Microsoft

Forefront Identity Manager 2010 Installation & Configuration

Introducing Synchronization Rules

Anthony Marsiglia & Kristopher Tackett

Microsoft Premier Field Engineering

MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

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

The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

Introducing Synchronization Rules

Prior to FIM 2010, which brought the concept of “Codeless Provisioning” (Using Sync rules) there was only 1 way to provision accounts / objects from 1 Directory to another. This was accomplished via “Custom Code” (Rules extensions) which we’re usually written with a Program such as Visual Studios which you would compile the code to build a .dll file which would then be added to the rules extension directory of the Synchronization Service. Additionally you would have to configure the Sync Engine to utilize this .dll when provisioning objects as well as every MA would require attribute flows associated with it. One negative consequence of rules extensions was the need to recompile the code whenever a change was needed to the Synchronization of attributes or resources. After recompiling the .DLL the rules extension would need to be reapplied and a Full Sync would need to be applied to commit the new or updated rules extension. With Synchronization Rules a change can be made to an individual Sync Rule and on the next import on the FIMMA you could run a preview on the change which is essentially a full sync on the individual (Sync Rule) object and from there you could continue normal run cycles. The difference is possible Minutes to commit a change as opposed to possibly hours. Synchronization rules are a bridge for data into and out of the Metaverse. Inbound Sync Rules flow data into the Metaverse and outbound Sync Rules flow data out of the Metaverse. Another option when creating Synchronization Rules is to create an Inbound/Outbound Synchronization rule that is a combined Sync Rule. In most cases I keep my Inbound and Outbound rules separate, not for any particular reason but it appears to be easier to read and discuss with management when reviewing the environment.

When creating a Sync Rule you associate the Sync Rule with an MA (Management Agent) and a resource. You can associate multiple Sync Rules to the same MA but be very careful not have multiple sync rules managing the same resources with the same attributes. If you have 2 Sync rules connected to the Active Directory Management Agent and both Sync Rules are managing User Resources, it is imperative that the same attributes are not being managed in each Sync Rule especially if 1 Sync rule is doing something different to the same attribute as another. For example if both of the Sync Rules previously discussed were outbound sync rules and the first sync rule built the displayName using (First Name Last Name) and the second Sync Rule built the displayName as (Last Name First name) you can depending on your Sync Rule Precedence which will be discussed later you could have these syncrules competing on updating the display name and each sync could potentially cause a situation where the attribute is continually being changed because of the order which the synchronization rules are being processed. Synchronization Rules always flow data into or out of the Metaverse in the direction of the associated Management Agent, for example an Inbound Synchronization rule always flows data from the Management Agent to the Metaverse. With that being said the Outbound Synchronization Rule always flows data from the Metaverse to the Management Agent. You will never create a Sync Rule that is associated with the FIMMA, this is for several reasons the first is once sync rules are created in the FIM Portal they need to exist in the Metaverse prior managing objects, if a Sync Rule was required to get data from the FIMMA to the Metaverse how would the Syncrule get into the metaverse. This FIMMA uses Direct Attribute Flows configured on the actual FIMMA (FIM Management Agent), It is also important to note that if you are using a Hybrid Sync Solution which includes Synchronization Rules and Rules Extensions and the same attribute is being managed for the same resource with each by default rules extensions always win.

In order to manage new objects they must first be projected into the Metaverse and then Provisioned into the Management Agent that the Sync Rule is associated with.

Provisioning into the FIM Portal is handled automatically once the object makes it in the Metaverse. Provisioning of objects is handled on the Outbound Sync Rule by selecting the check box Create Object in External Data source. Additionally in the Synchronization Service under options there is an Option to turn off Sync Rule Provisioning which will be discussed in detail later. On the Inbound Sync Rule you will have an option for creating objects in FIM and this must be checked in order to project new objects into the metaverse. Once objects are joined in the metaverse the sync rule will manage the objects it synchronizes regardless of the Checkbox selections.

There are 2 Types of Synchronization Rules:

Traditional Outbound Sync Rules

These Sync rules Require a Workflow that is associated with the Synchronization Rule and an MPR that when an event occurs such as a new user being created or an object transitions into a set, the MPR triggers the associated workflow which stamps an “ERE” (Expected Rule Entry) on the object which essentially tells the object which Sync rule in the metaverse will need to be applied to the object. It may help to think of the ERE is a Bus Ticket and without that ticket the object is stranded in the Metaverse or for this analogy the bus station.

Outbound Scoping Filter

Using an inclusive “Filter” to determine which objects within the Metaverse this particular sync rule will be applied to all objects that match the definition of the filter and does not require an ERE. It may help to think of this type of Sync rule as the person at the bus station that would normally collect those bus tickets but this time instead of asking the passengers for a bus ticket they allow passengers that meet a specified criteria such as all passengers wearing a blue shirt can ride this bus and if the passenger is wearing a yellow shirt they are not allowed on the bus. Because ERE’s are not needed so is the need for creating a workflow and MPR are also not required. It is important not to confuse this filter with the filter used on management agents. The management agent filter is exclusive which means that the MA (Management Agent) excludes all objects that match the criteria of the filter being applied to the MA and the Sync Rule Filter is Inclusive which means only objects that match the filter criteria are applied. Additionally the Sync rule filter can only be defined as an “AND” statement where are criteria would have to meet which is different from the MA Filter which can be configured as “AND / OR” You One of the key benefits of using this type of Sync Rule is the Traditional Sync Rule which add ERE’s to all objects in the Metaverse which in turn adds to the amount of objects that must be synchronized so if you have 100,000 users being synchronized you have 100,000 ERE’s at minimal to be associated with those objects which now doubles the amount of objects in your metaverse.

Traditional Inbound

Not like the Traditional Outbound Sync rule the Traditional Inbound only requires the Synchronization Rule to be created and does not require additional Workflows or MPR’s. By default when creating an inbound sync rule, all objects that enter the Metaverse that match the resource type of the inbound sync rule are controlled by the inbound sync rule of that resource type with the exception of applying a scoping filter which. (See Inbound Scoping Filter)

Inbound Scoping Filter

When configuring an inbound sync rule with a scoping filter only resource objects that match the filter criteria of that sync rule will be applied.