Implementing Forefront Identity Manager 2010

Student Manual

Module 4: The FIM Service and Portal

Table of Contents

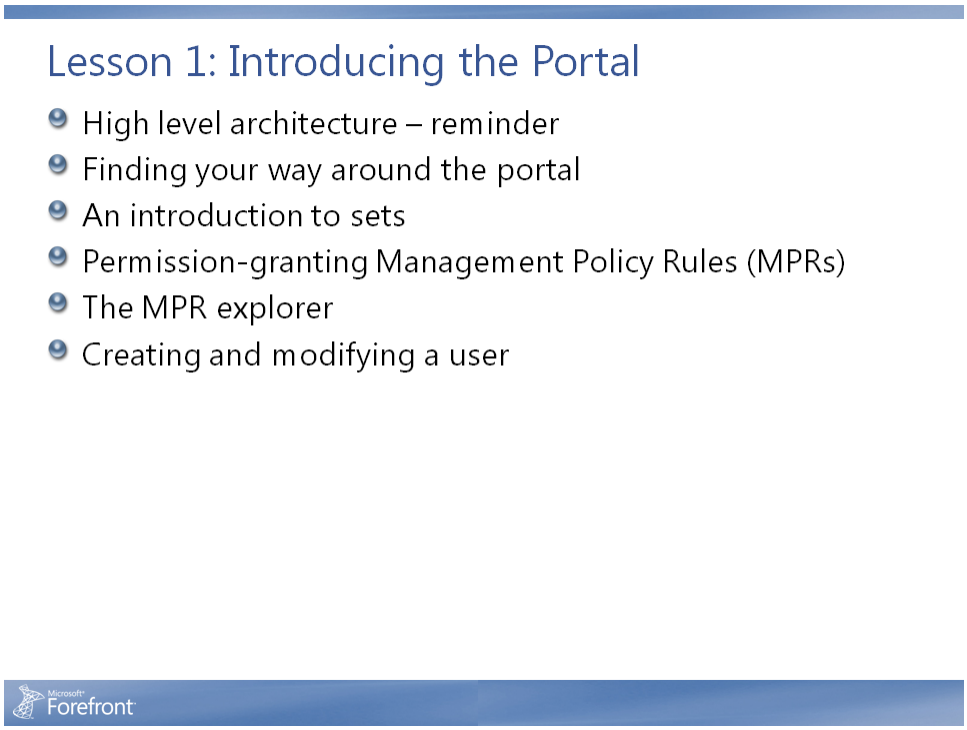
Module 4: The FIM Service and Portal

Module Overview

This module introduces the Forefront Identity Manager (FIM) Service with its associated portal and application database, initially as a standalone application, while covering the key concepts of Sets and Management Policy Rules (MPRs) through user management.

The module then looks at how you integrate the FIM Service with the FIM Synchronization Service, by using the FIM Service Management Agent (MA) to synchronize data.

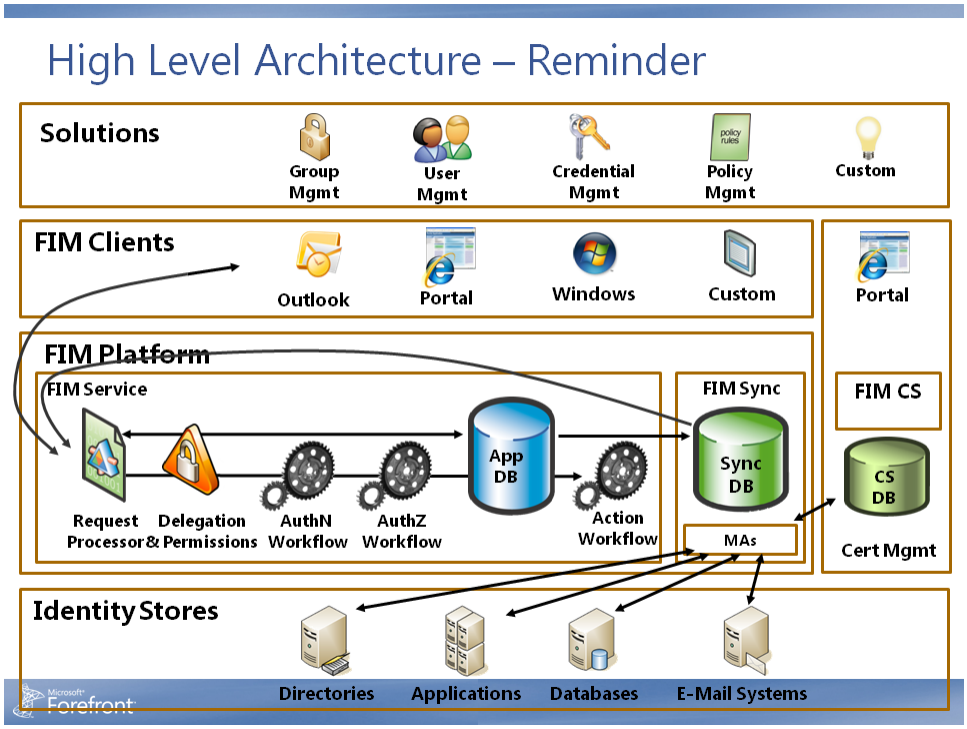
Lesson 1: Introducing the Portal



This lesson starts with a reminder about the FIM architecture, so that you can see the conceptual components of the FIM Service and the FIM Synchronization Service, and how they will fit together.

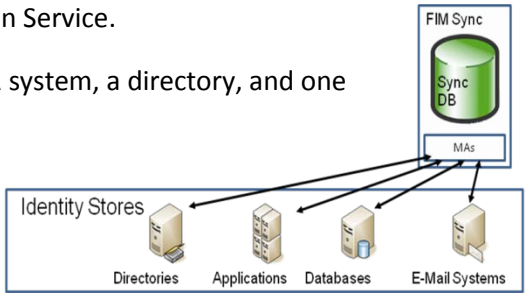
The rest of the lesson is spent on the portal—finding your way around, and understanding the key concepts, with a strong focus on the security model. The final topic concerns creating and modifying users.

High Level Architecture – Reminder



The FIM Synchronization Service

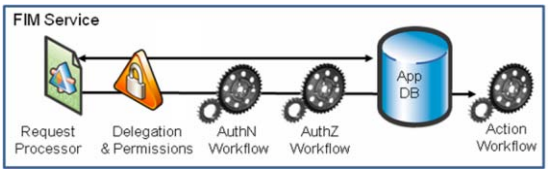
The last two modules have dealt with the FIM Synchronization Service.



The labs have connected up a number of data sources: an HR system, a directory, and one other source.

You now have a functioning Identity Management (IdM) system, albeit a very simple one. It synchronizes your identity data, but it doesn't have an interface for managing user, groups, or credentials. Nor is there anywhere from which you can manage policy in a formal manner.

The FIM Synchronization Service – Requests



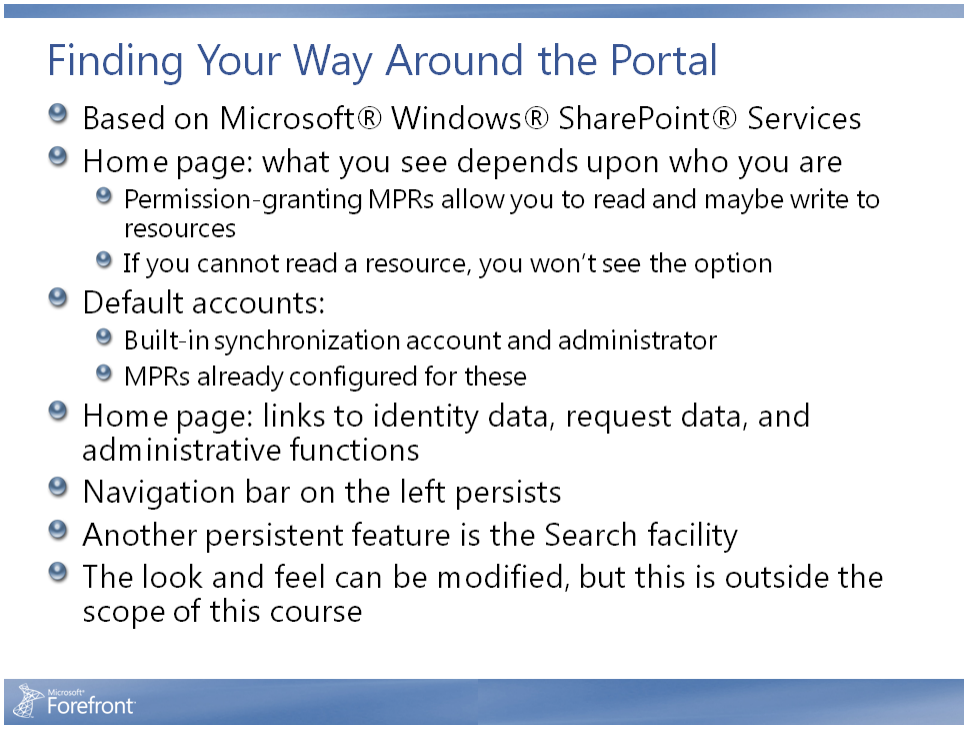
This lesson now looks at the FIM Service, and its associated portal and application database.

When you use the portal (through Windows® Internet Explorer®), any changes that you make are submitted in the form of a request to the FIM Service to read or write data to the FIM Service application database. Requests must pass through a number of checks, or gates, that happen in series. Failure at any point means that the process stops, and the modification request is rejected.

Read requests only need permission, they do not run workflows of any kind. Write requests can run workflows.

Requests are governed by Management Policy Rules (MPRs). Initially you will only look at MPRs that grant permissions (not workflows). MPRs work with sets of objects. These terms are explained further in the next few topics.

Finding Your Way Around the Portal



Based on SharePoint Services

The FIM portal is based on Microsoft® Windows® SharePoint® Services. This means that some of its built-in functionality is already familiar to an increasing number of users. Data is stored in a Microsoft® SQL Server® application database.

What you see on the home page of the portal depends upon who you are. Permission-granting MPRs allow you to read and perhaps write to resources. If you cannot read a resource, you will not see it. You will see only options (links) that are available to you.

The Default Accounts

To access the portal at all, you will have to be a user that is known to the portal. When you install FIM, two accounts are already there by default: the Built-In Synchronization Manager and the Administrator. Your first experience of the portal is likely to be as the administrator, and because MPRs have already been configured for you (for both accounts) you will see all the options.

Later the database will contain other users, and you can configure MPRs to govern what these users can do in the portal.

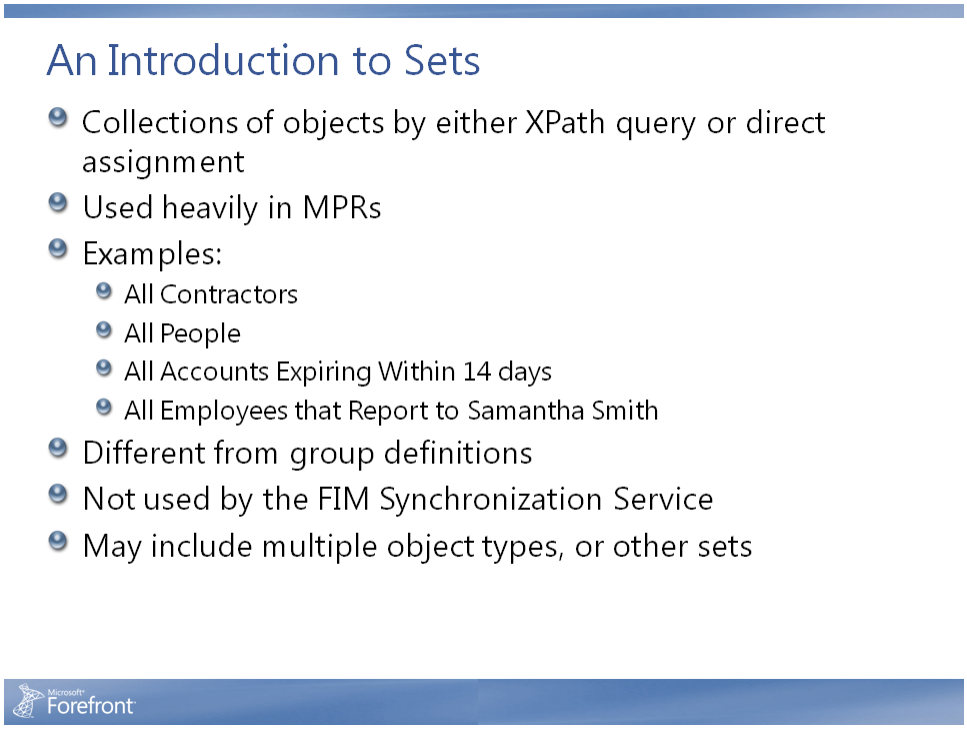
Home Page

The home page includes links to identity data (such as users and security groups), request data (for tracking requests and approvals), and administrative functions, which are listed in their own section. There is also an administrative page, which you access via the Administration link.

The navigation bar on the left persists on other pages, and becomes your primary navigation tool. The Search facility also persists, often with an associated menu bar containing options such as New and Delete. The search facility works in a straightforward manner, by default. For example, if you are on the Users page and search for **Max**, you will want to find users with display names containing a word starting with **Max**. Some pages offer a query-based advanced search, which allows sophisticated queries.

The look and feel can be modified, as can the navigation bar, home page, and search behavior; although these modifications are outside the scope of this course.

An Introduction to Sets



Portal Sets are collections of objects defined dynamically by a query, or statically by direct assignment. FIM comes with useful pre-defined sets, some of which are intended for internal use only, and others which allow you to define your own policies, such as MPRs.

MPRs Use Sets

Portal Sets are used a great deal in MPRs, in order to define who can run the MPRs (requestors), and to which object they apply (targets).

Other examples of portal sets are:

- All Contractors
- All People
- All Accounts Expiring within 14 days
- All Employees that Report to Samantha Smith

XPath

FIM makes extensive use of XML for data representation, so it is not surprising that set queries take the form of an XPath statement, since XPath is a language for selecting nodes from XML data.

You may enter simple queries, by using a query builder, which FIM then converts to an XPath statement. You can write your own XPath statements, and edit the ones created by FIM. You may find it useful to build a query and then review the XPath. You could define a set, for example, through the user interface, by using the query builder like this:

```
All user that match any of the following conditions
      EmployeeType is Contractor
      EmployeeType is Full Time Employee
```

This would be stored as an XPath query which includes

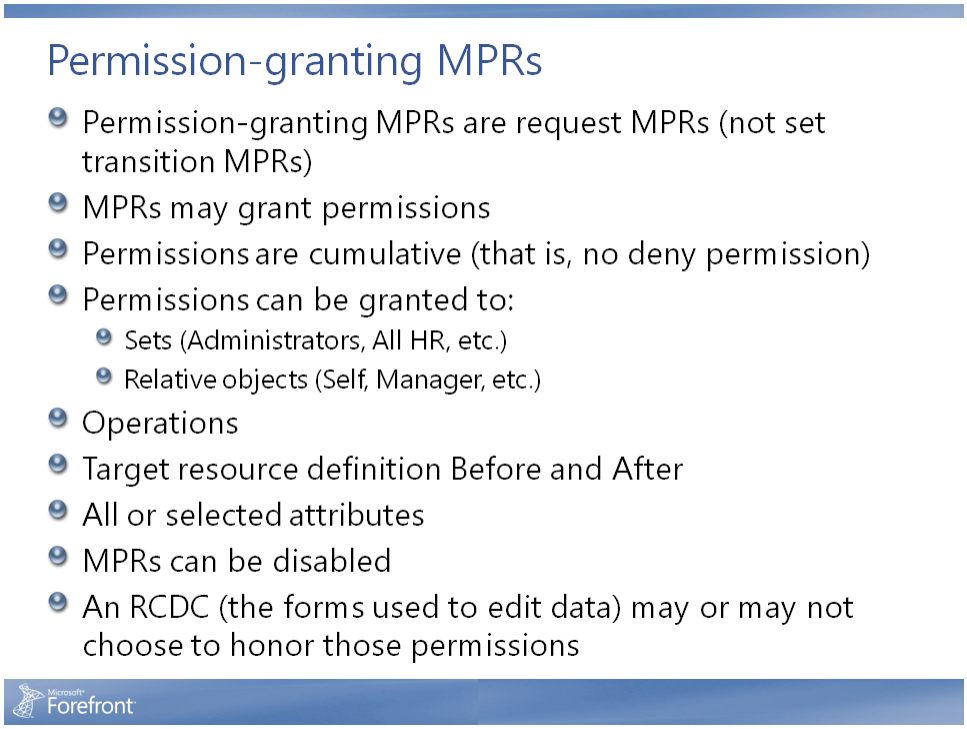
```
/Person[(EmployeeType = 'Contractor' or EmployeeType = 'Full Time Employee')]
```

Sets Are Not the Same as Groups

Sets are quite distinct from groups. Sets are intended to be used only within the portal, and, by default, they are not imported via the FIM MA (although, since they are objects, you could configure the MA to do so). Groups are provided to support corresponding group-like objects in other systems, like Active Directory®.

> **Note:** Sets may include multiple object types, or other sets. Sets may contain groups. Groups may contain users or other groups.

Permission-granting MPRs



Permission-granting MPRs Must Be Request MPRs

There are two types of MPRs:

- Request MPRs, which can grant permissions and run workflows.
- Set Transition MPRs, which can only run workflows.

For the moment you will only work with request MPRs, and you will ignore workflows, too.

Permissions Are Cumulative

There is no **deny** permission, therefore, your resultant permissions are built up from the permissions you do have. For example, you may have an MPR granting a set of users read permissions on all attributes for an object, another MPR granting create permissions on a subset, and yet another granting modify permissions on a sub-subset. When looking at permissions you have configured, you may not realize that another MPR exists that gives wider permissions. You should make permissions clear when naming MPRs, and also use the MPR Explorer (see next topic) to search for MPRs that give permissions to a set of interest—in order to ensure you haven't missed any.

Requestors Can Be Sets or Relative Objects

Permissions can be granted to requestors which are:

- Sets (Administrators, All HR, etc.).
- Relative objects (Self, Manager, etc.).

The second category is the basis of self-service and delegated permissions.

Operations

MPRs grant permissions for these operations:

- Create resource
- Read resource
- Modify resource
- Delete resource
- Add a value to a multivalued attribute
- Remove a value from a multivalued attribute

Target Resource Definition Before and After

A successful request can move a resource into or out of sets. For example:

- Creating users whose employee type is **intern** puts them in the All Interns set.
- Modifying users so that employee type is **contractor** puts them in the All Contractors set.

An MPR could be configured so as to allow creation, provided the target resource falls into a specific set, or to allow a modification such that the resource transitions between sets one way, but not the other.

> **Note**: For a read or delete action the **After** set cannot matter, while for a create action the **Before** set cannot matter.

All or Selected Attributes

You can specify which attributes of resources an MPR's permission-granting applies to. This is not only important for permission granting, but it would make the above transition example more efficient if FIM only had to worry about the Employee Type attribute changing.

MPRs Can Be Disabled

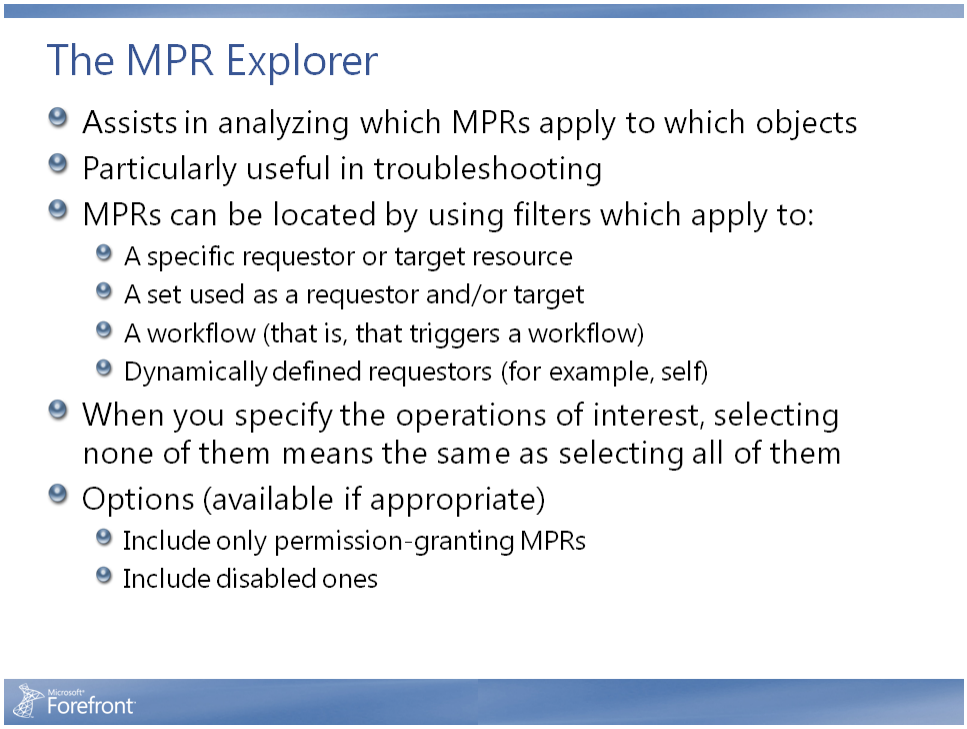
A number of MPRs are provided out-of-the-box, but many of them are disabled. You can enable them as you need. You may want to disable MPRs, for example, if:

- The MPR is not currently required, but may be required in future.
- The MPR is only required when you need to perform occasional risky admin functions, and you want to avoid accidental usage.
- You are testing to make sure that another MPR is not granting the same permission.

RCDCs May Further Restrict Access

Additionally, the forms used to present information—the Resource Control Display Configurations (RCDCs)—may or may not choose to honor those permissions. An RCDC could provide only read access, even though there is modify permission; but not vice-versa, of course.

The MPR Explorer



In the portal MPR window, the Explore button gives access to the MPR Explorer, which searches for MPRs based on the resources and sets of resources, and the operations to which they apply. This is particularly useful for troubleshooting purposes, but it can also used to verify that a configuration is secure and/or complete. For example, you might wish to know who can delete user objects, or who can modify security groups.

The MPR Explorer allows you to search for MPRs based on four different filter types.

A Requestor or Target Resource

You can specify an individual requestor and/or target resource and whichever operations are of interest. This allows you to ask questions like:

- Which MPRs allow Max Benson to modify users?
- Which MPRs allow anyone to delete Allison Brown? (This leads to “Who can delete Allison Brown?”)

When you specify the operations of interest, selecting none of them means the same as selecting all of them.

You can specify only permission-granting MPRs, and you can specify whether or not to include disabled MPRs.

A Set

A set may be part of a requestor definition, or a target resource definition, or both. For example:

- Which MPRs grant delete permissions for the Administrators set?
- Which MPRs refer to the target set Security Groups?

When you specify the operations of interest, selecting none of them means the same as selecting all of them.

You can specify only permission-granting MPRs, and you can specify whether to include disabled ones.

A Workflow

A workflow lets you find which MPRs refer to a particular workflow, with appropriate options, but these are not being covered yet.

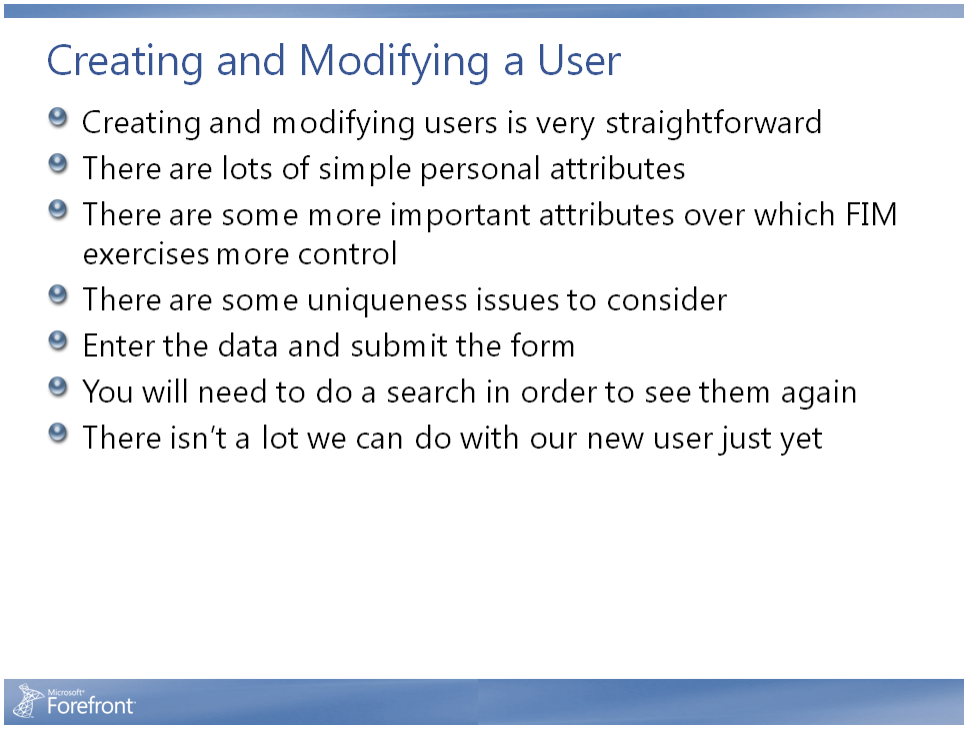
Dynamically Defined Relationships

A dynamically defined relationship allows searches where the resource in question depends on the resource being modified. For example, it would find a rule which specifies **users can modify groups they own** or **users can modify selected attributes of themselves**.

When you specify the operations of interest, selecting none of them means the same as selecting all of them.

You can specify only permission-granting MPRs, and you can specify whether to include disabled ones.

Creating and Modifying a User



Creating and modifying users is very straightforward.

Simple Personal Attributes

There are many attributes that contain simple personal identity data, most in the form of text strings (a notable exception is Photo).

More Important Attributes

There are attributes which might have a knock-on effect, such as:

- Causing a provisioning action (perhaps depending on Employee Type).
- Providing an account name in Active Directory.
- Providing an e-mail alias for Exchange Server to use.
- Choosing a department, which might affect group membership and hence permissions.
- Choosing an employee End Date, which could cause account expiry.
- Choosing a manager, which might affect group membership, or, in any case, be important.

For these attributes you are likely to want an element of control over format, or an enforced set of choices through a drop-down menu or lookup. You will find that you cannot complete a page if the e-mail alias includes a space, for example. The system could be configured for even more control by using workflows.

Uniqueness

As with your HR system, you do not at this stage have any automated method for generating unique values for Employee ID or Account Name. In a production environment, this might well be done with a workflow (which could decide on account name and e-mail alias without you actually entering them). FIM does have some built-in capability—the combination of Account Name and Domain must be unique in order for you to submit your update successfully.

Submitting Data

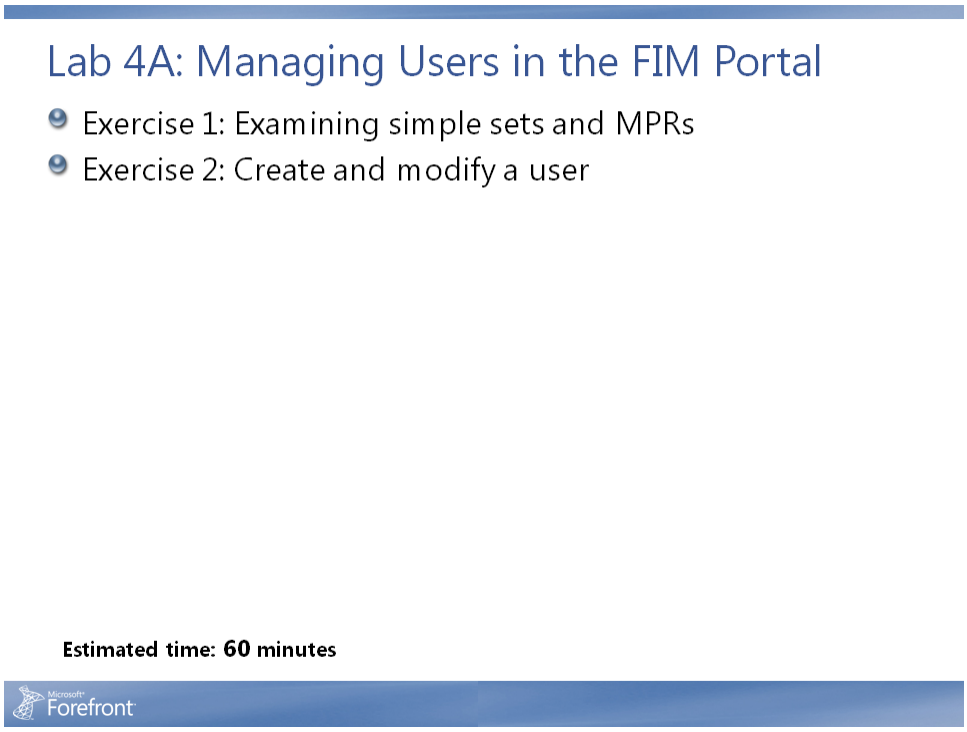
When you have entered the data, you must submit the form. A timeout occurs if you remain paused for a long time.

You will need to do a search in order to see the data again.

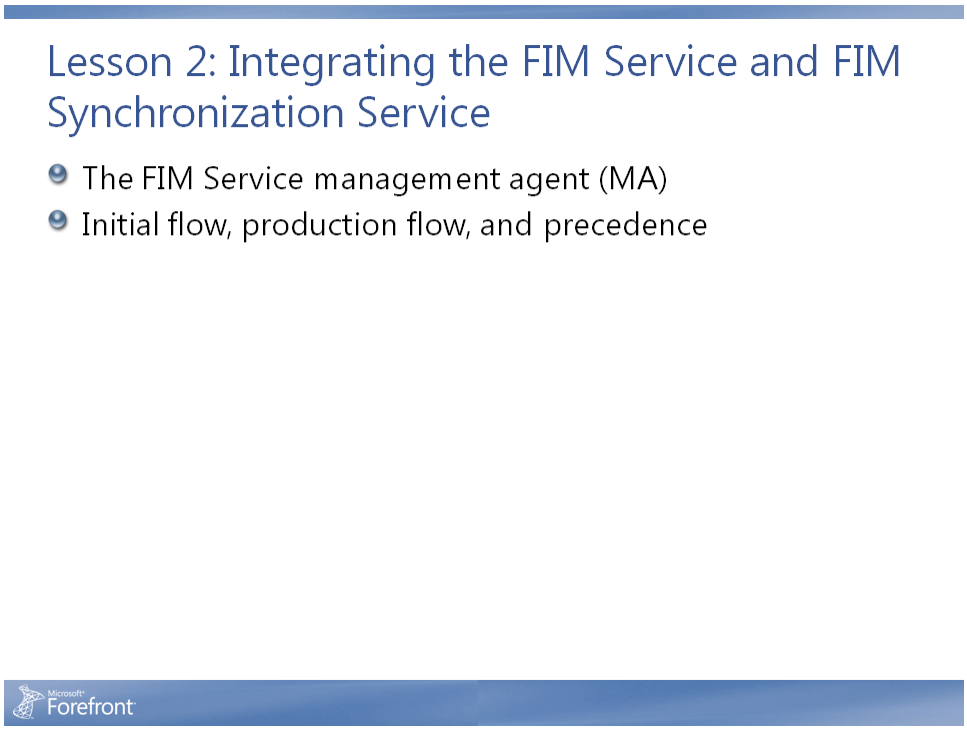
What Happens to the New User?

Not much happens to the new user at this stage. You still need to do a fair amount before the user gets an account in Active Directory, for example.

Lab 4A: Managing Users in the FIM Portal

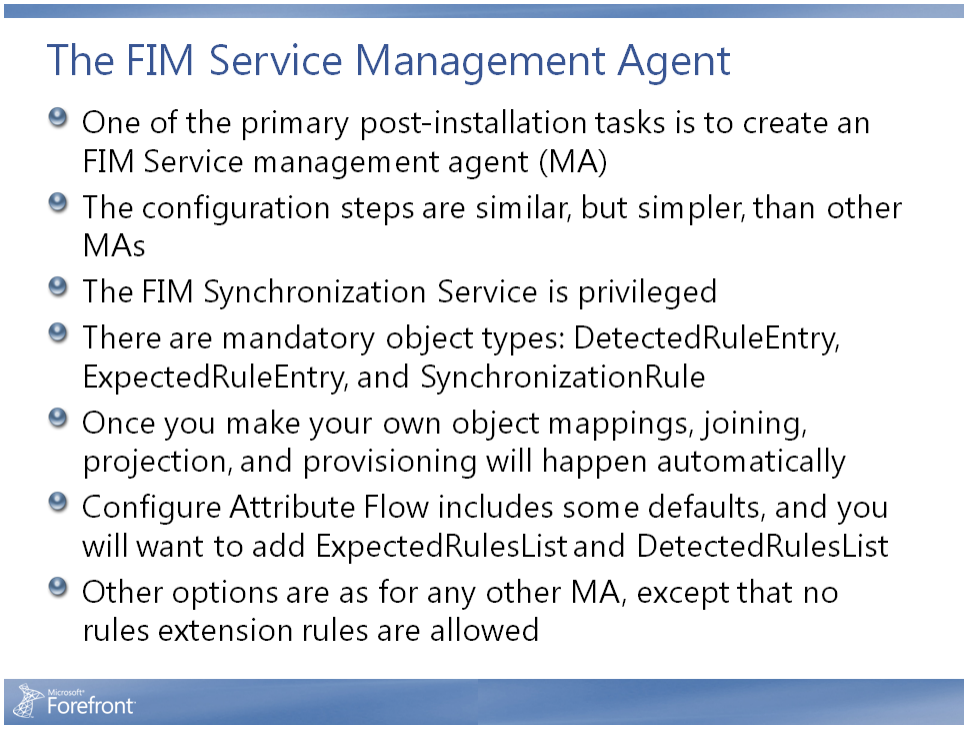


Lesson 2: Integrating the FIM Service and FIM Synchronization Service



Integrating the FIM Service and the FIM Synchronization Service is a matter of creating and configuring an FIM Service Management Agent (MA). The next task will be migrating existing data.

The FIM Service Management Agent



The FIM Synchronization Service Manager allows you to create many different types of MA, including the exceptional FIM Service MA.

The FIM Service Management Agent

In a sense, the FIM Synchronization Service sees the FIM Service as just another data source, and it has an MA type that it uses to connect to it. This MA is, however, unique. Its simple functionality is designed to shuttle objects and attributes between the FIM application database and the FIM Synchronization Service database, and not much else. In doing so, it not only integrates identity data, but it allows the FIM service to control the FIM Synchronization Service, by using some special object types.

Connecting the FIM Service with the FIM Synchronization Service

One of the primary post-installation tasks is to create an FIM Service MA because without it all you have is a sync engine on the one hand, and an attractive interface and database on the other. Once you have created and configured the MA, things start to happen.

The configuration steps are similar to those you have already seen, but they are easier because many of the options are not available.

Connect to Database

The FIM Synchronization Service, being a trusted user, is privileged. First, it is allowed to read directly from the application database. Second, although it writes via the FIM Service, it bypasses all authorization and authentication workflows—though it does still trigger action workflows.

For connection information, the FIM Synchronization Service needs a server and database for reading, and a web address for writing. It also needs credentials for a user that are specified during installation—the FIM MA Agent or Built-in Synchronization Account.

Select Object Types

Three object types are mandatory, and should be left well alone: DetectedRuleEntry, ExpectedRuleEntry, and SynchronizationRule. The exact purpose of these will become apparent in the next module.

You can select additional object types from those available. The out-of-the-box object types are Person and Group. By extending the schema, you can make additional object types available, such as contact, customer, and role.

Select Attributes

A number of attributes will already be selected, to support the mandatory object types. You will select additional attributes, to support your select object types. If you extend the schema and want to synchronize new attributes (or object types), you will need to update the schema, and select the additional attributes thus available.

Configure Connector Filter

Configuring the connector filter is identical to configuring that for any other MA.

Configure Join and Project – and Also Provision

There are no configuration options for join, project, and provision. FIM takes care of all of it.

Configure Object Type Mappings

Configuring object type mappings is the key step that makes everything happen.

You will find mappings already in place for the three mandatory object types and you must define mappings for your additional select object types. A perfectly normal mapping is Person (from the data source) to person (in the metaverse (MV)), and the same for Group to group.

Once you have defined your mappings, any persons imported from the FIM Service, for example, will be projected to the MV (unless there is already a MV object to join to—in which case it joins.) Any person projected into the MV by the HR system, for example, will be provisioned to the FIM Service.

If you already have persons in the MV, these will be provisioned on the next synchronization.

Configure Attribute Flow

Three attribute flows are already mapped for the three mandatory object mappings—do not change these. There are also three default flows for your additional mappings, which support FIM's needs, including the ability to join if anything breaks.

You will need to add flow for ExpectedRulesList and DetectedRulesList, if you plan using the portal to control synchronization of your object type. The exact purpose of these will become apparent in the next module.

The following would seem to be the minimum requirement (where you have to add the *italicized* entries):

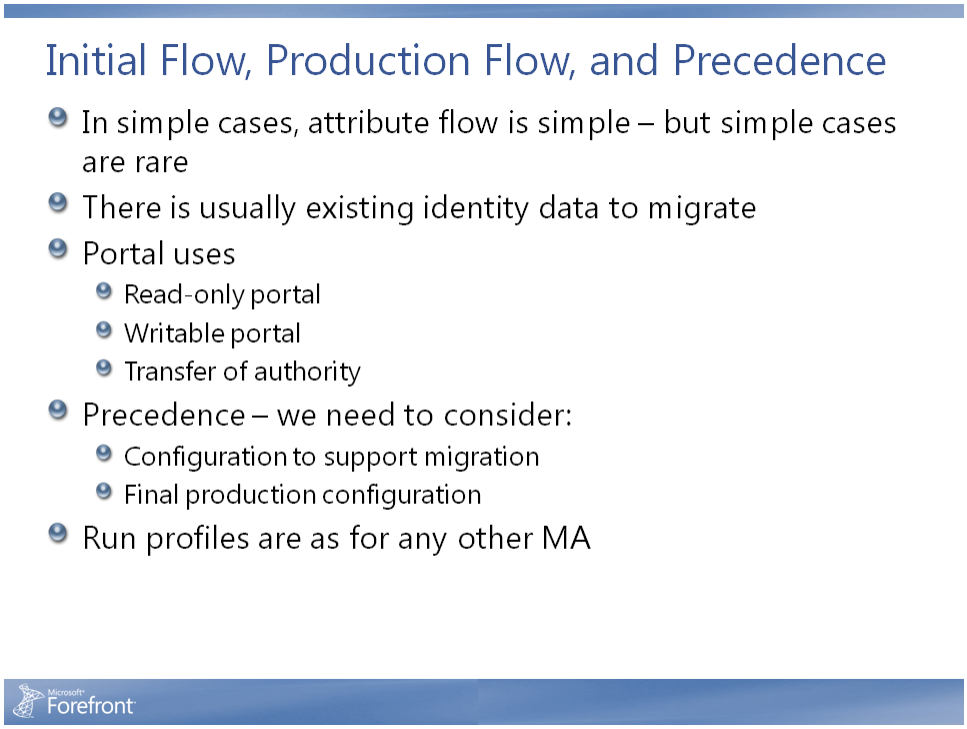
Data Source Attribute		Metaverse Attribute	Type
Dn	←		Sync-rule-mapping – expression
MVObjectID	←	<object-id>	Direct
DetectedRulesList	←	*DetectedRulesList*	*Direct*
<dn>	→	csObjectID	Direct
ExpectedRulesList	→	*ExpectedRulesList*	*Direct*

In the above table, Dn is simply an anchor GUID, and <object-id> is the MV GUID. You now add your import and export attribute flows as you would for any other MA. These will be simple, direct mappings; Advanced is not available.

Configure Deprovisioning

What you configure for deprovisioning at this stage depends on your overall strategy for deprovisioning. It is the same option as for any other MA, except that the FIM MA cannot use a rules extension.

Initial Flow, Production Flow, and Precedence



If you start by entering users in the portal and have them provisioned into other systems, there isn't much to say about attribute flow rules. You simply configure Import Attribute Flows for those attributes that are going to flow to the MV and beyond, and Export Attribute Flows for anything that you want to flow back again. Examples of the latter would be e-mail (perhaps generated by Exchange Server) and objectSID (needed if you want users to be able to log on to the portal.)

The process is not usually that simple, however.

Existing Identity Data

It is possible that identity data already exists in an HR (or similar) system. This system may have been authoritative in the past, but it is not certain what will happen in the future.

Read Only Portal

A major scenario is one in which an HR feed delivers person attributes to a read-only portal, which is then used only for administrative purposes, or used as a white pages system. In this way you can connect with useful authoritative data that you have carefully collected. The HR feed is still authoritative.

Writable Portal

A second scenario is where an HR feed delivers person attributes to a writable portal. This writable portal is used for: writeable white pages, group management, or self-service of person attributes. The HR feed is still authoritative for objects, and for some attributes.

Transfer of Authority

A final and very important scenario is that of temporary authority during initial load of data from existing sources. One or more such external systems could be authoritative while FIM is loaded with data, but following that, they transfer authority to FIM (although they may still remain connected).

Precedence

Precedence is a big issue in this situation. If there is a flow from an HR feed into, for example, displayName in the MV, and also one from the FIM service, how do you set the precedence? In the read-only scenario mentioned above, you could make it in favor of the HR Feed. In the writeable scenario you would reverse that *for some attributes*. In the last scenario you might reverse many of them. In all cases, however, you can only do this once you have migrated the existing data.

Suppose you have a lot of HR records, and, as yet, no corresponding FIM records. Suppose also that eventually you intend FIM to be authoritative for some attributes. You would have lots of export flows in the FIM Service MA to initially populate the portal database. Export flow will not flow against the precedence, however. Therefore, precedence must initially be set in favor of the HR system (or at least you should set equal precedence).

Once the portal database is populated, you can configure it as you want.

Run Profiles

The FIM Service MA is a typical MA in this respect, and it supports delta import.

Lab 4B: Creating the FIM MA and Synchronizing