# 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 via ActiveAdapter.

## Flödesbeskrivning

Grupphändelser hämtas från kön group-ad via receiveporten INT0003.Group.Events. 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

1. Orkestreringen *GroupEventsGroupToAD* lyssnar på MSMQ kön GROUP-AD via receive porten INT0003.Group.Events 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*
2. Inkommande grupp meddelande mappas om till ett AD anrop via mappningen InternalGroupEvent\_to\_ADAddGroupOutgoing. Gruppens namn skapas utifrån startperiod och utbildningstillfälleskod samt ett suffix, suffixet anges med en statisk parametrar i sendpipelinen XSLTransform.
3. AD anropet skickas ActiveDirectory via sendporten INT0003.AD.Send\_CreateADAcceptedGroup. Namn formatet på gruppen kommer vara [startperiod]\_[utbildningstillfälleskod]\_accepted.
4. AD anropet skickas ActiveDirectory via sendporten INT0003.AD.Send\_CreateADRegisteredGroup. Namn formatet på gruppen kommer vara [startperiod]\_[utbildningstillfälleskod]\_registered.
5. AD anropet skickas ActiveDirectory via sendporten INT0003.AD.Send\_CreateADReregisteredGroup. Namn formatet på gruppen kommer vara [startperiod]\_[utbildningstillfälleskod]\_reregistered.

### Personhändelser

1. Orkestreringen *GroupEventsIndividualToAD* lyssnar på MSMQ kön GROUP-AD via receive porten INT0003.Group.Events 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.
2. Orkestreringen ProcessGroupAffiliation kallar på Orkestreringen GetAccount som via sendporten INT0003.AD.Send\_UserQuery hämtar ett användarkonto från ADt.

Om fler än ett användarkonto hittas i ADt kommer Orkestreringen GetAccount kalla på Orkestreringen GetActiveAccount.

1. GroupEvent meddelandet mappas om till PersonEvent av typen ”PersonStudentActiveAccountRequest”.
2. PersonEvent meddelandet skickas till MSMQ kön IDENTITY-DISTRIBUTION via sendporten INT0003.Identity.distribution.Send\_ActiveAccountRequest.
3. PersonEvent meddelandet hämtas från MSMQ kön IDENTITY-RESPONSE via receiveporten INT0003.Identity.Response som innehåller svaret från föregående request (4).
4. Om ett användarkonto inte finns i ADt mappas GroupEvent meddelandet om till PersonEvent av typen ”PersonStudentAffiliationCreateRequestEvent”.
5. PersonEvent meddelandet skickas till MSMQ kön IDENTITY-DISTRIBUTION via sendporten INT0003.Identity.distribution.Send\_CreateUser. *AKKA kommer att plocka upp detta meddelande och skapa en post i ADt.*
6. AD kontot skapas och händelsen hämtas via receiveporten INT0003.AD.UserEvents.
7. AD meddelandet mappas om till ett PersonEvent via mappningen ADObjectAdded\_to\_ADNewIndividual.
8. När ett konto hittats skapas ett AD anrop via mappningen InternalGroupEvent\_to\_ADAddIndividualToGroup. Gruppens namn som kontot ska kopplas till skapas utifrån startperiod och utbildningstillfälleskod samt ett suffix, suffixet anges med en statisk parametrar i sendpipelinen XSLTransform.

Mappningen ser till att anropet blir en borttagning av kontot eller koppling av kontot till gruppen beroende på händelsen är av typen "GroupMembershipRemoveRequestEvent” eller inte.

1. Om grupphändelsen är av typen "ForvantatDeltagande" skickas anropet till ADt via sendporten INT0003.AD.Send\_UpdateAcceptedGroup. Skickas från orkestreringen UpdateGroupAffiliation.
2. Om grupphändelsen är av typen "Registrering" skickas anropet till ADt via sendporten INT0003.AD.Send\_Update RegisteredGroup. Skickas från orkestreringen UpdateGroupAffiliation.
3. Om grupphändelsen är av typen "Omregistrering" skickas anropet till ADt via sendporten INT0003.AD.Send\_Update RegisteredGroup samt sendporten INT0003.AD.Send\_Update RegisteredGroup. Skickas från orkestreringen UpdateGroupAffiliation.

Om händelsen är av typen "GroupMembershipRemoveRequestEvent" skickas anropet till ADt via samtliga sendportar portar 4,5,6. Vilket ska ta bort kontot från alla grupper.

Om något fel uppstår i något av flödena så skapas ett felmeddelande och skickas via mail. Försök att koppla bort ett konto från en grupp som inte finnas ska inte skapa ett felmeddelande.

## Receiveportar

### INT0003.Identity.Response

Typ: MSMQ  
Könamn: GROUP-AD

### 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.Identity.distribution.Send\_ActiveAccountRequest

Typ: MSMQ  
Könamn: IDENTITY-DISTRIBUTION  
Outbound map: InternalGroupEvent\_to\_PersonRequest  
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)~~