3

MIM Sync Configuration

If you have followed the previous chapters closely, you will now have newly installed MIM environment. In this chapter, we will discuss some of the basic configurations we need to look at, no matter how our environment looks or how we plan to use MIM.

We will focus on the initial configuration of the MIM Synchronization Service. Specifically, we will cover the following topics:

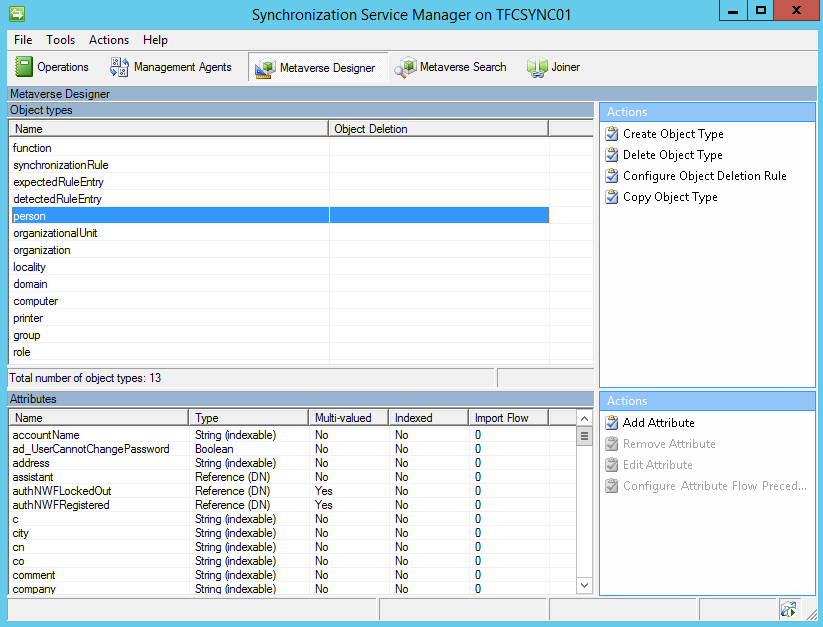
* MIM Synchronization Interface
* Creating Management Agents
* Schema management
* Initial load versus scheduled runs

Moving configuration from development to production

# MIM Synchronization Interface

Let’s start by examining the MIM Synchronization graphical interface and describing some of the tools and options available. Launching the Synchronization Service program will show an interface divided into five primary tools: Operations, Management Agents, Metaverse Designer, Metaverse Search, and Joiner. The basic features of these tools follow:

* The Operations tool provides connection status, details of new objects, object deletions, changes, errors, and internal MIM actions like projections, provisions, and joins.
* The Management Agents tool allows you to create, configure, control, and view management agents or the way we connect the synchronization engine to the various systems and pull and push data between those systems.



B04526\_03\_01.png

* The Metaverse Designer controls the Metaverse schema, the objects, the attributes associated to objects, object deletion rule, and controlling which system is authoritative for each attribute. Recall that the Metaverse is where MIM combines multiple connector space object attributes that are related to the same identity into a unique, single object.
* The Metaverse Search allows you to look at Metaverse objects and their details. Clauses can be specified that allow you to narrow down the search to a specific object or a group of objects.
* Finally, the Joiner tool enables you to manually create and destroy connections between connector space objects and their respective Metaverse objects.

# Creating Management Agents

Before we even start to use our MIM implementation to manage identities, we need to decide where the information about the identities will come from and where the information will go.It is best that we start off with the essential connections and then add other connections after verifying that the basics are working.

A very typical scenario is the one we have—The Financial Company has an HR (Human Resource) system that will, for the most part, work as the source of identity information. Then it has Active Directory, which is the primary system to receive identity information.

The basic flow will be HR - MIM - AD.

But that is only the basic flow, and as you will see later in this book, there are other sources of information and also other targets.

## Active Directory

Most MIM implementations have at least one Management Agent connected to an Active Directory.

There are a few things to consider before creating this Management Agent. First, you should have already sat down with business partners and technology teams and determined which systems you will be connecting to, which objects, which attributes, and how attribute should flow through MIM to other systems. These identity discovery and processing mapping discussions are extremely useful because you will effectively be configuring MIM to coincide with those business processes. Secondly, keep things as simple as possible and don’t try to do everything at once.

Do not try to implement everything from the beginning!

If, for example, your plan is to have MIM manage both users and groups in AD, start off by implementing the management of users and then add groups when the user part is working.

Are we interested in the whole AD or only parts?

Some businesses specifically exclude parts of Active Directory from MIM. There’s nothing wrong with excluding parts, but keep in mind this decision may impacts other requirements. For example, if a collection of users are excluded from MIM then those people will also be excluded in MIM group management. If Active Directory has group nesting, excluding a collection of groups could have repercussions.

Do I need a test environment?

Yes! You should always develop and verify your MIM configuration in a testing environment before applying the configuration to your production environment. What is the worst that can happen? A lot! Depending on what you’ve configured in MIM and the permissions your service accounts have, you could overwrite or clear data, mistakenly create new accounts, or inadvertently delete accounts. The authors have worked in support long enough to tell you that this is one lesson you do not want to learn the hard way.

### Least privileged

The Management Agent will use a service account to talk to Active Directory. The Financial Company is using the approach to have as few MA accounts as possible rather than having one account for each connected system.

In the case of The Company, the SVC-ADMA account will be the account that we will use to connect to Active Directory. What we need to do is to give this account the required permissions needed, to manage relevant objects in the AD.

You always want to apply a least privileged approach to all your accounts, especially service accounts such as the ones we will be using with our MIM management agents.

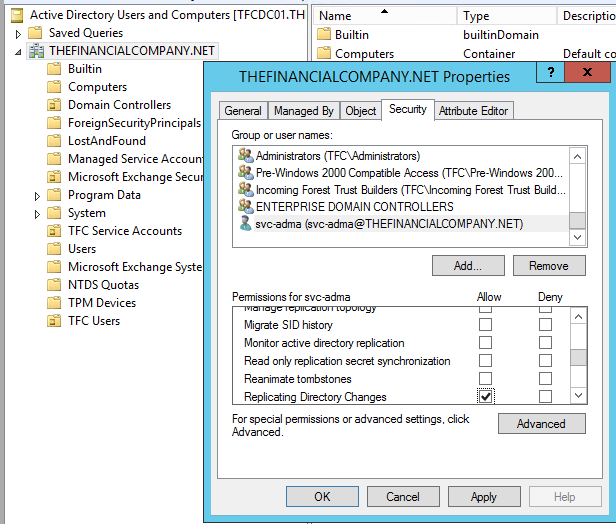
To keep things simple, our environment has user accounts in an OU named TFC Users. We then need to give MIM the required permissions to manage the   
objects. Right-click on the OU, and run the delegate control wizard.   
Give the AD MA account, SVC-ADMA, management permissions   
on user (and maybe group) objects. In some cases, the aforementioned wizard might give the AD MA account more permissions than needed. If, for example, MIM should only be able to create and manage the objects but not delete them, we need to adjust the permissions in order to use the least privileged approach.

### Directory replication

When importing (reading) information from AD, it is possible to use what is called delta. Delta means we only get the changes since the last time we checked. In order for the MIM Active Directory Management Agent to read only the changes—the delta information in AD, it needs a special permission called Replicating Directory Changes at the domain level. If you do not perform this step you will receive the error “Replication access was denied” when you attempt to read AD object data. You can read more about this at http://support.microsoft.com/kb/303972.

1. Open up the Security tab in the domain (ad.company.com for example).

You either create a group, if that is how you always do it, or you assign permission to the SVC-ADMA account(s). You need to check the Allow option for the Replicating Directory Changes permission:



B04526\_03\_02.png

Alternatively, a least privileged way is to go into the registry and create a DWORD value named ADMAUseACLSecurity and set it to 1 which will tell the AD Management Agent to use the AD ACL permissions rather than requiring the DIRSYNC permissions. You will need to create the value in SYSTEM\CurrentControlSet\Services\FIMSynchronizationService\Parameters

### Password reset

If you are implementing password synchronization and/or the Self-service Password Reset feature, you will need to assign permissions for that; details about this are given in Chapter 9, Password Management.

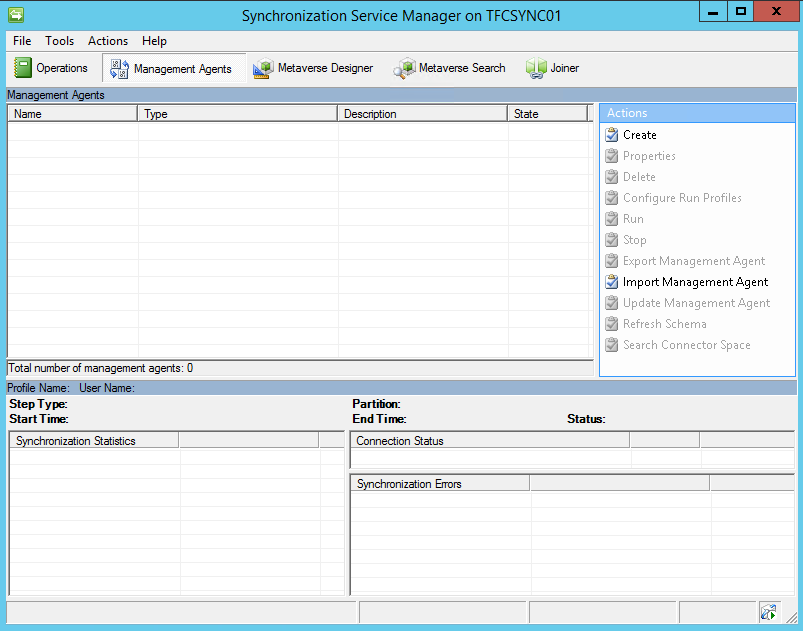
### Creating AD MA

In this segment, we will walk you through the steps for creating the Active Directory Management Agent. We will slowly work through some of the new terms, but trying to discuss every term is a sure way for beginners to get lost in the product. Some of these terms will be explained later on in this book as we start to use more advanced features.

If you are curious to know about some terms right away, you can click the Help button available on all the pages in the wizard.

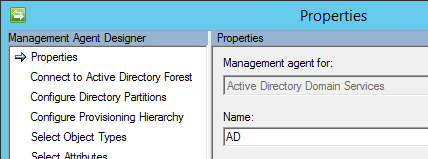
To begin, you need to log in to your MIM Synchronization server using an account that is a member of the MIMSyncAdmins group.

1. Start MIM Synchronization Service Manager.
2. Select the Management Agents tool, and click Create Management Agent in the Actions pane.



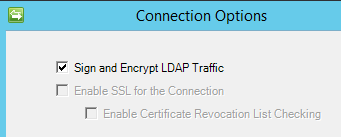
B04526\_03\_03.png

1. Select Active Directory Domain Services in the Management agent for: drop-down list:
2. Give the MA a descriptive name; at The Financial Company we simply call it AD:



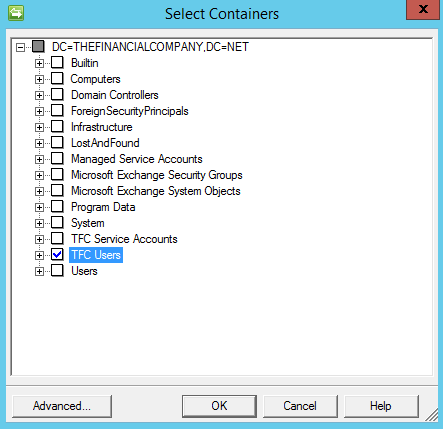
B04526\_03\_04.png

1. The AD MA connects to the Active Directory forest, and not to a specific domain in the forest. We decide later on which domain in the forest to connect to. When connecting to the AD forest, we configure the account used for the connection. We will use the SVC-ADMA account. The Options… button allows you to change the default LDAP connection options. It is recommended that you leave the default Sign and Encrypt LDAP Traffic option as it is:



B04526\_03\_05.png

1. In the Configure Directory Partitions section, select the domain partition DC=thefinancialcompany,DC=net. If you want MIM to use a preferred set of domain controllers, check Only use preferred domain controllers and click the Configure… button to choose the ones you want MIM to use. Specifying a preferred domain controller or domain controllers means MIM will only use those domain controller(s). If you were to specify a single domain controller and that domain controller is down for maintenance or decommissioned, MIM will need to be changed to add a usable domain controller.
2. The default is to work with the whole domain; but we do not want that, so let’s click the Containers… button. In the Select Containers dialog, uncheck the domain (top) level, thereby unselecting all the options. Then select the containers you want MIM to manage. In our example, we select the TFC Users OU:



B04526\_03\_06.png

1. On the Configure Provision Hierarchy page, we do not need to change anything; just click Next:

Provision Hierarchy means MIM can automatically create a missing OU if needed during provisioning. In our example, if we configured MIM to provision Active Directory accounts to an OU named TFC User Accounts, which does not exist, rather than throwing an error message MIM would create the OU.

1. In the Select Object Types page, select the object types, which you know MIM needs to manage. Keep the defaults and add only what you need. Do not deselect the default Container, domainDNS and organizationalUnit object types because these are required for MIM to know where in AD the objects reside. Do not select object types you have no need for. Initially, The Financial Company has no need for the contact object type so we do not select it. If we need any of these objects in the future we can only change the configuration.
2. Select the attributes that you know you need. Needs will be discussed in the following chapters, and we will make frequent changes to this configuration. If you check Show All, it will display your complete AD schema which includes any custom schema changes you have made. There are a few special attribute, objectSid and sAMAccountName, that are required if you want users to access the MIM Portal. For our basic demonstration make sure the following attributes are checked: department, displayName, employeeID, employeeType, givenName, manager, middleName, name, objectSid, pwdLastSet, sAMAccountName, sn, title, unicodePwd, userAccountControl, userPrincipalName

If for some reason we have configured the containers and object types in a way that we can reach objects we are not supposed to manage, we can make a connector filter to make sure these objects are out of scope. We will configure a connector filter in our MIM Service in the next chapter.

1. The Join and Projection rules will be configured using MIM Service in our environment. So, click Next.

If you are running only Synchronization Service or for some other reason using non-declarative (classic) synchronization, this is where you will configure your Join and Projection rules for the AD MA. We will discuss that later in the chapter.

1. Attribute flow will be configured using MIM Service in our environment. So, click Next.

If you are running only Synchronization Service or for some other reason using non-declarative (classic) synchronization, this is where you will configure your attribute flow rules for the AD MA. MIM supports the usage of both declarative and non-declarative attribute flows in your MAs.

1. On the Configure Deprovisioning page, there are a few things we need   
   to consider:
2. Deprovisioning is what happens when an object in the connector space is disconnected from its Metaverse object. We will look into how we can control this, later in this book. If you are uncertain, leave the default value as Make them disconnectors.
3. Stage a delete on the object for the next export run is what you will select if you want MIM to delete objects in AD when they are disconnected from the MV. To actually have deletes of users and groups in AD could cause a lot of problems, if they occur when they shouldn’t. In all cases, when we allow MIM to perform the deletes of objects in a CDS, we need to be very careful.
4. The Do not recall attributes contributed by objects from this management agent when disconnected checkbox might sometimes be useful if, for example, you are replacing a Management Agent with a new one and do not want the MV attributes to be deleted in the process.

Carol Wapshere wrote a great article explaining the FIM and MIM options for deprovisioning. Go to http://aka.ms/FIMDeprovisioning, and read it before you start using the options for deprovisioning.

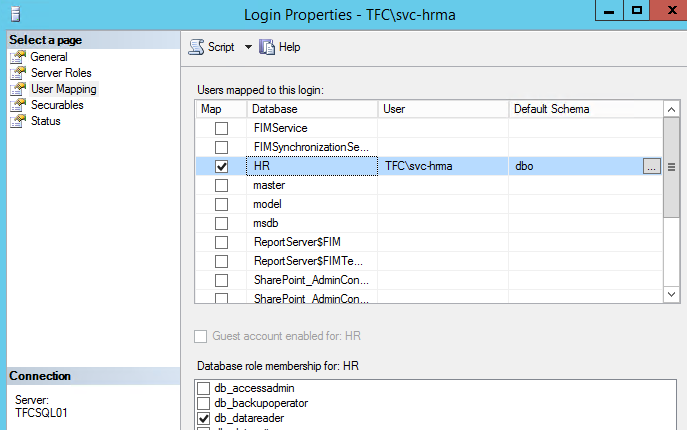
If you are doing a non-declarative (classic) synchronization using only Synchronization Engine and are using code to solve some problems, this is where you will configure which DLL contains your code. This is also where you will select the version of Exchange that you will use, if MIM is to provision users for Exchange. For now, we will leave this as No provisioning.

## HR (SQL Server)

The most popular MIM connection is Active Directory and the second most common connector is SQL Server. For those organizations that do not have identity data in SQL, there are occasions when creating a few SQL tables will assist in your identity management solution.

At The Financial Company, the HR system uses SQL Server as a database and we will interact with HR using a typical SQL MA. As with Active Directory, we should implement a least privileged approach when assigning permissions to the account that MIM is using to connect to SQL.

As the HR database (at present) is not supposed to receive any data, just send the data to MIM; we can assign the db\_datareader permissions to the SVC-HRMA account:



B04526\_03\_07.png

At The Financial Company, the HR data is in a database named HR.

If you want to filter what information is available to MIM in SQL, you can easily do that by creating a SQL view and configuring MIM to read from that view. Just remember that when MIM is using a SQL view to talk to SQL, updates become a little trickier. If you create a complex view for MIM to read and later on realize that MIM should also be able to update some column in some table, it may not be possible without redesigning the view.

Before we can configure our MA, we need to understand the data source we are connecting to. So, let’s take a quick look at how the HR database is built up.

In the HR table (named HRData), there is information about our users and organizational units. Note the relation we have between the column manager which references the objectID column.

If the SQL data has this kind of reference information, we will be able to use this to synchronize these to attributes in other CDSs, which also use reference attributes. For example, as the manager column in our HR data is a reference value, MIM can easily populate the manager attribute in AD, and also reference an attribute pointing to another object in the AD.

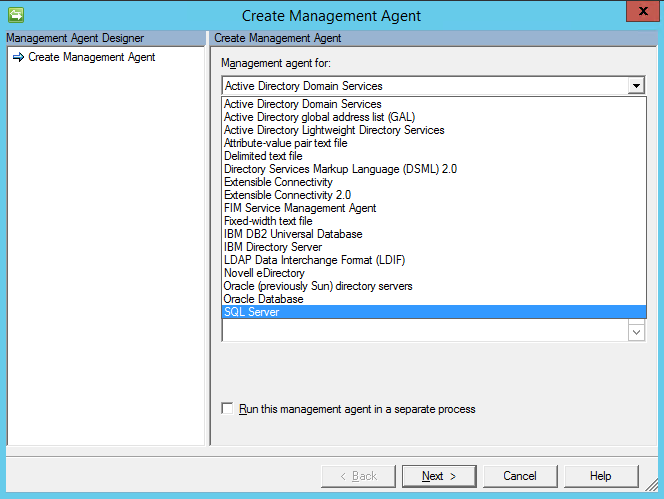


B04526\_03\_08.png

### Creating SQL MA

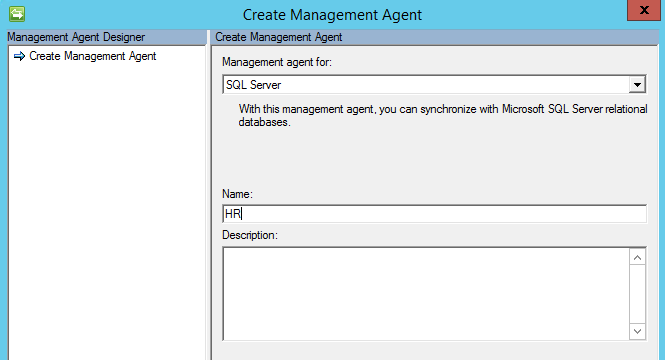
In this section, we will walk you through the process of creating the SQL MA for the HR system:

1. Start the Synchronization Service Manager.
2. Select the Management Agents tool, click Create Management Agent in the Actions pane.



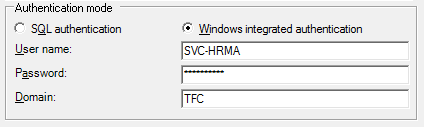
B04526\_03\_09.png

1. Select the SQL Server option in the Management agent and give the MA a descriptive name such as HR



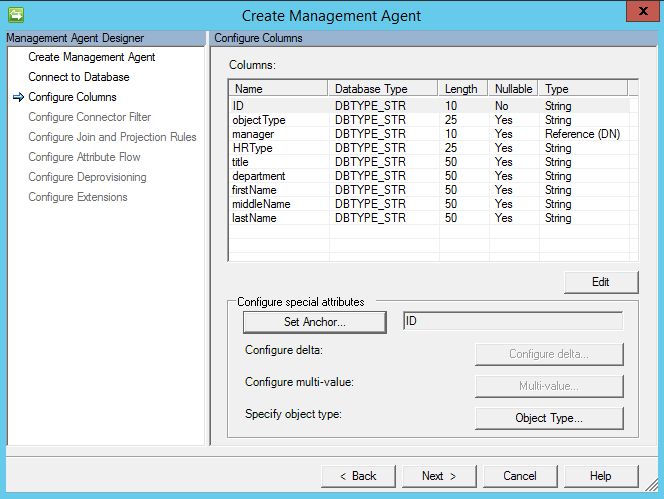
B04526\_03\_10.png

1. As we are using SQL aliases, we use the alias server name dbHR. The database is **HR** and the base table is HRData. We are using Windows integrated authentication with the SVC-HRMA account.



B04526\_03\_11.png

1. Clicking Next should show the SQL MA has retrieved the schema, the Columns, and the Database Types from the SQL database.

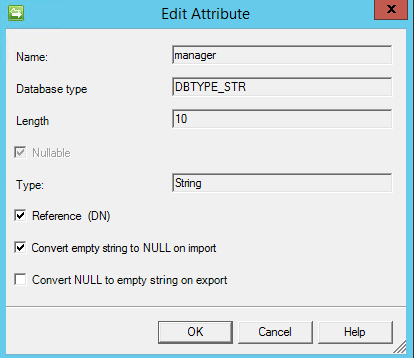


B04526\_03\_12.png

1. In our case, because the ID column is a primary key, the SQL MA automatically set the ID as an anchor. If you needed to modify the anchor in your environment, you would click on the Set Anchor… button, and set the anchor attributes accordingly.

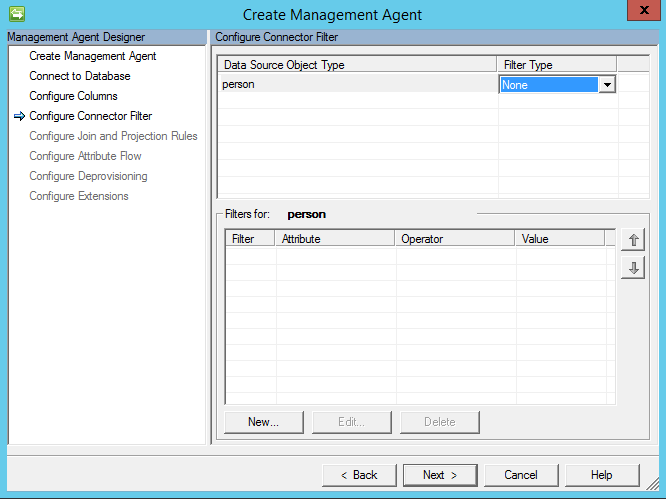
The anchor attribute consists of the column in the database that contains the unique value of each object, which does not change. Which attribute to be used as an anchor attribute in each of the CDSs, is an important decision to make. The anchor attribute value should **never** change for a specific object; the value should remain the same for the entire lifecycle of the object. If the anchor attribute changes, it will be detected as a delete of the old object and an addition of a new object by MIM, when importing information from the CDS.

1. Clicking the Object Type… button allows you to define if the SQL MA only contains one fixed object type or if the information about object type is stored in a column. If you can get this information as a column in the view or table, that would be better. This particular setting can only be configured during the creation of the MA; if you would like to change this later on, you will need to recreate the MA. In our HRData table, we have the object types in the column objectType. In order for MIM to detect the possible object types available, the table or view we look at must contain sample data with the possible object-type values.
2. There is one attribute in the list that need to be edited, as we need to tell MIM that it is of the Reference (DN) type. A Reference (DN) type tells MIM that the data in the column contains the ID value of some other object. Select manager, click the Edit… button, and check the Reference (DN) checkbox:



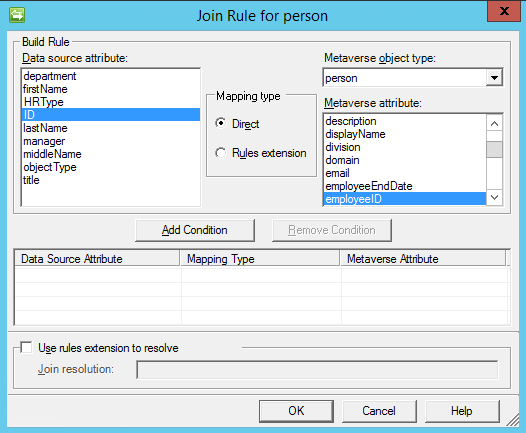
B04526\_03\_13.png

1. If for some reason we have configured the table or view used by the MA in a way that we reach objects we are not supposed to manage, we can configure a connector filter to make sure these objects are out of scope. MIM is essentially asking if there is attribute criteria that should filter or block connector space objects from connecting to their respective metaverse objects (MIM calls this process a join) or if MIM should block a connector space object from creating its own unique metaverse object (called a projection). In our example, everyone in the HR source system should be provisioned an Active Directory account therefore we keep the defaults of Filter Type **as** None and click Next.
2. We will configure a connector this in another management agent, so let’s leave it as it is:



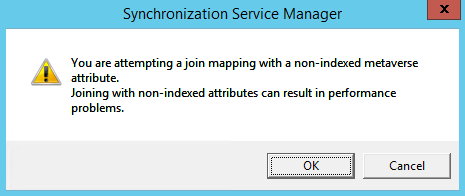
B04526\_03\_14.png

1. Configuring join and projection rules are next. Our anchor is ID therefore we should specify a join rule with ID. Click the New Join Rule button to open the join rule window:



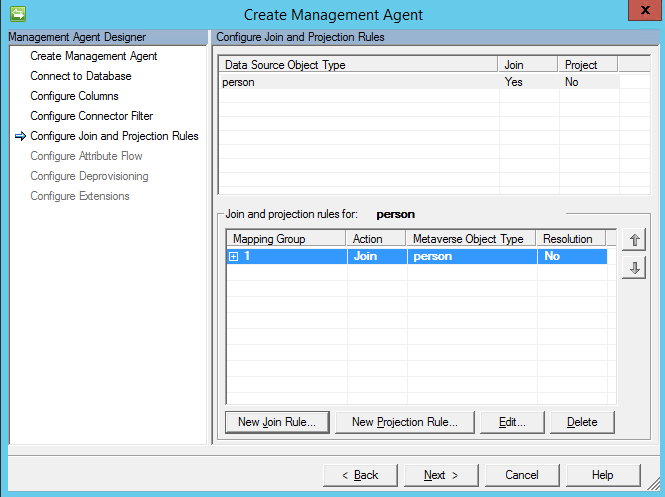
B04526\_03\_15.png

1. Change the Metaverse object type to person, select **ID** on the Data source attributesection, and employeeIDfor the Metaverse attribute. Click the Add Condition button and a non-index join warning message appears:



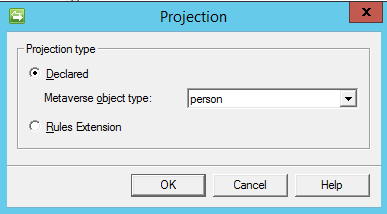
B04526\_03\_16.png

1. MIM is warning us that finding a matching employeeID value in the metaverse would be faster if we indexed that attribute. Click OK for now and we will show you where attribute index is later in this chapter. Click OK to finalize the ID to employeeID join and you will be back at the Configure Join and Projection Rules step:



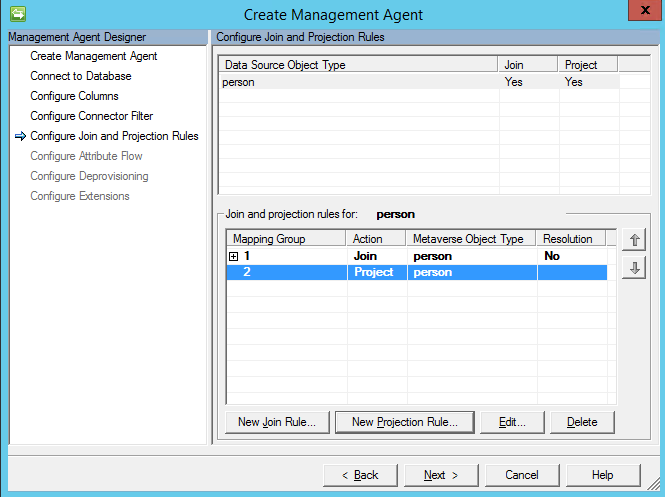
B04526\_03\_17.png

1. We now need to configure a project rule. This is one of the easiest things you will do today – click the New Project Rule button and the Projection type window will be displayed:



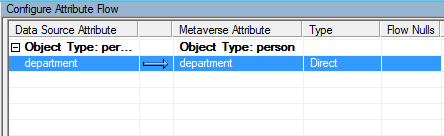
B04526\_03\_18.png

1. Accept the default person Metaverse object type and click OK.



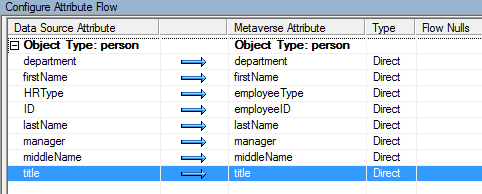
B04526\_03\_19.png

1. Click Next to move on to the next step.
2. We will now configure attribute flow which is mapping attributes in the connector space to attributes in the Metaverse. This means the connector space attribute value can be copied to the mapped Metaverse attribute. In our case, we are mapping the connector space object type person (left hand side) to a person object type in the metaverse (right hand side). On the Data source attribute section (left hand side), click on **department** and click on department in the Metaverse attribute (right hand side). Keep the Mapping Type set to Direct and Flow Direction set to Import. Click on the New button to add the mapping. You should see a new attribute flow like below:



B04526\_03\_20.png

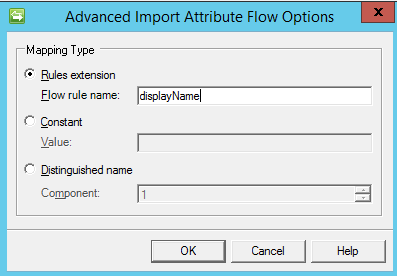
1. Perform the same steps to setup an import attribute flow as shown:



B04526\_03\_21.png

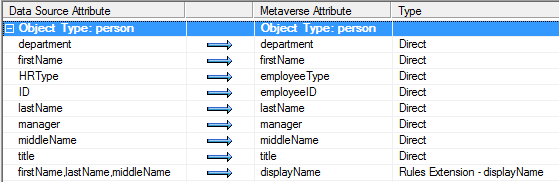
This is a good time to talk about attribute names. Often people new to identity management will get caught up on connector space attribute names not matching with the same attribute names in the metaverse. For example, the attribute HRType does not exist in the Metaverse. Should you change your HR system or create a new Metaverse attribute? Ultimately, it is your decision, but there is no reason to re-architect your source and target systems simply because attribute names do not match. In this case, something like employeeType effectively has the same function therefore it can be used. Non-matching attributes are expected in the identity world because of the disparate systems. Our advice? Get over it.

1. Let’s setup an import attribute flow for our display name. TFC would like identities to have a display name comprised of the first name, the first letter of the middle, then the last name. Notice our source system does not have a display name attribute, but we can build it with some simple code. Not a developer? Don’t panic! As you will see, it is not so bad. First, click on **Advanced** in the Mapping Type and click on firstName on the connector space (left hand side) section. Hold the control key down and click on middleName and lastName to select the other attributes needed to build the display name. On the metaverse attribute section (right hand side), click on displayName.
2. Click on the New button to bring up the advanced window and change the Flow rule name to displayName:



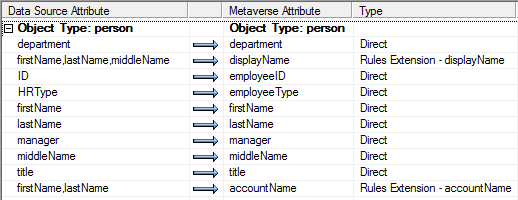
B04526\_03\_22.png

1. Click OK. The screen should now show the advanced import attribute flow:



B04526\_03\_23.png

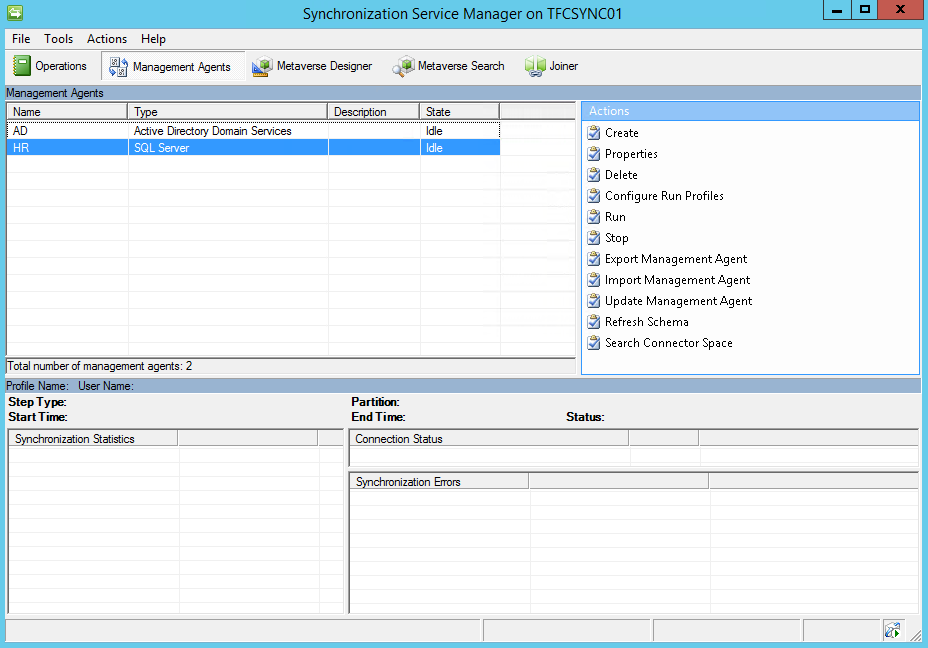
1. Add another advanced import attribute flow for accountname as shown below:



B04526\_03\_24.png

1. We will complete the displayName and accountName and rules after we finish the remaining two steps. Click Next to move on to the deprovisioning step.
2. The HR system is a source system which will not have deprovisioning. On the Configure Deprovisioning page we will click Next to keep the default Make them disconnectorsand click Next. If you are impatient and want to know what the different options provide, Carol Wapshere has written a great article explaining the options around FIM and MIM deprovisioning. Check it out at http://aka.ms/FIMDeprovisioning.

The final step specifies the rules extension name which was auto calculated as HRExtension.DLL. You could change the name if you want, but we will keep the default for the purpose of this example. This DLL that we will create will contain the coded displayName and accountName that we want to generate. Click the **Finish** button to complete the creation of the HR management agent. You should now see two management agents in the Service Manager console: AD and HR:

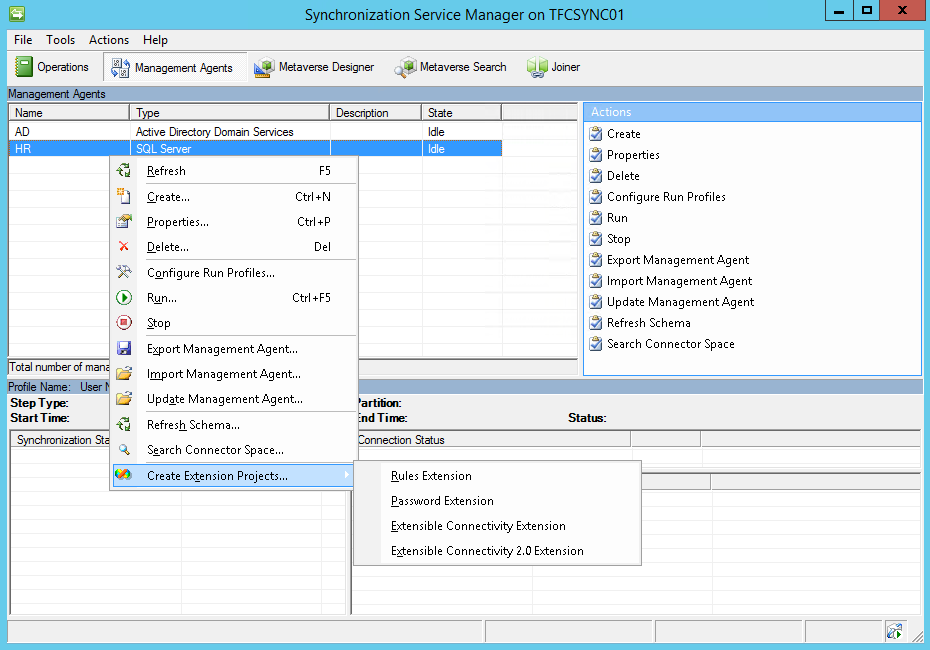


B04526\_03\_25.png

# Creating a rules extension

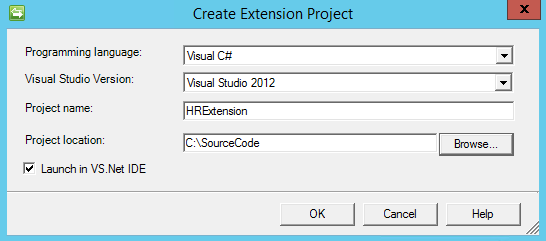
A rules extension supplements the MIM management agent and provides the flexibility for you to build customized rules. We will walk you through a simple (and common) example of building an attribute value from the values of other attributes. TFC wants displayName to be firstName, first initial of middleName, and lastName. Follow the steps below:

1. Let’s begin by right clicking on the HR management agent, hovering over Create Extension Projects, and selecting Rules Extension.



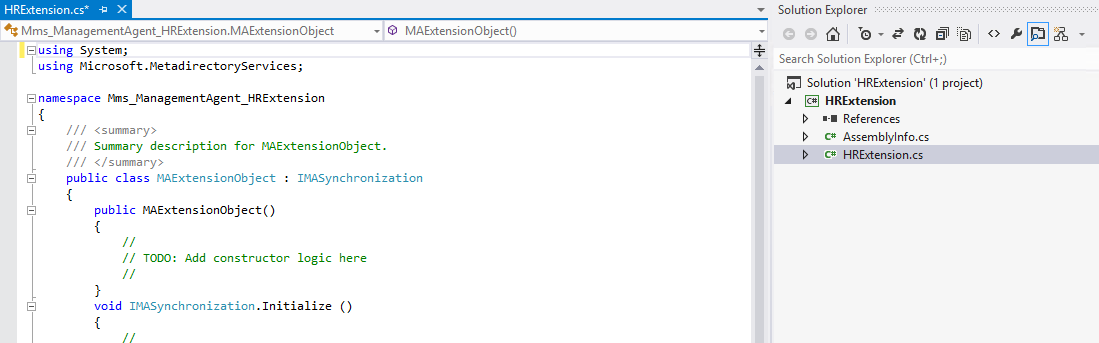
B04526\_03\_26.png

1. The Create Extension Project window appears. We will write our rules extension in Visual C# using Visual Studio 2012 and store our source code in C:\SourceCode. Click OK to launch Visual Studio:



B04526\_03\_27.png

1. The Visual Studio interface will load. Double click on the HRExtension.cs file so that the HRExtension.cs is opened:



B04526\_03\_28.png

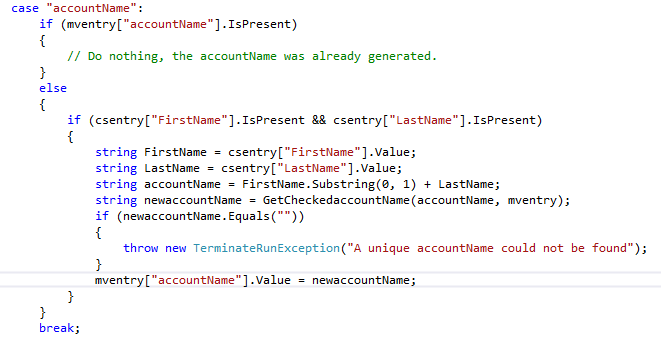
1. Scroll down until you see IMASynchronization.MapAttributesForImport. You should see case displayName: which matches, and not coincidentally, the same name we specified when we created the displayName Advanced Import Attribute Flow Options.

One way to accomplish our goal is to use the code below:



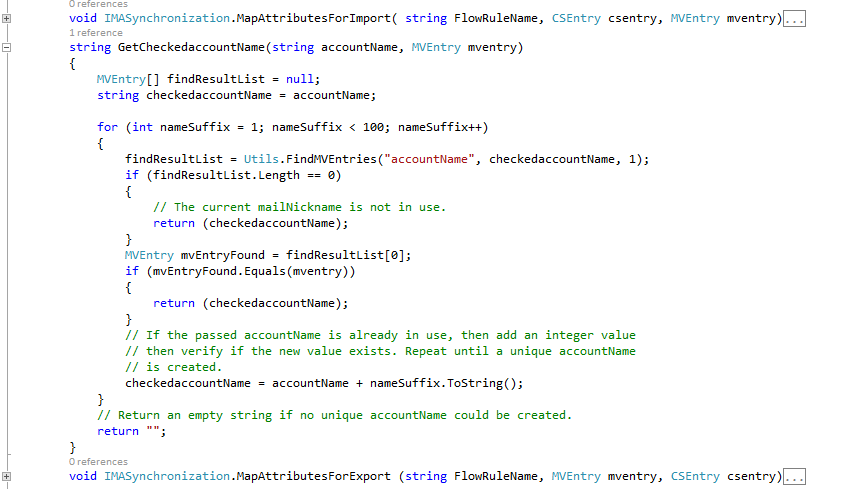
B04526\_03\_29.png

After the break we will have another case statement for the accountName. Here’s one way to handle that:



B04526\_03\_30.png

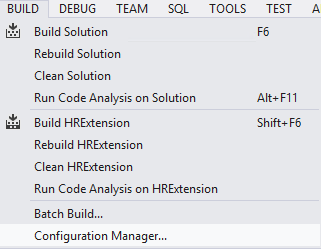
1. After the IMASynchronization.MapAttributesForImport and before IMASynchronization.MapAttributesForExport, add a new GetCheckedaccountName method like this:



B04526\_03\_31.png

All this second piece of code does is verify the accountName is unique and if not, adds an integer value to the end.

1. Click on Build, then Build Solution to compile the DLL. That’s it!

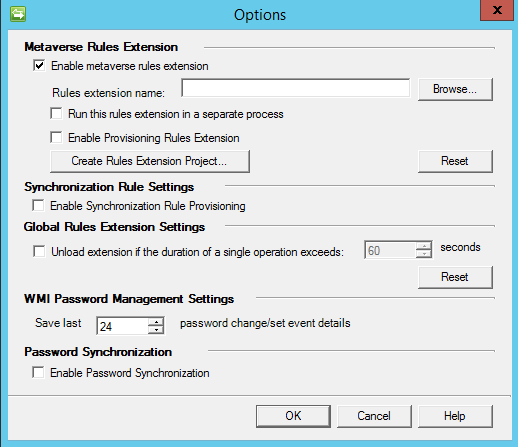


B04526\_03\_32.png

# The Metaverse rules extension

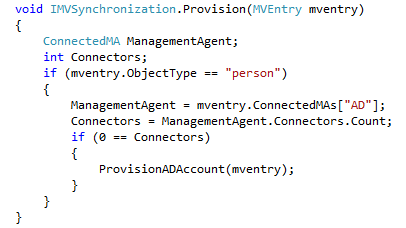
There’s one more rules extension we need to create, the Metaverse rules extension. A management agent rules extension, like the HR one we just created, is a DLL that allows us to manipulate data between the connector space and the Metaverse. The Metaverse DLL allows us to manipulate data between connector spaces. In our scenario, we want to push HR data to the Metaverse (this was done by setting HR management agent to “project”) and then from the Metaverse out to AD. Another, way to look at the need for a Metaverse rules extension is when you need to specify one-time or an initial value for one or more attributes. For example, if you were to create an AD object using any other tool, you would need to specify a password. We set our password and any other attributes that only need to be performed once in our Metaverse rules extention. Follow the steps as described below:

1. In the Management Agents tool, click on Tool then Options… check Enable metaverse rules extension and click Create Rules Extension Project.



B04526\_03\_33.png

1. Once Visual Studio opens the Metaverse solution, you will want to go to the Provision method and enter the following:



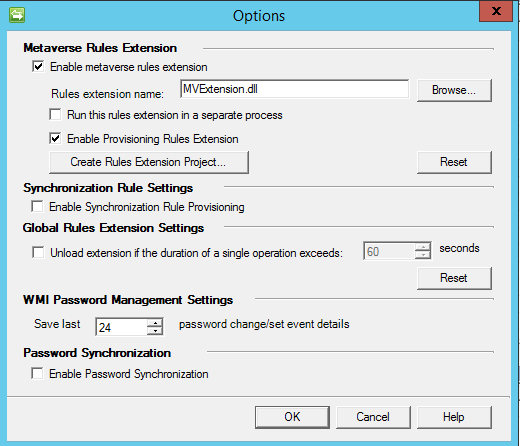
B04526\_03\_34.png

1. You can create a new ProvisionADAccount method like this:



B04526\_03\_35.png

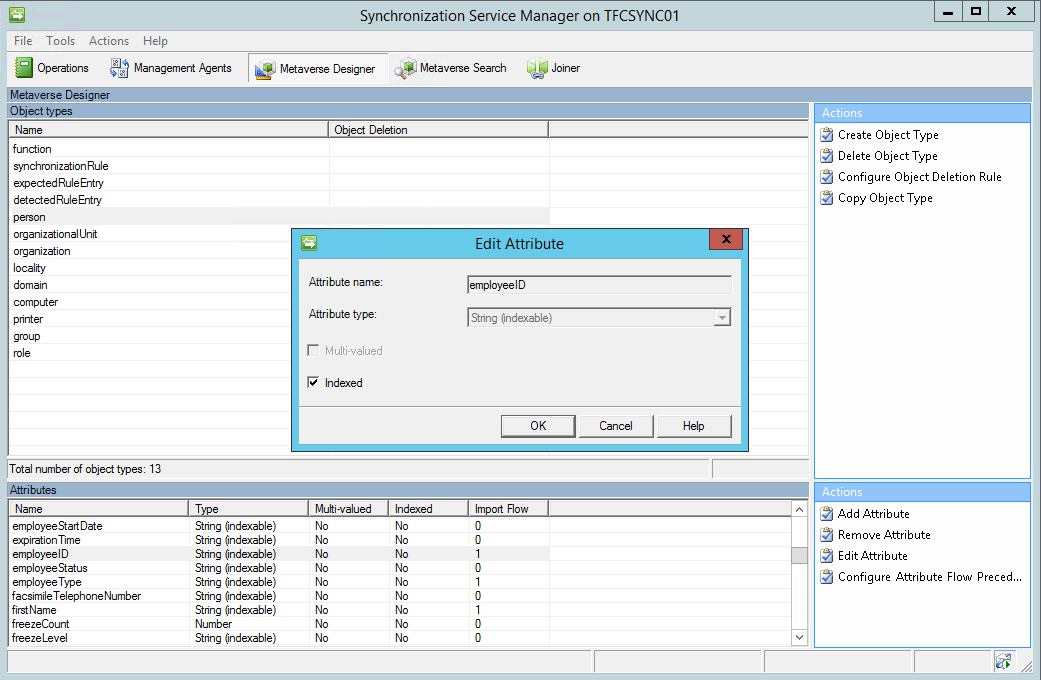
1. Build the solution, then go back to Tools | Options… and Check Enable Provision Rules Extension to allow MIM to fire the provision code you just wrote and compiled.



B04526\_03\_36.png

## Indexing Metaverse Attributes

Remember back when we created our HR management agent and created a join between the HR ID attribute and the Metaverse employeeID attribute? We received the error “You are attempting a join mapping with a non-indexed metaverse attribute. Joining with non-indexed attributes can result in performance problems.” To fix that problem, go to the **Metaverse Designer** tool, click on the person object type in the top pane and click on employeeID on the attributes or bottom pane. Next, click on Edit Attribute and check the Indexed box, as shown in the screenshot below:



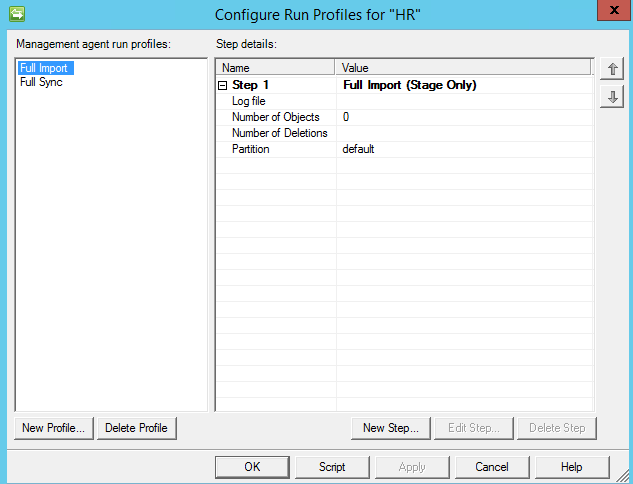
B04526\_03\_37.png

## Creating run profiles

In order for Synchronization Engine to do anything useful we need to create run profiles for each Management Agent, depending on our needs. A run profile is used for telling the MA to import, synchronize, or export the data that it has in its connector space.

In the help section of Synchronization Service Manager the concept is fully explained. In the Management Agents tool, click on the HR management agent and then click on Configure Run Profiles. Click on New Profil**e.** Then,enter Full Import, select Full Import (Stage Only), click Next, keep the default Partition, and finally click Finish.

You will need to create a full synchronization run profile called **Full Sync**. For the AD Management Agent create run profiles for Export, Full Import, Delta Import, and Delta Sync.



B04526\_03\_38.png

### Single or Multi step

When you create run profiles you have the option to use multi-step profiles.

You can, for example, create a profile that does import and synchronization, rather than having one profile doing import and then another doing synchronization. Initially, we recommend that you only use single-step profiles, as that will give you maximum control to begin with. Using only single step profiles, you will also avoid a combined-step problem.

When you configure a run profile with a single step of the type "Delta Import and Delta Synchronization", a condition can occur in which existing disconnector objects from a previous run are not processed. This condition occurs because the existing objects in the connector space that have not changed since the last run are ignored.

# Schema management

Very early on in our MIM deployment, we ran into discussions regarding the need for schema changes in MIM.

The default schema is, in almost every case, not sufficient and needs to be modified.

I will only give a short overview in this chapter about this, and will try to explain more in the coming chapters, as we look into the details of MIM implementation at The Company.

## MIM Sync versus MIM Service schema

One of the problems with the MIM Synchronization/MIM Service system is that it holds two schemas. We have one schema for the MIM Synchronization Service database and one for the MIM Service database.

Depending on our needs, we change one or both of these schemas. Whether the attributes or objects are required within MIM Service depends on whether or not they are managed using MIM Portal, or used in some policy. If not, we do not need them in the MIM Service schema.

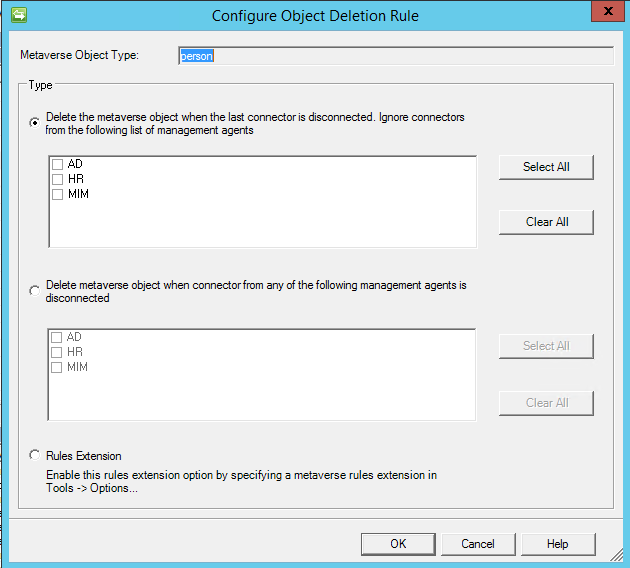
On the other hand, if an attribute or object type is used in a policy within MIM Service, but is never supposed to be synchronized to other data sources, we do not need to change the MIM Synchronization Service schema.

## Object deletion in MV

One type of schema configuration that we need to look at in our deployment is Object Deletion Rules in the MIM Synchronization Service database.

Open up the Synchronization Service Manager window, and select the Metaverse Designer tool; this is where you will configure the MV schema or, if you like, the MIM Synchronization Service database schema.

If you want to select an object type, you can select Configure Object Deletion Rule in the Action pane:



B04526\_03\_39.png

Here we can decide on what grounds the object should be deleted from the Metaverse.

The settings available in this dialog can be a bit confusing, but if you read the help section or look at http://social.technet.microsoft.com/wiki/contents/articles/understanding-deprovisioning-in-fim.aspx, you will find some explanation and ideas on when to use which method.

The default setting is that it will be deleted when the last connector is disconnected. It is vital to understand that an object cannot exist in the MV if it does not have a connector to an object in at least one connector space.

In many projects, object deletion is not meant to happen at all. The idea is that once an object is created within MIM, it should live on and just change its status. That said, every business is different and with MIM you have the flexibility of the .NET Framework to build a technology solution to meet those business requirements.

# Initial load versus scheduled runs

When we first start to import information into Synchronization Engine it is likely that information already exists in many or all of the connected systems.

We might need to create special synchronization rules just for the initial load, which are not used again unless we need to rebuild the data.

Let me give you an example. At The Financial Company, the basic idea is that users should be imported from the HR system and created in AD. But when we start, there might be existing users in AD and we would need to connect them using a Join rather than provisioning (creating) them in AD. During the initial load we would therefore turn off Provision in MIM, import users from both systems, project them into the MV, and join the users existing in both the systems.

Initial load is usually done manually; that is, we manually start the required run profiles for each MA.

If the environment is large, the initial load might take many hours due to the fact that, when we export our objects into the MIM Service database using the MIM Service MA, there might be many policies configured in the MIM Service that need to be applied for each object.

There are numerous ways of creating scheduled runs. I will show you a way that does not require any coding or third-party add-ons.

If you look at the run profile you would like to schedule, there is a Script button to create a script. It will generate a VB-script, which will start the run profile.

The task scheduler in Windows can then be used to create a schedule to run the script by using cscript runprofilevbscriptname. Just remember that the account (Network Service, for example) running the scheduled task needs to be a member of the MIMSyncOperators group, in order for it to be allowed to run the MA run profiles.

So far, we have the following requirements in our environment:

1. Import from HR
2. Synchronize the changes
3. Export to AD

Verify export to AD.

You will need to run a Full Import on the AD and MIM Management Agents to pull in the schema for those systems in order to provision user objects out to them. That is, click on the **AD** MA, click on Run, and select Full Import. Next, run a Full Import on the **MIM** MA. Now that you’ve brought in the schema to those systems, you can run the MAs in order of the data flow:

1. HR MA – Full Import
2. HR MA – Full Sync
3. AD MA – Export
4. AD MA – Delta Import
5. AD MA – Delta Sync

## Maintenance mode for production

Initially, while the MIM system is still being developed, we do not need to concern ourselves with someone working in the production environment. But later on, we need to make sure that no-one is working in the environment while we import new settings into the production servers.

One way of doing this is to put the servers into maintenance mode.

To place MIM Synchronization Service into maintenance mode, ensure that no Management Agents are running; that is, stop all schedules and make sure no MAs are running.

In order to place the MIM Service into maintenance mode, deny it access to port 5725. The steps to deny access to port 5725 are as follows:

1. Open Windows Firewall with Advanced Security. In order to do this:
2. Click Start, and type Windows Firewall with Advanced Security.
3. Once the search result appears on the Start menu, click Windows Firewall with Advanced Security.
4. In the console tree, click Inbound Rules.
5. In Inbound Rules, right-click on the Forefront Identity Manager Service (Webservice) rule, and then click Disable Rule.

In order to place MIM Portal into maintenance mode, disable MIM Portal with the following steps:

1. Open Internet Information Services (IIS) Manager, click Start, type Internet Information Services (IIS) Manager, and then click on it when the option appears on the Start menu.
2. Expand the objects in the console tree until you see SharePoint – 80.
3. Right-click SharePoint – 80, click Manage Web Site, and then click Stop.

When you are done importing the new configuration, I recommend that you do some manual testing before putting the system into production again.

### Disabling maintenance mode

No change is necessary to bring MIM Synchronization Service out of maintenance mode. If you have scheduled run profiles, you need to start the schedule again.

In order to return the MIM Service to normal operation, allow access to port 5725. The steps to allow access to port 5725 are as follows:

1. Open Windows Firewall with Advanced Security.
2. In the console tree, click Inbound Rules.
3. On the Inbound Rules page, right-click on the Forefront Identity Manager Service (Webservice) rule, and then click Enable Rule.

To return MIM Portal to normal operation, enable MIM Portal using the following steps:

1. Open Internet Information Services (IIS) Manager.
2. Navigate to SharePoint – 80.
3. Right-click on the site, click Manage Web Site, and then click Start.

# Summary

In this chapter, we have seen how The Financial Company configured their first Management Agents and prepared the MIM environment for further configuration.

One common source of error in a MIM environment is the lack of well-documented processes to make sure the development/test and production environments look the same. Learning and documenting how to move your configuration from development/test to production is vital as the configuration gets more complex.

If you take your time to make sure your basic configuration setup is satisfactory, it will save you many hours of troubleshooting later on. If you feel confident that your basic configuration is correct, moving on and making more complex configuration settings will be easier.

We are now ready to actually do something with our MIM environment. In the next chapter, we will start off by looking at how to configure the MIM Service.