# INT0003.AD.Distribution

## Beskrivning

Denna integrations syfte är att hämta grupphändelser från kön group-ad och utifrån dessa skriva information till AD:t. Meddelandena hämtas till BizTalk från en MSMQ-kö och skrivs till AD:t med en adapter från ActiveAdapter.

## Flödesbeskrivning

Grupphändelser hämtas från kön group-ad med en receiveport. Det är två orkestreringar som är kopplade till denna receiveport. Den ena orkestreringen hämtar de händelser som gäller kurstillfällen medan den andra hämtar de händelser som gäller registreringar eller avregistreringar av personer från kurstillfällen. De båda orkestreringarna har olika filter satta för att kunna hämta de händelser som är aktuella för rätt orkestrering. Se nedan för flöden gällande kurs- respektive personhändelser.

### Kurshändelser

Orkestreringen *GroupEventsGroupToAD* lyssnar på kön och läser in meddelanden av typen *GroupCreateRequestEvent.* Orkestreringen är uppsatt enligt ett singletonmönster med korrelation på inkommande grupps namn. Detta betyder att max en händelse per gruppnamn kan hanteras åt gången. Finns det fler händelser för samma grupp så kommer de att ligga i kön tills föregående händelse är färdighanterad. Kommer det in händelser för olika grupper så kan dessa hanteras samtidigt.

Den enda typen av kurshändelse som hanteras i nuläget är *KurstillfälleTillStatusEvent*. När en sådan händelse kommer in ska tre grupper skapas i AD:t. namnen på dessa grupper är [prefix]\_Accepted, [prefix]\_Registered och [prefix]\_ Reregistered, där [prefix] utgörs av startperiod sammanslaget med utbildningstillfälleskod. Skapandet av dessa grupper sker i tre parallella flöden med varsitt anrop mot AD:t. Om något fel uppstår i något av flödena så skapas ett felmeddelande och skickas med mail.

### Personhändelser

Orkestreringen *GroupEventsIndividualToAD* lyssnar på kön och läser in meddelanden av typen *GroupMembershipCreateRequestEvent* och *GroupMembershipRemoveRequestEvent.* Orkestreringen är uppsatt enligt ett singletonmönster med korrelation på personnummer på inkommande person. Detta betyder att max en händelse per person kan hanteras åt gången. Finns det fler händelser för samma person så kommer de att ligga i kön tills föregående händelse är färdighanterad. Kommer det in händelser för olika personer så kan dessa hanteras samtidigt.

Det första som händer är att personen som händelsen gäller ska kontrolleras. Ett anrop till *CheckAndWaitForADIndividual* görs. I denna orkestrering görs ett anrop mot AD:t för att verifiera att personen finns. Om personen finns sparas dess CN-värde och returneras tillbaka med personobjektet till huvudorkestreringen. Finns personen inte i AD:t så skickas en förfrågan till AKKA om att lägga till denna person. Detta meddelande läggs på kön identity-distribution för vidare hantering innan det når AKKA. Sedan ligger BizTalk och lyssnar på AD:t efter en personändring gällande aktuell person. När personen kommit in i AD:t och BizTalk fångat upp detta event så sparas CN-värdet från personen ur AD.t till personobjektet som returneras till huvudorkestreringen.

När sedan personen är verifierad och klar och det finns ett CN-värde fortsätter huvudorkestreringen sitt flöde.

Om typen på meddelandet är *GroupMembershipRemoveRequestEvent* så anropas orkestreringen *CallSendIndividualUpdateToAD* tre gånger, en för varje grupp som användaren ska tas bort ur. Den grupp som är aktuell skickas som parameter in till orkestreringen. I orkestreringen *CallSendIndividualUpdateToAD* kontrolleras sedan vilken grupp det gäller och rätt port anropas för att ta bort personen ur gruppen.

Om typen på meddelandet är *GroupMembershipCreateRequestEvent* så kontrolleras även grupptypen på meddelandet, d.v.s. vilken typ av registrering det är fråga om. Det finns tre möjligheter i nuläget: FörväntatDeltagande, Registrering och Omregistrering. Sedan anropas orkestreringen *AddIndividualToADGroup* med olika parametrar beroende på vilka grupper som påverkas. För FörväntatDeltagande är det gruppen accepted, för Registrering är det registered och för Omregistrering är det både registered och reregistered.

I orkestreringen *AddIndividualToADGroup* görs en fråga mot AD:t för att kontrollera om gruppen finns eller inte. Finns den så anropas *SendIndividualUpdateToAD* för att lägga in personen i den gruppen. Finns den inte så stannar BizTalkflödet och ligger och lyssnar på AD:t tills gruppen kommit in. Då kan flödet fortsätta och *SendIndividualUpdateToAD* anropas för att lägga in personen i gruppen.

## Receiveportar

### INT0003.Group.Events

Typ: MSMQ  
Könamn: GROUP-AD  
Pipeline DocumentSpecNames: INT0003.AD.Distribution.Schemas.GroupEvents.InternalGroupEvent, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a

### INT0003.AD.GroupEvents

Typ: ActiveAdapter  
Receive pipeline: SupressDebugInfo  
DocumentSpecNames:   
Query mode: Changes Only  
Container: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Filter: &(objectCategory=group)(cn=\*)  
Properties to return: cn

### INT0003.AD.UserEvents

Typ: ActiveAdapter  
Receive pipeline: SupressDebugInfo  
DocumentSpecNames: INT0003.AD.Distribution.Schemas.ADEvents.ADQueryResultsChangeObjectAdded, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Query mode: Changes Only  
Container: LDAP://exchange-test.its.uu.se:389/OU=Users,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Filter: &(objectCategory=user)(cn=\*)(whenCreated>=20160401000000.0Z)  
Properties to return: norEduPersonNIN, cn

### 

## Sendportar

### INT0003.Identity.distribution.Send\_CreateUser

Typ: MSMQ  
Könamn: IDENTITY-DISTRIBUTION  
Outbound map: InternalGroupEvent\_to\_InternalPersonEvent  
Send pipeline: PassThruTransmit

### INT0003.AD.Send\_UserUpdate

Typ: ActiveAdapter  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Send pipeline: PassThruTransmit

### INT0003.AD.Send\_UserQuery

Typ: ActiveAdapter  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Send pipeline: PassThruTransmit

### INT0003.AD.Send\_UpdateReregisteredGroup

Typ: ActiveAdapter  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Send pipeline: XSLTransform  
MapName: INT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddIndividualToGroup, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupSuffix=reregistered|URI=exchange-test.its.uu.se:389|path=OU EQ Users,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)

### INT0003.AD.Send\_UpdateRegisteredGroup

Typ: ActiveAdapter  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se  
Send pipeline: XSLTransform  
MapName: INT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddIndividualToGroup, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupSuffix=registered|URI=exchange-test.its.uu.se:389|path=OU EQ Users,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)

### INT0003.AD.Send\_UpdateAcceptedGroup

Typ: ActiveAdapter  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)  
Send pipeline: XSLTransform  
MapNameINT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddIndividualToGroup, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupSuffix=accepted|URI=exchange-test.its.uu.se:389|path=OU EQ Users,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)

### INT0003.AD.Send\_TransmissionError

Typ: nulladapter  
Send pipeline: PassThruTransmit  
Filters: ErrorReport.FailureCode == 0xc0c0167a And ErrorReport.MessageType == <http://UU.Integrations.Common.Schemas.InternalGroupEvent#GroupEvent> Or ErrorReport.FailureCode == 0xc0c01680 And ErrorReport.ReceivePortName == INT0003.AD.GroupEvents Or ErrorReport.ReceivePortName == INT0003.AD.UserEvents And ErrorReport.FailureCode == 0xc0c01680 Or ErrorReport.FailureCode == 0xc0c0167a And ErrorReport.MessageType == <http://UU.Integrations.Common.Schemas.ADSendAdapterOneWayUpdate#ActiveDirectoryUpdate> Or ErrorReport.FailureCode == 0xc0c01657 And ErrorReport.ReceivePortName == INT0003.AD.GroupEvents Or ErrorReport.ReceivePortName == INT0003.AD.UserEvents And ErrorReport.FailureCode == 0xc0c01657

### INT0003.AD.Send\_SupportMail

Typ: SMTP  
Send pipeline: PassThruTransmit

### INT0003.AD.Send\_CreateADReRegisteredGroup

Typ: ActiveAdapter  
Send pipeline: XSLTransform  
MapName: INT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddGroupOutgoing, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupPrefix=\_reregistered|EqualToValue=OU EQ Courses,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)

### INT0003.AD.Send\_CreateADRegisteredGroup

Typ: ActiveAdapter  
Send pipeline: XSLTransform  
MapName: INT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddGroupOutgoing, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupPrefix=\_registered|EqualToValue=OU EQ Courses,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)

### INT0003.AD.Send\_CreateADAcceptedGroup

Typ: ActiveAdapter  
Send pipeline: XSLTransform  
MapName: INT0003.AD.Distribution.Maps.InternalGroupEvent\_to\_ADAddGroupOutgoing, INT0003.AD.Distribution, Version=1.0.0.0, Culture=neutral, PublicKeyToken=b64e3957dd28061a  
Parameters: groupPrefix=\_accepted|EqualToValue=OU EQ Courses,OU EQ Student,OU EQ Akka,DC EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se EQ exchange-test,DC EQ its,DC EQ uu,DC EQ se (Test)  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)

### INT0003.AD.Send\_CourseQuery

Typ: ActiveAdapter  
Send pipeline: XMLTransmit  
ReceivePipeline: XMLTransmit  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)

### INT0003.AD.QueryGroup

Typ: ActiveAdapter  
Send pipeline: XMLTransmit  
ReceivePipeline: XMLTransmit  
WhereToBind: LDAP://exchange-test.its.uu.se:389/OU=Courses,OU=Student,OU=Akka,DC=exchange-test,DC=its,DC=uu,DC=se (Test)