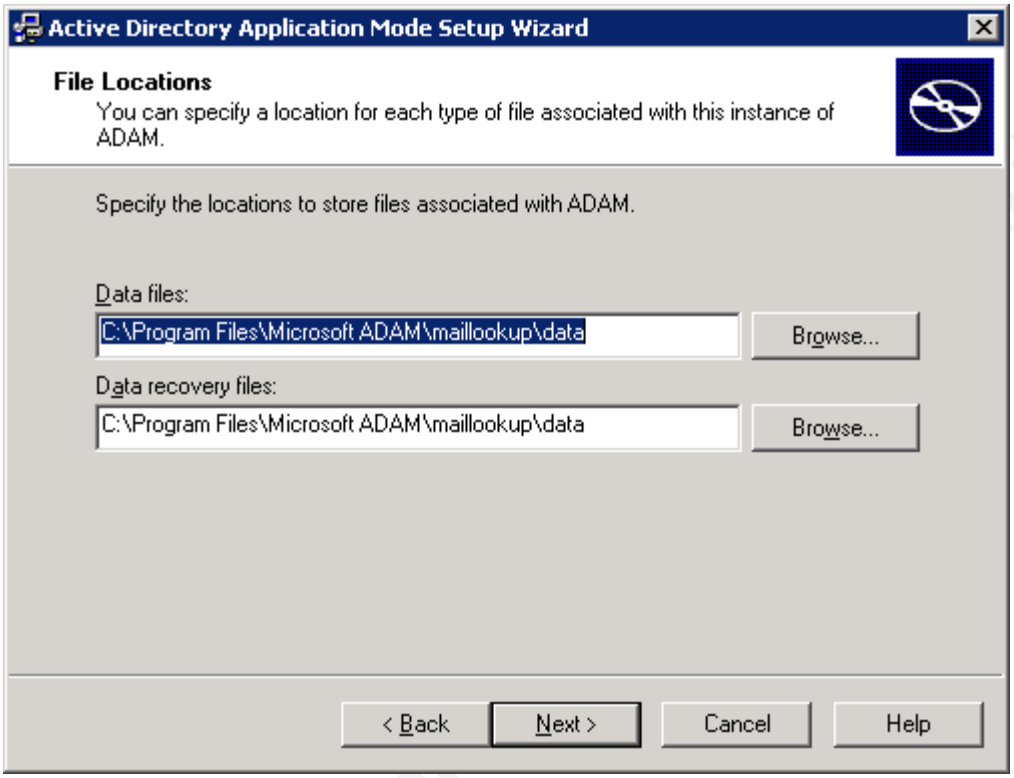


Provide a name

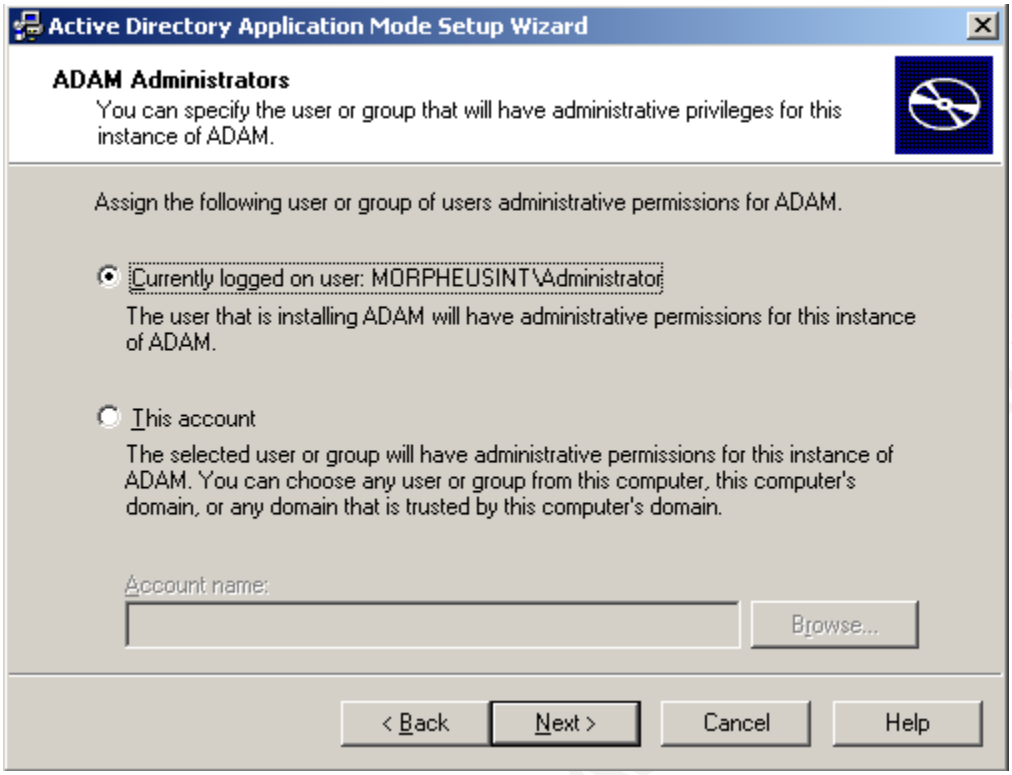




We must create a partition that will be dedicated to serving the MTA

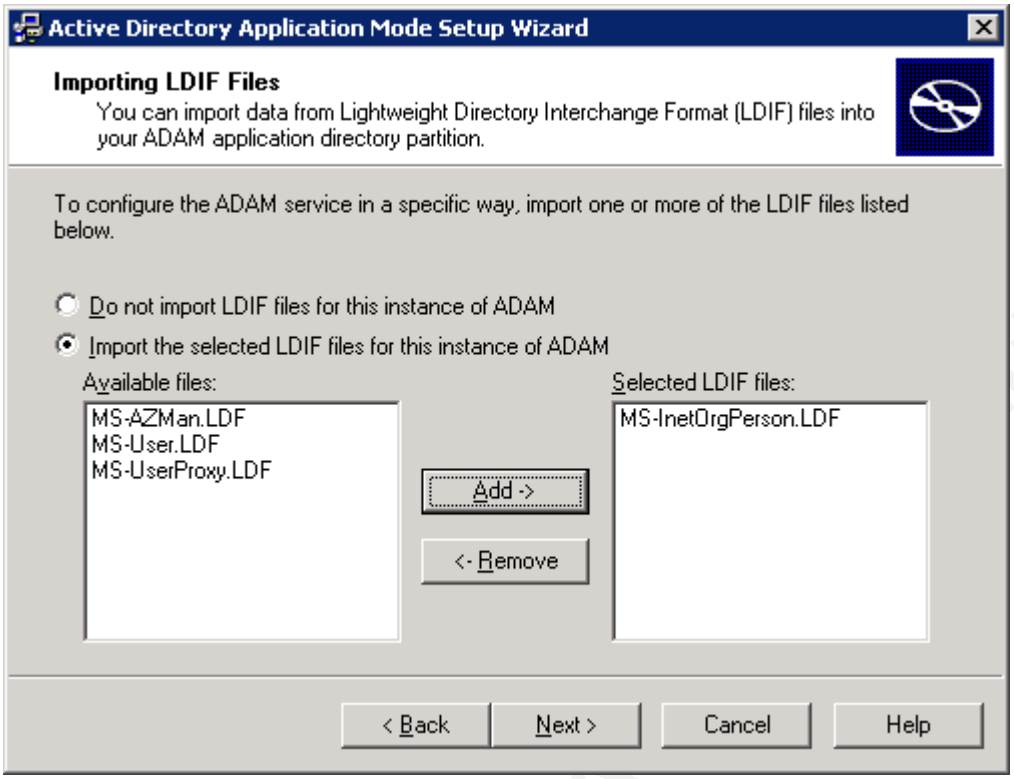


We apply the least privileged account principle: ADAM only needs "network service account"



Let's now import the required schema.

Select MS-InetOrgPerson



MIIS Design

Segregation of Duty and authorization model

To apply the segregation of duty principle the administrative model will be split into several business units. Typically the ADAM instances are managed by Security team while MIIS is maintained by Operation team. However the application owner is the security team.

The matrix below summarizes roles and responsibilities including the following levels:

R: Responsible

This is limited responsibility

A: Accountable

This is end to end complete responsibility

C: Consulted

This is being involved in the decision making

I: Informed

This is just being informed about decision and information process

	Operations Unit	Security Team
High Level Design	I	A,R
Business Rule Change	I	A,R
Patching	R	A
Backup	A	I

Strong PKI based authentication

Since the ADAM server is located in a DMZ we cannot and should not rely on Windows Domain Authentication and Authorization. Pre shared key or NTLMv2 hashes must be ruled out because they are easily defeated. The chosen authentication will be certificate Based (PKI technology).

Morpheus Inc runs an enterprise wide Certification authority whose purpose is to provide company servers with computer and user certificates.

MIIS will connect to the ADAM instance using LDAP over SSL. The server will use its certificate to authenticate to the ADAM server. The SSL session will be successful if both certificates trust the same root CA.

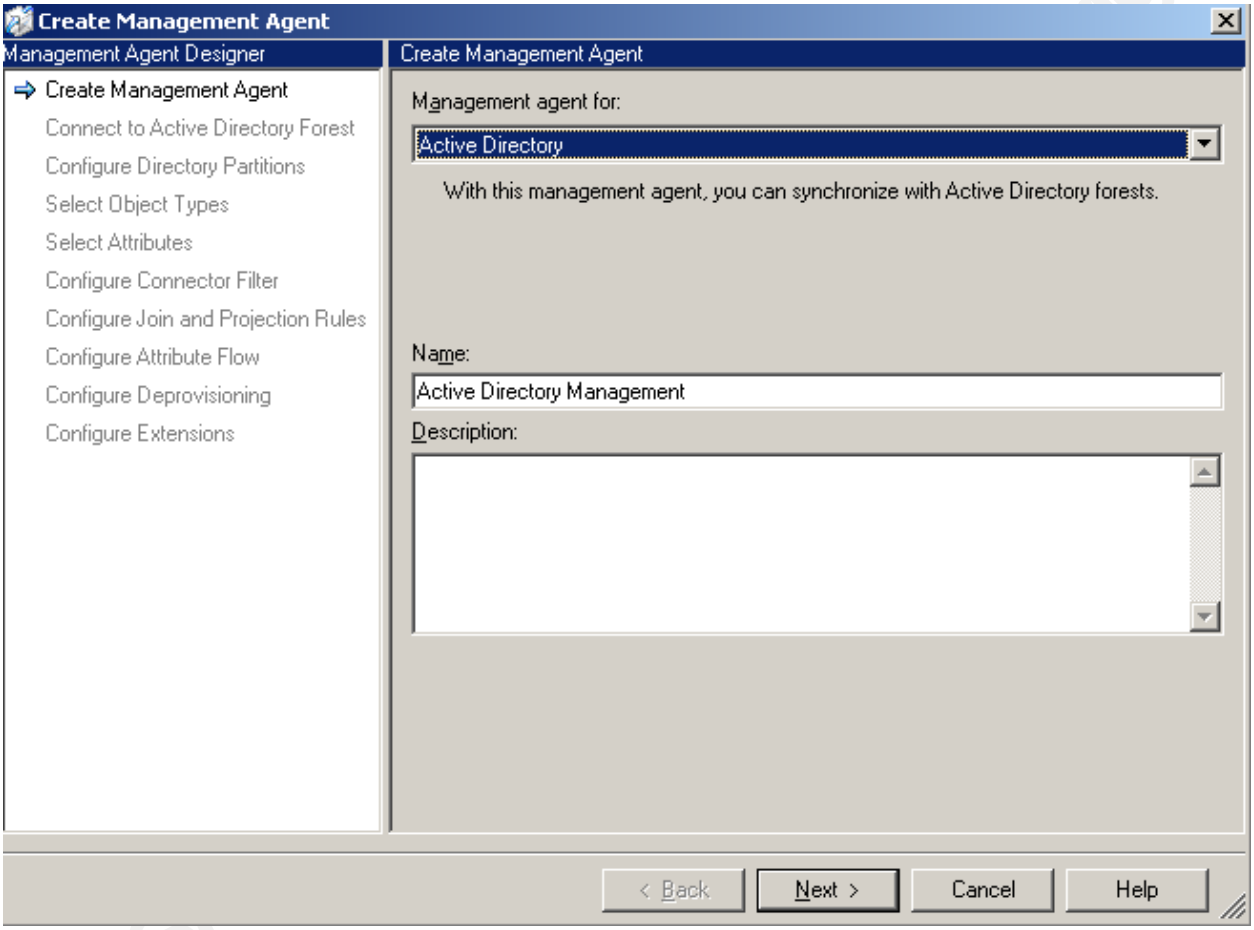
Defining Management Agents

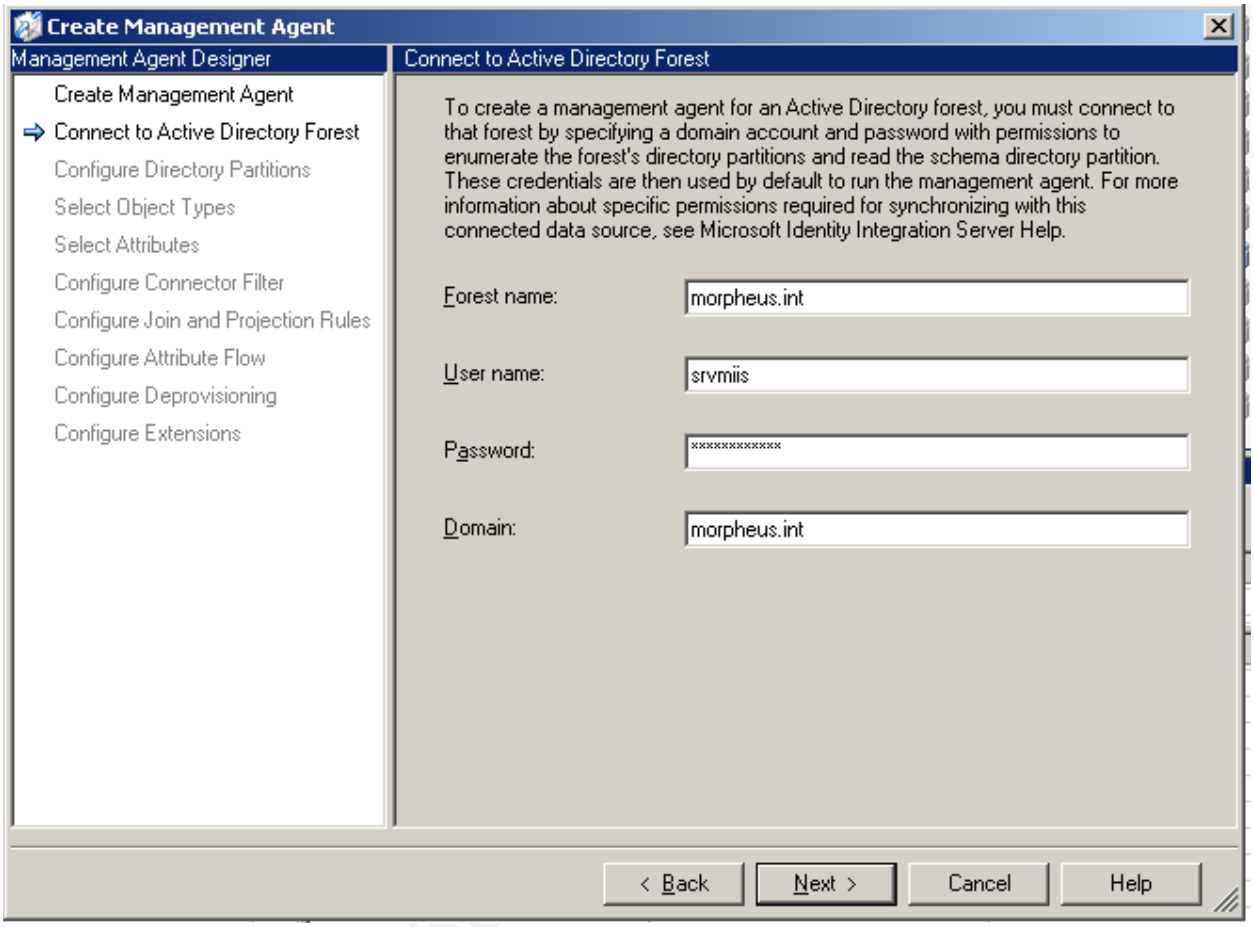
A management agent describes the connection between a data source (AD, ADAM) and some kind of temporary buffer area called

a connector space. The agent also controls data flow between the connector space and the metaverse.

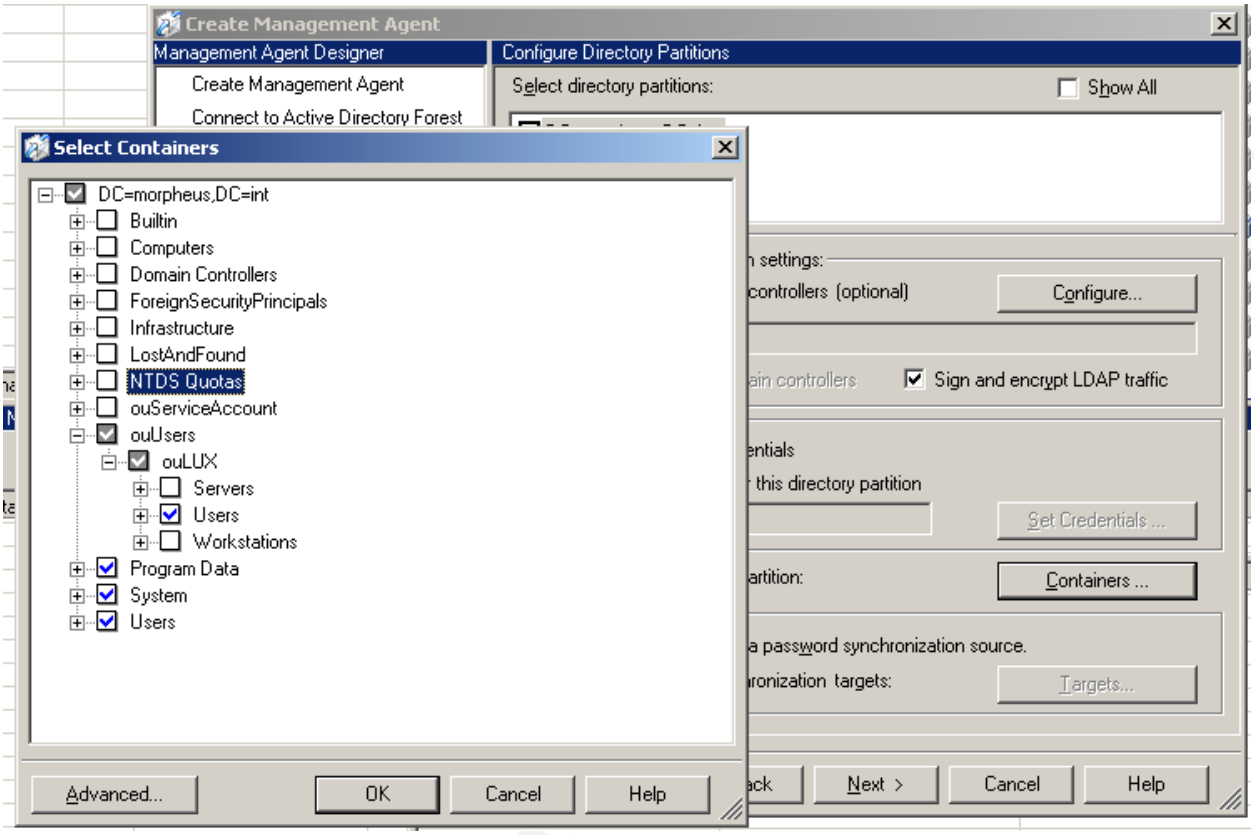
Active Directory Agent

Open the Identity Manager and select management agents- select create





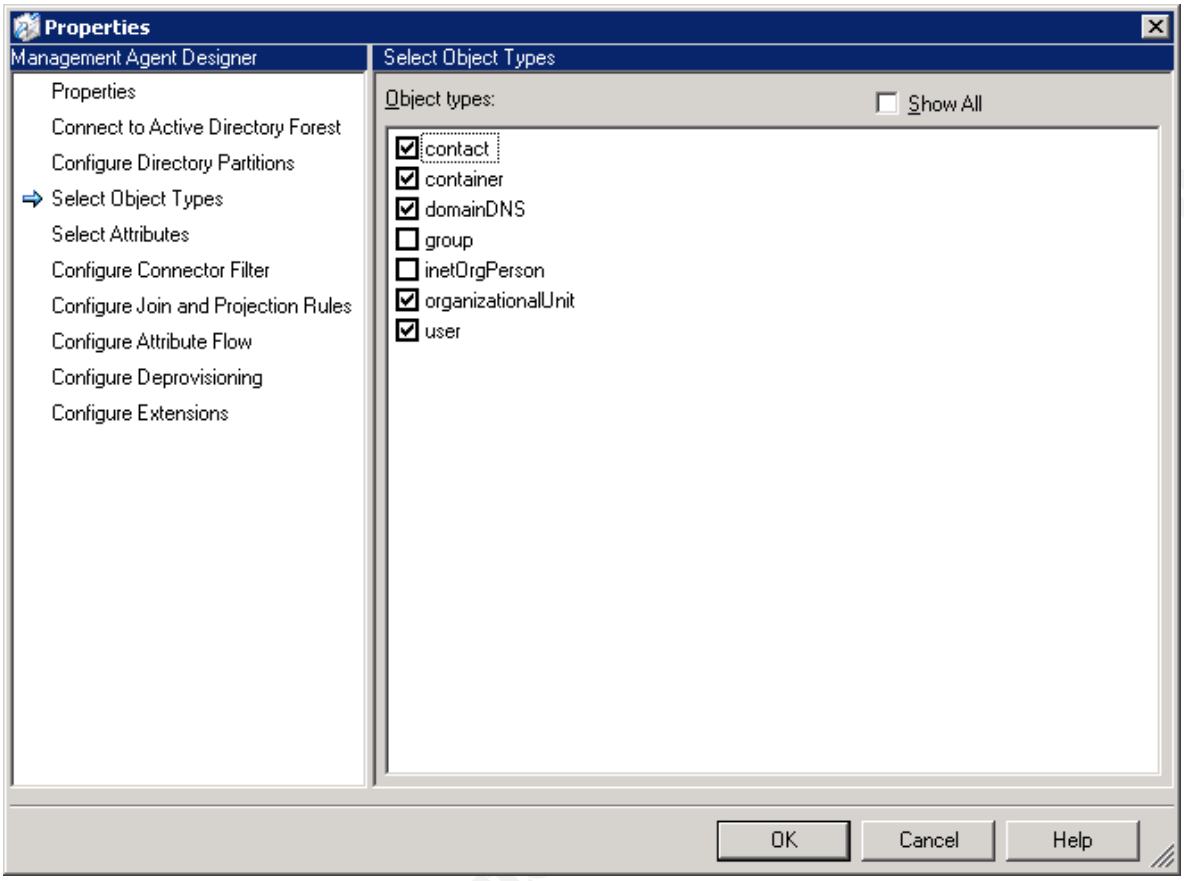
Click next



For security reasons we will instruct the agent to process users containers. This choice will not only increase the overall performance of the parsing engine but will also prevent unattended containers from being processed. In fact should support personnel create thousands of email addresses in new containers they will not be processed until an MIIS administrator (in our case the Security Team) updates the rule.

Object Types

Some objects are mandatory like containers or CN.



Next select the relevant attributes described in service view chapter

Attributes

CN

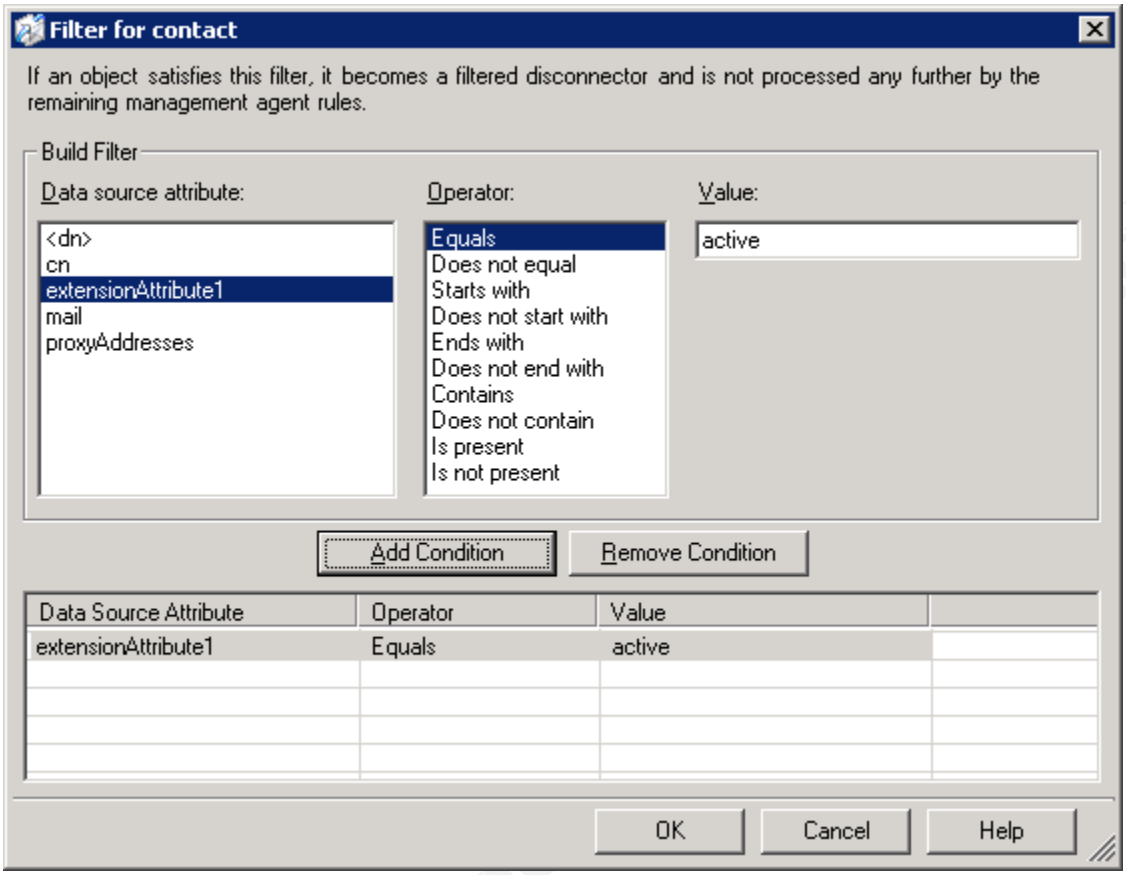
Mail

ProxyAddresses

Extensionattribute1

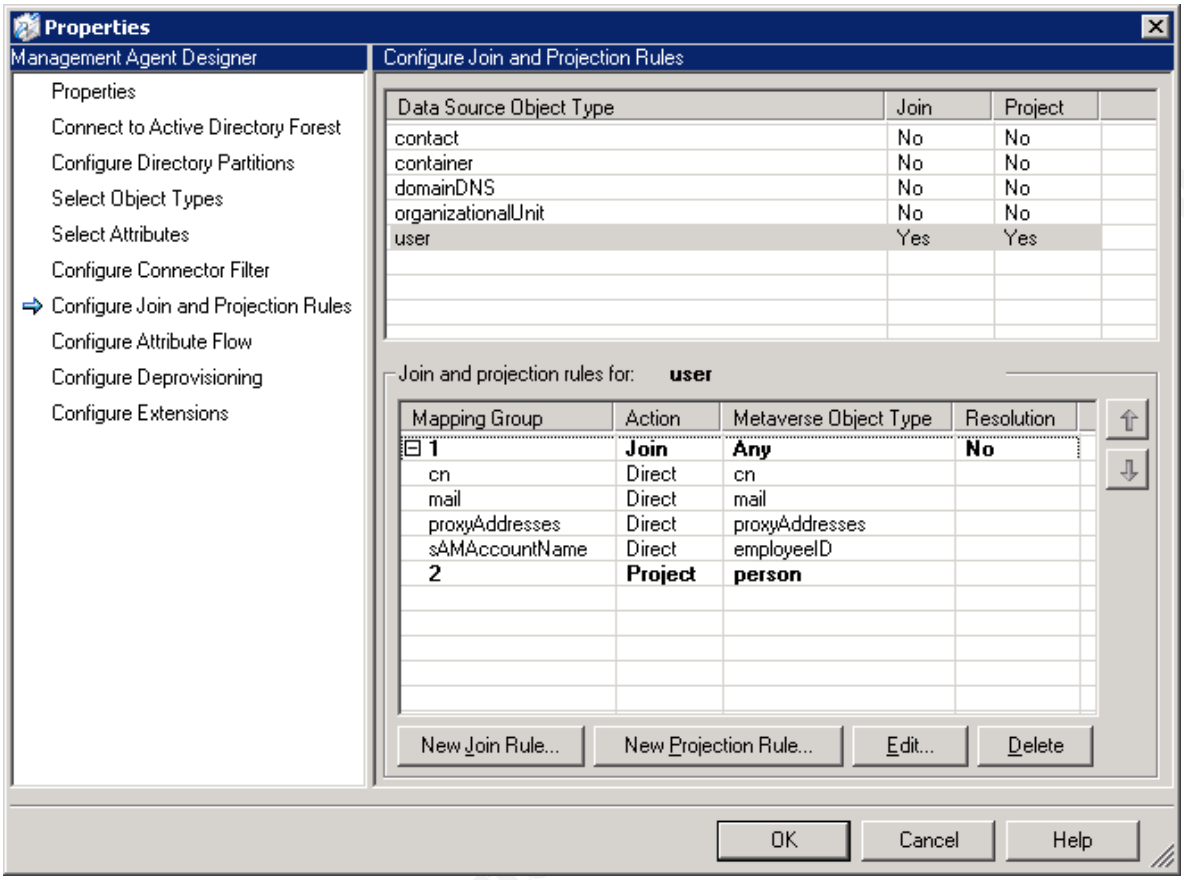
SAMAccount Name

Filters

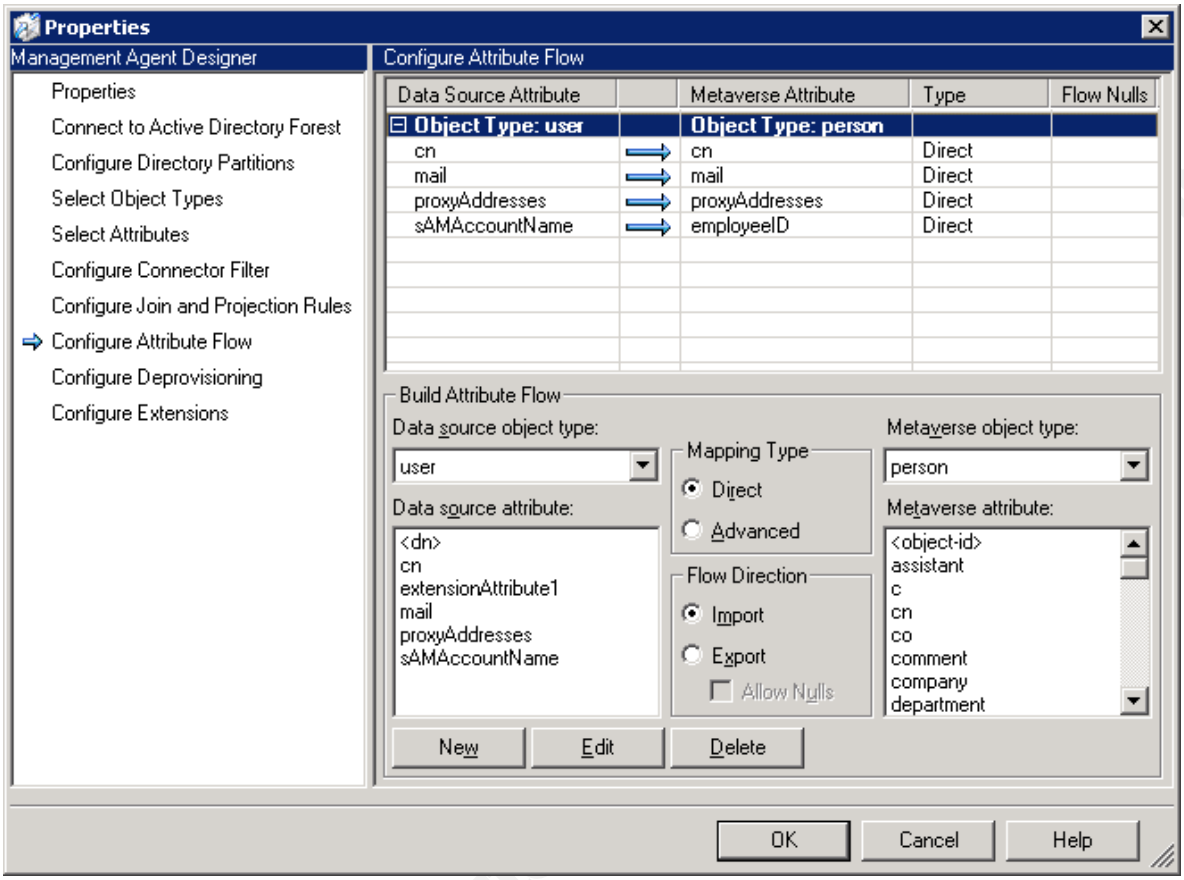


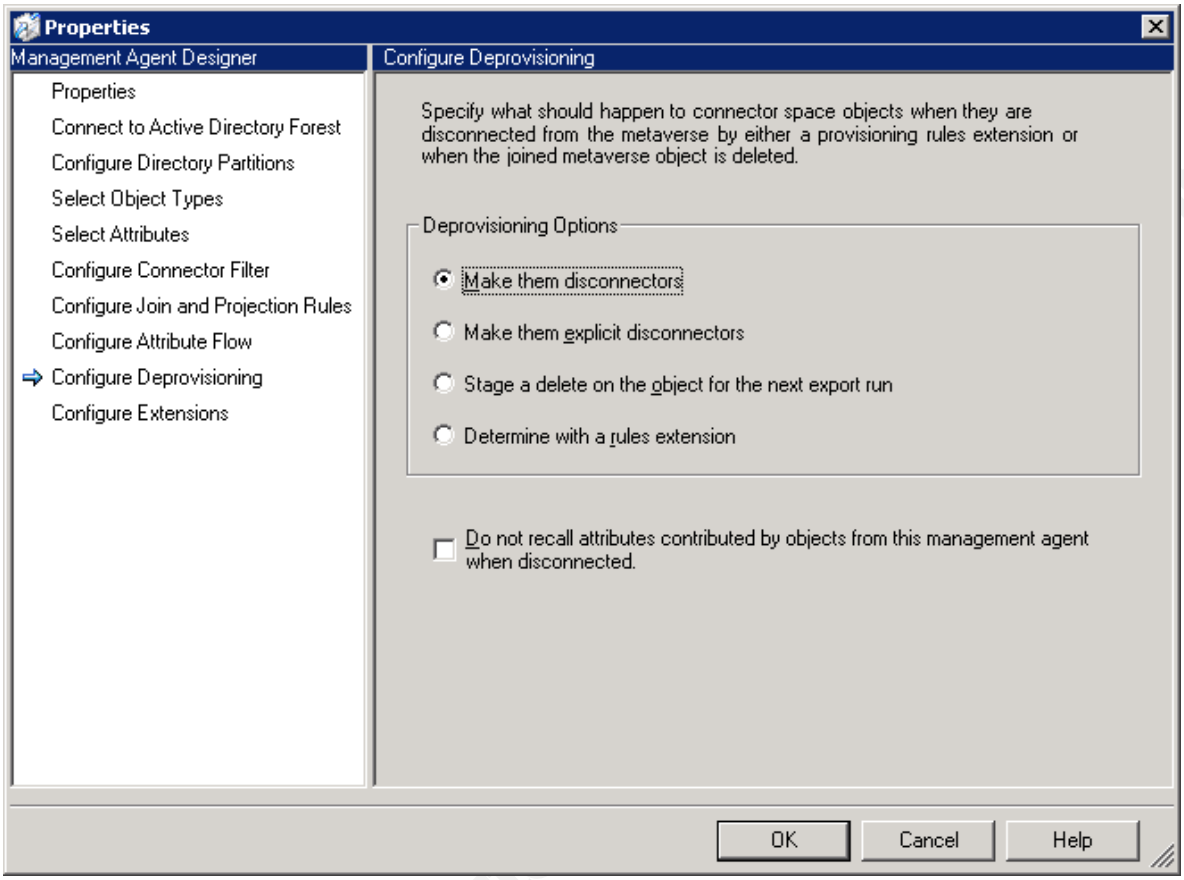
Email internet account must have the exchange extensionattribute1 set to "active" to be processed.

Join and Projection Rules

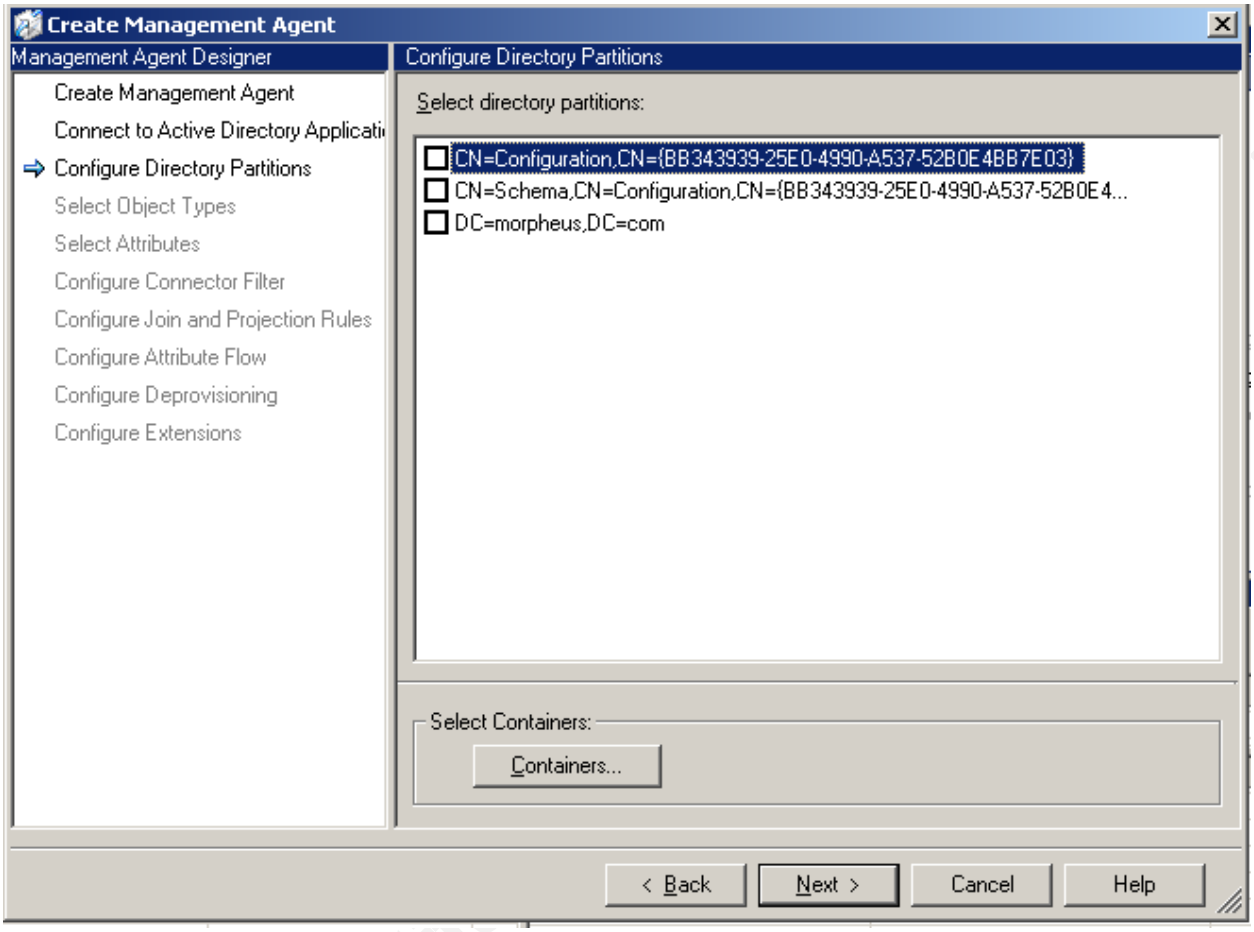


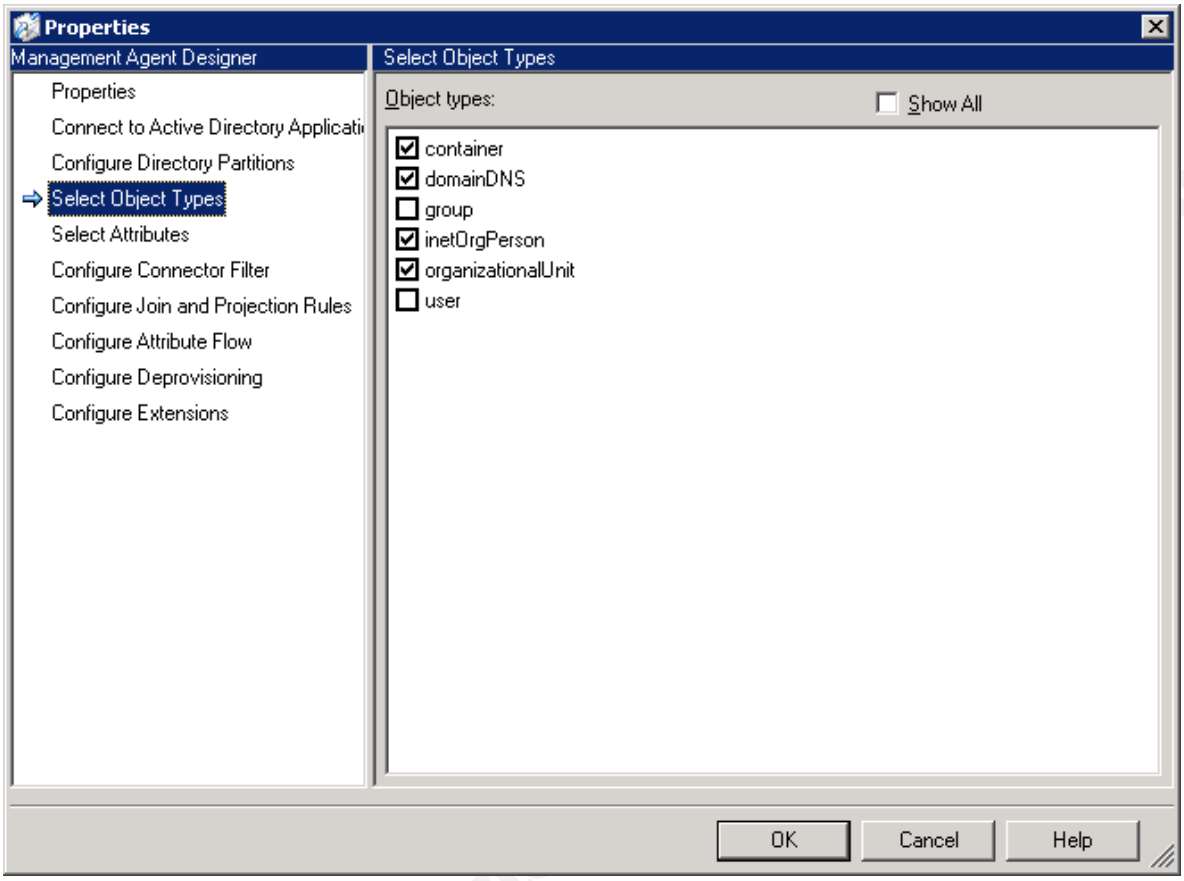
Attribute Flow

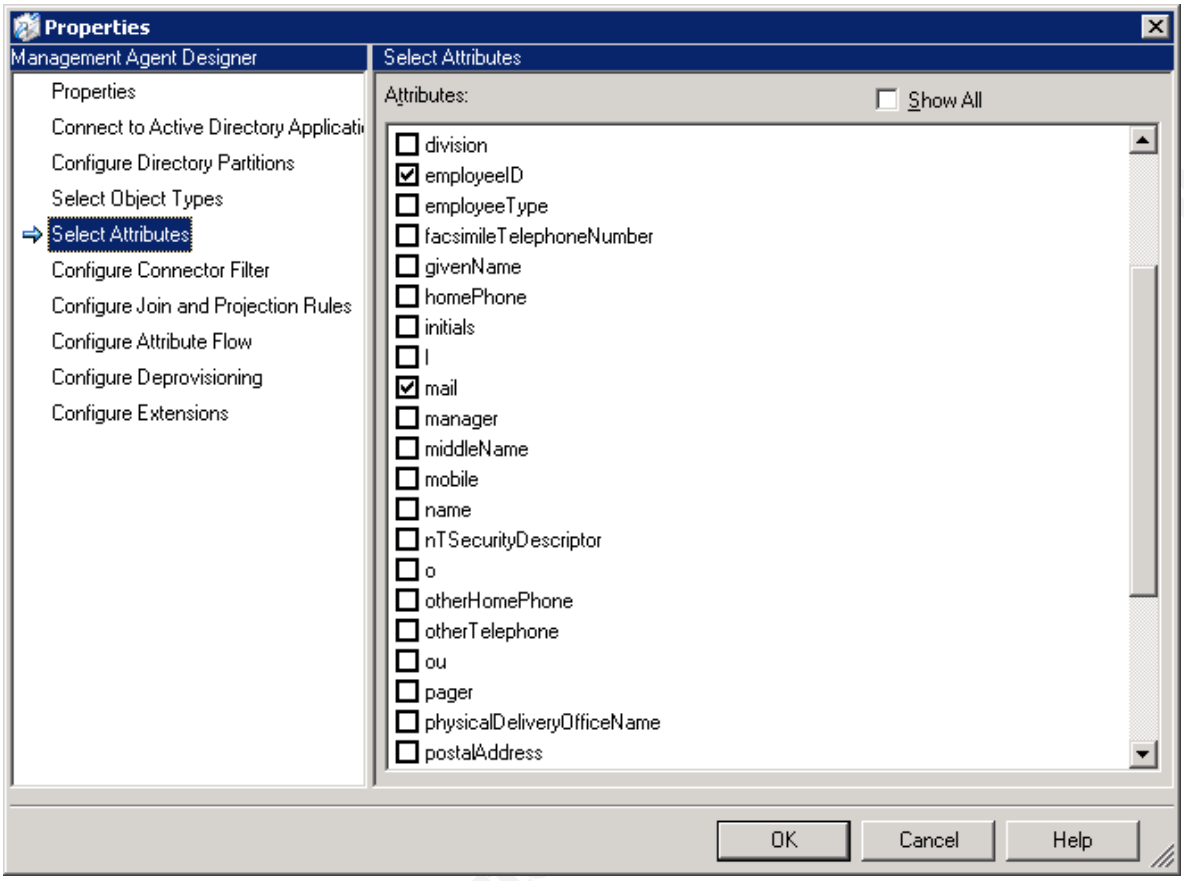




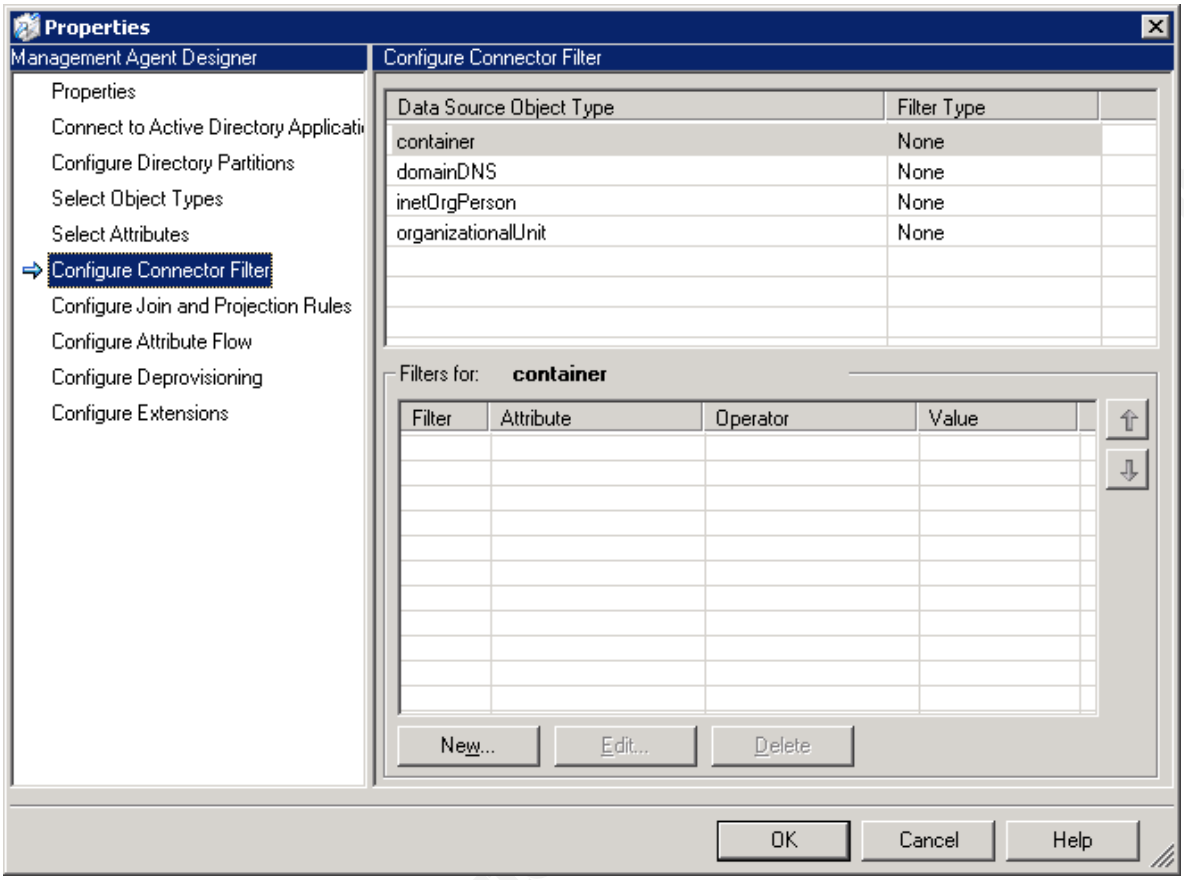
ADAM Agent

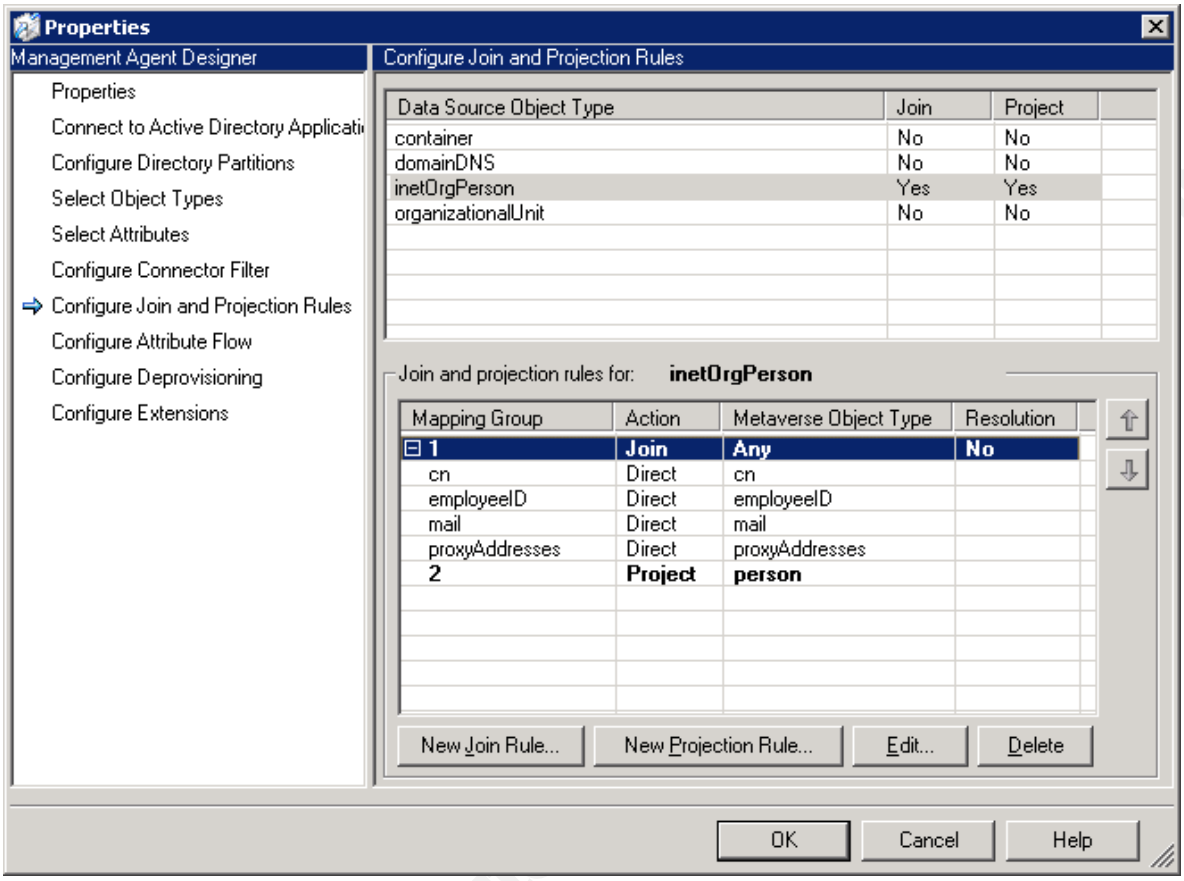


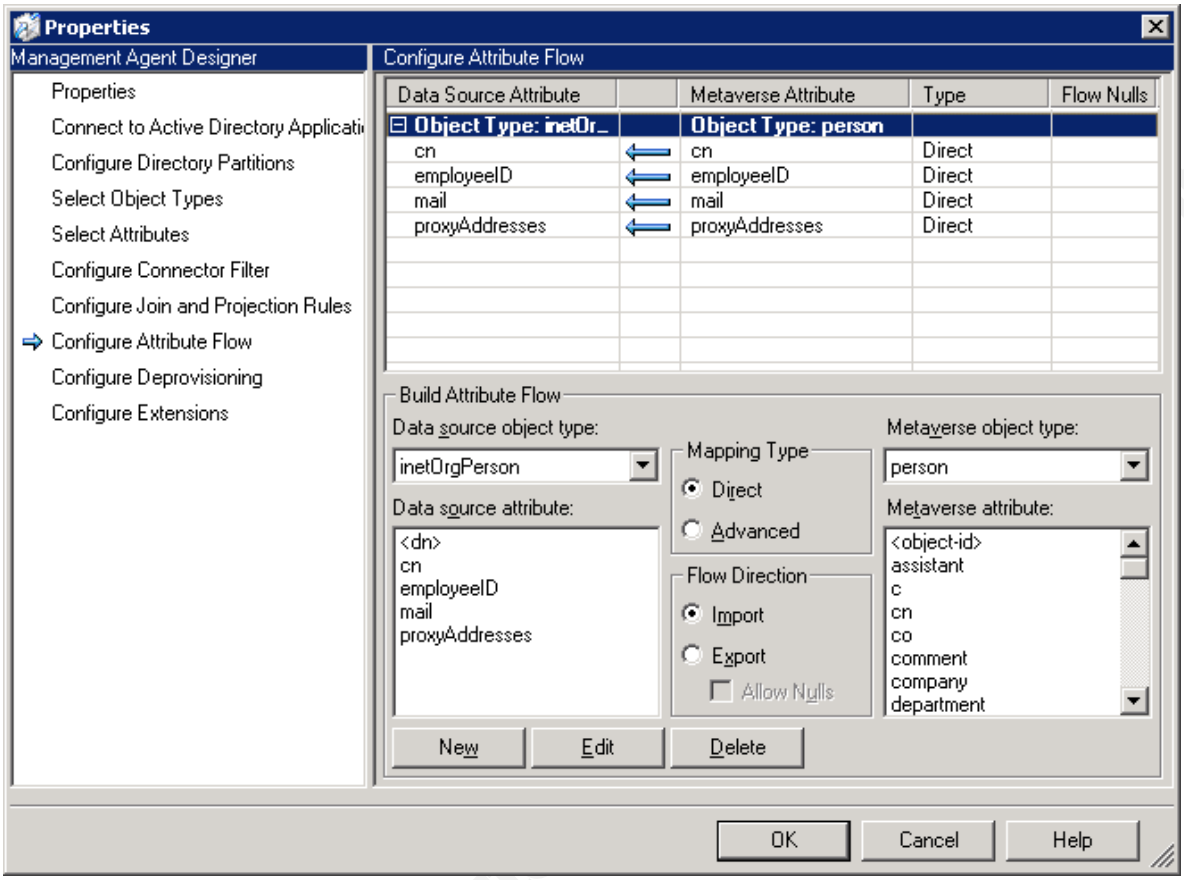


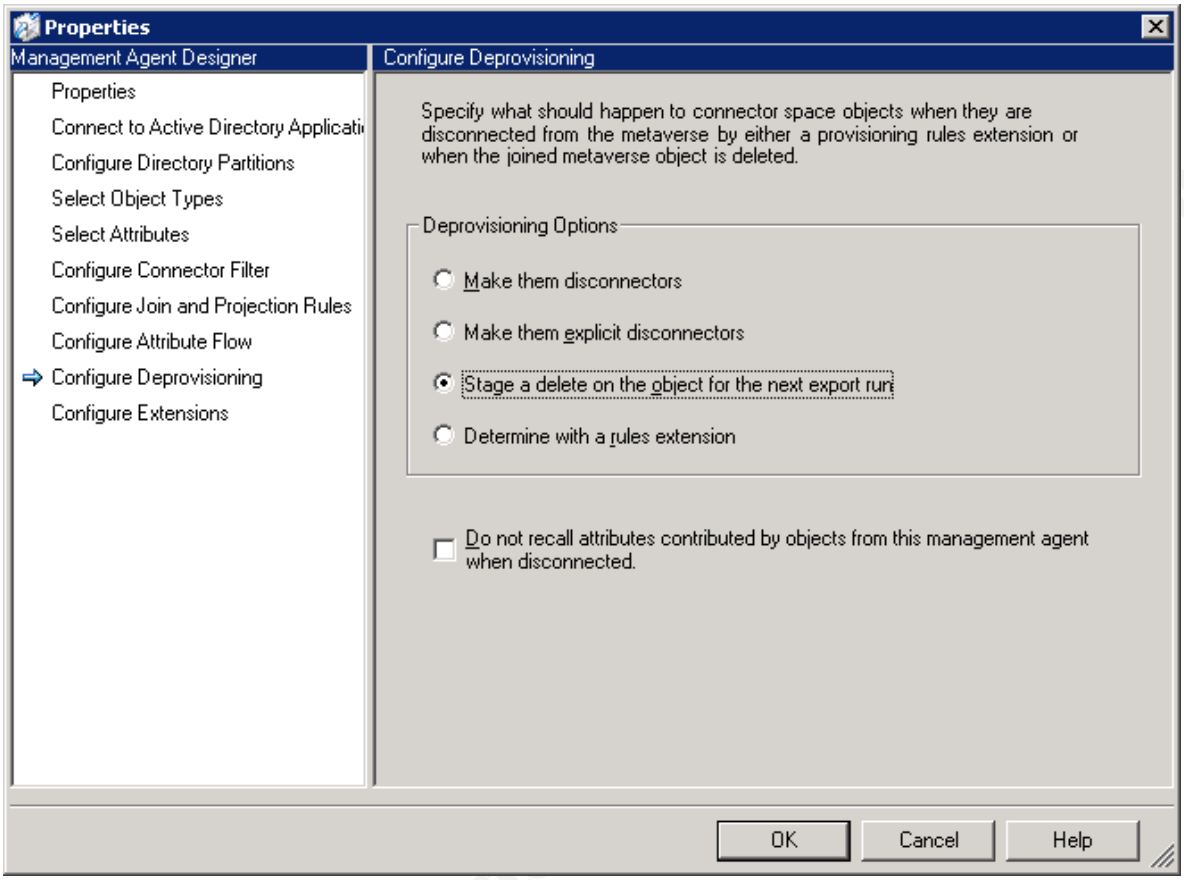


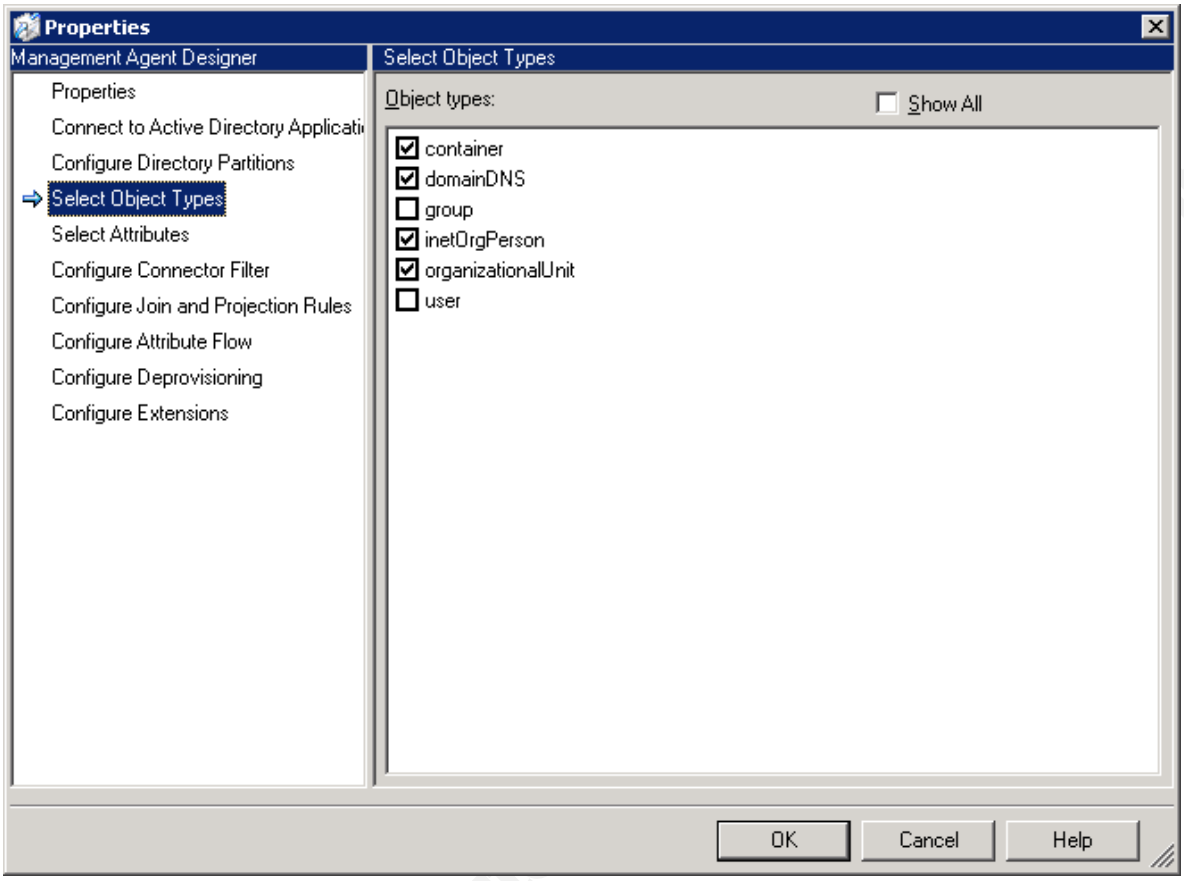
Select Mail, CN, Proxyaddresses, EmployeeID

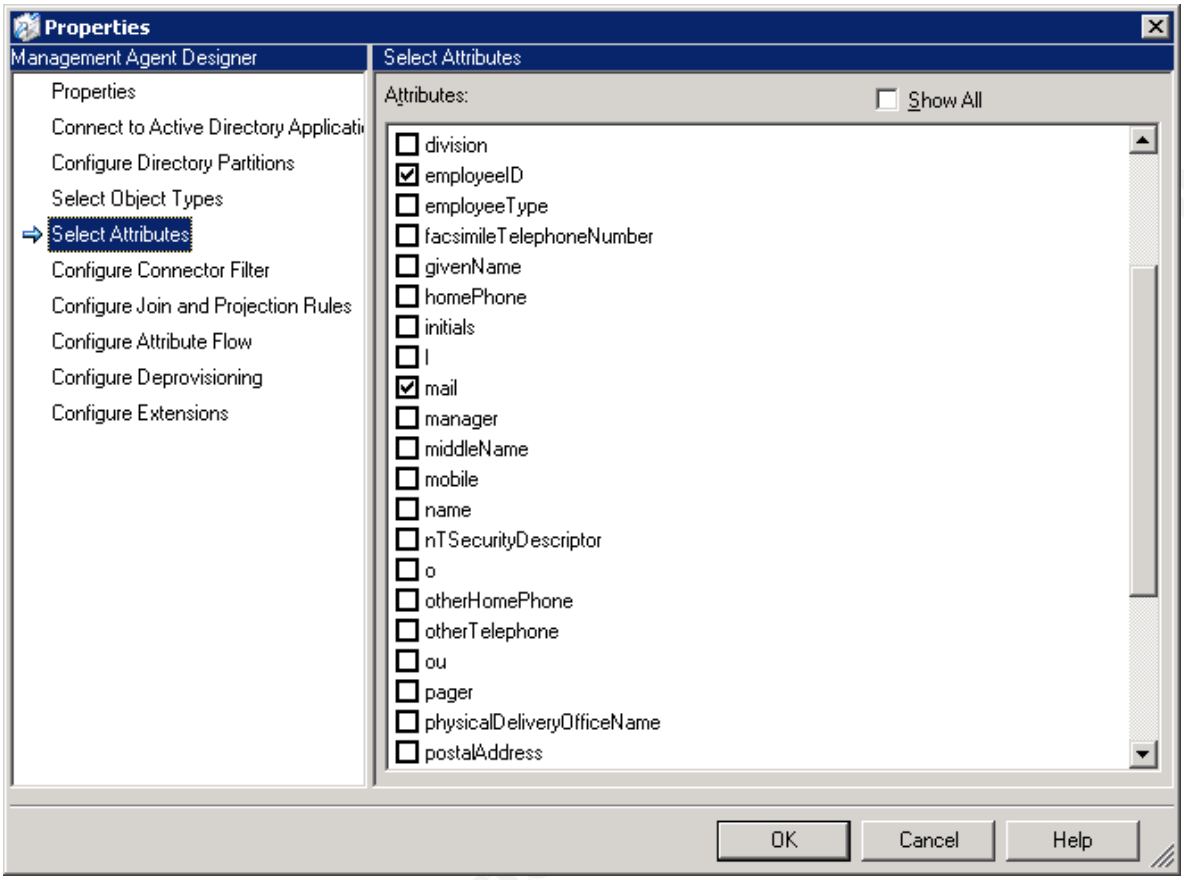




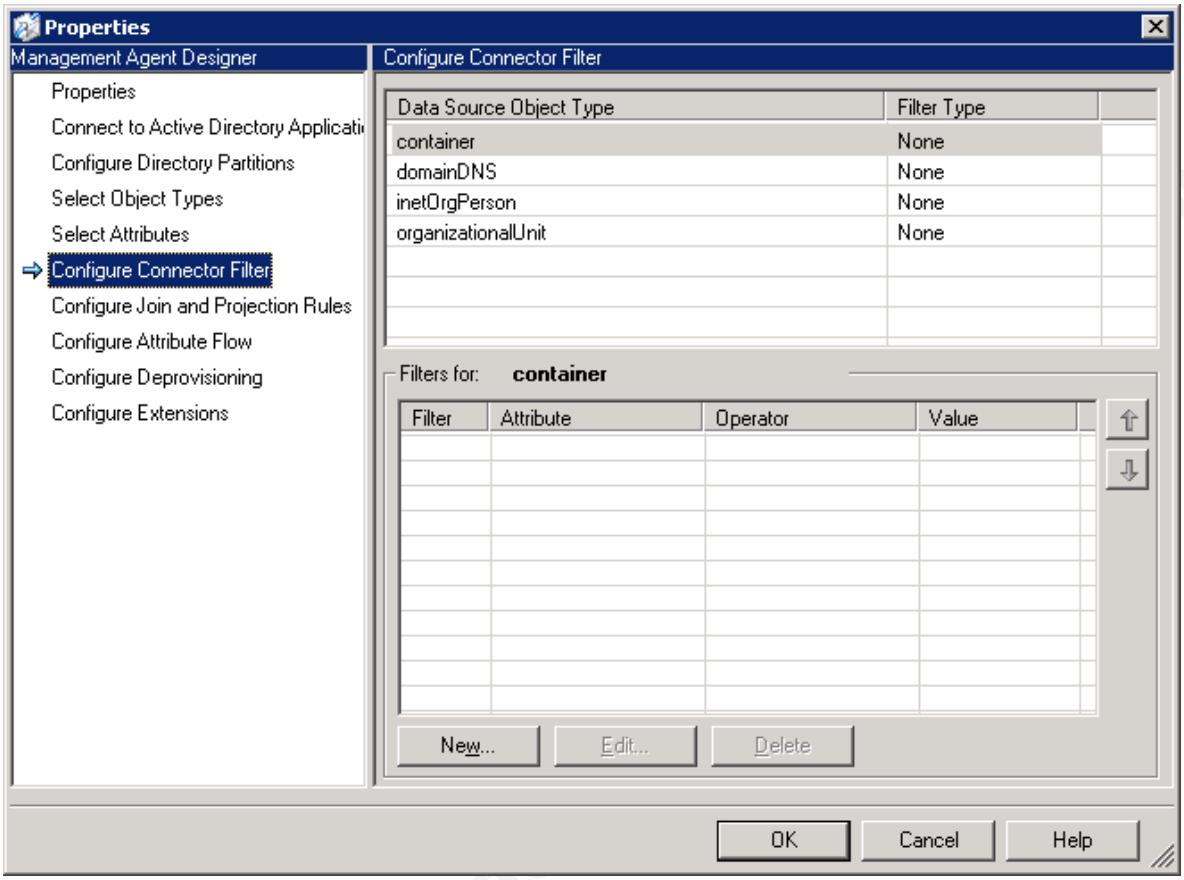


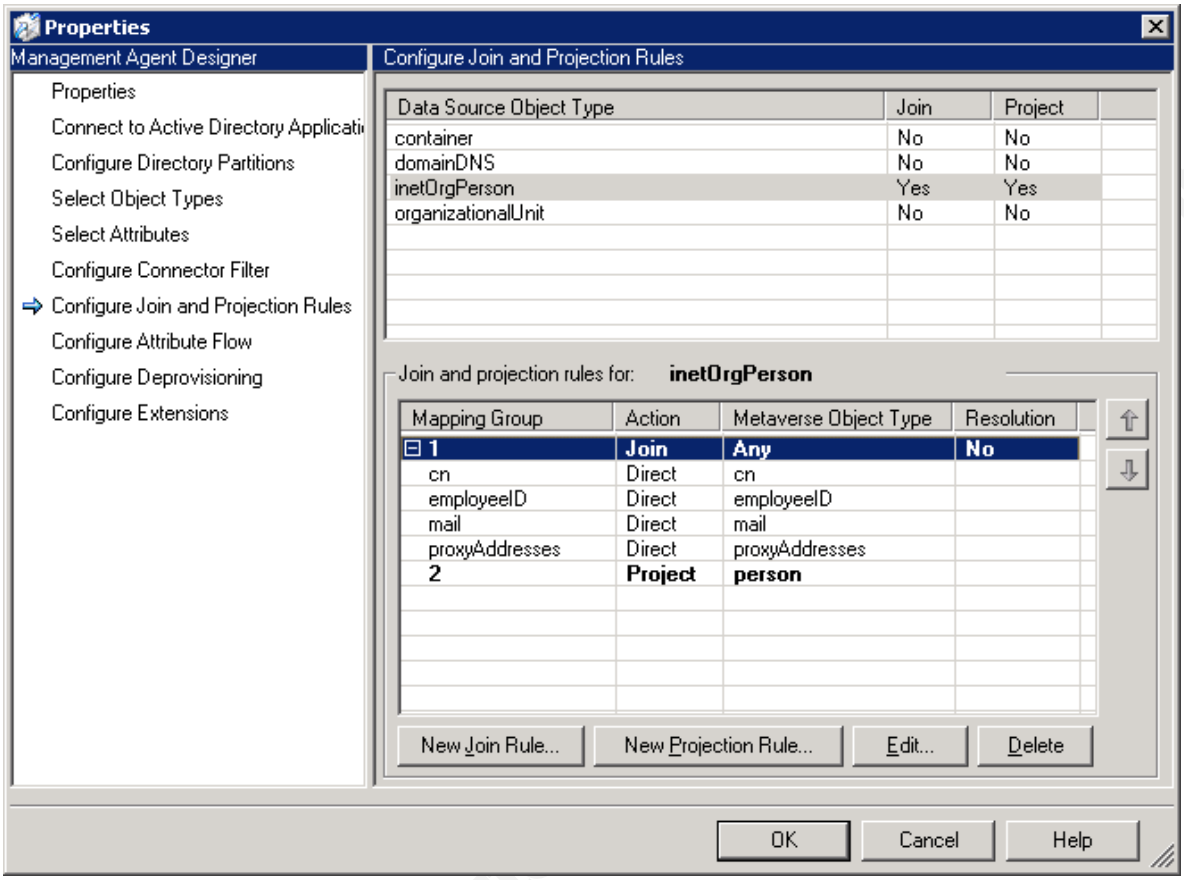


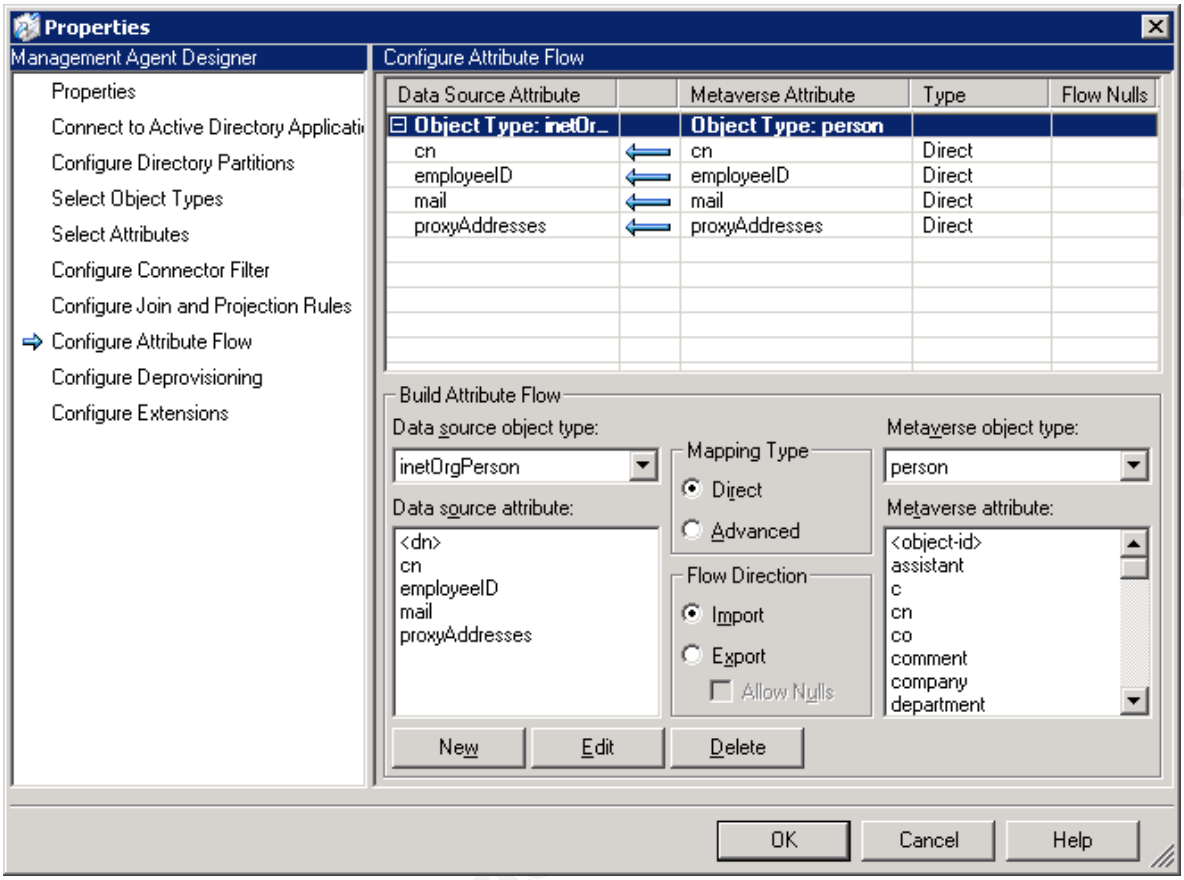


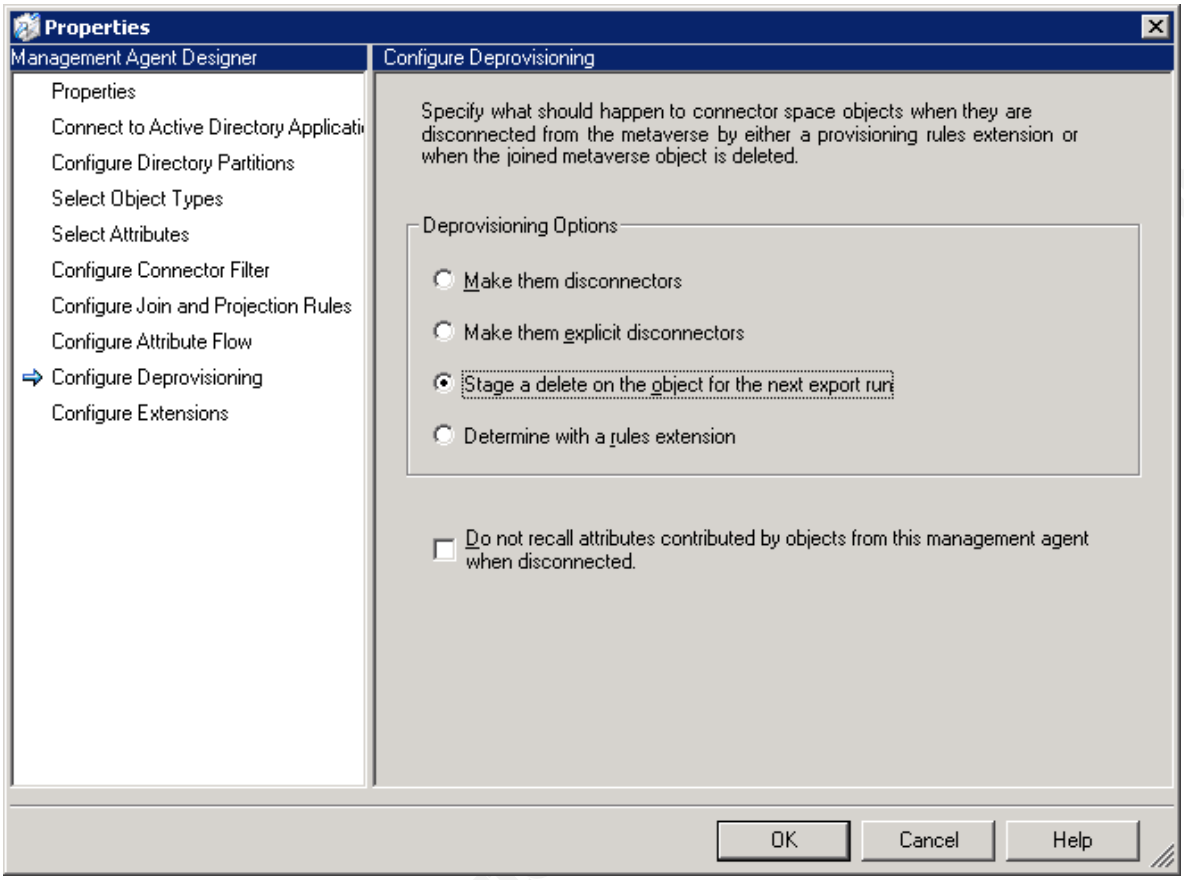


Select Mail, CN,Proxyaddresses, EmployeeID







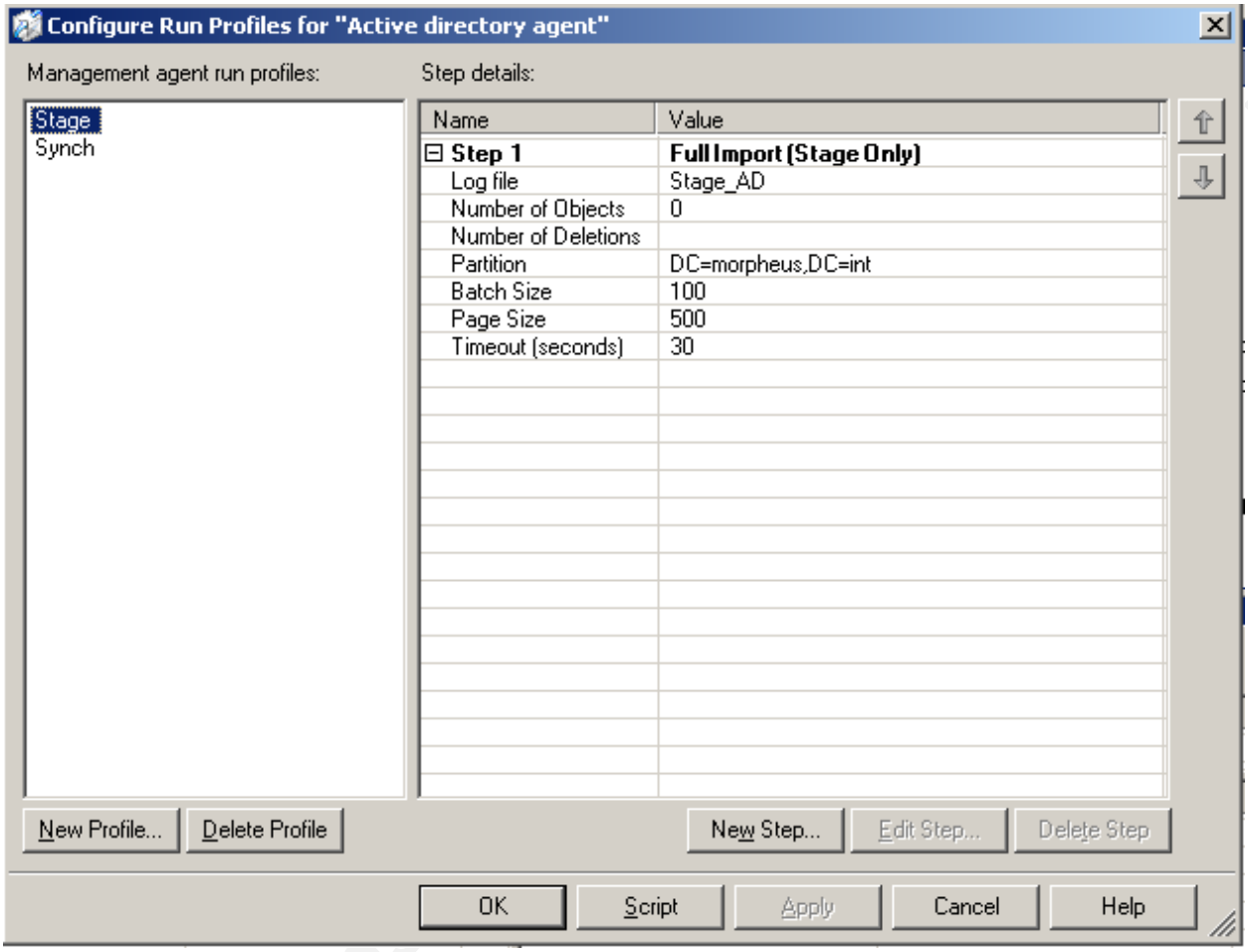


Agent Profiles

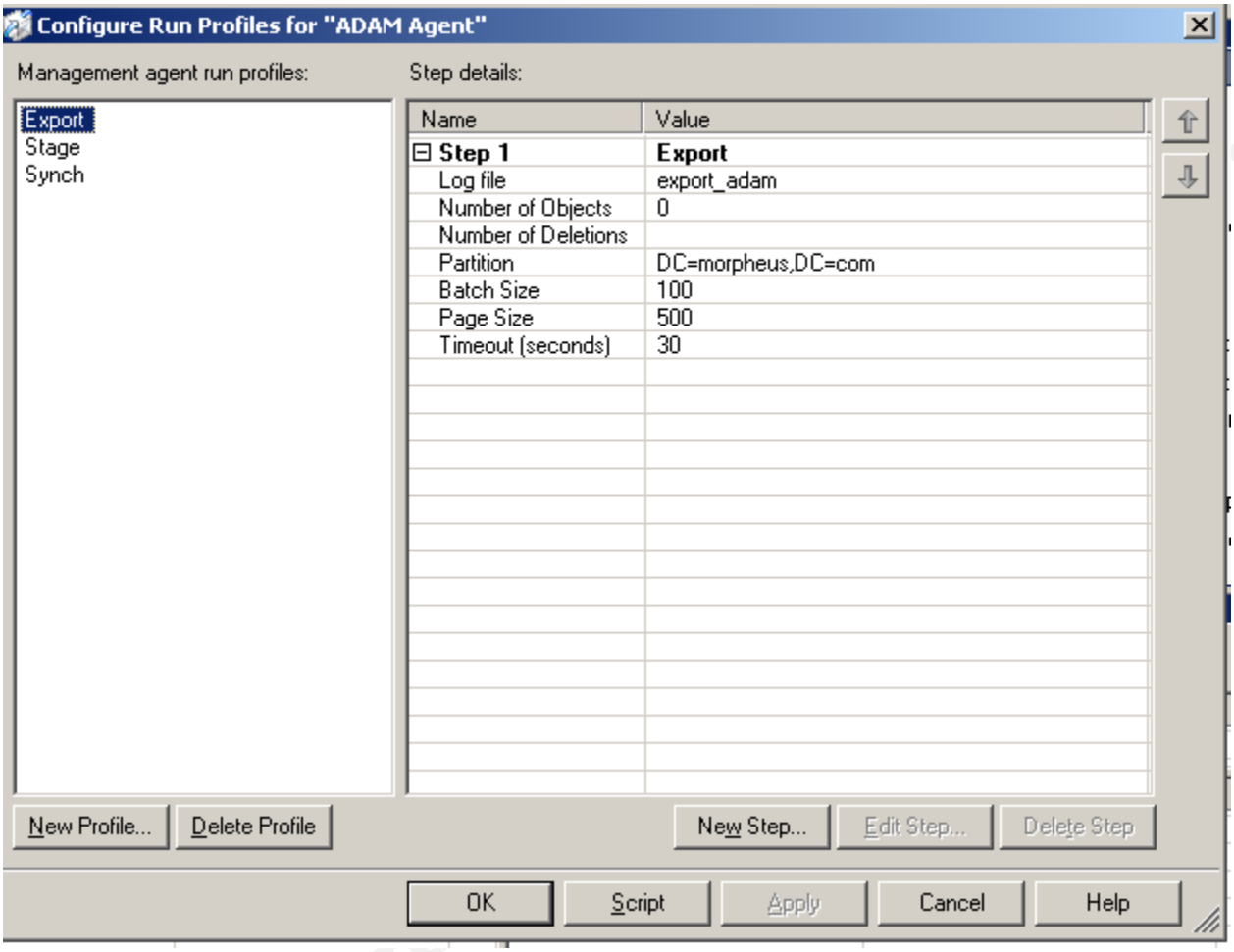
Now that management agents are defined we need to tell them which steps are required to do their job. In fact an agent will use what is called "profiles" to execute a number of steps. A "profile" contains partition target, tasks such as import/export along with number of objects to process.

A "profile" will always contain a staging and synchronizing steps. The ADAM Agent profile adds an "export" step.

Active Directory agent profiles

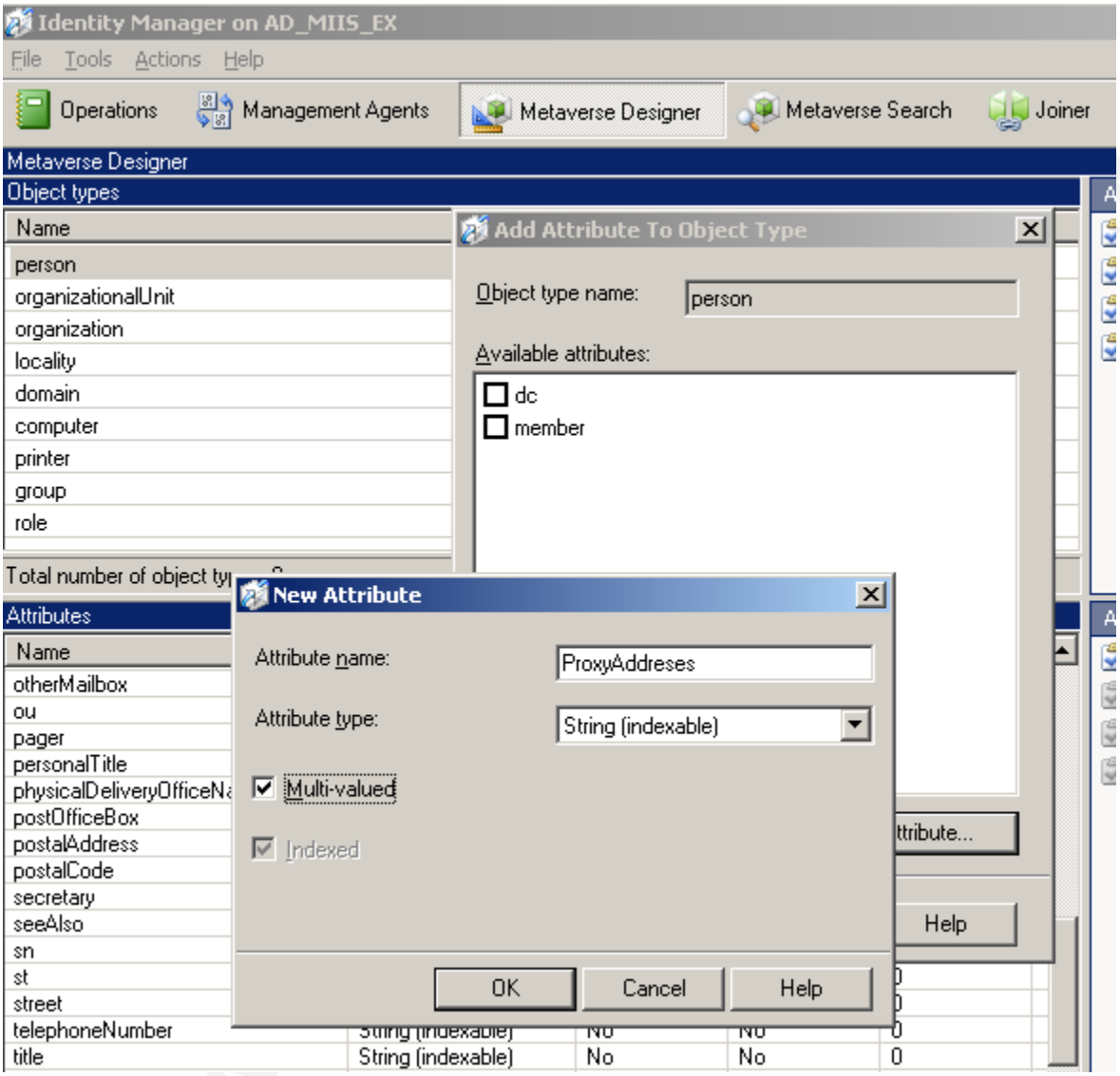


ADAM agent profile



Defining Metaverse Schema

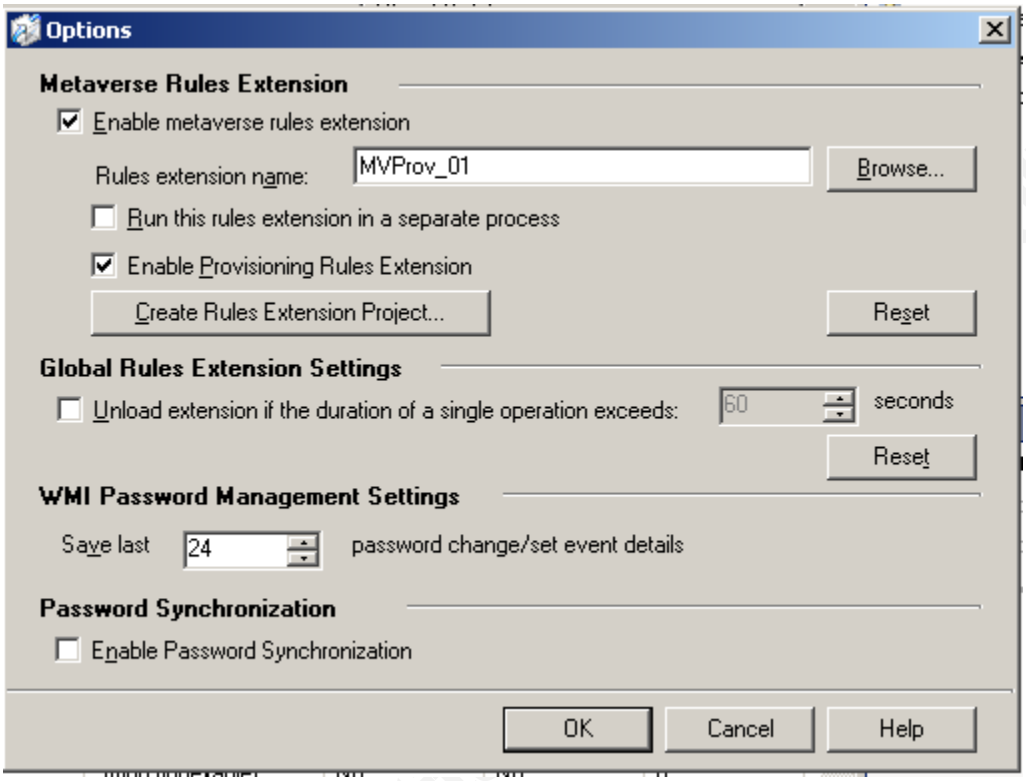
Let's add a "proxyAddress" attribute since it's missing in the default Metaverse schema. The main purpose will be to address multiple email addresses per user which is typical to any exchange infrastructure.



Provisioning engine rules

One of the main purpose of the processes we are putting in place is to provision and de-provision the ADAM instance. The metaverse provisioning rule will meet this purpose. This component must be defined as a .NET assembly DLL. Only one assembly can be defined. The .NET technology is beyond the scope

of this document. However this report will focus on the specific MIIS implementation. Further details can be found at www.microsoft.com/miis



First provisioning capability must be selected into the metaverse options pane.

A name of an assembly DLL must be submitted.

Provision Rule between the Metaverse and ADAM Connectors.

This rule is executed whenever a change occurs in the metaverse or joining or disconnecting occurs between MetaVerse and Space Connectors.

It means we only need to implement "provision" method from IMVSynchronization Interface

```
using System;
using Microsoft.MetadirectoryServices;



namespace Mms_Metaverse
{
      /// <summary>
      /// This implements the metaverse business rule to properly provision
ADAM space connector
      /// </summary>
      public class MVExtensionObject : IMVSynchronization
      {
            public MVExtensionObject()
            {
                  //
                  // TODO: Add constructor logic here
                  //
            }


            void IMVSynchronization.Initialize ()
            {
                  //
                  // TODO: Add initialization logic here
                  //
```

```
            }

            void IMVSynchronization.Terminate ()
            {
                  //
                  // TODO: Add termination logic here
                  //
            }

            void IMVSynchronization.Provision (MVEntry mventry)
            {

                  ConnectedMA ManagementAgent;  // Management Agent Object
                  ReferenceValue DN;            // Distinguished name
attribute
                  string Container;             // Container name
                  string RDN;                   // Relative distinguished
name strings
                  CSEntry csentry;              // Connector space entry
objects

                  // let's first determine the state of the metaverse object.
                  // let's check whether an employee has a mail attribute
                  if (mventry["mail"].IsPresent)
                        // ok employeestatus has been provisioned in the MV
                  {
```

```
                        Container = "DC=morpheus,DC=int";
                        RDN = "CN=" + mventry["cn"].Value;
                        try
                        {
                              // it's alright let's build container and
rdn

                              // let's work on ADAM
                              ManagementAgent =
mventry.ConnectedMAs["ADAM"];
                              DN=
ManagementAgent.EscapeDNComponent(RDN).Concat(Container);
                              csentry =
ManagementAgent.Connectors.StartNewConnector("inetOrgPerson");
                              csentry.DN = DN;
                              csentry.CommitNewConnector();




                        }
                        catch (ObjectAlreadyExistsException e)
                        {
                              // the object is already there we just
skip it
                        }
                  }

            }
```

```
            bool IMVSynchronization.ShouldDeleteFromMV (CSEntry csentry,
MVEntry mventry)
            {
                  //
                  // TODO: Add MV deletion logic here
                  //



                  throw new EntryPointNotImplementedException();
            }
      }
}
```

Deployment

The MIIS product does not feature a built in task scheduler. Specific actions must be either manually triggered or scripted through WMI or invoked through .NET Code. The next paragraph will focus on c# Generated code. The main reason is that the .NET framework security infrastructure will provide us with more advanced security options than the vbscript engine even though we will lose some flexibility. Furthermore this implementation is a perfect example of defense in-depth principles.

Code Security and "agent profile" invocation

We need to execute a number of steps for each agent at least daily.

The Active Directory management agent must execute a stage step once and a synch step. The ADAM management agent must process a stage once and synch and export step. We will rely on windows operating system scheduler but let's generate the c# code first. Open the management agent and select the relevant profile click

"script" and save your file selecting "c#". Here is the output file

```
using System;
using System.Management;


class Sample_ExecuteMA
{
    public static int Main( string[] args )
    {
        try
        {
            //
            // Credentials must only be specified when Microsoft Identity Integration Server is on remote system.
            //
            ConnectionOptions opt = new ConnectionOptions();
            opt.Authentication = AuthenticationLevel.PacketPrivacy;
            //
            // opt.Username = "Domain\\Me";
            // opt.Password = "MyPassword";
            // ManagementScope myScope = new ManagementScope( "\\\\MyServer\\root\\MicrosoftIdentityIntegrationServer", opt );
            //
            ManagementScope myScope = new ManagementScope( "root\\MicrosoftIdentityIntegrationServer", opt );
```

```
            SelectQuery myQuery = new SelectQuery(
"MIIS_ManagementAgent", "GUID='{3C222539-6284-4F18-94E4-
6EDF88364759}'" );

            ManagementObjectSearcher searcher = new
ManagementObjectSearcher( myScope, myQuery );

            foreach ( ManagementObject ma in searcher.Get() )

            {

                Console.WriteLine( "Active directory
agent.Execute( \"Synch\" )..." );

                ma.InvokeMethod( "Execute", new object[1] {
"Synch" } );

            }

        }

        catch ( Exception ex )

        {

            Console.WriteLine( "Error: " + ex.Message );

        }


        return 0;

    }

}
```

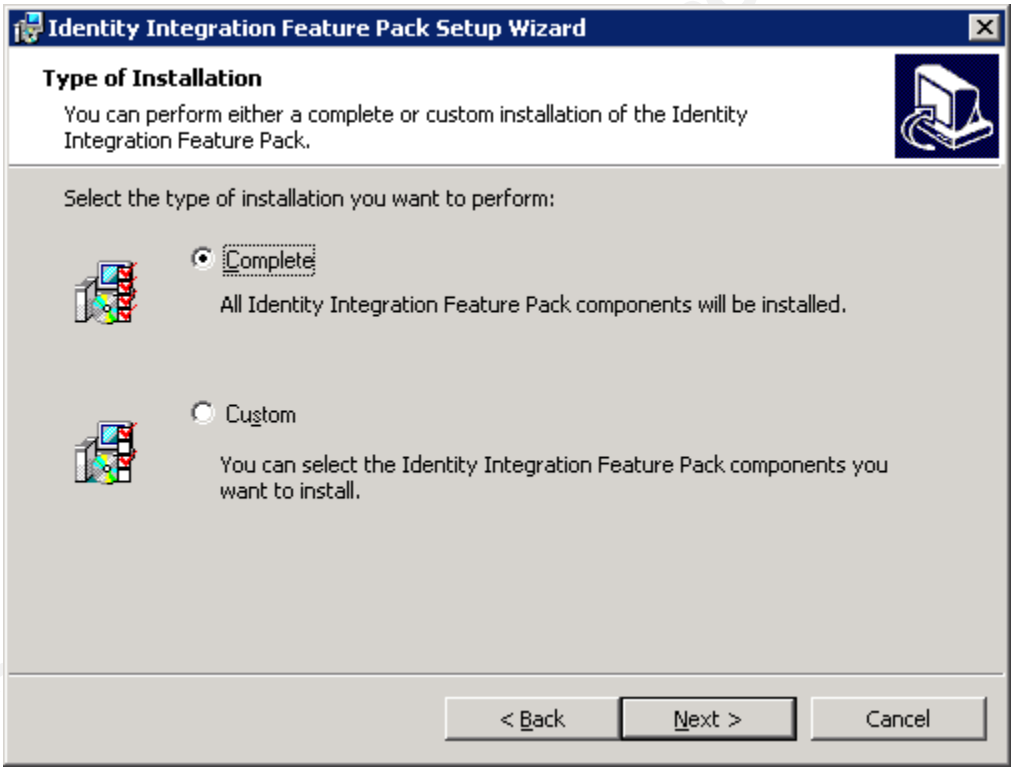
This working sample can be fine-tuned to more advance logging (log to event viewer, etc…). Once compiled you get an executable. You can then use the .NET security wizard to adjust the executable security. Furthermore this task will be executed by the scheduler using a limited service account. The only requirement is being part of the MIIS Operators local group.

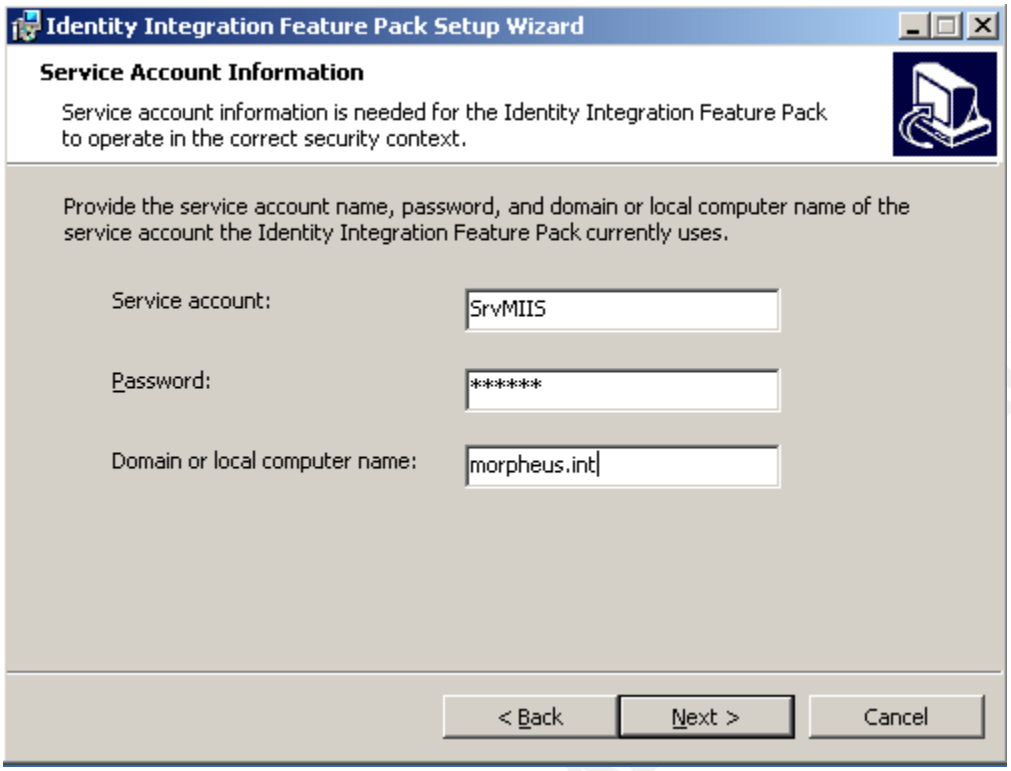
Monitoring and Logging and Troubleshooting

The MIIS Identity manager provides you with detailed statistical information about agent execution, object statistics, etc….You can also explore freely each connector space from the GUI.

In a production environment you would rely on professional products like (MOM, NetIQ, etc…) to gather statistics from either the event viewer, WMI or your own mechanism.

MIIS Installation Guide





MIIS will run under a service account that will also be used to authenticate to active directory

Scalability and Redundancy Factor

The ADAM software supports scalability through Microsoft Layer 3 NLB cluster implementation. This allows higher load and redundancy. It's also possible to deploy multiple instances of ADAM replicas. They will automatically replicate using the same mechanisms as the one used by Active Directory

MIIS being involved as an asynchronous process (it's not a real time process) it's not required to have high availability at the service level. What can be done is ensuring that the SQL database is hosted on highly available clusters.

Operational Readiness

The updated infrastructure brings updated processes. Mainly a separated staging environment will be used to qualify any

change. A separated lab environment will be setup as well as part of new change control.

Back-up

All components are backed up regularly on removable media. Furthermore annual exercises are practiced in the lab infrastructure to ensure data can be successfully recovered.

ADAM data is located in a single place while MIIS complete configuration lives at the SQL level.

Monitoring

ADAM LDAP directory is easy to monitor from an external source. The application runs as a service thus any windows operating system service monitoring tool is available.