



SailPoint Direct Connectors

Version 7.3

Administration and Configuration Guide

Copyright © 2018 SailPoint Technologies, Inc., All Rights Reserved.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend. All rights are reserved. No part of this document may be published, distributed, reproduced, publicly displayed, used to create derivative works, or translated to another language, without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and re-export of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or re-export outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Specially Designated Nationals (SDN) List; a party prohibited from participation in export or re-export transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

Copyright and Trademark Notices. Copyright © 2018 SailPoint Technologies, Inc. All Rights Reserved. All logos, text, content, including underlying HTML code, designs, and graphics used and/or depicted on these written materials or in this Internet web site are protected under United States and international copyright and trademark laws and treaties, and may not be used or reproduced without the prior express written permission of SailPoint Technologies, Inc.

“SailPoint Technologies & Design,” “IdentityIQ,” “IdentityNow,” “AccessIQ,” “Identity Cube,” “Managing the Business of Identity” and the SailPoint logo are registered trademarks of SailPoint Technologies, Inc. “SecurityIQ,” “SailPoint,” “Identity is Everything” and “The Power of Identity” are trademarks of SailPoint Technologies, Inc. None of the foregoing marks may be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

Revision History

The following table describes the revision history of *SailPoint Direct Connectors Administration and Configuration Guide* for version 7.3:

Version	Description
7.1	<p>Included the following important changes:</p> <ul style="list-style-type: none">• Deprecating support for Tenrox, Rally and ALES Connector• IQService: Support for Windows FIPS compliant mode• New Connectors: SCIM 2, SAP HANA, Web Services and RACF via LDAP• Workday Connector Enhancements:<ul style="list-style-type: none">- It is a Standard Deployment Connector now- Future data- Delta Aggregation- Extensible account schema including schema attributes• Administrator Permissions changes for SAP HANA Connector• Deleted the Password Interceptor appendix from the guide. For more information about Password Interceptor for Active Directory and IBM i, see https://community.sailpoint.com/docs/DOC-3257• Multiple group support in Tivoli, SunOne, and OpenLDAP LDAP Connectors• RSA Connector: Support for Extended Attributes such as mobile number• SAP HR/HCM Connector Enhancements<ul style="list-style-type: none">- support for future hire and future data- enhancement for supporting different models to detect employee's manager- Administrator permissions changes
7.1 Patch 1	<p>Includes the following important changes:</p> <ul style="list-style-type: none">• Changes in Workday Connector• Moved the following Connectors to the Integration Guide:<ul style="list-style-type: none">- Mainframe Integration Module: Agent Connectors, RACF LDAP- Enterprise Resource Planning Integration Modules: PeopleSoft, SAP Portal-User Management Web Service, Siebel, SAP, Oracle E-Business Suite and NetSuite Connector <p>The 'Component Interface' appendix related to PeopleSoft Connector also moved to Integration Guide.</p> <ul style="list-style-type: none">- Healthcare Integration Module: Epic Connector• RSA Connector: Support for RSA Authentication Manager 8.2• Azure Active Directory Connector: Support for Pass Through Authentication

Version	Description
7.2	<p>Includes the following important changes:</p> <ul style="list-style-type: none"> • Deprecating support for CyberArk Connector • Multiforest support for Active Directory • Administrator permissions update in SAP HR/HCM • Azure Active Directory: Support for Pass through Authentication • Sybase Performance Optimization • Changes in Workday Connector
7.3	<p>Includes the following important changes:</p> <ul style="list-style-type: none"> • Deprecated Connectors: AWS IAM and Microsoft Project Server New Integration Module for Amazon Web Services is available now. This is documented in the <i>SailPoint Integration Guide</i>. • Deprecated support for SharePoint Target and Lieberman Target Collector • New Connector: SharePoint Online, Okta, SuccessFactors and PeopleSoft Campus Solution module supported by PeopleSoft Direct Connector • ServiceNow Connector: <ul style="list-style-type: none"> - Kingston support - Support for Helsinki is dropped. • Active Directory Connector: Support for Simple Authentication and Security Layer (SASL) • Azure Active Directory Connector: Support for Delta Aggregation (Accounts and Groups) and Partitioning Delta Aggregation (Accounts) • Google Apps Connector: <ul style="list-style-type: none"> - Support for Role assignments, aggregation and provisioning of custom schema attributes, delta aggregation - G Suite is the new name for formerly called Google Apps. • Web Services Connector: Support for paging, OAuth2 authentication type, paging, XML based request • New platform support for various connectors • SailPoint Oracle Connector now supports managing Oracle Database hosted on Amazon Web Services Relational Database Service (AWS RDS) • SailPoint Microsoft SQL Server Connector now supports managing SQL Server hosted on Amazon Web Services Relational Database Service (AWS RDS) • PeopleSoft Connector: Support for Databases (MSSQL, Oracle, MySQL and so on)

Table of Contents

Revision History	3
Overview	1
Connector basics	1
Connector Licensing	1
Working of Connectors	1
Retryable mechanism	2
Mainframe Connectors	2
What are Direct Connectors	3
Application Types for Connectors	3
Viewing the available connectors	6
Connector selection	6
Chapter 1: SailPoint Active Directory Connector	7
Overview	7
Supported features	8
Supported Managed System	9
Pre-requisites	10
Administrator permissions	10
Configuration parameters	12
Forest Configuration	12
Domain Configuration	13
Exchange Configuration	14
Additional configuration parameters	14
Configuring searchDNs	18
Schema attributes	18
Account attributes	19
Group attributes	25
Provisioning Policy attributes	25
Active Directory Recycle Bin	30
Pre-requisites	30
Configuring Recycle Bin	30
Additional information	31
Unstructured Target Collector	31
Creating TLS communication	32
Using Strong Authentication (SASL)	33
Chapter 2: SailPoint AIX Connector	35
Overview	35
Supported features	35
Supported Managed Systems	36
Pre-requisites	36
Administrator permissions	37
Configuration parameters	38
Additional configuration parameters for SSH configuration	38
Public key authentication configuration	39
Schema attributes	39
Account attributes	39
Group attributes	45

Provisioning policy attributes	45
Account attributes	45
Group attributes	46
Additional information	47
Upgrade considerations	47
Unstructured Target Collector	47
Troubleshooting	48
Chapter 3: SailPoint Azure Active Directory Connector	51
Overview	51
Supported features	51
Pre-requisites	52
Administrator permissions	53
Configuration parameters	54
Additional configuration parameters	55
Schema attributes	56
Account attributes	56
Group attributes	57
Provisioning Policy attributes	58
Create Account Policy	58
Create Group Policy	59
Update Group Policy	59
Additional information	60
Managing licenses	60
Connector Reconfigure	60
Excluding Provisioning Policy attributes	60
Chapter 4: SailPoint BMC Remedy Connector	61
Overview	61
Supported features	61
Supported Managed Systems	62
Pre-requisites	62
Administrator permission	62
Configuration parameters	62
Schema attributes	62
Account attributes	62
Group attributes	63
Provisioning policy attributes	64
Create account attributes	64
Create group attributes	64
Update policies	64
Additional information	64
Enable/Disable Account	65
Troubleshooting	65
Chapter 5: SailPoint BMC Remedy IT Service Management Suite Connector	67
Overview	67
Supported features	67
Supported Managed Systems	68
Pre-requisites	68
Administrator permission	68
Configuration parameters	68
Schema attributes	69

Account attributes	69
Group attributes	70
Provisioning policy attributes	70
Create account attributes	70
Create group attributes	71
Update policies	72
Additional information	72
Enable/Disable Account	72
Add Entitlement operation for ITSM	72
Troubleshooting	73
Chapter 6: SailPoint Box Connector	75
Overview	75
Supported features	75
Supported Managed Systems	76
Pre-requisites	76
Administrator permissions	76
Configuration parameter	76
Schema attributes	77
Account attributes	77
Group attributes	78
Provisioning Policy attributes	78
Additional information	79
Upgrade considerations	79
Troubleshooting	79
Chapter 7: SailPoint DB2 Windows Connector	81
Overview	81
Supported features	82
Supported Managed Systems	82
Pre-requisites	82
Administrator permissions	82
Configuration parameters	83
Schema Attributes	83
Account attributes	83
Roles attributes	84
Provisioning Policy attributes	85
Additional information	85
Create user	86
Delete user	86
Delete Role	86
Troubleshooting	86
Chapter 8: SailPoint Delimited File Connector	89
Overview	89
Configuration parameters	89
Schema attributes	91
Chapter 9: SailPoint Dropbox Connector	93
Overview	93
Supported features	93
Supported Managed System	94
Pre-requisites	94

Administrator permissions	94
Configuration parameters	94
Additional configuration parameter	94
Schema attributes	95
Account attributes	95
Group attributes	95
Provisioning Policy attributes	96
Account attributes	96
Delete Provisioning Policy	96
Chapter 10: SailPoint Duo Connector	99
Overview	99
Supported features	99
Pre-requisites	99
Administrator permissions	100
Configuration parameters	100
Schema Attributes	101
Account attributes	101
Group attributes	101
Provisioning Policy attributes	102
Account attributes	102
Additional information	103
Upgrade considerations	103
Behavioral changes	104
Troubleshooting	104
Chapter 11: SailPoint GE Centricity Connector	105
Overview	105
Supported features	105
Prerequisites	106
Administrator permissions	106
Configuration parameters	106
Additional configuration parameters	106
Schema attributes	107
Account attributes	107
Group attributes	108
Provisioning Policy attributes	108
Troubleshooting	109
Chapter 12: SailPoint Google Apps (G Suite) Connector	111
Overview	111
Supported features	111
Pre-requisites	112
Administrator permissions	113
Configuration parameters	113
Additional configuration parameters	113
Schema attributes	114
Account attributes	114
Group attributes	116
Role attributes	119
Provisioning Policy attributes	119
Additional information	126

Performance improvement	126
Managing Custom Schema attributes	127
Complex Provisioning Policy attributes	127
Troubleshooting	128
Chapter 13: SailPoint GoToMeeting Connector	131
Overview	131
Supported features	131
Pre-requisites	131
Administrator permissions	132
Configuration parameter	132
Schema attributes	132
Account attributes	132
Group attributes	132
Provisioning Policy attributes	133
Chapter 14: SailPoint IBM i Connector	135
Overview	135
Supported features	135
Supported Managed Systems	136
Pre-requisites	136
Administrator permissions	137
Configuration parameters	137
Schema Attributes	138
Account and Account - Group attributes	138
Provisioning policy attributes	140
Additional information	140
Direct Permissions	140
Upgrade Consideration	141
Create TLS 1.2 Communication between IdentityIQ and IBM i system	141
Aggregation of Group Profile as part of Account Aggregation	142
Chapter 15: SailPoint JDBC Connector	143
Overview	143
Supported features	143
Supported Managed Systems	144
Pre-requisites	144
Administrator permissions	144
Configuration parameters	144
Additional configuration parameters	147
Schema Attributes	147
Troubleshooting	148
Chapter 16: SailPoint Jive Connector	151
Overview	151
Supported features	151
Pre-requisites	152
Administrator permission	152
Configuration parameters	152
Schema attributes	152
Account attributes	152
Group attributes	154
Provisioning Policy attributes	154

Create account attributes	154
Create group attributes	155
Additional information	155
Configuration settings	155
Upgrade considerations	156
Troubleshooting	156
Chapter 17: SailPoint LDAP Connector	157
Overview	157
Supported features	157
Supported Managed Systems	158
Pre-requisites	160
Administrator permissions	160
Configuration parameters	160
Additional configuration parameter	161
Configuring Account Search Scope	161
Configuring Group Search Scope	163
Schema attributes	163
Account attributes	164
Group attributes	167
posixgroup and nisNetgroup Attributes	168
Group Membership attribute	169
Group Entitlement attribute	169
Provisioning Policy attributes	170
Configuring group provisioning policy for new group	171
Additional information	171
Adding additional group types	171
Using Novell eDirectory as a Pass-through Authentication Source	173
Troubleshooting	173
Chapter 18: SailPoint LDIF Connector	175
Overview	175
Configuration parameters	175
Schema Attributes	176
Account attributes	176
Group attributes	179
Chapter 19: SailPoint Logical Connector	181
Overview	181
Configuration parameters	181
Schema attributes	181
Additional information	182
Logical Connector - Tiers Tab	182
Defining Logical Connectors	184
Logical Application Filtering	184
Chapter 20: SailPoint Lotus Domino Connector	187
Overview	187
Supported features	187
Supported Managed Systems	188
Pre-requisites	188
Administrator permissions	189
Configuration parameters	189

Additional configuration parameters	190
Schema attributes	190
Account attributes	190
Group attributes	193
Provisioning policy attributes	193
Create account attributes	193
Create group attributes	196
Update policies	196
Additional information	198
ID Vault functionalities	198
Password management	198
Troubleshooting	199

Chapter 21: SailPoint Linux Connector 201

Overview	201
Supported features	201
Supported Managed Systems	202
Pre-requisites	202
Administrator permissions	203
Configuration parameters	204
Additional configuration parameters for SSH configuration	204
Public key authentication configuration	205
Schema attributes	205
Account attributes	206
Group attributes	206
Provisioning policy attributes	206
Account attributes	207
Group attributes	207
Additional information	208
Unstructured Target Collector	208
Troubleshooting	209

Chapter 22: SailPoint Mainframe Connector 215

Overview	215
Configuration parameters	215
Schema attributes	216
Account attributes	216

Chapter 23: SailPoint Microsoft SQL Server 217

Overview	217
Supported features	217
Supported Managed Systems	218
Pre-requisites	218
Administrator permissions	218
Configuration parameters	220
Schema attributes	221
Account attributes	221
Group attributes	222
Provisioning Policy attributes	222
Additional information	223
Upgrade considerations	223
Direct permission	224
Amazon Web Services Relational Database Service (AWS RDS)	224

Troubleshooting	225
Chapter 24: SailPoint Microsoft SharePoint Server Connector	227
Overview	227
Supported features	227
Supported Managed system	228
Pre-requisites	228
Application Account permissions	229
Configuration parameters	229
Additional configuration parameter	230
Schema attributes	230
Account attributes	231
Group attributes	231
Provisioning Policy attributes	232
Additional information	232
Certifications	232
Performance improvement	232
Troubleshooting	233
Chapter 25: SailPoint Microsoft SharePoint Online Connector	235
Overview	235
Supported features	235
Prerequisites	236
Administrator permissions	236
Configuration parameters	236
Schema attributes	237
Account attributes	237
Group attributes	238
Provisioning Policy attributes	238
Chapter 26: SailPoint Okta Connector	241
Overview	241
Supported features	241
Pre-requisite	242
Configuration parameters	242
Schema attributes	243
Account attributes	243
Group attributes	244
Application attributes	245
Provisioning Policy attributes	245
Create Account Policy	245
Create Group Policy	246
Disable Policy	246
Additional information	247
Supported Aggregation Filters	247
Account Status Mapping	248
Delta Aggregation	248
Troubleshooting	249
Chapter 27: SailPoint Oracle Connector	251
Overview	251
Supported features	251
Supported Managed Systems	252

Pre-requisites	252
Administrator permissions	252
Configuration parameters	254
Additional configuration parameter	255
Schema attributes	255
Account attributes	255
Group attributes	256
Provisioning policy attributes	257
Additional information	257
Amazon Web Services Relational Database Service (AWS RDS)	257
Troubleshooting	258
Chapter 28: SailPoint Oracle HRMS Connector	259
Overview	259
Supported features	259
Supported Managed Systems	259
Pre-requisites	259
Administrator permissions	260
Configuration parameters	261
Schema attributes	262
Account attributes	262
Chapter 29: SailPoint PeopleSoft HCM Database Connector	265
Overview	265
Supported features	265
Supported Managed Systems	266
Pre-requisites	266
Administrator permission	266
Configuration parameters	267
Schema attributes	269
Account attributes	269
Additional information	270
Configuring Component Interface Security	270
Creating PeopleSoft HRMS Jar File	271
Troubleshooting	272
Chapter 30: SailPoint RACF Connector	275
Overview	275
Supported features	275
Configuration parameters	275
Schema Attributes	277
Account attributes	277
Group attributes	281
Chapter 31: SailPoint Remedyforce Connector	283
Overview	283
Supported features	283
Configuration parameters	284
Additional configuration parameters	285
Schema attributes	285
Account attributes	286
Profile attributes	288
Provisioning Policy attributes	288

Troubleshooting	289
Chapter 32: SailPoint RSA Authentication Manager Connector	291
Overview	291
Supported features	292
Supported Managed Systems	292
Pre-requisites	292
Administrator permissions	293
RSA Token PIN Reset	294
Configuration parameters	294
Schema attributes	295
Account attributes	295
Group attributes	296
Provisioning Policy attributes	296
Additional information	297
Active Directory configured as an identity source	297
Chapter 33: SailPoint Salesforce Connector	299
Overview	299
Supported features	300
Administrator permissions	300
Pre-requisites	301
Configuration parameters	301
Additional configuration parameters	302
Schema attributes	303
Account attributes	303
Role attributes	305
Group attributes	306
Profile attributes	306
Provisioning Policy attributes	307
Additional information	307
Upgrade Considerations	307
Query filter for PermissionSet	308
Troubleshooting	309
Chapter 34: SailPoint SAP HR/HCM Connector	313
Overview	313
Supported features	313
Supported Managed Systems	314
Pre-requisites	314
Administrator permissions	314
Configuration parameters	317
Schema Attributes	319
Account attributes	319
Additional information	325
Upgraded Application	325
Support for Custom BAPI Invocation	326
Input and Output parameters for BAPI	327
Troubleshooting	330
Chapter 35: SailPoint SAP HANA Connector	333
Overview	333
Supported features	333

Supported Managed Systems	334
Pre-requisites	334
Administrator permissions	334
Configuration parameters	335
Additional Configuration parameters	336
Schema Attributes	337
Account attributes	337
Group attributes	338
Provisioning Policy attributes	338
Create account attributes	338
Additional information	339
Enabling SSL connection to SAP HANA database through IdentityIQ	339
Delete provision request	339
Troubleshooting	340
Chapter 36: SailPoint ServiceNow Connector	341
Overview	341
Supported features	341
Supported Managed System	342
Pre-requisites	342
User permissions	343
Configuration parameters	343
Schema attributes	344
Account attributes	345
Group attributes	346
Role attributes	347
Provisioning Policy attributes	348
Additional information	349
Upgrade	349
Session management	349
Custom attributes	350
Troubleshooting	351
Chapter 37: SailPoint Solaris Connector	355
Overview	355
Supported features	355
Supported Managed Systems	356
Pre-requisites	356
Administrator permissions	357
Configuration parameters	358
Additional configuration parameters for SSH configuration	358
Public key authentication configuration	359
Schema attributes	359
Account attributes	359
Group attributes	361
Provisioning policy attributes	361
Account attributes	361
Group attributes	362
Additional information	363
Unstructured Target Collector	363
Troubleshooting	364

Chapter 38: SailPoint SuccessFactors Connector	369
Overview	369
Supported features	369
Pre-requisites	370
Administrator permissions	371
Configuration parameters	371
Additional configuration parameter	373
Schema attributes	374
Account attributes	374
Additional Information	376
Support for Additional Parameters	376
Troubleshooting	377
Chapter 39: SailPoint SQL Loader Connector	379
Overview	379
Supported features	379
Supported Managed Systems	380
Administrator permissions	380
Configuration parameters	380
Schema Attributes	382
Troubleshooting	382
Chapter 40: SailPoint System for Cross-Domain Identity Management Connector 2.0	385
Overview	385
Supported features	385
Administrator permissions	386
Supported Managed System	386
Configuration parameters	386
Additional configuration parameter	387
Schema attributes	387
Provisioning Policy attributes	388
Update account attributes	388
Provisioning of extended attributes	388
Additional information	389
Upgrade considerations	389
Troubleshooting	389
Chapter 41: SailPoint System for Cross-Domain Identity Management Connector	391
Overview	391
Supported features	391
Administrator permissions	392
Configuration parameters	392
Additional configuration parameters	393
Schema attributes	396
Account attributes	396
Group attributes	400
Provisioning Policy attributes	401
Create account attributes	401
Update group attributes	401
Additional information	401
Upgrade considerations	402

Troubleshooting	402
Chapter 42: SailPoint Sybase Connector	405
Overview	405
Supported features	406
Supported Managed Systems	406
Pre-requisites	406
Administrator permissions	407
Configuration parameters	408
Schema attributes	409
Account attributes	409
Group attributes	410
Provisioning policy attributes	410
Additional information	411
Identity and Entitlement representation	411
Performance Optimization	412
Troubleshooting	412
Chapter 43: SailPoint Tivoli Access Manager Connector	415
Overview	415
Supported features	415
Supported Managed System	416
Pre-requisites	416
Configuration parameters	418
Schema attributes	418
Account attributes	418
Group attributes	419
Provisioning Policy attributes	419
Create account attributes	419
Create group attributes	420
Additional information	420
Unstructured Target Collector	420
Troubleshooting	421
Chapter 44: SailPoint Top Secret Connector	423
Overview	423
Supported features	423
Configuration parameters	424
Schema Attributes	425
Chapter 45: SailPoint UNIX Connector	437
Overview	437
Supported features	437
Configuration parameters	437
Schema attributes	438
Account attributes	438
Group attributes	438
Chapter 46: SailPoint Web Services Connector	441
Overview	441
Supported features	441
Supported Managed Systems	442
Pre-requisites	442
Administrator permissions	442

Configuration parameters	442
(General Settings) Basic configuration parameters	442
(Connector Operations) Operation specific configuration parameters	446
Schema attributes	451
Additional information	451
Upgrade considerations	451
Web Services Class used in Before/After operation Rule	452
Use of Quotes	458
Pagination	459
Configuration for Response	464
Configuration for Multiple endpoints	466
Configuring Multiple Entitlement Requests	467
Other Operations	467
Chapter 47: SailPoint WebEx Connector	475
Overview	475
Supported features	475
Pre-requisites	475
Administrator permissions	476
Configuration parameters	476
Schema attributes	476
Account attributes	476
Group attributes	478
Provisioning Policy attributes	479
Chapter 48: SailPoint Windows Local Connector	481
Overview	481
Supported features	481
Supported Managed Systems	482
Pre-requisites	482
Administrator permissions	482
Configuration parameters	482
Additional configuration parameters	483
Schema attributes	483
Account attributes	483
Group attributes	484
Provisioning Policy attributes	485
Install and register IQService	485
Additional information	486
Unstructured Target Collector	486
Troubleshooting	487
Chapter 49: SailPoint Workday Connector	491
Overview	491
Supported features	491
Prerequisites	492
Administrator permissions	492
Configuration parameters	495
Additional configuration parameter	496
Configuration to fetch the Custom attributes in Workday	501
Configuration to update Custom attributes in Workday	503
Schema attributes	504
Account attributes	504

Additional information	510
Future dated Workers	511
Fetch Workers by Organization type	512
Upgrade considerations	512
Troubleshooting	516
Chapter 50: SailPoint XML Connector	523
Overview	523
Supported features	523
Configuration parameters	523
Additional information	524
1 - Using XML Schema Definition (XSD)	524
2 - Using Document Type Definition (DTD)	525
Chapter 51: SailPoint Yammer Connector	527
Overview	527
Supported features	527
Pre-requisites	527
Administrator permissions	527
Configuration parameter	528
Schema attributes	528
Account attributes	528
Group attributes	529
Appendix	531
Appendix A: Delta Aggregation	533
Overview	533
Delta aggregation for Active Directory	533
Pre-requisite	533
Configuring server for Delta Aggregation	534
Testing Delta Aggregation	534
Delta aggregation for Azure Active Directory	535
Delta aggregation for ADAM, SunOne and Tivoli	535
Pre-requisite	535
Configuring server for Delta Aggregation	536
Testing Delta Aggregation	536
Delta aggregation for JDBC	537
Delta aggregation for Lotus Domino	538
Pre-requisites	538
Delta aggregation for Google Apps	538
Pre-requisites	538
Appendix B: Partitioning Aggregation	539
Overview	539
Partitioning Aggregation for JDBC Connector	540
Partitioning Aggregation for Active Directory Connector	540
Configuring partitions manually	540
Partitioning Aggregation LDAP Connector	541
Partitioning Aggregation for Delimited Connector	542
Partitioning Aggregation for IBM i Connector	543
Partitioning Aggregation for GoogleApps	543
Partitioning Aggregation for Tivoli Access Manager	544
Partitioning Aggregation for Azure Active Directory Connector	545

Appendix C: Before and After Provisioning Action	547
Overview	547
Before and After Provisioning Action for AIX/Linux/Solaris Connectors	547
Pre-requisite	547
Creating Before and After Provisioning Action	547
Before and After Provisioning Action for IBM i Connector	549
Pre-requisites	549
Creating CL scripts	549
Appendix D: IQService.....	553
Install and register the IQService for Windows	553
Installing and registering IQService	555
Upgrading IQService	557
IQService Public Key Exchange Task	557
IQService Before/After Scripts	558
Writing a script	558
Creating a Rule	560

Overview

The following topics are discussed:

Connector basics	1
Retryable mechanism	2
What are Direct Connectors	3
Application Types for Connectors.....	3
Viewing the available connectors.....	6

Connectivity is critical to successful IAM deployments. SailPoint is committed to providing design, configuration, troubleshooting and best practice information to deploy and maintain connectivity to target systems. SailPoint has modified the structure of this document to aid customers and partner deployments. The focus of this document is product configuration and integration. For more details on design, troubleshooting and deployment best practices, refer to the Connector and Integration Deployment Center in Compass.

This document describes the different types of connectors available for integration with external applications.

Connector basics

There are several different types of connectors. Connectors are commonly grouped by the ways in which they can communicate. There are:

- read-only connectors that can only communicate data from an external application (Governance)
- read-write connectors that can read data from external applications and write data out to them (Gateway and Direct)

Connector Licensing

Customers who licensed IdentityIQ on or after July 15, 2013 are entitled to all IdentityIQ connectors for aggregation, provisioning and password management use cases.

Customers who licensed IdentityIQ before July 15, 2013 are entitled to use connectors for reading data (aggregation). However, entitlement to provisioning and password management functionality through the IdentityIQ connectors (whether Direct, Gateway or Agent based) requires the purchase of the SailPoint Provisioning Engine.

Working of Connectors

This section describes how the connectors work. The Direct Connectors are of the following types:

- **Read/Write Connectors:** These connectors have read and write capabilities on external application and allow data to send in both directions.
- **Read only Connectors:** These connectors are very simple in design, they make a direct read-only connection to the external application through the connection parameters specified on the Application Definition.

Retryable mechanism

Account rename operation

IdentityIQ does not fully support processing requests to Move/Rename accounts or groups on native system. Connectors supporting Move/Rename accounts via attributes update require customization in IdentityIQ to initiate such requests and update the native identity of the link once the request is successfully processed by the connector.

The Account Move/Rename requests must not be merged with any other requests/updates.

Group provisioning

To enable group provisioning for existing application after upgrading to version 7.3, perform the following:

- Add **GROUP_PROVISIONING** to the featureString from debug page
- Define **CreateGroup** and **EditGroup** provisioning policies.

For more information on the provisioning policies defined for the connector from `connectorRegistry.xml`, see the “Provisioning Policy” section of the respective connectors for the required attributes.

Retryable mechanism

For availing the advantage of some of the logic around retryable situations, add the retryable error messages list to the attributes map on an application. The **retryableErrors** entry is a list of strings through which the connector searches when it receives a message from the managed application. If one of the strings in the entry exists in the error, the connector attempts to retry the connection. When the configured error string is not a part of the error message returned from the connector, then IdentityIQ would not attempt a retry.

Here is an example of this entry:

```
<entry key="retryableErrors">
  <value>
    <List>
      <String>Server is not operational</String>
    </List>
  </value>
</entry>
```

Precaution: Avoid using error messages which contain a date/time, sequence id, SM packets/messages, and so on, as these are very specific. Error codes or error message substrings would be good candidates for inclusion.

Mainframe Connectors

The Mainframe connectors (agent-based connectors) now retries every account provisioning operation even if a test connection is not performed or it fails and if the error message is configured in retryable error list in application debug page.

Note: **Mainframe Connectors would attempt retry connection for account provisioning even if the following type of error messages that appear are not configured in retryableErrors list:**

- **TimeoutException**
- **ConnectException**
- **NoRouteToHostException**

What are Direct Connectors

- **Simple to configure and use:** Direct connectors are simple to configure, few configuration details required to use the connector and no extra steps to deploy agents on the end managed systems.
- **Less moving parts:** Direct connectors do not require Connector Gateway (CG), Connector Manager (CM), Provisioning Modules (PM) to be deployed to get the setup done. Installing and (Re)configuring each component is not required. Data caching and sequencing on transactions not required.
- **Increased performance:** The performance of direct connectors is improved compared to the old FULL or Gateway based connectors. It is recommended to move to direct connector to get maximum benefit of per transactions.
- **No single point of failure:** Earlier if one component failed in the connector model, then it required re-cycling of the Connector and Connector Gateway. Such issues do not exist in direct connector architecture.
- **Less hardware:** New direct connectors do not require any agent installation on end managed system or other computer. The overall hardware requirement for connectors setup is reduced due to this new architecture.

Application Types for Connectors

The following table lists the application type of Connectors:

Table 1—Application Types for Connectors

Connector	Application Type	Details
Active Directory	Active Directory - Direct	Chapter 1:SailPoint Active Directory Connector 7
AIX	AIX - Direct	Chapter 2:SailPoint AIX Connector 35
Azure Active Directory	Azure Active Directory	Chapter 3:SailPoint Azure Active Directory Connector 51
Remedy	BMC Remedy - Direct	Chapter 4:SailPoint BMC Remedy Connector 61
Remedy ITSM	BMC ITSM - Direct	Chapter 5:SailPoint BMC Remedy IT Service Management Suite Connector 67
Box	Box	Chapter 6:SailPoint Box Connector 75
DB2 Windows	DB2 Windows - Direct	Chapter 7:SailPoint DB2 Windows Connector 81
Delimited	DelimitedFile	Chapter 8:SailPoint Delimited File Connector 89
Dropbox	Dropbox	Chapter 9:SailPoint Dropbox Connector 93
Duo Connector	Duo	Chapter 10:SailPoint Duo Connector 99
GE Centricity	GE Centricity	Chapter 11:SailPoint GE Centricity Connector 105

Application Types for Connectors

Table 1—Application Types for Connectors

Connector	Application Type	Details
Google Apps	GoogleApps - Direct	Chapter 12:SailPoint Google Apps (G Suite) Connector 111
GoToMeeting	GoToMeeting	Chapter 13:SailPoint GoToMeeting Connector 131
IBM i	IBM i	Chapter 14:SailPoint IBM i Connector 135
JDBC	JDBC	Chapter 15:SailPoint JDBC Connector 143
Jive	JIVE	Chapter 16:SailPoint Jive Connector 151
LDAP	LDAP	Chapter 17:SailPoint LDAP Connector 157
LDIF	LDIF	Chapter 18:SailPoint LDIF Connector 175
Logical	Logical	Chapter 19:SailPoint Logical Connector 181
Lotus Domino	IBM Lotus Domino - Direct	Chapter 20:SailPoint Lotus Domino Connector 187
Linux	Linux - Direct	Chapter 21:SailPoint Linux Connector 201
Mainframe	Mainframe	Chapter 22:SailPoint Mainframe Connector 215
Microsoft SQL Server	Microsoft SQL Server - Direct	Chapter 23:SailPoint Microsoft SQL Server 217
Microsoft SharePoint Server	Microsoft SharePoint Server	Chapter 24:SailPoint Microsoft SharePoint Server Connector 227
Microsoft SharePoint Online	Microsoft SharePoint Online	Chapter 25:SailPoint Microsoft SharePoint Online Connector 235
Okta	Okta	Chapter 26:SailPoint Okta Connector 241
Oracle	Oracle Database - Direct	Chapter 27:SailPoint Oracle Connector 251
Oracle HRMS	Oracle HRMS	Chapter 28:SailPoint Oracle HRMS Connector 259
PeopleSoft HCM Database	PeopleSoft HCM Database	Chapter 29:SailPoint PeopleSoft HCM Database Connector 265
RACF	RACF	Chapter 30:SailPoint RACF Connector 275
Remedyforce	Remedyforce	Chapter 31:SailPoint Remedyforce Connector 283
RSA Authentication Manager	RSA Authentication Manager - Direct	Chapter 32:SailPoint RSA Authentication Manager Connector 291
Salesforce/Remedyforce	RemedyForce	Chapter 33:SailPoint Salesforce Connector 299

Table 1—Application Types for Connectors

Connector	Application Type	Details
SAP HR/HCM	SAP HR/HCM	Chapter 34:SailPoint SAP HR/HCM Connector 313
SAP HANA	SAP HANA Database	Chapter 35:SailPoint SAP HANA Connector 333
ServiceNow	ServiceNow	Chapter 36:SailPoint ServiceNow Connector 341
Solaris	Solaris - Direct	Chapter 37:SailPoint Solaris Connector 355
SuccessFactors	SuccessFactors	Chapter 38:SailPoint SuccessFactors Connector 369
SQL Loader	SQLLoader	Chapter 39:SailPoint SQL Loader Connector 379
System for Cross-Domain Identity Management 2.0	SCIM 2	Chapter 40:SailPoint System for Cross-Domain Identity Management Connector 2.0 385
System for Cross-Domain Identity Management	SCIM	Chapter 41:SailPoint System for Cross-Domain Identity Management Connector 391
Sybase	Sybase - Direct	Chapter 42:SailPoint Sybase Connector 405
Tivoli Access Manager	IBM Tivoli Access Manager	Chapter 43:SailPoint Tivoli Access Manager Connector 415
Top Secret	TopSecret	Chapter 44:SailPoint Top Secret Connector 423
UNIX	Unix	Chapter 45:SailPoint UNIX Connector 437
Web Service	Web Services	Chapter 46:SailPoint Web Services Connector 441
Webex	Webex	Chapter 47:SailPoint WebEx Connector 475
Windows Local	Windows Local - Direct	Chapter 48:SailPoint Windows Local Connector 481
Workday	Workday	Chapter 49:SailPoint Workday Connector 491
XML	XML	Chapter 50:SailPoint XML Connector 523
Yammer	Yammer	Chapter 51:SailPoint Yammer Connector 527

Viewing the available connectors

Connectors may be added, removed, or modified in any release, including patch releases. Existing defined applications will continue to use the connector specified during their initial creation, and changes to the connector will not affect existing applications unless those changes are manually applied to the application definition. However, the **ConnectorRegistry** entry for the connectors does change with new releases. The list of available connectors, with their current set of available features, can be retrieved from the Connector Registry within the Debug Pages.

Select **Configuration** in the Objects list and click **List**, then select **ConnectorRegistry** to view the XML for all the connectors.

The **featuresString** value on each connector indicates the functionality that connector is capable of providing; when **PROVISIONING** is specified in the **featuresString**, the connector is a write-capable connector. The "`<entry key="MscsType" value="[MSCS-Type-Name]" />`" attribute requests the name for specific connector's MSCS Type value (also listed in the previous section here).

The out-of-the-box connector specifications can also be found in the **ConnectorRegistry.xml** file in the **[IdentityIQ Installation Directory]/WEB-INF/config** directory.

Connector selection

Often there is more than one connector that can communicate with a single external application, which may raise questions as to which one is the best choice. When multiple connectors exist for a single application, they are always of different types. The best choice is dictated by the needs (and license limitations) of the organization. More information on connector selection is provided in the introductory section for each application within this document.

Chapter 1: SailPoint Active Directory Connector

The following topics are discussed in this chapter:

Overview.....	7
Supported features	8
Supported Managed System.....	9
Pre-requisites	10
Administrator permissions	10
Configuration parameters.....	12
Forest Configuration	12
Domain Configuration	13
Exchange Configuration.....	14
Additional configuration parameters	14
Configuring searchDNs.....	18
Schema attributes	18
Account attributes	19
Group attributes.....	25
Provisioning Policy attributes	25
Active Directory Recycle Bin	30
Additional information	31
Unstructured Target Collector	31
Creating TLS communication.....	32

Overview

This connector manages multiple domains or forests for users, contacts, groups, Exchange mailbox, mail users, mail contacts and Skype users from single application. The scope of the application can be defined in terms of containers, domains and forests. The connector also manages group memberships, foreign security principals (FSP), terminal services, and Dial-in Attributes. The connector supports restoring deleted objects from Active Directory Recycle Bin. The connector mainly uses the LDAP, ADSI and PowerShell interfaces to communicate with end system.

IQService, a Windows service, is an integral part of this connector. Most of the features that this connector supports requires IQService to be deployed on a Windows system in the environment. For more information, see “Appendix D: IQService”.

For large environments, for faster aggregation of the users and contacts, the connector supports partitioned aggregation. For more information, see “Appendix B: Partitioning Aggregation”.

For large environments, for faster delta aggregation of the accounts, the connector supports partition delta aggregation. When delta aggregation is performed, the connector aggregates only the changes made to the Active Directory accounts and groups since last aggregation using Active Directory synchronization (DirSync) control.

In addition, the connector supports managing **Microsoft Exchange** user mailboxes, mail users, mail contacts and distribution lists. While this connector reads Exchange Server attributes by connecting to the Active Directory, for provisioning, connector uses remote Powershell via IQService.

Overview

The connector also supports **Microsoft Skype for Business Server** user management. The connector uses **Microsoft Skype for Business Server** administrative tools to read and provision to Microsoft Skype for Business Server. The connector supports create, update, delete, enable/disable, setting policies, and managing PIN for Microsoft Skype for Business user.

Supported features

The Active Directory connector provides the ability to provision users, groups, contacts, and entitlements. The connector supports the following features:

- Account Management
 - Active Directory Users
 - Manages Active Directory Users as Accounts
 - Aggregation, Delta Aggregation, Partitioned Aggregation, Refresh Account, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements (includes Foreign Security Principals)
 - Terminal Services, Dial-in Attributes
 - Create, Update, Delete Exchange User Mail Box
 - Create, Update, Delete Exchange Mail User
 - Create, Update, Delete Skype for Business user
 - Enable/disable, setting policies for Skype for Business user
 - Reset Skype for Business user PIN
 - Password Interceptor
 - Active Directory Contacts
 - Manages Active Directory Contacts as Accounts
 - Aggregation, Partitioned Aggregation, Refresh Account
 - Create, Update, Delete
 - Add/Remove Entitlements
 - Create, Update, Delete Exchange Mail Contact
- Account - Group Management
 - Manages Active Directory Groups as Account-Groups
 - Aggregation, Delta Aggregation, Refresh Group
 - Create, Update, Delete
 - Create, Delete Exchange Distribution List

- Permission Management

- Application can be configured for following unstructured target collectors to read permissions from the following end system:

Windows File Share: Read Windows File Share permissions directly assigned to accounts and groups.

- Supports automated revocation of the aggregated permissions and creates work items for requests only when the default provisioning action is overridden and **Manual Work** Item is selected as the provisioning action.

- Other

- Restore deleted objects (Active Directory Accounts and Groups) using 'Active Directory Recycle Bin'
- Supports executing native before/after scripts for provisioning requests
- Provides support for Simple Authentication and Security Layer (SASL) when binding to Active Directory
- From version 7.2 onwards, by default Active Directory Connector provides support for Serverless configuration

For more information, see "Pre-requisite for Active Directory Connector" on page 554.

References

- "Active Directory Recycle Bin" on page 30
- "Unstructured Target Collector" on page 31
- "Appendix A: Delta Aggregation"
- "Appendix B: Partitioning Aggregation"
- "Appendix D: IQService"

Supported Managed System

- Supported Active Directory Domain Services (AD DS) functional levels
 - Microsoft Windows Server 2016
 - Microsoft Windows Server 2012 R2
 - Microsoft Windows Server 2012
 - Microsoft Windows Server 2008 R2
 - Microsoft Windows Server 2008
 - Microsoft Windows Server 2003
- Supported Microsoft Exchange Servers
 - Microsoft Exchange Server 2016
 - Microsoft Exchange Server 2013
 - Microsoft Exchange Server 2010
- Supported Microsoft Lync\Skype for Business Servers
 - Microsoft Skype for Business 2015 Server
 - Microsoft Lync Server 2013

Pre-requisites

1. Before you start using the connector, it is required that the IQService is installed and registered on any Windows system with any of the supported Operating System. For more information on installing and registering IQService, see "Appendix D: IQService".
Note: **If 'Authentication Type' is set to 'Strong' then IQService host must be in the same domain or in trusted domain.**
2. For Exchange and Skype for Business management, Exchange PowerShell version 3.0 or above must be configured on system running IQService.
3. For managing Terminal Services (Remote Desktop Services profile) attributes, install the IQService on a Server class Windows Operating System.
4. For application managing multiple domain trees either from same or different forests, there must be two way trust relationship between them.

Administrator permissions

- The Service Account must have appropriate rights on the Active Directory. The Domain Controller must be accessible from the IQService host computer.

Note: **The rights discussed in the following section grant limited account creation privileges to a user. This user can create and modify most accounts. It cannot manage the Administrator user account, the user accounts of administrators, the Server Operators, Account Operators, Backup Operators, and Print Operators. To manage these user types you must assign the appropriate security rights or add the user to groups having higher permissions. For example, domain administrators.**

The service account specified in the application must be the member of the Account Operators group.

More granular rights can be assigned to users for specific portions of the directory, but this is discouraged by Microsoft best practices for Active Directory access control. The required rights will depend on the use cases that are implemented, but could include

- Read All Properties
 - Write All Properties
 - Create User Objects
 - Delete User Objects
 - Change Password
 - Reset Password
 - Read Members
 - Write Members
- For Strong authentication (SASL), single service account can be used for multiple domains/forest. For this:
 - The domains must have two way trust
 - The service account must have delegated permissions across other domains for user, contact and group objects.

Permissions must be delegated to Service Account using Delegation Control Wizard as follows:

- a. Open Active Directory Users and Computers.

- b. Right click on the domain and select **Delegate Control** to open **Delegation of Control Wizard** and click **Next**.
- c. Add Service Account user using **Add** button and click on **Next**.
- d. Select **Create a custom task to delegate** and click **Next**.
- e. Select Only the following objects in the folder option and select User Objects, Contact Objects and Group Objects and Create/Delete the selected objects in the folder.
- f. On the next screen, select **Full Control** under **Permissions** and click on **Next** and then **Finish**.

For Foreign Security Principals (FSPs) to be aggregated, created, and modified single service account must have full delegated permissions on FSP container. Permissions must be delegated to Service Account using **Delegation Control Wizard** as follows:

- a. Open **Active Directory Users and Computers**.
 - b. Right click on the domain and select **Delegate Control** to open **Delegation of Control Wizard** and click **Next**.
 - c. Add Service Account user using **Add** button and click on **Next**.
 - d. Select **Create a custom task to delegate** and click **Next**.
 - e. Select This folder, existing objects in this folder and creation of objects in this folder.
 - f. On the next screen, select **Full Control under Permissions** and click on **Next** and then **Finish**.
- For managing Exchange Server, the Service Account must be a member of Recipient Management group.

Note: Application user for provisioning of Exchange Server must be Remote shell enabled. To enable remote Shell for a user, set the 'RemotePowerShellEnabled' parameter to \$True on the Set-User cmdlet.

For example, Set-User UserName -RemotePowerShellEnabled \$True

- For managing contacts, the contacts must be delegated to **Account Operators** group using Delegation Control Wizard as follows:
 - a. Open **Active Directory Users and Computers**.
 - b. Right click on the domain and select **Delegate Control** to open **Delegation of Control Wizard** and click **Next**.
 - c. Select **Account Operators** group and click **Next**.
 - d. Select **Create a custom task to delegate** and click **Next**.
 - e. Select Only the following objects in the folder option and select **Contact Objects** and **Create and Delete selected objects in the folder**.
 - f. On the next screen, select **Full Control** under **Permissions** and click on **Next** and then **Finish**.

- For Microsoft Skype for Business Server user management, service account must be a member of RTCUniversalServerAdmins and **CSUserAdministrator** domain groups. The account must also be a member of local Administrator group on the system running IQService.

OR

For Microsoft Skype for Business Server user management, service account must be a member of custom group with **SQL permission** and **CSUserAdministrator** domain groups. The account must also be a member of the local Administrator group on the system running IQService.

Permissions required for **Custom group** and **CSUserAdministrator** domain group in SQL:

Database Instance	Security login	Database Role Membership	Databases
RTCLOCAL	Group required to be added in SQL server: Custom Group and CSUserAdministrator	DB_Owner	RTC, XDS, RTCDYN

Configuration parameters

Database Instance	Security login	Database Role Membership	Databases
RTC	Group required to be added in SQL server: Custom Group and CSUserAdministrator	DB_Owner	RTCXDS, XDS

Configuration parameters

This section contains the information that the connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Active Directory connector uses the following connection parameters:

Attributes	Description
IQService Host	Host name of the system where IQService is installed.
IQService Port	Port number used by the IQService.

Forest Configuration

Forest Configuration consists of the details of all the forests that must be managed.

Configuring the Global Catalog details also helps improve the pass-through authentication performance. The Active Directory Connector provides preference to connect to the Global Catalog if details are provided, else uses Server configured for respective domains to authenticate the users.

The following table lists the attributes that must be configured for each domain that the application is managing:

Attributes	Description
Forest Name	Logical name of the forest used in the organization.
GC Server	Global Catalog Server information in following format: GC IP/FQDN:Port
User	Service account to manage forest's with appropriate permissions mentioned in the "Administrator permissions" on page 10. For Strong Authentication (SASL) to work, username must be in userPrincipalName format that is, UserName@DNSDomainName.com . For more information, see "Using Strong Authentication (SASL)" on page 33.
Password	Password of service account.

Attributes	Description
Authentication Type	<p>Authentication Type to be used when binding to Active Directory.</p> <ul style="list-style-type: none"> Simple: the account to authenticate is identified by the DN of the entry for that account, and the proof identity comes in the form of a password. Recommend to Use TLS with Simple Authentication Strong: Strong Authentication bind is performed which uses Kerberos or NTLM depending upon whether the IdentityIQ is in a network (of service account domain) or outside network. SASL has implicit security layer for Data Encryption.
Use TLS	<p>(Applicable only when Authentication Type is Simple) Indicates whether this is a TLS communication.</p> <p>Note: For more information on enabling the TLS communication, see "Creating TLS communication" on page 32.</p>
Manage All Domains	<p>Manages all domains under that forest using the forest credential. If selected, domain configuration section is not required to be configured. In this case, domains that the application will manage can be previewed with Preview button.</p> <p>If not selected, domains in this forest can be enumerated in the Domain Configuration by clicking Discover button.</p>

Domain Configuration

Domain Configuration consist of details to connect domain/s such as distinguished name of the domain, Username and Password. Domain settings must be configured for all domains that this application is expected to contact.

The following table lists the attributes that must be configured for each domain that the application is managing.

Attributes	Description
Forest Name*	Name of the forest of the domain. This forest must also be configured in Forest Configuration.
Domain*	Distinguished name of the domain.
User*	<p>User of the domain in Domain\User format with appropriate rights required to read and provision.</p> <p>For Strong Authentication (SASL) to work, username must be in userPrincipalName format that is, UserName@DNSDomainName.com. For more information, see "Using Strong Authentication (SASL)" on page 33.</p>
Password*	Password of the user mentioned for User field.
Servers	Domain Controller Servers (IP or FQDNs) that can be configured.

Configuration parameters

Attributes	Description
Authentication Type	Authentication Type to be used when binding to Active Directory. <ul style="list-style-type: none">• Simple: the account to authenticate is identified by the DN of the entry for that account, and the proof identity comes in the form of a password. Recommend to Use TLS with Simple Authentication.• Strong: Strong Authentication bind is performed which uses Kerberos or NTLM depending upon whether the IdentityIQ is in a network (of service account domain) or outside network. SASL has implicit security layer for Data Encryption.
Use TLS	(Applicable only when Authentication Type is Simple) Indicates whether this is a TLS communication. Note: For more information on enabling the TLS communication, see "Creating TLS communication" on page 32.

Note: Attributes with asterisk mark (*) are the mandatory attributes.

Exchange Configuration

Exchange Configuration consist of details to connect Exchange Server such as Exchange Forest, Exchange Hosts, UserName, Password, Account Forests and whether the communication should be over TLS. If this application needs to manage Exchange mailboxes, mail users or distribution lists, following Exchange Configurations must be defined.

Attributes	Description
Exchange Forest*	Name of the forest where exchange is installed. This forest should also be configured in Forest Configuration.
Exchange Hosts*	FQDN or IP of Exchange Server Host/s.
User*	User in Domain\User format with appropriate rights.
Password*	Password of the user mentioned for User field.
Account Forest*	Name of accounts (user) forests served by this exchange.
Use TLS	Indicates whether this is a TLS communication. Note: For more information on enabling the TLS communication, see "Creating TLS communication" on page 32.

Note: Attributes with asterisk mark (*) are the mandatory attributes.

Additional configuration parameters

The following attributes can be added into the application debug page:

Attributes	Description
unlockOnChangePassword	The default behavior of unlocking the account on change password can be turned off by setting the <code>unlockOnChangePassword</code> attribute to false . Default: true
setAttributeLevelResult	Set it to true to enable attribute request level results. Default: False Note: Enabling this parameter would marginally increase the time taken to process the request.
aggregationMaxRetries	Count of maximum retry attempts for Active Directory aggregation in case of failures with any of the retry-able errors. Default: 5
aggregationRetryThreshold	Delay in seconds between each retry attempt of aggregation Default: 10 seconds
manageLync	Microsoft Lync\Skype for Business Server to be managed by the application. Add the <code>manageLync</code> attribute as follows in the application debug page: <pre><entry key="manageLync"> <value> <Boolean>true</Boolean> </value> </entry></pre>
authSearchAttributes	List of attributes which would be used to search user during Pass Through Authentication. The <code>authSearchAttributes</code> attribute can be changed as follows in the application debug page: <pre><entry key="authSearchAttributes"> <value> <List> <String>sAMAccountName</String> <String>msDS-PrincipalName</String> <String>mail</String> </List> </value> </entry></pre>
memoryStoreSizeInElements	Defines the number of cache elements to be stored in memory (RAM). If all elements must be stored in-memory and nothing on the disk, specify the value as -2 as follows: <pre><entry key="memoryStoreSizeInElements" value="-2"/></pre>

Configuration parameters

Attributes	Description
disableComputePreloading	<p>To disable auto detection of group membership pre-loading for forests, set the value to true as follows:</p> <pre data-bbox="706 382 1258 508"><entry key="disableComputePreloading"> <value> <Boolean>true</Boolean> </value> </entry></pre> <p>Default: false</p>
useSingleThreadedCookieSearch	<p>During full aggregation, dirsync cookies are fetched as per domain basis using concurrent threads. To fetch cookies sequentially on a single thread, set the value to true as follows:</p> <pre data-bbox="706 705 1312 756"><entry key="useSingleThreadedCookieSearch" value="true"/></pre>
displayAttributeForContacts	<p>CN is used as default for display name of contact objects in IdentityIQ. To use any other schema attribute, define the name of the attribute as the as value of this attribute:</p> <pre data-bbox="706 895 1282 946"><entry key="displayAttributeForContacts" value="firstName"/></pre>
disableFspAggregation	<p>To disable aggregating foreign memberships of any user, set the value to true as follows:</p> <pre data-bbox="706 1043 1209 1170"><entry key="disableFspAggregation"> <value> <Boolean>true</Boolean> </value> </entry></pre> <p>Default: false</p>
skipDeletedObjScopeCheckInDelta	<p>If set to true as follows during account delta aggregation, connector does not make a call to Active Directory to check whether deleted object was in scope of the application.</p> <pre data-bbox="706 1360 1344 1419"><entry key="skipDeletedObjScopeCheckInDelta" value="true"/></pre> <p>If the deleted object is present in the IdentityIQ database, it gets deleted from the database. If the deleted object is not there in the IdentityIQ database not then no further action would be performed.</p> <p>Default: false</p>

Attributes	Description
skipObjTypeCheckForMembersInDelta	<p>If set to true as follows during account delta aggregation, connector does not make a call to Active Directory to check if objectType of member is added/removed to a group:</p> <pre data-bbox="703 397 1372 449"><entry key="skipObjTypeCheckForMembersInDelta" value="true"/></pre> <p>If object is present in the IdentityIQ database, then membership would get updated.</p> <p>Default: false</p>
skipBindUsingDNS	<p>If set to true as follows, DNS server would not be used to find out Domain Controller for any given domain in serverless configuration:</p> <pre data-bbox="703 720 1348 751"><entry key="skipBindUsingDNS" value="true"/></pre> <p>Connector would always call IQService to find domain controller.</p> <p>Default: false</p>
skipGetObjInMembershipDelta	<p>If set to true as follows, Connector would not make a call to Active Directory to get additional attributes of the changed object intercepted during delta aggregation.</p> <pre data-bbox="703 963 1286 1015"><entry key="skipGetObjInMembershipDelta" value="true"/></pre> <p>These additional attributes are fetched by connector if user has entitlement changes along with attribute change(s) or if users have add, remove or both entitlement changes.</p> <p>Hence, when skipGetObjInMembershipDelta is set to true, the Resource Object is sent to IdentityIQ containing only the attributes intercepted during delta aggregation.</p> <p>Default: false</p>
Caching Ports	
Port numbers for caching mechanism to replicate the cached data across different task servers.	
Note: SailPoint recommends that the ports are open and not in use by any other application.	
cacheRmiPort	The default value is 40001
cacheRemoteObjectPort	The default value is 40002
cacheReplicationTimeout	<p>Maximum time in minutes to wait for membership cache replication on task server. Default: 10 minutes</p> <pre data-bbox="703 1660 1416 1691"><entry key="cacheReplicationTimeout" value="20"/></pre>
cacheSocketTimeoutMillis	<p>Maximum time in milliseconds to wait for the client sockets to send messages to a remote listener. Default: 2000 milliseconds.</p> <pre data-bbox="703 1786 1241 1839"><entry key="cacheSocketTimeoutMillis" value="5000"/></pre>

Schema attributes

Note: Active Directory connector supports all jndi system properties. For more information, see <https://docs.oracle.com/javase/jndi/tutorial/ldap/connect/config.html>

Following are examples with sample values:

```
<entry key="com.sun.jndi.ldap.connect.pool.maxsize" value="10"/>
<entry key="com.sun.jndi.ldap.connect.pool.protocol" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.timeout" value="20000"/>
<entry key="com.sun.jndi.ldap.connect.pool.initsize" value="5"/>
<entry key="com.sun.jndi.ldap.connect.pool.authentication" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.debug" value="fine"/>
<entry key="com.sun.jndi.ldap.connect.pool" value="true"/>
<entry key="com.sun.jndi.ldap.read.timeout" value="120000"/>
```

Configuring searchDNs

The searchDNs define list of distinguished names of the containers along with other relevant attributes which defines scope for this application. Each of these searchDNs is also considered as a partition for partitioned full aggregation. Users, Contacts and Groups can have different set of searchDNs to define different scope for each of them. In case the scope is not defined for Groups, it follows **Accounts Search Scope***. Defining one search DN to the minimum is required to successfully configure application.

Attributes to be defined for searchDNs are as follows:

Attributes	Description
Allow Auto Partitioning	Auto Partitioning discovers and forms partitions which are evenly distributed. Note: SailPoint recommends use of 'Allow Auto Partitioning' feature for configuring partitions. For configuring partitions manually, see "Configuring partitions manually" on page 540.
Search DN*	Distinguished Name of the container.
Iterate Search Filter	LDAP filter that defines scope for accounts/groups from this container.
Group Membership Search DN	(Applicable for users and contact search scope) Distinguished Names of the containers/domains separated by semicolon (;) that defines search scope when looking for group membership for the accounts. If not defined, connector considers the searchDN in order.
Group Member Filter String	(Applicable only for user search scope) LDAP Search filter to apply while fetching user's group membership.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, accounts (users and contacts) and group. Account objects are used when building identities Link objects. The group schema is used when building Account_Group objects which are used to hold entitlements shared across identities.

Note: The Schema tab is used to define the attributes for each object type in the application being configured. The schema attributes can be defined as Entitlement, Multi-Valued and Indexed. For more information on the schema tab, see *SailPoint IdentityIQ Administration Guide*.

Account attributes

Attributes	Description
businessCategory	The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: “engineering”, “finance”, and “sales”.
carLicense	This attribute type contains the license plate or vehicle registration number associated with the user.
cn	This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: “Martin K Smith”, “Marty Smith” and “printer12”.
distinguishedName	This attribute contains the distinguished name by which the user is known.
departmentNumber	This attribute contains a numerical designation for a department within your enterprise.
description	This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: “Updates are done every Saturday, at 1am.”, and “distribution list for sales”.
destinationIndicator	This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: “AASD” as a destination indicator for Sydney, Australia. “GBLD” as a destination indicator for London, United Kingdom. Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.
displayName	This attribute contains the preferred name to be used for this person throughout the application.
employeeNumber	This attribute contains the numerical identification key for this person within your enterprise.
employeeType	This attribute contains a descriptive type for this user, for example, contractor, full time, or part time.
externalEmailAddress	This attribute contains external email address of the mail user. Mail user is an AD user having mailbox outside of organization.

Schema attributes

Attributes	Description
facsimileTelephoneNumber	This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute.
givenName	This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. Examples: "John", "Sue", and "David".
homePhone	This attribute contains the employees home phone number.
homePostalAddress	This attribute contains the employees mailing address.
homeMDB	Exchange mailbox store DN. Required for mailbox creation.
initials	This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute. Examples: "J. A." and "J"
internationalISDNNumber	This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. Example: "0198 444 444".
l	This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. Examples: "Austin", "Chicago", and "Brisbane".
mail	This attribute type contains the RFC822 mailbox for the user.
manager	This attribute type contains the distinguished name of the manager to whom this person reports.
mailNickname	Exchange Alias.
mobile	This attribute type contains the mobile telephone number of this person.
msExchHideFromAddressLists	Hide from Exchange address lists.
msNPAllowDialin	Indicates whether the account has permission to dial in to the RAS server.
msNPCallingStationID	If this property is enabled, the server verifies the caller's phone number. If the caller's phone number does not match the configured phone number, the connection attempt is denied.
msRADIUSCallbackNumber	The phone number that is used by the server is set by either the caller or the network administrator. If this property is enabled, the server calls the caller back during the connection process.

Attributes	Description
msRADIUSFramedRoute	Define a series of static IP routes that are added to the routing table of the server running the Routing and Remote Access service when a connection is made.
msRADIUSFramedIPAddress	Use this property to assign a specific IP address to a user when a connection is made.
o	This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute.
ou	This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies".
objectguid	Globally unique identifier of the object.
pager	This attribute type contains the telephone number of this persons pager.
objectType	Indicates type of the Active Directory objects. For example, User, Contact
physicalDeliveryOfficeName	This attribute type contains names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E".
postOfficeBox	This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27".
postalAddress	This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA".
postalCode	This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA.
preferredDeliveryMethod	This attribute type contains an indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone".
preferredLanguage	This attribute type contains the preferred written or spoken language of this person.
registeredAddress	This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$XYZ Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA".

Schema attributes

Attributes	Description
roomNumber	This attribute type contains the room or office number or this persons normal work location.
secretary	This attribute type contains the distinguished name of this persons secretary.
seeAlso	<p>This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute.</p> <p>Example: The person object “cn=Elvis Presley,ou=employee,o=XYZ\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=XYZ\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=XYZ\, Inc.”. Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values.</p>
sn	<p>This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute.</p> <p>Example: “Smith”.</p>
st	<p>This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute.</p> <p>Example: “Texas”.</p>
street	<p>This attribute type contains site information from a postal address (that is, the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute.</p> <p>Example: “15 Main St.”.</p>
telephoneNumber	<p>This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute.</p>
teletexTerminalIdentifier	<p>The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute.</p>
telexNumber	<p>This attribute type contains sets of strings that are a telex number, country code, and answer back code of a telex terminal. Each set is one value of this multi-valued attribute</p>
title	<p>This attribute type contains the persons job title. Each title is one value of this multi-valued attribute.</p> <p>Examples: “Vice President”, “Software Engineer”, and “CEO”.</p>
uid	<p>This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute.</p> <p>Examples: “s9709015”, “admin”, and “Administrator”.</p>
objectClass	<p>The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”.</p>
memberOf	<p>This attribute type contains the account group membership for this person on the application.</p>
objectSid	Windows Security Identifier

Attributes	Description
sAMAccountName	This attribute type contains the sAMAccoutName for this user.
msDS-PrincipalName	Name of the entity in the following format: NetBIOS domain name\sAMAccountName
sIDHistory	(Optional) User can add this attribute manually to view the data in a readable string format.
TS_TerminalServicesProfilePath*	The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server.
TS_TerminalServicesHomeDrive*	The root drive for the user.
TS_TerminalServicesHomeDirectory*	The root directory for the user.
TS_TerminalServicesInitialProgram*	The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server.
TS_TerminalServicesWorkDirectory*	The working directory path for the user.
TS_EnableRemoteControl*	A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session.
TS_AllowLogon*	A value that specifies whether the user is allowed to log on to the RD Session Host server.
TS_BrokenConnectionAction*	A value that specifies the action to be taken when a Remote Desktop Services session limit is reached.
TS_ReconnectionAction*	A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed.
TS_ConnectClientDrivesAtLogon*	A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started.
TS_ConnectClientPrintersAtLogon*	A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled.
TS_DefaultToMainPrinter*	A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled.
TS_MaxConnectionTime*	The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated.
TS_MaxDisconnectionTime*	The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated.

Schema attributes

Attributes	Description
TS_MaxIdleTime*	The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated.
Microsoft Lync\Skype for Business Server attributes	
msRTCSIP-UserEnabled	Whether the user is currently enabled for Microsoft Lync\Skype for Business Server.
LineServerURI	The line server URL.
EnabledForFederation	Whether a user is enabled for federation.
PublicNetworkEnabled	Whether a user is enabled for access outside network.
EnterpriseVoiceEnabled	Whether a user EnterpriseVoiceEnabled service is enabled.
LineURI	The line Uniform Resource Identifier (URI).
SipAddress	This attribute contains the SIP address of a given user.
VoicePolicy	The name of Voice Policy.
MobilityPolicy	The name of Mobility Policy.
ConferencingPolicy	The name of Conferencing Policy.
PresencePolicy	The name of Presence Policy.
VoiceRoutingPolicy	The name of VoiceRouting Policy.
RegistrarPool	The name of registrar pool.
LocationPolicy	The name of Location Policy.
ClientVersionPolicy	The name of ClientVersion Policy.
ClientPolicy	The name of Conferencing Policy.
ExternalAccessPolicy	The name of ExternalAccess Policy.
HostedVoicemailPolicy	The name of HostedVoicemail Policy.
PersistentChatPolicy	The name of PersistentChat Policy.
UserServicesPolicy	The name of UserServices Policy.
ExperiencePolicy	The name of Experience Policy.
ArchivingPolicy	The name of Archiving Policy.
LegalInterceptPolicy	The name of LegalIntercept Policy.
PinPolicy	The name of Pin Policy.
LyncPinSet	Whether a user pin is set.
LyncPinLockedOut	Whether a user pin is locked.

Note: Attributes with asterisk mark (*) are the Terminal Services/Remote Desktop Services attributes. By default, these attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes, add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Group attributes

Table 1—Active Directory Connector - Group Attributes

Name	Description
cn	This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: "Martin K Smith", "Marty Smith" and "printer12".
dn	This attribute type contains the directory path to the object.
owner	This attribute type contains the name of the owner of the object.
objectguid	Globally unique identifier of the object.
description	This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: "Updates are done every Saturday, at 1am.", and "distribution list for sales".
mailNickname	Exchange Alias.
memberOf	This attribute type contains the account group membership for this person on the application.
objectSid	This attribute type contains the Windows Security Identifier for this user.
sAMAccountName	sAMAccountName
groupType	Group Type. Allowed values are: 1. Security 2. Distribution
groupScope	Group Scope. Allowed values are: 1. Domain local 2. Global 3. Universal.
msDS-PrincipalName	Name of the entity in the following format: NetBIOS domain name\sAMAccountName

Provisioning Policy attributes

The following table lists the provisioning policy attributes:

Attribute	Description
Provisioning policy attributes for User Creation	

Provisioning Policy attributes

Attribute	Description
ObjectType	Type of the account to be created. Default value: User. For creating Contact, object type must be contact.
distinguishedName	Distinguished name of the user to be created.
sAMAccountName	sAMAccountName of the user to be created.
password	Password of the user to be created.
lIQDisabled	A boolean attribute, set to true to create a disabled user.
primaryGroupDN	Default group of the user to be created.
description	Description of the user to be created.
msNPAllowDialin	Indicates whether the account has permission to dial in to the RAS server.
msNPCallingStationID	If this property is enabled, the server verifies the caller's phone number. If the caller's phone number does not match the configured phone number, the connection attempt is denied.
msRADIUSCallbackNumber	The phone number that is used by the server is set by either the caller or the network administrator. If this property is enabled, the server calls the caller back during the connection process.
msRADIUSFramedRoute	Define a series of static IP routes that are added to the routing table of the server running the Routing and Remote Access service when a connection is made.
msRADIUSFramedIPAddress	Use this property to assign a specific IP address to a user when a connection is made.
TS_TerminalServicesProfilePath*	The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server.
TS_TerminalServicesHomeDrive*	The root drive for the user.
TS_TerminalServicesHomeDirectory*	The root directory for the user.
TS_TerminalServicesInitialProgram*	The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server.
TS_TerminalServicesWorkDirectory*	The working directory path for the user.
TS_EnableRemoteControl*	A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session.
TS_AllowLogon*	A value that specifies whether the user is allowed to log on to the RD Session Host server.

Attribute	Description
TS_BrokenConnectionAction*	A value that specifies the action to be taken when a Remote Desktop Services session limit is reached.
TS_ReconnectionAction*	A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed.
TS_ConnectClientDrivesAtLogon*	A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started.
TS_ConnectClientPrintersAtLogon*	A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled.
TS_DefaultToMainPrinter*	A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled.
TS_MaxConnectionTime*	The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated.
TS_MaxDisconnectionTime*	The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated.
TS_MaxIdleTime*	The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated. ^a
preferredServer	The preferred server (Domain Controller) on which this request must be executed.
Delete provisioning policy attribute for non-leaf user object	
deleteSubTree	<p>To delete the non-leaf user objects set the value of the attribute to true (boolean).</p> <p>For example,</p> <pre><ProvisioningPlan nativeIdentity="Adam" targetIntegration="AD-Direct"> <AccountRequest application="AD-Direct" nativeIdentity="CN=Adam,CN=Users,DC=SPDomain,DC=local" op="Delete"> <AttributeRequest name="deleteSubTree" op="Add"> <Value> <Boolean>true</Boolean> </Value> </AttributeRequest> </AccountRequest> </ProvisioningPlan></pre>

Provisioning Policy attributes

Attribute	Description
Special provisioning attributes for Move/Rename request	
AC_NewName	A string attribute to rename the user. For example, CN=abc
AC_NewParent	A string attribute to move the user to new OU. For example, OU=xyz,DC=pqr,DC=com
Provisioning Exchange	
homeMDB	(Optional) Exchange mailbox store DN. Required for mailbox creation. Send this attribute with new mailbox store DN to move the mailbox to another mailbox store.
mailNickname	Exchange alias for mailbox or mailuser. Required for mailbox creation and to update or disable the mailbox. Send this attribute with no value to disable the mailbox.
exch_externalEmailAddress	External email address. Required for mailuser creation and to update or disable the mailuser. Send this attribute with no value to disable the mailuser.
msExchHideFromAddressLists	(Optional) Hide from Exchange address lists.
DomainController	(Optional) Fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Note: Active Directory Connector also supports provisioning other Microsoft Exchange attributes other than mentioned above.	
Provisioning policy attributes for Microsoft Lync\Skype for Business user	
SipAddress	To assign the user a specific SIP address.
RegistrarPool	Registrar pool where the user's Microsoft Lync\Skype for Business Server account will be homed. Mandatory attribute. Send this attribute with no value to remove the user from Microsoft Lync\Skype for Business Server.
SipAddressType	Select one of the SipAddressType from supported values: SamAccountName, FirstName, EmailAddress. Microsoft Lync\Skype for Business Server generates a SIP address for the new user when SipAddressType is provided in combination with SipDomain.
SipDomain	The SIP domain for the user account being enabled. Microsoft Lync\Skype for Business Server generates a SIP address for the new user when SipAddressType is provided in combination with SipDomain.
msRTCSIP-UserEnabled	Send this attribute with true/false to enable/disable Microsoft Lync\Skype for Business.
Pin	Dial-in Conferencing PIN number to be set for the Microsoft Lync\Skype for Business.

Attribute	Description
LyncPinLockedOut	Send this attribute with true/false to lock/unlock Microsoft Lync\Skype for Business user's Dial-in Conferencing PIN.
Note: Active Directory Connector also supports provisioning other Microsoft Lync\Skype for Business attributes other than mentioned above.	
	To set any other Microsoft Lync\Skype for Business attributes, edit application xml file to add lyncAttributes application attribute of string type with comma separated name of Microsoft Lync\Skype for Business attributes added in provisioning policy.
Provisioning policy attributes for CreateGroup	
distinguishedName	Group in the distinguished name format.
sAMAccountName	sAMAccountName
Provisioning policy attributes for UpdateGroup	
description	A description of the group.
groupType	Group Type. Allowed values are: 1. Security 2. Distribution
groupScope	Group Scope. Allowed values are: 1. Domain local 2. Global 3. Universal.
mailNickname	Alias and is required if want to create Distribution Group on exchange. Only Universal type of group can be created on exchange.

a. * - Attributes with asterisk mark (*) are the Terminal Services/Remote Desktop Services attributes. By default, these attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes, add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Note: **To skip plan attributes getting processed by connector, add excludeAttributesFromProvisioning attribute to application with value listing names of such attributes. For example,**

```
<entry key="excludeAttributesFromProvisioning">
    <value>
        <List>
            <String>region</String>
        </List>
    </value>
</entry>
```

Active Directory Recycle Bin

A new feature ‘Recycle Bin’ introduced by Microsoft in Windows Server 2008 R2 provides support for restoring deleted users, groups with all their attributes and group memberships. SailPoint Active Directory Connector support this feature. Using this feature, any deleted objects (Accounts and Groups) can be restored.

Pre-requisites

Note: Recycle Bin feature should be enabled on Active Directory.

1. IQService can be installed on Windows system with one of the following Operating System:

- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2

For more information on installing and registering IQService, see “Appendix D: IQService”.

2. Install **Active Directory module for Windows PowerShell** on the computer where IQService is installed.

Note: By default, this module is installed on all DCs.

For non-DC but server class Operating System computer, open Windows PowerShell Console and execute the following commands:

- Import-Module ServerManager
- Add-WindowsFeature -Name "RSAT-AD-PowerShell" -IncludeAllSubFeature

3. Run the following PowerShell command on all domain controllers (DCs) in the forest which must be managed:

Enable-PSRemoting

Note: If multiple servers are managed, run the above command on all the servers present under the “domainSettings”.

Configuring Recycle Bin

1. Open the Console and import `IIQHOME\WEB-INF\config\configManageDeletedObjects.xml` file. The `configManageDeletedObjects.xml` file creates the **Manage Recycle Bin** quick link on the dashboard and adds the **Restore Deleted Objects** workflow.
2. Modify **manageRecycleBin** attribute in the Active Directory application with the value set to **true**.


```
<entry key="manageRecycleBin">
    <value>
        <Boolean>true</Boolean>
    </value>
</entry>
```
3. After account and account-group aggregation, the deleted object would be visible under the **Manage Recycle Bin** quick link. Accounts/Groups can be restored individually or all together.
4. The **DirSync** delta aggregator also supports detecting deleted objects.

Additional information

This section describes the additional information related to the Active Directory Connector.

Unstructured Target Collector

Unstructured target information is used to define unstructured data sources from which the connector is to extract data. Unstructured data is any data that is stored in a format that is not easily readable by a machine. For example, information contained in an Excel spread sheet, the body of an email, a Microsoft Word document, or an HTML file is considered unstructured data. Unstructured targets pose a number of challenges for connectors, because not only is the data stored in a format that is hard to extract from, the systems and directory structures in which the files reside are often difficult to access.

The unstructured target collector that can be configured with Active Directory application is Windows file share.

Note: Active Directory Connector supports automated revocation of the Target Permissions.

Windows File Share

Windows file share target collector can be configured on Active Directory application to read and correlate file share permissions on Active Directory entities. To correlate the aggregated permissions, ensure that the following attribute is marked as Correlation Key in respective schema:

- **objectSid** for Accounts and Groups

This target collector requires a the IQService to be installed on a machine that has visibility to the directory or share to include in the target scan. Refer to the Installation Guide for information on installing and registering the IQService.

The unstructured targets defined on this tab are used by the Target Aggregation task to correlate targets with permissions assigned to identities and account groups for use in certifications.

The Unstructured Targets tab contains the following information:

Table 2—Application Configuration - Unstructured Targets Tab field descriptions

Field	Description
Attributes: The required settings for connecting to the IQService.	
IQService Host	The host on which the IQService resides.
IQService Port	The TCP/IP port where the IQService is listening for requests.
Number of targets per block	Number of targets (files) to include in each block of data returned.
File Shares: The required information for each share.	
Path	UNC Style path to a share or local directory. You can target a specific file or a directory and its sub-directories containing multiple files from which to extract the required data. If you target a directory, use the Wildcard and Directory Depth fields to narrow the query if possible.
Directories Only	Use to instruct to the collector to ignore files and just report back directory permission information.

Additional information

Table 2—Application Configuration - Unstructured Targets Tab field descriptions

Field	Description
Directory Depth	The sub-directory depth from which to extract data. The Directory Depth field enables you to extend your query up to ten (10) sub-directories below the one specified in the Path field.
Wildcard	Use wild cards to target a particular file type or naming scheme. For example, to search only Excel spread sheets, use * .xls or to search only files with names beginning with finance_, use finance_*.*
Include Inherited Permissions	Use to instruct the collector to not report permissions unless they are directly assigned. Only directly assigned permissions will be returned
Administrator	The administrator that has access to this share so you can collect permissions. This value should be the users principal user@xyz.com name or a fully qualified domain user name in the domain\\user format.
Password	The password associated with the specified administrator. Note: The service will be running as System or can be configured to be run as any user, so the Administrator/Password fields may not be required in all cases.
Rules: Specify the rules used to transform and correlate the targets. Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.	
Creation Rule	The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ.
Correlation Rule	The rule used to determine how to correlate accounts (users and contacts) information from the application with identity cubes in IdentityIQ.
Provisioning related Attributes: Select the settings for provisioning to the share.	
Override Default Provisioning	Select it to override the default provisioning action for the collector.
Provisioning Action	The overriding provisioning action for the collector.

To revoke permissions for Active Directory users and/or groups using Windows File Share Target Collector, perform the following:

1. Add the following attributes under target source configuration:

```
<entry key="searchAttrForAcct" value="msDS-PrincipalName"/>
<entry key="searchAttrForGrp" value=" msDS-PrincipalName "/>
```
2. Remove the NO_PERMISSIONS_PROVISIONING feature string from the application configuration.

Creating TLS communication

If you want secure TLS connection for Active Directory, TLS communication must be enabled between IdentityIQ and Active Directory Server. For a Java client to connect using TLS and self-signed certificates, you have to install the certificate into the JVM keystore.

To create TLS communication between IdentityIQ and Active Directory Server, perform the following:

1. Export server certificate and copy the exported .cer file to the Java client computer (IdentityIQ computer).
2. At the client computer execute the following command from the bin directory of JDK:

```
keytool -importcert -trustcacerts -alias aliasName -file <absolute path of certificate> -keystore <JAVA_HOME>/jre/lib/security/cacerts
```

In the preceding command line, *aliasName* is the name of the alias.

3. Login to IdentityIQ.
4. Create the application for Active Directory application type and provide all the required values after selecting the **Use TLS** checkbox.
5. Click on **Test Connection** and save the application.

Using Strong Authentication (SASL)

1. If server is configured in application, then FQDN for that server must be present in DNS record.
2. The service account must be in userPrincipalName format, that is **UserName@DNSDomainName.com**. The **DNSDomainName.com** must have a DNS record/s.
3. For Strong Authentication, user performing PTA must have values of userPrincipalName attribute in Active Directory.

Note: If Authentication Type is selected as follows, then connections only to GC would use Strong Authentication:
- Strong Authentication for forest
- Simple Authentication is selected for all domains in that forest

Additional information

Chapter 2: SailPoint AIX Connector

The following topics are discussed in this chapter:

Overview.....	35
Supported features	35
Supported Managed Systems	36
Pre-requisites	36
Administrator permissions	37
Configuration parameters.....	38
Additional configuration parameters for SSH configuration	38
Public key authentication configuration.....	39
Schema attributes	39
Account attributes	39
Group attributes.....	45
Provisioning policy attributes.....	45
Account attributes	45
Group attributes.....	46
Additional information	47
Upgrade considerations.....	47
Unstructured Target Collector	47
Troubleshooting	48

Overview

The SailPoint AIX Connector manages the accounts and groups on AIX computer.

Supported features

SailPoint AIX Connector provides support for the following features:

- Account Management
 - Manages AIX Users as Accounts
 - Aggregation, Refresh Account
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages AIX Groups as Account-Groups
 - Aggregation, Refresh Group
 - Create, Update, Delete

Overview

- Permissions Management
 - Application can be configured to read file permissions directly assigned to accounts and groups using Unstructured Target Collector.
 - The connector supports automated revocation of the aggregated permissions for accounts and groups.

Note: AIX connector supports MD5, SHA-1, and SHA-2 cryptographic hash functions.

References

- "Unstructured Target Collector" on page 47
- "Appendix C: Before and After Provisioning Action"

Supported Managed Systems

The AIX connector supports the following versions of the operating system:

- AIX 7.2
- AIX 7.1

Pre-requisites

- SSH should be installed on AIX computer.
- The **sshj-0.23.0.jar** and **ganymed-ssh2-build209-1.0.jar** files must be present in \WEB-INF\lib folder
- For Sudo users and permissions
 - The administrator user must have rights to execute /usr/bin/awk command.

Update /etc/sudoers file entry for the administrator user with /usr/bin/awk command.

- User and group schema must add new multi valued schema attribute as **sudoCommands** which would collect all the necessary user commands and store it as a part of this attribute.
- If end user wants to aggregate the sudo commands from multiple sudo files then user must provide list of files as a separate configuration attribute.

For example, `<entry key="sudoCmdFiles" value="/etc/sudoers.d/special_user.conf,/etc/sudoers.d/special_group.conf"/>`

Note: The default command which would collect the sudo commands is as follows:

`awk '/[^#]/' /etc/sudoers.`

In the above command, the commented lines are skipped and the remaining content of /etc/sudoers file are aggregated in to a temporary file on AIX computer.

The temporary file of AIX computer would get copied to local IdentityIQ computer and processes all the sudo user and group commands.

If the end user wants to provide new command for aggregating file data, then it can be configured as a part of application xml file.

For example: `key: sudoUserCommand and value : awk '/[^#]/' /etc/sudoers`

Administrator permissions

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

Rights to execute the following commands with root privilege:

```
/usr/sbin/lsuser, /usr/sbin/lsgroup, /usr/bin/chmod, /usr/bin/mkuser,
/usr/sbin/userdel, /usr/bin/chuser, /usr/bin/chgroup, /usr/bin/mkgroup,
/usr/sbin/rmgroup, /usr/bin/passwd, /bin/rm -f spt_tmp_*, /bin/echo,
/usr/bin/find, /usr/bin/pwdadm, /usr/bin/awk
```

An entry in /etc/sudoers file should look similar to the following:

```
username ALL = (root) PASSWD: /usr/sbin/lsuser, /usr/sbin/lsgroup, /usr/bin/chmod,
/usr/bin/mkuser, /usr/sbin/userdel, /usr/bin/chuser, /usr/bin/chgroup,
/usr/bin/mkgroup, /usr/sbin/rmgroup, /usr/bin/passwd, /bin/rm -f spt_tmp_*, 
/bin/echo, /usr/bin/find, /usr/bin/pwdadm, /usr/bin/awk
```

Note: All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on AIX computers as they might differ from the values mentioned here.

Note: If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

Read Only permissions

If you want to use sudo user to perform read only operations, the sudo user must be configured with the following rights and permissions:

- **For Account Aggregation only**

Rights to execute the following commands with root privilege:

```
/usr/sbin/lsuser, /bin/echo, /bin/rm -f spt_tmp_*
```

An entry in /etc/sudoers file must look similar to the following:

```
username ALL = (root) PASSWD: /usr/sbin/lsuser, /bin/echo, /bin/rm -f spt_tmp_*
```

- **For Group Aggregation only**

Rights to execute the following commands with root privilege:

```
/usr/sbin/lsgroup, /bin/echo, /bin/rm -f spt_tmp_*
```

An entry in /etc/sudoers file must look similar to the following:

```
username ALL = (root) PASSWD: /usr/sbin/lsgroup, /bin/echo, /bin/rm -f spt_tmp_*
```

Note: If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry must be performed. Verify the command paths on AIX computers as they might differ from the values mentioned here.

Configuration parameters

Supported Authentication methods

The AIX Connector supports the following authentication methods for root and sudo user:

- publickey
- username and password

Configuration parameters

The following table lists the configuration parameters of AIX Connector:

Parameters	Description
Unix Server Host	Host Name/IP address of AIX computer. Note: For IdentityIQ version 6.4 Patch 4 and above, the format of the application XML has been changed from <code><entry key="UnixServerHost" value="<hostname>" /></hostname></code> to <code><entry key="host" value="<hostname>" /></hostname></code>
SSH Port	SSH port configured. Default value: 22
Not a 'root' user	If User ID specified is not root, check this parameter.
User Name	User ID on AIX computer that you want to use for connector operations.
User Password	Password of the target system user account that you want to use for connector operations.
Private Key File Path	Path to Private Key File. Private/Public key authentication will have precedence over password authentication.
Passphrase For Private Key	Passphrase provided for creating Private Key.

Additional configuration parameters for SSH configuration

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

Note: These additional configuration parameters must be added in the Application/Target Source debug page.

1. Following is the default command for setting shell prompt on UNIX computer:
`<entry key="SetPrompt" value="PS1='SAILPOINT>' />`
In the above command, "SetPrompt" is the application/target source attribute and PS1='SAILPOINT' is the value of the application/target source attribute.
If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:
For example: For tcsh shell, the entry value would be:
`<entry key="SetPrompt" value="set prompt='SAILPOINT>' />`
2. For executing the commands, verify that the default shell is present on your system.

If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

```
<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>
```

Public key authentication configuration

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

1. Generate Private and Public key's. For more information of the standard steps, see "4 - Test connection fails for key based authentication with an error" on page 49.
2. Append contents of public key file to `~/.ssh/authorized_keys` as shown below.
`cat <public key file> >> ~/.ssh/authorized_keys`
3. Copy private key file to a location which is accessible by IdentityIQ server.
4. Provide path of private key file in application configuration.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
User Name	Name of the user on AIX computer that you want to use for connector operations.
gecos	The General Electric Comprehensive Operating System (GECOS) information for User. The user's name, phone numbers, and other generic personal information are stored here.
id	User ID
pgrp	Primary group of user.
groups	Secondary groups of user.
home	Home directory of user.
shell	Default shell of user.
login	Indicates whether the user can log in to the system with the login command. Possible values are: <ul style="list-style-type: none"> • true: The user can log in to the system. Default. • false: The user cannot log in to the system.

Schema attributes

Attributes	Description
su	Indicates whether another user can switch to the specified user account with the su command. Possible values are: <ul style="list-style-type: none"> true: Another user can switch to the specified account. Default false: Another user cannot switch to the specified account.
rlogin	Permits access to the account from a remote location with the telnet or rlogin commands. Possible values are: <ul style="list-style-type: none"> true: The user account can be accessed remotely. Default false: The user account cannot be accessed remotely.
daemon	Indicates whether the user specified by the <i>Name</i> parameter can execute programs using the cron daemon or the src (system resource controller) daemon. Possible values are: <ul style="list-style-type: none"> true: The user can initiate cron and src sessions. Default false: The user cannot initiate cron and src sessions.
admin	Defines the administrative status of the user. Possible values are: <ul style="list-style-type: none"> true: The user is an administrator. Only the root user can change the attributes of users defined as administrators. false: The user is not an administrator. Default
dce_export	Allows the DCE registry to overwrite the local user information with the DCE user information during a DCE export operation. Possible values are: <ul style="list-style-type: none"> true: Local user information will be overwritten false: Local user information will not be overwritten
sugroups	Lists the groups that can use the su command to switch to the specified user account. The <i>Value</i> parameter is a comma-separated list of group names, or a value of ALL to indicate all groups. An ! (exclamation point) in front of a group name excludes that group. If this attribute is not specified, all groups can switch to this user account with the su command.
admgroups	Lists the groups the user administers. The <i>Value</i> parameter is a comma-separated list of group names. For additional information on group names, see the adms attribute of the /etc/security/group file.
tpath	Indicates the user's trusted path status. The possible values are: <ul style="list-style-type: none"> always: The user can only execute trusted processes. This implies that the user's initial program is in the trusted shell or some other trusted process. notsh: The user cannot invoke the trusted shell on a trusted path. If the user enters the secure attention key (SAK) after logging in, the login session ends. nosak: The secure attention key (SAK) is disabled for all processes run by the user. Use this value if the user transfers binary data that may contain the SAK sequence. Default on: The user has normal trusted path characteristics and can invoke a trusted path (enter a trusted shell) with the secure attention key (SAK).

Attributes	Description
ttys	Lists the terminals that can access the account specified by the <i>Name</i> parameter. The <i>Value</i> parameter is a comma-separated list of full path names, or a value of ALL to indicate all terminals. The values of RSH and REXEC also can be used as terminal names. An ! (exclamation point) in front of a terminal name excludes that terminal. If this attribute is not specified, all terminals can access the user account. If the <i>Value</i> parameter is not ALL, then /dev/pts must be specified for network logins to work.
expires	Identifies the expiration date of the account. The <i>Value</i> parameter is a 10-character string in the MMDDhhmmmyy form, where MM = month, DD = day, hh = hour, mm = minute, and yy = last 2 digits of the years 1939 through 2038. All characters are numeric. If the <i>Value</i> parameter is 0, the account does not expire. The default is 0. See the date command for more information.
auth1	<p>Lists additional mandatory methods for authenticating the user. The auth1 attribute has been deprecated and may not be supported in a future release. The SYSTEM attribute should be used instead. The authentication process will fail if any of the methods specified by the auth1 attribute fail.</p> <p>The <i>Value</i> parameter is a comma-separated list of <i>Method;Name</i> pairs. The <i>Method</i> parameter is the name of the authentication method. The <i>Name</i> parameter is the user to authenticate. If you do not specify a <i>Name</i> parameter, the name of the user being authenticated is used. Valid authentication methods for the auth1 and auth2 attributes are defined in the /etc/security/login.cfg file.</p>
auth2	<p>Lists additional optional methods for authenticating the user. The auth2 attribute has been deprecated and may not be supported in a future release. The SYSTEM attribute should be used instead. The authentication process will not fail if any of the methods specified by the auth2 attribute fail.</p> <ul style="list-style-type: none"> • The <i>Value</i> parameter is a comma-separated list of <i>Method;Name</i> pairs. • The <i>Method</i> parameter is the name of the authentication method. • The <i>Name</i> parameter is the user to authenticate. If you do not specify a <i>Name</i> parameter, the name of the user being authenticated is used.
umask	Determines file permissions. This value, along with the permissions of the creating process, determines a file's permissions when the file is created. The default is 022.
registry	Defines the authentication registry where the user is administered. It is used to resolve a remotely administered user to the local administered domain. This situation may occur when network services unexpectedly fail or network databases are replicated locally. Example values are files or NIS or DCE.
loginretries	<p>Defines the number of unsuccessful login attempts allowed after the last successful login before the system locks the account. The value is a decimal integer string. A zero or negative value indicates that no limit exists. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's <i>unsuccessful_login_count</i> attribute in the /etc/security/lastlog file to be less than the value of loginretries. To do this, enter the following:</p> <pre>chsec -f /etc/security/lastlog -s username -a \ unsuccessful_login_count=0</pre>

Schema attributes

Attributes	Description
pwdwarntime	Defines the number of days before the system issues a warning that a password change is required. The value is a decimal integer string. A zero or negative value indicates that no message is issued. The value must be less than the difference of the maxage and minage attributes. Values greater than this difference are ignored, and a message is issued when the minage value is reached.
account_locked	Indicates if the user account is locked. Possible values include: <ul style="list-style-type: none"> • true: The user's account is locked. The values yes, true, and always are equivalent. The user is denied access to the system. • false: The user's account is not locked. The values no, false, and never are equivalent. The user is allowed access to the system. Default
minage	Defines the minimum age (in weeks) a password must be before it can be changed. The value is a decimal integer string. The default is a value of 0, indicating no minimum age.
SYSTEM	Defines the system authentication mechanism for the user. The value may be an expression describing which authentication methods are to be used or it may be the keyword NONE. <p>The SYSTEM mechanism is always used to authenticate the user, regardless of the value of the auth1 and auth2 attributes. If the SYSTEM attribute is set to NONE, authentication is only performed using the auth1 and auth2 attributes. If the auth1 and auth2 attributes are blank or ignored, as with the TCP socket daemons (ftpd, rexecd and rshd), no authentication will be performed.</p> <p>The method names compat, files and NIS are provided by the security library. Additional methods may be defined in the <code>/usr/lib/security/methods.cfg</code> file.</p> <p>Specify the value for SYSTEM using the following grammar:</p> <pre> "SYSTEM" ::= EXPRESSION EXPRESSION ::= PRIMITIVE "(" "EXPRESSION" ")" EXPRESSION OPERATOR EXPRESSION PRIMITIVE ::= METHOD METHOD "[" "RESULT" "]" RESULT ::= "SUCCESS" "FAILURE" "NOTFOUND" "UNAVAIL" "*" OPERATOR ::= "AND" "OR" METHOD ::= "compat" "files" "NONE" [a-z,A-Z,0-9]* </pre> <p>An example of the syntax is:</p> <pre> SYSTEM = "DCE OR DCE[UNAVAIL] AND compat" </pre>
maxage	Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is a decimal integer string. The default is a value of 0, indicating no maximum age.

Attributes	Description
maxexpired	Defines the maximum time (in weeks) beyond the maxage value that a user can change an expired password. After this defined time, only an administrative user can change the password. The value is a decimal integer string. The default is -1, indicating no restriction is set. If the maxexpired attribute is 0, the password expires when the maxage value is met. If the maxage attribute is 0, the maxexpired attribute is ignored.
minalpha	Defines the minimum number of alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.
minother	Defines the minimum number of non-alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.
logintimes	<p>Specifies the times, days, or both, the user is allowed to access the system. The value is a comma-separated list of entries of the following form:</p> <p>[!]:time-time -or- [!]day[-day][:time-time] -or- [!]date[-date][:time-time]</p> <p>The <i>day</i> variable must be one digit between 0 and 6 that represents one of the days of the week. A 0 (zero) indicates Sunday and a 6 indicates Saturday.</p> <p>The <i>time</i> variable is 24-hour military time (1700 is 5:00 p.m.). Leading zeroes are required. For example, you must enter 0800, not 800. The <i>time</i> variable must be four characters in length, and there must be a leading colon (:). An entry consisting of only a time specification applies to every day. The start hour of a time value must be less than the end hour.</p> <p>The <i>date</i> variable is a four digit string in the form <i>mmdd</i>. <i>mm</i> represents the calendar month and <i>dd</i> represents the day number. For example 0001 represents January 1. <i>dd</i> may be 00 to indicate the entire month, if the entry is not a range, or indicating the first or last day of the month depending on whether it appears as part of the start or end of a range. For example, 0000 indicates the entire month of January. 0600 indicates the entire month of June. 0311-0500 indicates April 11 through the last day of June.</p> <p>Entries in this list specify times that a user is allowed or denied access to the system. Entries not preceded by an ! (exclamation point) allow access and are called ALLOW entries. Entries prefixed with an ! (exclamation point) deny access to the system and are called DENY entries. The ! operator applies to only one entry, not the whole restriction list. It must appear at the beginning of each entry.</p>
mindiff	Defines the minimum number of characters required in a new password that were not in the old password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.
maxrepeats	Defines the maximum number of times a character can be repeated in a new password. Since a value of 0 is meaningless, the default value of 8 indicates that there is no maximum number. The value is a decimal integer string.

Schema attributes

Attributes	Description
minlen	Defines the minimum length of a password. The value is a decimal integer string. The default is a value of 0, indicating no minimum length. The maximum value allowed is 8. This attribute is determined by the minalpha attribute value added to the minother attribute value. If the sum of these values is greater than the minlen attribute value, the minimum length is set to the result.
histexpire	Designates the period of time (in weeks) that a user cannot reuse a password. The value is a decimal integer string. The default is 0, indicating that no time limit is set.
histsize	Designates the number of previous passwords a user cannot reuse. The value is a decimal integer string. The default is 0.
pwdchecks	Defines the password restriction methods enforced on new passwords. The value is a list of comma-separated method names and is evaluated from left to right. A method name is either an absolute path name or a path name relative to <code>/usr/lib</code> of an executable load module.
dictionlist	Defines the password dictionaries used by the composition restrictions when checking new passwords. The password dictionaries are a list of comma-separated, absolute path names that are evaluated from left to right. All dictionary files and directories must be write-protected from all users except root. The dictionary files are formatted one word per line. The word begins in the first column and terminates with a new-line character. Only 7-bit ASCII words are supported for passwords. If text processing is installed on your system, the recommended dictionary file is the <code>/usr/share/dict/words</code> file.
default_roles	Specifies the default roles for the user. The Value parameter, a comma-separated list of valid role names, can only contain roles assigned to the user in the roles attribute. You can use the ALL keyword to signify that the default roles for the user are all their assigned roles.
fsize	Identifies the soft limit for the largest file a user process can create or extend.
cpu	Sets the soft limit for the largest amount of system unit time (in seconds) that a user process can use.
data	Identifies the soft limit for the largest process data segment for a user process.
stack	Specifies the soft limit for the largest process stack segment for a user process.
core	Specifies the soft limit for the largest core file a user process can create.
rss	Sets the soft limit for the largest amount of physical memory a user process can allocate. This limit is not enforced by the system.
nofiles	Sets the soft limit for the number of file descriptors a user process may have open at one time.
stack_hard	Specifies the largest process stack segment for a user process.
roles	Contains the list of roles for each user.
time_last_login	Specifies the number of seconds since the epoch (00:00:00 GMT, January 1, 1970) since the last successful login. The value is a decimal integer.
tty_last_login	Specifies the terminal on which the user last logged in. The value is a character string.

Attributes	Description
host_last_login	Specifies the host from which the user last logged in. The value is a character string.
unsuccessful_login_count	Specifies the number of unsuccessful login attempts since the last successful login. The value is a decimal integer. This attribute works in conjunction with the user's loginretries attribute, specified in the /etc/security/user file, to lock the user's account after a specified number of consecutive unsuccessful login attempts. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's unsuccessful_login_count attribute to be less than the value of loginretries. To do this, enter the following: chsec -f /etc/security/lastlog -s username -a \\\nunsuccessful_login_count=0

Group attributes

The following table lists the group attributes:

Attributes	Description
users	Identifies a list of one or more users which are associated with group.
admin	Specifies whether administrative group or not.
registry	Specifies where the user or group identification information is administrated.
Group Name	Name of group
id	Group ID

Provisioning policy attributes

This section lists the different policy attributes of AIX Connector.

Account attributes

The following table lists the provisioning policy attributes for Create and Update Account:

Attributes	Description
User Name	(Only for Create Account) User ID on AIX computer that you want to use for connector operations.
id	User ID
pgrp	Primary group of user
home	Home directory of user
shell	Default shell of user

Provisioning policy attributes

Attributes	Description
login	Indicates whether the user can log in to the system with the login command. Possible values are: <ul style="list-style-type: none"> • true: The user can log in to the system. Default. • false: The user cannot log in to the system.
su	Indicates whether another user can switch to the specified user account with the su command. Possible values are: <ul style="list-style-type: none"> • true: Another user can switch to the specified account. Default • false: Another user cannot switch to the specified account.
rlogin	Permits access to the account from a remote location with the telnet or rlogin commands. Possible values are: <ul style="list-style-type: none"> • true: The user account can be accessed remotely. Default • false: The user account cannot be accessed remotely.
admin	Defines the administrative status of the user. Possible values are: <ul style="list-style-type: none"> • true: The user is an administrator. Only the root user can change the attributes of users defined as administrators. • false: The user is not an administrator. Default
sugroups	Lists the groups that can use the su command to switch to the specified user account. The <i>Value</i> parameter is a comma-separated list of group names, or a value of ALL to indicate all groups. An ! (exclamation point) in front of a group name excludes that group. If this attribute is not specified, all groups can switch to this user account with the su command.
admgroups	Lists the groups the user administers. The <i>Value</i> parameter is a comma-separated list of group names. For additional information on group names, see the adms attribute of the /etc/security/group file.
umask	Determines file permissions. This value, along with the permissions of the creating process, determines a file's permissions when the file is created. The default is 022.
default_roles	Specifies the default roles for the user. The <i>Value</i> parameter, a comma-separated list of valid role names, can only contain roles assigned to the user in the roles attribute. You can use the ALL keyword to signify that the default roles for the user are all their assigned roles.
Password	Initial password for newly created user account.

Group attributes

The following table lists the provisioning policy attributes for Create and Update Group:

Attributes	Description
Group Name	(Only for create group) Name of group
users	Identifies a list of one or more users which are associated with group.
Id	Group ID

Additional information

This section describes the additional information related to the AIX Connector.

Note: To enable logging, specify the logging

log4j.logger.openconnector.connector.unix.UnixConnector **and**
log4j.logger.openconnector.connector.unix.AIXConnector **in the**
log4j.properties **file.**

For example, log4j.logger.openconnector.connector.unix.UnixConnector=debug
log4j.logger.openconnector.connector.unix.AIXConnector=debug.

Upgrade considerations

When upgrading IdentityIQ to version 7.3, the **lastPasswordUpdate** schema attribute must be added manually to the application debug page as follows:

```
<AttributeDefinition name="lastPasswordUpdated" remediationModificationType="None"
type="string">
<Description>Specifies the time when user's password last updated.</Description>
</AttributeDefinition>
```

Unstructured Target Collector

AIX uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group **identityAttribute** for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint IdentityIQ User’s Guide*.

For AIX target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

AIX Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated Users and Groups. For UNIX platforms direct access means ownership of file or directory.

Table 1—Unstructured Target Configuration parameters

Attributes	Description	Possible values
Unix File System Path(s)*	Absolute path(s) which are to be scanned for resources.	Multiple paths can be mentioned with comma separated values. For example, /etc/tmp
Application Name*	Name of the application with which Unstructured Target will be correlated.	

Note: Attributes marked with * sign are the mandatory attributes.

Troubleshooting

Note: If Unstructured Configuration is configured before upgrading to version 7.3 from version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

Correlation Rule: The rule used to determine how to correlate account and group information from the application with identity cubes.

Note: For version 6.2 onwards, the default schema does not have correlation keys defined. Update correlation rule in Unstructured Target Configuration accordingly.

Provisioning related parameters

Select the settings for provisioning to the box.

- **Override Default Provisioning:** Overrides the default provisioning action for the collector.
- **Provisioning Action:** The overriding provisioning action for the collector.

Troubleshooting

1 - Test connection fails for managed systems

Test Connection fails for managed systems with the following error when SSH login prompt appears with some delay:

```
[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout occurred while setting the shell 'sh'
```

The above error occurs when connector tries to login to target managed system with SSH and execute the **sh** command. The **sh** command fails because of delay on target managed system for SSHLogin prompt to appear.

Resolution: To resolve this issue, tune the following time out parameters according to your need in the Application Debug page:

- **sshWaitTime:** Default value: 500 ms
This time out parameter is responsible for wait to receive command output.
- **SSHTimeOut:** Default value: 120000 ms
This time out parameter is responsible to tune maximum time for which a ssh command execution should be allowed. After this time out even if the command execution is in progress on target host, the connection will be dropped out and the operation will be timed out.

2 - Aggregation/test connection fails with timeout error

Aggregation/test connection fails with the following timeout error:

```
Exception during aggregation of Object Type account on Application <application name>
Reason: Unable to create iterator sailpoint.connector.TimeoutException:
[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout occurred while reading command response.
```

Test Connection fails with following timeout error:

[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout occurred while reading output stream for the executed command.

Resolution: Change the value of the **sshWaitTime (in millisecond)** application attribute as per your requirement in the debug page of the application:

```
<entry key="sshWaitTime" value="500"/>
```

3 - After target aggregation resources are not getting correlated with Account Groups

After target aggregation the resources are not getting correlated with Account Groups.

Resolution: Ensure that your correlation rule populates "Correlator.RULE_RETURN_GROUP_ATTRIBUTE" as follows:

```
....  
if ( isGroup ) {  
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE,"nativeIdentity");  
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);  
}  
....
```

4 - Test connection fails for key based authentication with an error

Test connection fails for key based authentication with the following error.

Login failed. Error while connecting to host:<hostname>. Cannot read key file.

Resolution: Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using open ssl. This method can be used for any version of SSH.
 - a. Create private key using the following command:


```
openssl <gendsa/genrsa> -des3 -out <private_key> 1024
```
 - b. Change the permission on the <private_key> file as follows:


```
chmod 0600 <private_key>
```
 - c. Create public key from private_key


```
ssh-keygen -y -f <private_key> > <public_key>
```
 - d. Use the <private_key> and <public_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
 - a. Create private and public key using the following command


```
ssh-keygen -t <dsa/rsa> -b 1024
```

 By default files with name `id_dsa/id_rsa` and `id_dsa.pub/id_rsa.pub` will be created.
 - b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:


```
openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3
```
 - c. Change the permission on the <new_private_key> file as follows:


```
chmod 0600 <new_private_key>
```
 - d. Create public key file using the new private key as follows:


```
ssh-keygen -y -f <new_private_key> > <new_public_key>
```
 - e. Use the <new_private_key> and <new_public_key> files for authentication.

Troubleshooting

5 - Test connection fails with an error when sudo user is configured for public key authentication

Test connection fails with the following error when sudo user is configured for public key authentication:

[InvalidConfigurationException] [Possible suggestions] a) Verify the private key file is correct for specified user. b) Verify the private key Passphrase is correct for specified user. c) Verify the private/public key file permissions are correct on the given unix host. [Error details] Failed to authenticate the ssh credentials for the user: to the host:

Resolution: Verify the sudo user's password specified in application configuration, password must be correct for certificate based authentication.

Chapter 3: SailPoint Azure Active Directory Connector

The following topics are discussed in this chapter:

Overview.....	51
Supported features	51
Pre-requisites	52
Administrator permissions	53
Configuration parameters.....	54
Additional configuration parameters	55
Schema attributes	56
Account attributes	56
Group attributes.....	57
Provisioning Policy attributes.....	58
Create Account Policy	58
Create Group Policy	59
Update Group Policy	59
Additional information	60
Managing licenses	60
Connector Reconfigure	60
Excluding Provisioning Policy attributes.....	60

Overview

SailPoint Azure Active Directory connector manages the users and groups in Windows Azure Active Directory. SailPoint Azure Active Directory Connector can be used to manage Users and Groups as Windows Azure Active Directory is the directory for all cloud based organizational Microsoft directory services including Microsoft Office365.

SailPoint Azure Active Directory connector can also be used to provision users into a federated domain in Azure Active Directory.

The SailPoint Azure Active Directory connector uses **Azure AD Graph API** to manage users, groups and licenses.

Supported features

The Azure Active Directory connector supports the following features:

- Account Management
 - Aggregation, Delta Aggregation, Get Account, Partitioning Aggregation, Pass Through Authentication
 - Create user in Azure Activity Directory,
 - Create user in a federated domain in Azure Active Directory.
 - Update, Delete users in Azure Active Directory
 - Enable\Disable users,

Overview

- Set password
- Add\Remove Entitlements:
 - Add\Remove individual license plans
 - Add\Remove license packs
 - Add\Remove Roles
 - Add\Remove user's group membership
- Account - Group Management
 - Aggregation, Delta Aggregation, Get operation for Security groups and Mail Enabled Security Groups
 - Create, Update Security Groups
 - Delete Security Groups, Mail Enabled Security Groups
- Other
 - Supports executing native before/after scripts for provisioning requests

This feature requires installation and registration of IQService. For more information, see “Installing and registering IQService” on page 555.

References

- “Appendix A: Delta Aggregation”
- “Appendix B: Partitioning Aggregation”

Pre-requisites

- To use Graph API, a client application must be registered on the Azure management portal. This application is responsible for calling Web APIs on behalf of the connector. The application's client ID and client secret key are required while configuring the application.

To register an application on Azure, perform the following:

- a. User can use any of the following Azure management portal to do the configuration:

<https://portal.azure.com>

OR

<https://aad.portal.azure.com>

- b. Select **Azure Active Directory** in the left pane.
- c. Click on **App registrations**.
- d. Click **New application registration**.
- e. Enter the required details as follows and click **Create**:

Fields	Description
Name*	Name for the application. For example, SailPointAzureADManagement
Application Type	Select Web application and/or Web API.

Fields	Description
Sign-On URL*	<p>Enter the base URL. For example, https://localhost:43434</p> <p>For Application ID URL, enter url as, http://domainName/GraphWebapp.</p> <p>Note: Connector does not use these URLs, the above value are just place holders and do not impact connector functionality.</p>

- f. Retrieve the Application ID (Client ID) and Key (Secret Key) using the following steps:
 - From Settings, copy **Application ID**.
 - To copy Secret Key use following steps:
 - Open the application.
 - Navigate to **Settings** and click on **Keys**.
 - Provide a key description and validity duration and click **Save**.
 - Copy the displayed key.
- Note: The displayed key would be displayed only once.**
- To enable Pass Through Authentication for an existing and new applications, add the following configuration attributes in the application:
 - a. Add **AUTHENTICATE** in the application's **featureString**.
 - b. Add **authSearchAttributes** configuration attribute as follows:

```

<entry key="authSearchAttributes">
  <value>
    <List>
      <String>userPrincipalName</String>
    </List>
  </value>
</entry>

```

Administrator permissions

Following permissions must be granted to the client application created in Azure:

- Read Directory data
- Read and Write Directory data

To grant permissions to the client application:

- Navigate to client application which is created using above mentioned steps.
- Navigate to **Setting** and select **Required Permissions** option.
- Click on **Windows Azure Active Directory** option under **API** tab.
- Under **APPLICATION PERMISSION** provide permission to **Read directory data** and **Read and write directory data**.
- Save application.
- Click on **Grant Permissions** in **Required permissions**.

Configuration parameters

- Click on **Yes** button in the pop up window.
- Assign **User Account Management** role to the application using **Windows Azure Active Directory Module for Windows PowerShell**. This role is required to delete users.

Note: If client application is configured using the following portal, then ensure to click on 'Grant Permissions' option under 'Required permissions' option of 'API Access' section:
<https://portal.azure.com>
OR
<https://aad.portal.azure.com>

Above permissions do not allow connector to manage users with administrative roles. To manage such users, the application created on Azure must have **Company Administrator** role assigned. This role can be assigned via PowerShell commands. Following are the prerequisite for executing the PowerShell commands.

Note: These prerequisites are not required for the connector to function. These can be installed on any system for temporary use to give required role to the application on Azure.

- Microsoft Online Services Sign-In Assistant for IT Professionals RTW
- Windows Azure Active Directory Module for Windows PowerShell

After installing the pre-requisites, open **Windows Azure Active Directory Module for Windows PowerShell** console and execute the following commands:

- Connect-msolservice, press enter, provide Azure administrator credentials.
- Execute `Get-MsolServicePrincipal | ft DisplayName, AppPrincipalId -Autosize`
- Locate your application name and copy the **ObjectId** value.
- Execute `$ClientObjID = <copied objectId of the application in the previous step>`
- Execute `$webApp = Get-MsolServicePrincipal -AppPrincipalId $ClientObjID`
- Execute `Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType ServicePrincipal -RoleMemberObjectId $webapp.ObjectId`

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Azure Active Directory connector uses the following connection parameters:

Attributes	Description
Azure AD application client ID*	ID of the application created on the Azure Active Directory for using Graph REST API.
Azure AD application client secret key*	Client secret of the Azure Active Directory application.
Azure AD domain name*	Name of the Azure Active Directory domain to be managed. For example, contoso.onmicrosoft.com
Page Size	Number of records per page. Default: 500
IQService Host	Host name of the system where IQService is installed.
IQService Port	Port number used by the IQService.

Note: Attributes marked with * sign are the mandatory attributes

Note: To enable native before/after script execution for provisioning requests, IQService Host and IQService Port parameters must be configured.

Additional configuration parameters

Attributes	Description
createAccountTimelag	<p>Time in seconds to wait after create account and before calling get account. Default: 20 seconds</p> <p>For example, <code><entry key="createAccountTimelag" value="20"></code></p>
maxReadTimeout	<p>Time in seconds to wait for getting response from the REST call, in the read operation, before operation gets timed out. Default: 180 seconds.</p> <p>For example, <code><entry key=" maxReadTimeout" value="200"></code></p>
maxRetryCount	<p>Indicates the number of time read operation must be retried on retry errors for the read operations. Default: 5</p> <p>For example, <code><entry key=" maxRetryCount" value="6"></code></p>
retryableErrorsOnAgg	<p>List of error which must be retried if occurred during aggregation or get operation. Type: List of strings</p>
userPartitions	<p>List of filters to be applied during account aggregation to limit set of data.</p> <p>For more information, see "" on page 544.</p>
userFilters	<p>Filter that defines the scoping condition for Accounts to be applied during account aggregation to limit set of data.</p> <p>Following is an example for userFilters configuration attribute:</p> <pre><entry key="userFilters" value="(startswith(displayName, 'A') and accountEnabled eq true)"/></pre>
groupFilters	<p>Filter that defines the scoping condition for Groups to be applied during account-group aggregation to limit set of data.</p> <p>Following is an example for groupFilters configuration attribute:</p> <pre><entry key="groupFilters" value="(startswith(displayName, 'A') or startswith(displayName, 'B'))"/></pre>
partitionHost	<p>(Applicable only for partitioned account delta aggregation only) The partitionHost is the host on which all delta partitions must be executed.</p> <p>Following is an example for partitionHost configuration attribute:</p> <pre><entry key="partitionHost" value= "myhost"></pre>

Schema attributes

This section describes the different schema attributes.

Note: In addition to the schema attributes listed in the following tables, the connector supports managing the extended attributes that are registered on the client application on Azure.

Account attributes

The following table lists the account attributes:

Name	Description
accountEnabled	True if the account is enabled; otherwise, false
assignedLicenses	List of the licenses that are assigned to the user
assignedPlans	Plans that are assigned to the user (Entitlement).
city	City in which the user is located.
country	Country/region in which the user is located.
department	Name for the department in which the user works.
dirSyncEnabled	Indicates whether this object was synced from the on-premises directory.
disabledPlans	Plans that are not assigned to user.
displayName	Name displayed in the address book for the user.
facsimileTelephoneNumber	Telephone number of the user's business fax machine.
givenName	First name of user.
groups	Groups assigned to a user (Entitlement).
immutableId	Property used to associate an on-premises Active Directory user account to their Azure AD user object.
jobTitle	User's job title.
lastDirSyncTime	Indicates the last time at which the object was synchronized with the on-premises directory.
mail	The SMTP address for the user. For example, john@contoso.onmicrosoft.com
mailNickname	The mail alias for the user.
manager	Manager of the user. (Type: String) By default this attribute is not added to the schema for performance optimization.
mobile	Primary cellular telephone number for the user.
objectId	Unique identifier for the user.
onPremisesSecurityIdentifier	Contains the on-premises security identifier (SID) for the user that was synchronized from on-premises to the cloud.

Name	Description
otherMails	A list of additional email addresses for the user.
passwordPolicies	Specifies password policies for the user.
physicalDeliveryOfficeName	Office location in the user's place of business.
postalCode	ZIP OR postal code for the user's postal address.
preferredLanguage	Preferred written or spoken language for a person.
proxyAddresses	Proxy addresses. For example, ["SMTP: bob@contoso.com", "smtp: bob@sales.contoso.com"]
roles	Administrator Role assigned to user (Entitlement).
sipProxyAddress	Specifies the voice over IP (VOIP) session initiation protocol (SIP) address for the user.
state	The state or province in the user's address.
streetAddress	The street address of the user's place of business.
surname	Last name of the user.
telephoneNumber	Primary telephone number of the user's place of business.
usageLocation	A two letter country code indicating usage location.
userPrincipalName	The user principal name (UPN) of the user.
userType	Type of the user.

Group attributes

Name	Description
description	Description for the group.
dirSyncEnabled	Indicates whether this object was synced from the on-premises directory.
displayName	Display name for the group.
lastDirSyncTime	Indicates the last time at which the object was synced with the on-premises directory.
mail	SMTP address for the group.
mailEnabled	Specifies whether the group is mail-enabled
mailNickname	The mail alias for the group.
objectId	Group ID.
onPremisesSecurityIdentifier	Contains the on-premises security identifier (SID) for the group that was synchronized from on-premises to the cloud.
owners	Owner of the group. By default not present in the schema, Type: String, Multi-Valued
proxyAddresses	Proxy addresses of the group.

Provisioning Policy attributes

Name	Description
securityEnabled	Specifies whether the group is a security group.

Provisioning Policy attributes

This section lists different policy attributes for Azure Active Directory Connector.

Note: The attributes marked with * sign are required attributes.

Create Account Policy

Following table describes various attributes in the create account policy.

Attribute	Description
userPrincipalName*	user principal name (UPN) of the user. For example, jeff@contoso.onmicrosoft.com
password*	Password for the new user.
displayName*	Display name of the user.
mailNickname*	Mail alias for the user.
accountEnabled	Set it to false to create disabled account. Default: True
forceChangePasswordNextLogin	If true, asks user to change password on next login. Default: True
department	Department in which the user works.
jobTitle	User's job title.
isFederatedDomain	Set it true to create federated domain user. If this is checked and immutableId is not set then random immutableId value will be used.
immutableId	This property is used to associate an on-premises Active Directory user account to their Azure AD user object; Populate this attribute with objectGUID of account from on-premises Active Directory to create federated user synchronized with on –premises Active Directory user.
passwordPolicies	Specifies password policies for the user For example: DisablePasswordExpiration, DisableStrongPassword
otherMails	Additional email addresses for the user.
givenName	First name of the user.
surname	Surname of the user.
usageLocation	A two letter country code (ISO standard 3166). Required for users that will be assigned licenses.

Attribute	Description
country	The country/region in which the user is located. For example, US or UK
state	The state or province in the user's address.
city	The city in which the user is located.
streetAddress	The street address of the user's place of business.
postalCode	The postal code for the user's postal address.
physicalDeliveryOfficeName	The office location in the user's place of business.
preferredLanguage	Preferred language for the user. Should follow ISO 639-1 Code. For example, en-US
telephoneNumber	Primary telephone number of the user's place of business.
mobile	Primary cellular telephone number for the user.
facsimileTelephoneNumber	Telephone number of the user's business fax machine.
userType	A string value that can be used to classify user types in your directory, such as Member and Guest .

Create Group Policy

Following table describes various attributes in the create group policy.

Attribute	Description
displayName*	Display name of the group.
mailNickname*	The mail alias for the group.

Update Group Policy

Following table describes various attributes in the update group policy.

Attribute	Description
description	Description of the group.
owners	Owner of the group. Read only.
mailEnabled	True if it is mail enabled security group. Read only.

Additional information

This section describes the additional information related to the Azure Active Directory Connector.

Managing licenses

Azure Active Directory Connector supports assigning different Azure services licenses to the users. Connector provides options to assign license either by individual plan or as a whole license pack.

- **Assigning license plan:** Office 365 license pack consist of licenses for individual services. For example, Exchange Online, SharePoint Online and so on.
- The connector models **assignedPlans** attribute from account schema as an entitlement. It can be requested as an entitlement during **Create** or **Update** operations for Identities.
- **Assigning license pack:** To assign license pack, set **assignedLicenses** attribute from account schema as **Managed, Entitlement, Multi-Valued**, So that it request able as an entitlement.

Note: It is recommended that 'assignedPlans' or 'assignedLicenses' must be promoted as an entitlement to avoid conflicts.

Note: To provision the licenses or plans to user, set the user's 'usageLocation' property correctly.

Connector Reconfigure

Existing Microsoft Office365 application can be reconfigured to Azure Active Directory application to preserve the data present in the IdentityIQ.

Excluding Provisioning Policy attributes

To skip plan attributes from getting processed by Azure Active Directory Connector, add **excludeAttributesFromProvisioning** attribute to the application debug page with value listing names of such attributes.

For example,

```
<entry key="excludeAttributesFromProvisioning">
    <value>
        <List>
            <String>city</String>
        </List>
    </value>
</entry>
```

Chapter 4: SailPoint BMC Remedy Connector

The following topics are discussed in this chapter:

Overview.....	61
Supported features	61
Supported Managed Systems	62
Pre-requisites	62
Administrator permission	62
Configuration parameters.....	62
Schema attributes	62
Account attributes	62
Group attributes.....	63
Provisioning policy attributes.....	64
Create account attributes	64
Create group attributes	64
Update policies.....	64
Additional information	64
Enable/Disable Account.....	65
Troubleshooting.....	65

Overview

SailPoint BMC Remedy Connector manages the accounts and groups contained in BMC Remedy Action Request System.

Supported features

SailPoint BMC Remedy Connector supports the following features:

- Account Management
 - Manage BMC Remedy Users as Accounts
 - Aggregation, Refresh Account, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manage BMC Remedy Groups as Account - Groups
 - Aggregation, Refresh Group
 - Create, Update, Delete

Configuration parameters

References

- “Enable/Disable Account” on page 65

Supported Managed Systems

- BMC Remedy Action Request System Server version 18.05
- BMC Remedy Action Request System Server version 9.1
- BMC Remedy Action Request System Server version 9.0

Pre-requisites

- You must copy the `arapi<v>.jar` file from the location where the server is installed (`install-Folder\BMC Software\ARSystem\midtier\WEB-INF\lib`) to the lib folder of the connector installation (`\webapps\identityiq\WEB-INF\lib`).
- Add the location of `arapi<v>.jar` file to the CLASSPATH system variable (`\webapps\identityiq\WEB-INF\lib\arapi<v>.jar`) of the computer where IdentityIQ installed.
- Provide the appropriate read and write permissions to the Administrator to perform the user and group provisioning operations.

Administrator permission

The Application User should be a member of the **Administrator** group.

Configuration parameters

The following table lists the configuration parameters of BMC Remedy Connector:

Parameters	Description
Remedy Server name or IP Address	IP address of the computer on which the Remedy server is installed.
Administrator Name	Name of the Remedy administrator.
Administrator Password	Password of the administrator.
Server Port	Remedy Server port number.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
RequestID	RequestID of the user.
LoginName	Remedy login name.
ForcePasswordChangeOnLogin	Set to Yes if the user should be asked to change his password on next login else to No.
FullName	Full name of the user.
Status	Status of the user.
AccountDisabledDate	Account disabled date of user.
ApplicationLicense	Application license of user.
AppliedDaysAfterExpirationUntilDisablement	Applied days after expiration until disablement of user.
AppliedNewUserMustChangePassword	Is set to Yes if the new user must change password.
AppliedNo.DaysbeforeExpiration	The number of days before expiration.
AppliedNumberofWarningDays	The number of warning days.
AppliedPasswordEnforcementEnabled	Is set to Yes if password enforcement is enabled.
Creator	Creator of the user.
LastModifiedBy	Name of the user who last modified the user.
LicenseType	License Type of user.
UniquelIdentifier	Unique Identifier of the user.
Groups	Groups connected to the user.

Group attributes

The following table lists the group attributes:

Attributes	Description
RequestID	RequestID of group.
Comments	Comments about group.
GroupCategory	Category of group.
GroupID	ID of group.
GroupName	Name of the group.
GroupType	Type of group.
LastModifiedBy	Name of the user who last modified the group.
LongGroupName	Long name of the group.
ParentGroup	Parent group of the group.
Status	Status of the group.
UniquelIdentifier	Unique Identifier of group.

Provisioning policy attributes

This section lists the different policy attributes of BMC Remedy Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

Attributes	Description
Login Name	Remedy login name of the user.
Full Name	Full name of the user.
Force Password Change On Login	Is set to Yes if the user should be asked to change his password on next login.
License Type	License Type of the user.
Password	Password of the user.

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

Attributes	Description
Group Name	Name of the group to be created.
Group ID	ID of the group. It should be a numeric value.
Group Type	Type of the group.
Long Group Name	Long name of the group.
Group Category	Category of the group. If the category of the group is Computed , ComputedGroupDefinition needs to be added in the provisioning policy.

Update policies

The following table lists the attributes for enable/disable a user:

Attributes	Description
ResetPassword	The new password to be set.

Additional information

This section describes the additional information related to the BMC Remedy Connector.

Enable/Disable Account

For disabling a user, a password not known to the user should be provided by the administrator. The **Status** attribute of the user will be set to **Disabled**. All users which have a status other than **Current** will be marked as **Disabled**.

For enabling a user, a password should be provided by the administrator which can be communicated to the user after successful password change. The **Status** attribute of the user will be set to **Current**.

Troubleshooting

- When an attribute is to be added to the schema, the attributes ID should be added as an **internalName** of the attribute in the schema.
- When an attribute (which is not present in the schema) is to be added to the provisioning policy, the ID of the attribute should be provided as the **name** of the attribute.
- While creating a Remedy Group having GroupType value **Computed**, ensure that the **ComputedGroupDefinition** attribute is added to the provisioning policy.

For example, `<Field displayName="ComputedGroupDefinition" name="121" type="string"/>`

Troubleshooting

Chapter 5: SailPoint BMC Remedy IT Service Management Suite Connector

The following topics are discussed in this chapter:

Overview.....	67
Supported features	67
Supported Managed Systems	68
Pre-requisites	68
Administrator permission	68
Configuration parameters.....	68
Schema attributes	69
Account attributes	69
Group attributes.....	70
Provisioning policy attributes.....	70
Create account attributes	70
Create group attributes	71
Update policies.....	72
Additional information	72
Enable/Disable Account.....	72
Add Entitlement operation for ITSM	72
Troubleshooting.....	73

Overview

SailPoint BMC Remedy IT Service Management Suite Connector manages the accounts and groups contained in a BMC Remedy IT Service Management Suite (ITSM).

Supported features

SailPoint BMC Remedy ITSM Connector supports the following features:

- Account Management
 - Manage BMC Remedy ITSM Users as Accounts
 - Aggregation, Refresh Account, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add/Remove Entitlements

Configuration parameters

- Account - Group Management
 - Manage BMC Remedy ITSM Support Groups as Account - Groups
 - Aggregation, Refresh Group
 - Create, Update, Delete

References

- “Add Entitlement operation for ITSM” on page 72
- “Enable/Disable Account” on page 72

Supported Managed Systems

- BMC Remedy IT Service Management Suite version 18.05
- BMC Remedy IT Service Management Suite version 9.1
- BMC Remedy IT Service Management Suite version 9.0

Pre-requisites

1. You must copy the **arapi<v>.jar** file from the location where the server is installed (*installFolder\BMC Software\ARSystem\midtier\WEB-INF\lib*) to the lib folder of the connector installation (*\webapps\identityiq\WEB-INF\lib*).
2. Add the location of **arapi<v>.jar** file to the CLASSPATH system variable (*\webapps\identityiq\WEB-INF\lib\arapi<v>.jar*) of the computer where IdentityIQ installed.
3. Provide the appropriate read and write permissions to the Administrator to perform the user and group provisioning operations.

Administrator permission

The Application User must be a member of the **Administrator** group.

Configuration parameters

The following table lists the configuration parameters of Remedy ITSM Connector:

Parameters	Description
Remedy Server name or IP Address	IP address of the computer on which the Remedy ITSM server is installed.
Administrator Name	Name of the Remedy ITSM administrator.
Administrator Password	Password of the administrator.
Server Port	Remedy ITSM Server port number.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
PersonID	PersonID of the user
RemedyLoginID	Remedy Login Name
FirstName	First name of the user
LastName	Last name of the user
InternetEmail	Internet Email of user
Status	Status of the user
AccountingNumber	Accounting Number of user
ClientSensitivity	Sensitivity of client
ClientType	Type of Client
Company	Company of user
CorporateID	Corporate ID of user
BusinessPhoneNumber	Business Phone Number of the user
FullTextLicenseType	Full Text License Type of user
JobTitle	Job Title of user
LastModifiedBy	Name of the user who last modified the user attributes
LicenseType	Type of License
Region	Region information
Site	Site information
SiteAddress	Site Address information
SiteGroup	Site Group information
Submitter	Name of the submitter
SupportStaff	If user is part of Support Staff
VIP	If user is VIP
UnrestrictedAccess	If user has unrestricted access
Groups	Groups connected to user

Group attributes

The following table lists the group attributes:

Attributes	Description
SupportGroupID	Support Group ID
Company	Support Company information
Description	Group description
DisableGroupNotification	If group notification is disabled
GroupNotificationEmail	Group notification email id
instanceId	Instance id of group
LastModifiedBy	Name of the user who last modified the group
Status	Status of the group
Creator	Creator of the group
SupportGroupName	Name of the group
SupportGroupRole	Support Group role name
SupportOrganization	Support Group organization name
UsesOLA	If the group uses OLAs
UsesSLA	If the group uses SLAs
VendorGroup	If group is a Vendor Group
OnCallGroup	If group is a On Call Group

Provisioning policy attributes

This section lists the different policy attributes of Remedy ITSM Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Accounts:

Attributes	Description
FirstName	First name of the user
LastName	Last name of the user

Attributes	Description
ClientType	Type of Client. Following are the allowed values: <ul style="list-style-type: none"> • Office-Based Employee • Field-Based Employee • Home-Based Employee • Contractor • Customer • Prospect • Vendor
ClientSensitivity	Sensitivity of Client. Following are the allowed values: <ul style="list-style-type: none"> • Sensitive • Standard
VIP	Following are the allowed values: <ul style="list-style-type: none"> • Yes • No
Company	Company name of the user
BusinessPhoneNumber	Business phone number of the user
RemedyLoginID	Remedy login ID
Password	Remedy Password for the login ID
SupportStaff	Following are the allowed values: <ul style="list-style-type: none"> • Yes • No <p>If value is Yes, AssignmentAvailability attribute must be added. Allowed values: Yes or No.</p>
UnrestrictedAccess	Following are the allowed values: <ul style="list-style-type: none"> • Yes • No

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

Attributes	Description
SupportCompany	Support Company name of group
SupportOrganization	Support organization of group
SupportGroupName	Support group name
SupportGroupRole	Support Group role name

Additional information

Attributes	Description
VendorGroup	Following are the allowed values: <ul style="list-style-type: none">• Yes• No
OnCallGroup	Following are the allowed values: <ul style="list-style-type: none">• Yes• No

Update policies

The following table lists the attributes for different update policies:

Attributes	Description
Enable/Disable a user	
ResetPassword	The new password to be set.
Create an ITSM Account and Group Connection	
AC_1000000017	Full name of the user.
AC_4	Remedy Login ID of the user.
AC_1000000401	Support Group Association Role name.

Note: The connection attributes should have 'AC_' prefixed to the field id of the attribute.

Additional information

This section describes the additional information related to the BMC Remedy ITSM Suite Connector.

Enable/Disable Account

For disabling a user, a password not known to the user should be provided by the administrator. The **Profile Status** attribute of the user will be set to **Obsolete**. All users which have a status other than **Enabled** will be marked as **Disabled**.

For enabling a user, a password should be provided by the administrator which can be communicated to the user after successful password change. The **Profile Status** attribute of the user will be set to **Enabled**.

Add Entitlement operation for ITSM

To add a user to a group in BMC Remedy ITSM, there are some mandatory attributes to be provided which are a part of the connection between the user and the group. Hence, for Remedy ITSM, an entitlement will have mandatory attributes which will be a part of the update provisioning policy. All entitlements added will have the same connection attributes.

Troubleshooting

- When an attribute is to be added to the schema, the attributes ID should be added as an **internalName** of the attribute in the schema.
- When an attribute (which is not present in the schema) is to be added to the provisioning policy, the ID of the attribute should be provided as the **name** of the attribute.
- For connection attributes, ensure that the ID of the attribute is prefixed with **AC_**.
- While creating an ITSM Account having SupportStaff value **Yes**, ensure that the **AssignmentAvailability** attribute is added to the provisioning policy.

For example,

```
<Field displayName="AssignmentAvailability" name="1000000346"
reviewRequired="true" type="string">
    <AllowedValues>
        <String>Yes</String>
        <String>No</String>
    </AllowedValues>
</Field>
```

- For account creation in BMC Remedy ITSM Suite version 7.5.00.001 required mandatory attribute **InternetEmail**.

Troubleshooting

Chapter 6: SailPoint Box Connector

The following topics are discussed in this chapter:

Overview.....	75
Supported features	75
Supported Managed Systems	76
Pre-requisites	76
Administrator permissions	76
Configuration parameter.....	76
Schema attributes	77
Account attributes	77
Group attributes.....	78
Provisioning Policy attributes.....	78
Troubleshooting.....	79

Overview

The Box Connector manages enterprise users and groups of Box server. Box Connector is a read/write connector that can retrieve enterprise users and groups of a particular network, activates/inactivates the users and assign enterprise user memberships to groups.

Supported features

SailPoint Box Connector provides support for the following features:

- Account Management
 - Manage Box Users as Accounts
 - Aggregate, Refresh Account
 - Create, Update, Delete
 - Enable, Disable
 - Add/Remove Entitlements
- Account - Group Management
 - Manage Box Groups as Account Group
 - Aggregate, Refresh Group
 - Create, Update, Delete

Supported Managed Systems

SailPoint Box Connector supports the following managed system:

- BOX API Version 2.0 supported by Box Server

Pre-requisites

- Box Connector application must support **OAuth 2.0 with JWT (Server Authentication)**. Perform the following steps to provide support for OAuth 2.0 with JWT:
 - a. Go to <https://app.box.com/developers/console> and select your application and the configuration.
 - b. Under Authentication Method select **OAuth 2.0 with JWT (Server Authentication)**.
 - c. Under Add and Manage Public Keys section click on **Add a Public Key**.
 - d. Upload the generated public key (Public Key ID).
- Generate public/private key pair using the following respective commands:

Note: These commands can be executed on Windows and Linux system.

 - Private Key: `openssl genrsa -aes256 -out private_key.pem 2048`
 - Public Key: `openssl rsa -pubout -in private_key.pem -out public_key.pem`

Note: Above command provides the keys with 256 bit encryption. If encryption is set to more than 128 bits, replace `local_policy.jar` and `US_export_policy.jar` of JRE with the latest jars available on <http://www.oracle.com/technetwork/java/javase/downloads>.
- On Weblogic Server update `java.security` file (located in `C:\Program Files\Java\jre1.8.0_111\lib\security`) and replace the value of `security.provider.1` to `security.provider.1=com.rsa.jsafe.provider.JsafeJCE`

Note: If Box Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

Administrator permissions

The administrator should be able to create and retrieve enterprise users, enterprise user memberships, and Box groups.

The Box application created on box system must have **Manage an enterprise** scope selected as the permission.

Note: For a Box application existing prior to version 6.2 patch 2, the corresponding application must be deleted. A new application must be created by selecting the application type as “Box”. After the application is configured the aggregation must be performed.

Configuration parameter

This section contains the information that this connector uses to connect and interact with the application.

The Box Connector uses the connection attributes listed in the following table:

Parameters	Description
Enterprise Id	Box Enterprise ID for a token specific to an enterprise.
Public Key Id	Box Public Key ID generated by Box and provided upon submission of a Public Key (as mentioned in the “Pre-requisites” on page 76).
Client Id	Box Server application Id.
Client Secret	Box application secret key.
Private Key	Private key text used for encrypting the JWT assertion.
Private Key Password	Password for decrypting private key.
Page Size	BoxNet page size.

Schema attributes

The application schema is used to map Box server user or group objects to IdentityIQ account and group objects. The following Box managed user and group object attributes are mapped to Account and Group objects respectively.

Account attributes

The following table lists the account attributes ([Table 1—Account attributes](#)):

Table 1—Account attributes

Attributes	Description
id	User ID as assigned by Box server.
name	Name of the enterprise user.
login	Email ID used to login.
role	User role as co-administrator/user.
memberof	Member of
space_amount	Space allocated to the user in GigaBytes.
max_upload_size	The maximum individual file size in bytes this user can have.
is_sync_enabled	Whether the user can use synchronization as Yes/No.
can_see_managed_users	Whether the user can see other users in the enterprise in their contact lists as Yes/No.
is_exempt_from_device_limits	Whether to exempt this user from Enterprise device limits as Yes/No.
is_exempt_from_login_verification	Whether or not this user must use two-factor authentication.
job_title	The user’s job title displayed on their profile page.

Provisioning Policy attributes

Table 1—Account attributes (Continued)

Attributes	Description
phone	The user's phone number displayed on their profile page.
address	User address.
language	Two letter representation of user language. for example, en for English.
status	User is enabled or disabled.
enterprise	User enterprise.

Group attributes

The following table lists the group attributes ([Table 2—Group attributes](#)):

Table 2—Group attributes

Attributes	Description
group_id	Group ID as assigned by Box server.
group_name	Group name.

Box group membership and group access: Whenever an entitlement is requested for a user, by default the Box connector adds a user into a group with **member** as the access value. This is true for all entitlement requests.

Note: To disable the fetching of group members in account aggregation, set the value of the following attribute to true:

```
<entry key="disable_group_membership" value="true"/>
```

By default the value of this attribute is false.

Note: From IdentityIQ version 7.3 onwards, if user does not want to fetch Account - Group membership in account aggregation then delete 'memberof' (group membership) attribute from account schema.

Provisioning Policy attributes

The following table lists the provisioning policy attributes ([Table 3—Provisioning Policy attributes](#)):

Table 3—Provisioning Policy attributes

Attributes	Description
name	Name for the enterprise user.
role	Box user role as co-administrator/user.
Login Id	login ID of the user.
space_amount	Space to be allocated to the user in GigaBytes.
Inactive account	Inactive the user account.
Unlimited Storage	Unlimited space amount.

Additional information

This section describes the additional information related to the Box Connector.

Upgrade considerations

When upgrading to IdentityIQ version 7.3 delete the Refresh Token entry from the application debug page to support JWT Authentication on Box Connector.

Troubleshooting

1 - During enable/disable of accounts the following error message appears

During enable/disable of accounts the following error message appears:

openconnector.ConnectorException: api.box.com

Resolutions: Verify the network connections.

2 - When performing the Test Connection on WebLogic Server an error message appears

When performing the Test Connection on WebLogic Server after upgrading to IdentityIQ version 7.3, the following error message appears:

SSLWLSWildcardHostnameVerifier

Resolution: The default host name verifier for WebLogic Server does not support SSL certificates where the CN contains a wildcard for the hostname.

Perform the following to change WebLogic's hostname verifier, and WebLogic ships with a class that can verify CNs with wildcards:

1. Navigate to **WebLogic Administrator Console ==> Environment ==> Servers ==> Customer server ==> Configuration ==> SSL** and click on **Lock & Edit**.
2. Open the **Advanced** tab and change **Hostname Verification** from **BEA Hostname Verifier** to **Custom Hostname Verifier**.
3. Set **Custom Hostname Verifier** to **weblogic.security.utils.SSLWLSWildcardHostnameVerifier**.
4. Click **Save** and then **Activate Changes**.
5. Restart the server.

Troubleshooting

Chapter 7: SailPoint DB2 Windows Connector

The following topics are discussed in this chapter:

Overview.....	81
Supported features	82
Supported Managed Systems	82
Pre-requisites	82
Administrator permissions	82
Configuration parameters.....	83
Schema Attributes	83
Account attributes	83
Roles attributes	84
Provisioning Policy attributes	85
Additional information	85
Create user	86
Delete user	86
Delete Role	86
Troubleshooting.....	86

Overview

IBM DB2 is a relational model database server developed by IBM. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications. Following are the main products of the DB2 family:

- DB2 for LUW (Linux, Unix, and Windows)
- DB2 for z/OS (mainframe)
- DB2 for iSeries (formerly OS/400)

The DB2 LUW product runs on multiple Linux, UNIX distributions (such as Red Hat Linux, SUSE Linux, AIX, HP/UX, and Solaris) and Windows systems. DB2 also powers the IBM InfoSphere Warehouse edition, which is DB2 LUW with DPF (Database Partitioning Feature), a massive parallel share-nothing data warehousing architecture.

Basically DB2 server manages all access on DB2 server for Windows users and the Windows authentication are also used to login to DB2 server.

Note: SailPoint DB2 Connector supports DB2 Windows flavor only.

SailPoint DB2 Windows Connector manages the following entities of DB2 Server:

- Database Users
- Roles

Supported features

SailPoint DB2 Windows Connector supports the following features:

- Account Management
 - Manage DB2 Users as Accounts
 - Aggregate, Refresh Account, Discover Schema
 - Create, Update, Delete
 - Add/Remove Entitlement
- Account - Group Management
 - Manage DB2 Roles as Account Group
 - Aggregate, Refresh Group
 - Delete
- Permissions Management
 - Permissions directly assigned to accounts and groups as direct permissions during account and group aggregation.
 - Automated revocation of the aggregated permissions.

Supported Managed Systems

SailPoint DB2 Windows Connector supports the following versions of DB2 Server:

- IBM DB2 Enterprise Server version 11.1
- IBM DB2 Enterprise Server version 10.5
- IBM DB2 Enterprise Server version 10.1

Pre-requisites

The compatible DB2 Windows JDBC drivers must be used in the classpath of IdentityIQ for connecting to DB2 Server. For example, db2jcc.jar

Administrator permissions

The Administrator login must have the SYSADM (Authority) as the minimum privilege and must be able to perform the following operations on DB User and Roles:

- Search
- Create
- Update
- Delete

Note: To run the **CREATE ROLE rolename** and **DROP ROLE rolename** query in the following DB2 versions, the respective specified authorities are required:

- 10.1 and 10.5: SECADM, SYSCTRL, or SECADM authority

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The DB2 Windows Connector uses the following connection attributes:

Attribute	Description
URL*	A valid URL of DB2 Server in the following format: jdbc:db2:// [serverName] [:portNumber] / [databaseName] where: <ul style="list-style-type: none">• jdbc:db2:// is known as the sub-protocol and is constant.• serverName: address of the server to connect to. This could be a DNS, IP address, a localhost, or 127.0.0.1 for the local computer.• portNumber: is the port to connect to on <i>serverName</i>. Default is 50000.• databaseName: is the database to which you want to connect.
User*	A Domain or Local login to the operating system through which you want to connect the DB2 Server. The login name should have minimum privileges to log in to the DB2 Server.
Password*	Authentication details of login and should have a valid password of that login.
JDBC Driver*	Name of the driver class supported by JDBC. For example, com.ibm.db2.jcc.DB2Driver

Schema Attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attribute name	Description
GRANTEE	Database user name.
GRANTEETYPE	Database user Type. U=grantee is an individual user.
GRANTOR	Grantor of the authority.
GRANTORTYPE	S=Grantor is system U=Grantor is an individual
BINDADDAAUTH	Authority to create packages.

Schema Attributes

Attribute name	Description
CONNECTAUTH	Authority to connect to the database. N=Not held Y=Held
CREATETABAUTH	Authority to create tables. N=Not held Y=Held
DBADMAUTH	DBADM authority. N=Not held Y=Held
EXTERNALROUTINEAUTH	Authority to create external routines. N=Not held Y=Held
IMPLSCHEMAAUTH	Authority to implicitly create schemas by creating objects in non-existent schemas. N=Not held Y=Held
LOADAUTH	Authority to use the DB2 load utility. N=Not held Y=Held
NOFENCEAUTH	Authority to create non-fenced user-defined functions. N=Not held Y=Held
QUIESCECONNECTAUTH	Authority to access the database when it is quiesced. N=Not held Y=Held
SECURITYADMAUTH	Authority to monitor and tune SQL statements. N=Not held Y=Held
roles	Roles connected to user.

Roles attributes

The following table lists the Roles attributes:

Attribute name	Description
ROLENAMES	Name of the role.
ROLEID	Identifier for the role.

Attribute name	Description
CREATE_TIME	Time when the role was created.
HierarchicalRoles	<p>List of inherited roles.</p> <p>Note: The hierarchical roles display the child roles of parent roles in the Group object properties of Entitlement Catalog. For existing applications which are getting upgraded, mark entitlement as true to display the child roles in Entitlement grid of Group object properties.</p>
AUDITPOLICYID	Identifier for the audit policy.
AUDITPOLICYNAME	Name of the audit policy.
REMARKS	User-provided comments or null.

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Account:

Attribute name	Description
GRANTEE*	Database user name.
CONNECTAUTH	Authority to connect to the database.
BINDADDAUTH	Authority to create packages.
CREATETABAUTH	Authority to create tables.
NOFENCEAUTH	Authority to create non-fenced user-defined functions.
IMPLSCHEMAAUTH	Authority to implicitly create schemas by creating objects in non-existent schemas.
DBADMAUTH	DBADM authority.
LOADAUTH	Authority to use the DB2 load utility.
QUIESCECONNECTAUTH	Authority to access the database when it is quiesced.
EXTERNALROUTINEAUTH	Authority to create external routines.
SECURITYADMAUTH	Authority to monitor and tune SQL statements.

Additional information

This section describes the additional information related to the DB2 Windows Connector.

Troubleshooting

Following targets are supported:

- SCHEMA
- TABLE
- INDEX
- TABLE SPACE
- PACKAGE
- FUNCTION
- PROCEDURE
- METHOD

Note: **SCHEMA** is appended before Schema name. Similar appending is done for other objects.

Create user

The DB2 server manages all access on DB2 server for Windows users and the Windows authentication is used to login to DB2 server. Hence a user already existing in windows box/domain is required.

To create user in DB2, perform the following steps:

1. Create identity and navigate to request access.
2. Select the identity.
3. Select the entitlement and checkout it.
4. Enter a User Name.
5. Select **Y** to any one authorizations listed by default in the create template policy.

Note: **User creation fails if any one authorization is not selected as 'Y'.**

Delete user

In order to perform the delete operation, set the value of the **DeleteDatabaseUserBYdefault** parameter to **Y** as follows:

```
<entry key="DeleteDatabaseUserBYdefault" value="Y"/>
```

Delete Role

In order to perform the delete role operation, set the value of the **DeleteRoleBYdefault** parameter to **Y** as follows:

```
<entry key="DeleteRoleBYdefault" value="Y"/>
```

Troubleshooting

1 - Revoke permission is not working

If db2jcc4.jar file is used, then the revoke permission will not work.

Resolution: If revoke permission is not working then copy or download the correct version of **db2jcc.jar** file.

Troubleshooting

Chapter 8: SailPoint Delimited File Connector

The following topics are discussed in this chapter:

Overview.....	89
Configuration parameters.....	89
Schema attributes	91

Overview

The SailPoint Delimited File Connector is a *read only* and rule driven connector. This connector has rules that can be customized to handle the complexity of the data that is being extracted.

This connector can be configured to enable the automatic discovery of schema attributes. See “Schema attributes” on page 91.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Delimited File connector uses the following account and application object type “group” connection attributes. The group attributes are optional and the settings default to settings from the Account if they are not specifically defined.

Table 1—Delimited File Connector - Account Tab Descriptions

Parameters	Descriptions
File:	
Parsing Type	Enter which type of parsing technique should be used when parsing the contents of the data file. Note: The parsing type is only applicable for Account Attributes.
File Path	Enter the path and name of the data file that should be parsed.
File Encoding	Specify the encoding that was used when saving the data file. If this is left blank the application's server default encoding will be used when parsing the file.
Delimiter	Enter the character that should be used as a delimiter. If the delimiter is a unicode character use the \\u format. For example, \\u0009 is used to specify for the tab character.
File has column header on first line	Only available for Delimited Parsing Type. Select this option if the data file has a header defined on the first line of the file.

Configuration parameters

Table 1—Delimited File Connector - Account Tab Descriptions

Parameters	Descriptions
Fail on column length mismatch	Only available for Delimited Parsing Type. Select this option if you want the connector to fail if all of the columns are not part of each line. Sometimes the last token is left out of the data. If this is the case in your file, select this option.
Regular Expression	Only available for Regular Expression Parsing Type. Enter the regular expression using regular expression groups that can be used to break the data into tokens.
Columns	Enter the names of the columns that will be used while parsing the file. If you are using the Regular Expression Parsing Type, field is required. If you are using the Delimited Parsing Type, you only have to configure this field if there is not a header defined or you want to rename of the columns that will be used in the buildMap rule.
Transport:	
Note: Transport attributes only apply to Accounts.	
File Transport	Specify how the file will be transferred. If the file resides locally on the application server, select Local .
Host	Specify the host name where the file is located
User	Specify the username that will be used during the file transfer.
Password	Specify the password for the user that will be used during the file transfer.
Filtering:	
Number of lines to skip	Enter the number of lines to skip from the top of the data file before parsing begins.
Filter Empty	Select this option if you want to filter out any objects that parse but have no attributes.
Comment Character	Enter a comment character used in the data file. Any line starting with this character will be skipped.
Filter String	Enter the string representation of an filter object. Any object matching the filter will be filtered out of the dataset and will not be returned. For example, a filter that will filter out all objects from the Manufacturing department is written as follows: <code>department == "Manufacturing"</code>
Merging:	
Data needs to be merged	Select this option if the data for a single object spans multiple lines.
Index Column	Enter the name of the index column that will be used when finding like objects in the dataset.

Table 1—Delimited File Connector - Account Tab Descriptions

Parameters	Descriptions
Data sorted by the indexColumn(s)?	Select this option if the data is sorted by the index columns. If the data is not sorted, an in-memory representation of the data is built and used.
Which Columns should be merged?	Enter the names of the columns from the file from which values should be merged.
Connector Rules:	
Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.	
Build Map Rule	A rule that is called for each row in the data file. This rule is used to convert the string tokens from the data file into a <code>java.util.Map</code> object. If a rule is not specified the connector builds a map with the contents keyed by the column name.
PreIterate Rule	A rule that is called before the iteration process begins and provides a hook for things like checking the file, building an alternate feed, or returning a stream object.
PostIterate Rule	A rule that is called after the iteration process has completed.
Map To ResourceObject Rule	A rule that is called for each unique <code>java.util.Map</code> created from the data file. This rule's job is used to convert a <code>java.util.Map</code> object, built from the data file, into a <code>ResourceObject</code> . If a rule is not specified the connector builds a <code>ResourceObject</code> using the schema.
MergeMaps Rule	A rule that is called during merging for each row that has a matching index column. The rule will receive the existing map along with the newly parsed map that has to be merged. If a rule is not specified the connector builds a combined <code>java.util.Map</code> using the original object and merges the attributes specified in the <code>mergeColumns</code> configuration option.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. In version 7.3, this connector now supports multiple types of objects, account and any number of group application object types. Account objects are used when building identities Link objects. The Additional Schema definitions can be used when building AccountGroup objects which are used to hold entitlements shared across identities.

For delimited file connectors the schema is usually dictated by the data in the file. If this connector is configured to use the automatic discovery function and you've specified column names (`columnNames`, `group.columnNames`, `account.columnNames`), those names are used to populate the schema. If there is a header in the file and the `hasHeader` option is enabled the columns are pulled directly from the file and populate the schema. All automatically generated schema attributes are marked as type String.

Schema attributes

Chapter 9: SailPoint Dropbox Connector

The following topics are discussed in this chapter:

Overview.....	93
Supported features	93
Supported Managed System.....	94
Pre-requisites	94
Administrator permissions	94
Configuration parameters.....	94
Schema attributes	95
Account attributes	95
Group attributes.....	95
Provisioning Policy attributes	96
Account attributes	96

Note: SailPoint Dropbox Connector does not support upgrade from any previous version of Dropbox Business API version 1 to Dropbox Business API version 2. User must define new application of Dropbox in IdentityIQ version 7.3.

Overview

The SailPoint Dropbox Connector manages enterprise users and groups of Dropbox for Business. Dropbox Connector is a read/write connector that can retrieve enterprise users and groups of a particular business, update the permission of users and group, assign enterprise user memberships to groups.

Supported features

SailPoint Dropbox Connector provides support for the following features:

- Account Management
 - Manage Dropbox Users as Accounts
 - Aggregate, Refresh Account
 - Create, Update, Delete
 - Add/Remove Entitlements
- Account - Group Management
 - Manage Dropbox Groups as Account - Group
 - Aggregate, Refresh Group

Configuration parameters

Supported Managed System

SailPoint Dropbox Connector supports the following managed system:

- Dropbox Business API Version 2 supported by Dropbox Server

Pre-requisites

User must generate the access token for business Dropbox application. This has to be acquired by the Dropbox administrator from Dropbox business server using OAUTH2 mechanism to allow IdentityIQ to interact with Dropbox server. The access tokens are valid until the user uninstalls the application, or explicitly revokes the grant via the Dropbox page.

Note: For more information, see the OAuth Guide in Dropbox documentation.

Administrator permissions

Application created in the Dropbox requires the following permission (This permission has both read and update rights):

- Team member management: Team information, and the ability to add, edit, and delete team members

Note: The default role for any user in the Drop is Members_only and it cannot be deleted.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Note: The attributes marked with * sign are the required attributes.

Table 1—Delimited File Connector - Account Tab Descriptions

Parameters	Descriptions
Dropbox Url*	The location where Dropbox server is present. For example, https://api.dropbox.com
Dropbox Access Token*	Access token specific to business application.
Member Page size	Number of records per page. Default: 1000 (maximum)

Additional configuration parameter

The default value of Dropbox api version is 2. If version must be changed, add the following attribute in the application debug page:

```
<entry key="dpWsapiVersion" value="Version_name"/>
```

For example, `<entry key="dpWsapiVersion" value="V2"/>`

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
team_member_id	Member ID
status	Status of the member whether active or invited.
surname	Member's surname.
given_name	Member's first name.
email	Email address of the member.
email_verified	Boolean attribute describes whether user's email is verified to be owned by the user .
role	Roles on the Dropbox, it can be administrator or member.
groups	List of groups connected to member.
external_id	External ID.
familiar_name	Locale-dependent name. In the US, a person's familiar name is their given_name, but elsewhere, it could be any combination of a person's given_name and surname.
display_name	A name that can be used directly to represent the name of a user's Dropbox account.
abbreviated_name	An abbreviated form of the person's name. Their initials in most locales.
account_id	A user's dropbox account identifier.
joined_on	The date and time the user joined as a member of a specific team.
persistent_id	Persistent ID that a team can attach to the user. The persistent ID is unique ID to be used for SAML authentication.

Group attributes

The following table lists the group attributes:

Attributes	Description
group_id	Group ID
group_external_id	This is an arbitrary ID that an admin can attach to a group.
group_name	Name of the group.
member_count	Total count of the members connected to the group.
members	List of group members.

Provisioning Policy attributes

Attributes	Description
group owners	List of group owners.
group_management_type	The group type determines how a group is managed that is, user_managed/company_managed

Provisioning Policy attributes

This section lists the different policy attributes of Dropbox Connector.

Account attributes

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
Email	Member email.
First Name	Member given name.
Surname	Member surname.
External Id	Member external Id.
Welcome Email	Welcome email sent to members.

Delete Provisioning Policy

If required user can add the following delete provisioning policy for Dropbox application:

```
<Form name="account delete" objectType="account" type="Delete">
    <Attributes>
        <Map>
            <entry key="IIQTemplateOwnerDefinition">
                <value>
                    <DynamicValue value="" />
                </value>
            </entry>
        </Map>
    </Attributes>
    <Description>account delete</Description>
    <Field displayName="email_id of member to transfer file" filterString=""
name="email_id of member to transfer error" reviewRequired="true"
type="string"/>
    <Field displayName="transfer_admin_email_id" filterString=""
name="transfer_admin_email_id" reviewRequired="true" type="string"/>
</Form>
```


Provisioning Policy attributes

Chapter 10: SailPoint Duo Connector

The following topics are discussed in this chapter:

Overview.....	99
Supported features	99
Pre-requisites	99
Administrator permissions	100
Configuration parameters.....	100
Schema Attributes	101
Account attributes	101
Group attributes.....	101
Provisioning Policy attributes.....	102
Account attributes	102
Behavioral changes	104
Troubleshooting.....	104

Overview

The SailPoint Duo Connector is a Cloud based Connector. The Duo authentication secures login's by two factor authentications and protects users, data and applications from credential theft and breaches with a focus on streamlined usability.

In Duo Connector the users who have an account on Duo managed system are used for Account provisioning. You can configure the Duo connector to use any of the attributes of user / group which are supported by Duo APIs.

Supported features

SailPoint Duo Connector supports the following features:

- Account Management
 - Aggregation, Refresh Accounts
 - Create, Delete
 - Enable, Disable, Unlock
 - Add/Remove Entitlement
 - Add/Remove Phone attribute
- Account - Group Management
 - Aggregation

Pre-requisites

JRE version 1.7 or higher must be installed.

Administrator permissions

1. Login as an administrator user and log into the **Duo Administrator Panel**.

Note: Ensure that the administrator creating the new Administrator API integration has the following permissions:
- Grant Administrators
- Grant Applications
- Grant Read Resource
- Grant Write Resource

2. Create a new Admin API integration.

This will generate an integration key and secret key for you to use as mentioned in the following “Configuration parameters” section.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Duo Connector uses the following connection attributes:

Attribute	Description
Admin API Credentials	
API hostname *	The API hostname is unique to account and shared with all integrations. Uses https, unsecured http is not supported.
Integration Key *	Identifies integration. Required to configure your system to work with Duo.
Secret Key *	The secret key is treated like a password. Identifies integration. Required to configure your system to work with Duo.
Auth API Credentials	
API hostname	Enter the Auth API hostname (used in multi-factor authentication).
Integration Key	Enter the Auth API Integration Key (used in multi-factor authentication).
Secret Key	Enter the Auth API Secret Key (used in multi-factor authentication).
Note: The Auth API Credentials are used by IdentityIQ and not by Connector.	

Schema Attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
username	Users name.
status	The users status: <ul style="list-style-type: none"> Active: User must complete secondary authentication Bypass: User will bypass secondary authentication after completing primary authentication. Disabled: User will not be able to login. Locked out: User has been automatically locked out due to excessive authentication attempts.
email	Users email address.
user_id	Users unique ID generated by Duo system.
realname	Users real name.
notes	Notes about the user. Seen in Duo administrative interface.
groups	List of groups to which user belongs. Contains description and name of the group.
phones	A list of phones the user can use. Contains phone_id, number, extension, name, postdelay, predelay, type, capabilities, platform, activated, sms_passcodes_sent.
last_login	The last time the user logged in, as a UNIX timestamp or null if the user has not logged in.
tokens	A list of tokens the user can use. Contains serial, token_id, type.
desktoptokens	A list of tokens the user can use.
alias1	Username Alias 1
alias2	Username Alias 2
alias3	Username Alias 3
alias4	Username Alias 4

Group attributes

The following table lists the account attributes:

Provisioning Policy attributes

Attributes	Description
name	The groups name.
desc	The groups description
status	<p>The groups authentication status.</p> <ul style="list-style-type: none"> • Active: User must complete secondary authentication • Bypass: User will bypass secondary authentication after completing primary authentication. • Disabled: User will not be able to login. • Locked out: User has been automatically locked out due to excessive authentication attempts.
group_id	The groups ID.
voice_enabled	<p>If true, users in the group would be able to authenticate with a voice callback. If false, users in the group would not be able to authenticate with a voice callback.</p> <p>Note: This setting has no effect if voice callback is disabled globally.</p>
sms_enabled	<p>If true, users in the group would be able to use SMS passcodes to authenticate. If false, users in the group would not be able to use SMS passcodes to authenticate.</p> <p>Note: This setting has no effect if SMS passcodes are disabled globally.</p>
mobile_otp_enabled	<p>If true, users in the group would be able to use mobile otp password to authenticate. If false, users in the group would not be able to use mobile otp password to authenticate.</p> <p>Note: This setting has no effect if mobile otp passwords are disabled globally.</p>
push_enabled	<p>If true, users in the group would be able to use Duo Push to authenticate. If false, users in the group would not be able to use Duo Push to authenticate.</p> <p>Note: This setting has no effect if Duo Push is disabled globally.</p>

Provisioning Policy attributes

This section describes the various provisioning policy attributes for Account.

Account attributes

The following table lists the provisioning policy attributes for Create, Enable and Unlock Account:

Attribute name	Description
Create Account	
User Name*	Name of user to be created.

Attribute name	Description
Real Name	Real name of the user.
Email ID	Email address will be used for enrollment of the user.
Phone	Phone numbers for the user. Note: After upgrading IdentityIQ version 6.4 to version 7.3, add the Phone attribute manually.
Phone_Type	(Optional and applicable only when single mobile number is assigned to account) Type of phone (mobile or landline). Note: For multiple phone numbers assigned to the account, the default phone type is set as 'mobile'.
Phone_Platform	(Optional and applicable only when single mobile number is assigned to account) Platform of the mobile. Note: For multiple phone numbers assigned to the account, the default phone platform is set as 'generic smartphone'.
Enable and Unlock Account	
User Name*	Name of the user to be enabled/unlocked.
status	Users status: <ul style="list-style-type: none">• Active: User must complete secondary authentication• Bypass: User will bypass secondary authentication after completing primary authentication.

Additional information

This section describes the additional information related to the Duo Connector.

Upgrade considerations

After upgrading IdentityIQ to version 7.3, to support alias user name for the upgraded existing Duo application, add the alias schema attributes manually to the application debug page as follows:

```

<AttributeDefinition name="alias1" type="string">
    <Description>Username Alias 1</Description>
</AttributeDefinition>

<AttributeDefinition name="alias2" type="string">
    <Description>Username Alias 2</Description>
</AttributeDefinition>

<AttributeDefinition name="alias3" type="string">
    <Description>Username Alias 3</Description>
</AttributeDefinition>

<AttributeDefinition name="alias4" type="string">
    <Description>Username Alias 4</Description>
</AttributeDefinition>
```

Behavioral changes

- During Aggregation
 - If a users status is active and has one or more groups connected to the user and if one of the groups (for example, Firewall Admins) is disabled, then the users status would get disabled from Firewall Admins group. This would also disable the user from all the groups assigned to that user.
In the above scenario, if Account is aggregated, the user status is still displayed as active.
 - On Duo system the Last login for some of the users is displayed as **Never authenticated**. In this case, during aggregation, the last Login is displayed as blank.
 - For the **Phone numbers of Account** attribute, the API returns the phone number as +19405429053 but on Duo Connector UI it is displayed as 9405429053.
- For provisioning
 - When performing provisioning on **AD_SYNC user(s)** on Duo, it fails with the following error message:
`openconnector.ConnectorException: disable failed. Duo error code (40010): User is synchronized with Active Directory. Some attributes may be read-only.`

Troubleshooting

1 - During provisioning policy while configuring phone_type or phone_platform attribute an error message is displayed

During provisioning policy while configuring phone_type or phone_platform attribute the following error message is displayed:

`openconnector.ConnectorException: Both attributes 'phone_type' and 'phone_platform' are required.`

Resolution: During provisioning policy, configure both the attributes **phone_type** and **phone_platform**.

Chapter 11: SailPoint GE Centricity Connector

The following topics are discussed in this chapter:

Overview.....	105
Supported features	105
Prerequisites	106
Administrator permissions	106
Configuration parameters.....	106
Additional configuration parameters	106
Schema attributes	107
Account attributes	107
Group attributes.....	108
Provisioning Policy attributes	108
Troubleshooting.....	109

Overview

Centricity is a brand of 27 healthcare IT software solutions from GE Healthcare, a division of General Electric. It includes software for independent physician practices, academic medical centres, hospitals and large integrated delivery networks. The various modules perform practice management, revenue cycle management, electronic medical records, medical imaging, and other functions.

GE Healthcare acquired IDX Systems in 2006 and re-released its products under the Centricity brand. The GE-branded solution formerly produced by IDX acts as an interface engine for receiving and exchanging HL7 and other types of records.

SailPoint GE Centricity Connector manages GE Centricity accounts, roles/rights, applications and user system. The connector manages GE centricity Web portal interface. Changes to the character cell interface must be done manually.

Supported features

SailPoint GE Centricity Connector supports the following features:

- Account Management
 - Manage GE Centricity accounts as Accounts
 - Aggregate, Refresh Account, Pass Through Authentication
 - Create, Update
 - Enable, Disable
 - Change Password - Supported for change/reset password in IDX Web Portal. Need to manually synchronize password on IDX Backend Console,
 - Add/Remove Entitlements - While additional entitlements can be requested for UserSystems, UserApplications, and Roles, remove Entitlement is supported only for Roles.

Configuration parameters

- Account Group Management
 - Manage GE Centricity Roles as Account- Groups
 - Aggregate
- Permissions Management
 - Application reads permissions directly assigned to accounts and groups as direct permissions during account and group aggregation.
 - The connector supports automated revocation of the aggregated account permissions. Work items are created for requests to revoke group permissions.

Prerequisites

The DataURL must be configured in the GE client application. The URL (IP Address and Port) with Username and password is required for using the connector.

No third party jars are required.

Administrator permissions

The GE IDX Connector administrator should have **Administrator - View all, edit users** role.

Configuration parameters

The following table lists the configuration parameters of GE Centricity Connector:

Parameters	Description
GECentricity URL	Specifies the data url containing host and port of GE Centricity instance you are using.
Username	Specifies the administrator or the name of the user which has Admin level privileges to perform aggregation and provisioning operation on GE Centricity system.
Password	Defines the password of the username.
Manage Active Accounts Only	If selected will aggregate only active accounts.

Additional configuration parameters

Attribute	Description
columnDelimiter	Column separator for multi column attribute value. For example, UserApplications. Default: #.
requireForceChangePswdFlag	Indicates whether force change password should be turned on for helpdesk password reset. Default: Y.

Attribute	Description
skipAppUserPswdChange	Defined when change of password for all user system applications is not required. Use the following format to configure in the application: <entry key="skipAppUserPswdChange" value="true"/>
skipSetPswdForAppUserStartsWith	Lists the system application usernames which should not override the original password. For example, the usernames like msi , msitrain which need to be skipped. If the attribute is not added in the configuration, it will update all the user applications with new password. The following format is used to be configured in the application: <entry key="skipSetPswdForAppUserStartsWith" value="msitest,msitrain"/>

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
Key	By default, this attribute is the connectors default nativeIdentity and display name attributes. It's the user ID. For example, SailPoint
Name	User's full name
Email	User's department
Password	User's password
Roles	List of Roles assigned to user.
UserSystems	User's system connections. It is multivalued attribute with SystemName, SystemID, and MenuKey.
UserApplications	User's application connections. It is multivalued attribute. It will consist of SystemId, SystemName, ApplicationID, ApplicationName and Username.
PasswordExpired	Flag that indicates if the user's password expired
PasswordFailures	Integer value indicates of user's failure attempts
PasswordLastChanged	Date timestamp of user's last password changed.
PasswordNeverExpires	Flag indicates the user's neverexpires password.
LastLogin	Date timestamp of user's last login.
Authentication	Authentication of which CF have been used.
DefaultRoleKey	Default Role of user's in CF application.

Provisioning Policy attributes

Attributes	Description
DefaultSystemID	Default System of user's in CF application.
LastFailedLogin	Date timestamp of user's last failed login.
directPermission	List of Rights assigned to User.

Group attributes

Roles are aggregated during account group aggregation, following are the attributes returned by the group aggregation process. Right API brings rights as well as roles. Roles have TypeID as **H** while for Rights, TypeID is **L**.

Attributes	Description
key	By default, this attribute is the connectors default Group Attribute It's the role Id.
Name	Name of the Role in CF application.
ProductID	Name of the Product mapped to Role in CF application.
ApplicationID	Name of the Application mapped to Role in CF application.
VTBMenuKey	Menukey which is mapped to each Role.
ID	Represents the unique Role of the IDX.
Description	Shows detail description of Roles.
directPermissions	The rights will be treated as directPermission for Role (Group) entity.

Provisioning Policy attributes

The SailPoint GE Centricity Connector has a default Provisioning Policy defined which allows creation of accounts. The provisioning policy can be edited to fit specific customer environments.

Attributes	Description
Create user policy	
Key	The userid of the GE centricity application
password	The user's password.
Disabled	False (Default): Will show active user True : Will show inactive user
Name	Defaults to the identity's Lastname/FullName/FirstName.
Email	Defaults to the identity's department.
Force Password Change	Indicates the last changed password of identity.
Password Never Expires	Defaults to identity's password NeverExpire Flag.

Attributes	Description
Default System	Defaults to UserSystems .
Default Role	Defaults to Role/rights of identity.

Troubleshooting

1 - If Authentication field is added on create/update user policy, the API displays an error message

If Authentication field is added on create/update user policy, the API displays the following error message:

Error updating into table "Users"

Resolution: Remove Authentication field from policies.

2 - Change/Reset password works from IdentityIQ, but not able to login in GE IDX

Change/Reset password works if the password is changed from GE Web console and IDX Backend Console (character Cell Interface). The IdentityIQ change/reset password calls the WebAPI, hence the password is changed for GE Web console. By the time the password is synchronized with IDX Backend Console, IdentityIQ stops working.

Resolution: After the password is changed and the user is able to login, synchronize the password in specific User System applications.

3 - When creating user from IdentityIQ/Web console, user is unable to login through Web portal of GE

When creating user from IdentityIQ/Web console, user is unable to login through Web portal of GE.

Resolution: Create user in IDX character cell interface.

Troubleshooting

Chapter 12: SailPoint Google Apps (G Suite) Connector

The following topics are discussed in this chapter:

Overview.....	111
Supported features	111
Pre-requisites	112
Administrator permissions	113
Configuration parameters.....	113
Schema attributes	114
Account attributes	114
Group attributes.....	116
Role attributes	119
Provisioning Policy attributes.....	119
Additional information	126
Performance improvement.....	126
Managing Custom Schema attributes.....	127
Complex Provisioning Policy attributes	127
Troubleshooting.....	128

Note: G Suite is the new name for formerly called Google Apps.

Overview

SailPoint Google Apps (G Suite) Connector now manages accounts, groups and roles from all available domains of Google Apps for Business, Education, or ISP provided that service account mentioned in application configuration has sufficient access to those domains.

The connector consists of a number of features like managing Gmail Delegates for accounts, moving user from one Organizational Unit to another, managing large number of account and group attributes.

Supported features

The SailPoint Google Apps Connector supports the following features:

- Account Management
 - Manage Google Apps Users as Accounts
 - Delta Aggregation, Aggregate, Partitioning Aggregation, Refresh Accounts, Aggregation and Provisioning of Custom Schema attributes
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add/Remove Entitlements
 - Manages Delegated Administrators and Alias on Accounts
 - Move user to other Organization Unit

Overview

- Account - Group Management
 - Manage Google Apps Groups as Account - Groups
 - Aggregate, Refresh Group
 - Create, Update, Delete
- Account - Role Management
 - Manage Google Apps Role as Account - Roles
 - Aggregate, Refresh Role
 - Create, Update, Delete
- Transfer data from one account to another before deleting the account
For more information on the attributes to be configured for data transfer, see “Additional configuration parameters” on page 113.

References

- “Appendix A: Delta Aggregation”
- “Appendix B: Partitioning Aggregation”

Pre-requisites

Note: If GoogleApps Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

1. Enable API access from **Admin Console**.
Select **Security => API Reference => API Access** and select **Enable API access**.
2. Create a project in **Google Developers Console**.
3. Select **APIs & auth** in the left side bar.
In the list of displayed APIs, ensure that the Admin SDK and Groups Settings API status is set to ON.
4. Select **APIs & auth => Credentials** in the left side bar.
Create a Client ID for Web Application and note the Client ID and Client Secret.
5. Acquire refresh token for the following scopes:

Scope	Purpose
https://www.googleapis.com/auth/admin.directory.group	Group Provisioning
https://www.googleapis.com/auth/admin.directory.user	User Provisioning
https://www.googleapis.com/auth/apps.groups.settings	Group Settings APIs
https://apps-apis.google.com/a/feeds/emailsettings/2.0/	Gmail Delegate Provisioning
https://www.googleapis.com/auth/admin.reports.audit.readonly	For Account and Group Delta Aggregation
https://www.googleapis.com/auth/admin.datatransfer	For Data transfer before deletion of Account
https://www.googleapis.com/auth/admin.directory.rolemanagement.readonly	For getting and listing roles, privileges, and role assignments

Scope	Purpose
https://www.googleapis.com/auth/admin.directory.rolemanagement	For all roles management operations, including creating roles and role assignments

6. Navigate to **Google Apps Admin Console => Dashboard => Google Apps => Gmail => User settings => Mail Delegation.**

Verify if **Let users delegate access to their mailbox to other users in the domain** is selected.

For more information about the above listed pre-requisites, see the following respective links:

- To get a refresh token
<https://developers.google.com/accounts/docs/OAuth2WebServer>
- Pre-requisites for using user and group APIs
<https://developers.google.com/admin-sdk/directory/v1/guides/prerequisites>
- For more details about pre-requisites for using groups settings APIs
<https://developers.google.com/admin-sdk/groups-settings/prerequisites>

Administrator permissions

The application user should be configured to have the **Super Admin** role.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Google Apps Connector uses the mandatory connection attributes listed in the following table:

Parameters	Description
Client ID	Client ID of user application.
Client Secret	Client Secret of user application.
Refresh Token	Refresh token value.
Read Group Details	<ul style="list-style-type: none"> • Y: Reads all group attributes during group aggregation. This will take longer to complete the aggregation. • N: Reads minimum group attributes during group aggregation.

Note: For customers who do not want to use the multiple domain feature of Google Apps Connector, add the 'domainName' parameter in the application configuration as follows:

```
<entry key="domainName" value="test.sailpoint.com"/>
```

Additional configuration parameters

Configure the following attributes to transfer data of Google Applications before account deletion:

Schema attributes

Attribute	Description
excludeAppsFromTransfer	List of Google Apps which need to be excluded from data transfer. By default it contains Calendar and None . At least one entry (None) must be present in the list.
enableDataTransferonDelete	Boolean flag which indicates if data transfer is to be performed after deletion of account.
relationsForDataTransfer	List of relations which are required to be checked as owner of data as per provided sequence. This attribute would be considered when newOwner is not provided in the provisioning form. For example, <pre><entry key="relationsForDataTransfer"> <value> </List> <String>manager</String> <String>assistant</String> <String>father</String> </List> </value> </entry></pre>

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.
- **Role:** The Role schema is used when building Account-Role objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

Attributes	Description
objectID	Unique ID of the user
primaryEmail	Primary E-mail ID of user. Note: Domain of email id must be a valid domain accessible to service account provided in application configuration.
name	Full name of the user.
isAdmin	Is user an administrator.

Attributes	Description
isDelegatedAdmin	Is user a delegated administrator.
lastLoginTime	Last login time of user.
suspended	Is user suspended.
suspensionReason	Reason for suspension.
changePasswordAtNextLogin	Indicates if the user is forced to change password at next login.
ipWhiteListed	Indicate if user's IP address is white listed.
ims	The user's Instant Messenger (IM) accounts.
emails	A list of the user's E-mail addresses.
externalIds	A list of external IDs for the user, such as an employee or network ID.
relations	A list of the user's relationships to other users.
addresses	A list of the user's addresses.
organizations	List of organizations the user belongs to
phones	A list of the user's phone numbers.
aliases	List of the user's alias E-mail addresses.
nonEditableAliases	List of the user's non-editable alias E-mail addresses.
customerId	The customer ID to retrieve all account users.
orgUnitPath	The full path of the parent organization associated with the user.
isMailboxSetup	Indicates if the user's Google mailbox is created.
includeInGlobalAddressList	Indicates if the user's profile is visible in Global Address List when the contact sharing feature is enabled for the domain.
thumbnailPhotoUrl	Photo Url of the user
delegatedAdmins	Delegated administrators of a user.
Groups	Groups connected to the user.
customSchemas	Aggregates custom schema attributes.
Roles	Roles connected to the user.

Custom Schema attributes

SailPoint Google Apps Connector now provides support for aggregating custom schema attributes by adding the schema attributes in the following format:

```
customSchemas . SCHEMANAME . FIELDNAME
```

For example, If the custom schema's are as follows:

```
customSchemas = {SSO {"ADID":"101", "SAPID":"102"}, Address {"Postal_Code":"411018"}}
```

To display the attributes of the above custom schema's, customer must add the following attributes in account schema:

```
customSchemas.SSO.ADID
```

Schema attributes

`customSchemas.SSO.SAPID`

`customSchemas.Address.Postal_Code`

Note: A schema cannot contain another schema in target system as follows:

`customSchemas= {SSO{Address{"Postal_Code":"411018"}}}`

Group attributes

The following table lists the group attributes:

Attribute	Description
<code>objectId</code>	ID of group
<code>email</code>	Group E-mail address.
<code>name</code>	Name of the group.
<code>directMembersCount</code>	Number of group members.
<code>description</code>	Description of the group.
<code>nonEditableAliases</code>	List of the group's non-editable alias E-mail addresses that are outside of the account's primary domain or sub domains. These are functioning E-mail addresses used by the group. This is a read-only property.
<code>adminCreated</code>	Whether it is created by administrator.
<code>aliases</code>	Group aliases.
<code>members</code>	Members of the group.
<code>whoCanJoin</code>	Permissions to join the group. Possible values: <ul style="list-style-type: none">• <code>ALL_IN_DOMAIN_CAN_JOIN</code>: Anyone in the account can join.• <code>ANYONE_CAN_JOIN</code>: Anyone outside your domain can join.• <code>CAN_REQUEST_TO_JOIN</code>: Non group members can request an invitation to join.• <code>INVITED_CAN_JOIN</code>: Candidates for membership can be invited to join.
<code>whoCanViewMembership</code>	Permissions to view membership. Possible values - <ul style="list-style-type: none">• <code>ALL_IN_DOMAIN_CAN_VIEW</code>: Anyone in the account can view the group members list.• <code>ALL_MANAGERS_CAN_VIEW</code>: The group managers can view group members list.• <code>ALL_MEMBERS_CAN_VIEW</code>: Group members can view the group members list

whoCanViewGroup	<p>Permissions to view group.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • ALL_IN_DOMAIN_CAN_VIEW: Anyone in your account can view this group's messages. • ALL_MANAGERS_CAN_VIEW: Any group manager can view this group's messages. • ALL_MEMBERS_CAN_VIEW: All group members can view the group's messages. • ANYONE_CAN_VIEW: Any Google Apps user can view the group's messages.
whoCanInvite	<p>Permissions to invite members.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • ALL_MANAGERS_CAN_INVITE: Only managers can invite a new member. this includes the group's owner. • ALL_MEMBERS_CAN_INVITE: Managers and members can invite a new member candidate.
allowExternalMembers	A Boolean indicating if Google Apps users external to your account can view or become members of this group.
MANAGERS	Managers of a group.
OWNERS	Owners of a group.
whoCanPostMessage	<p>Permissions to post messages to the group.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ALL_IN_DOMAIN_CAN_POST: Anyone in the account can post a message. • ALL_MANAGERS_CAN_POST: Managers, including group owners, can post messages. • ALL_MEMBERS_CAN_POST: Any group member can post a message. • ANYONE_CAN_POST: Any Google Apps user outside the account can access the Google Groups service and post a message. • NONE_CAN_POST: The group is disabled and archived. No one can post a message to this group.
allowWebPosting	A Boolean indicating if any member allowed to post to the group web forum.
primaryLanguage	Language tag for a group's primary language.
maxMessageBytes	The maximum size of a message.
isArchived	A Boolean indicating if the contents of the group to be archived.
archiveOnly	A Boolean indicating if the group to be only archived.

Schema attributes

messageModerationLevel	<p>Moderation level for messages.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • MODERATE_ALL_MESSAGES: All messages are approved by owner before the message is sent to the group. • MODERATE_NEW_MEMBERS: All messages from new members are approved by owner before the message is sent to the group. • MODERATE_NONE: No moderator approval is required. • MODERATE_NON_MEMBERS: All messages from non-group members are approved by owner before the message is sent to the group.
spamModerationLevel	<p>Moderation levels for messages detected as spam.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • ALLOW: Post the message to the group. • MODERATE: Send the message to the moderation queue. • SILENTLY_MODERATE: Send the message to the moderation queue, but do not send notification to moderators. • REJECT: Immediately reject the message.
replyTo	<p>The default reply to a message is sent here.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • REPLY_TO_CUSTOM: For replies to messages, use the group's custom E-mail address. • REPLY_TO_IGNORE: Group users individually decide where the message reply is sent. • REPLY_TO_LIST: Reply message is sent to the group. • REPLY_TO_MANAGERS: Reply message is sent to the group's managers. • REPLY_TO_OWNER: Reply is sent to the owner(s) of the group. • REPLY_TO_SENDER: Reply is sent to author of message.
customReplyTo	An E-mail address used when replying to a message.
sendMessageDenyNotification	A Boolean indicating if the members are notified if his message is denied by owner.
defaultMessageDenyNotificationText	Text sent to the message's author as part of rejection notification.
showInGroupDirectory	A Boolean indicating if group is listed in the Groups directory.
allowGoogleCommunication	A Boolean allowing Google to contact group administrators.
membersCanPostAsTheGroup	A Boolean indicating if members can post using the group E-mail address.

messageDisplayFont	Default message's display font. Possible values are: <ul style="list-style-type: none">• DEFAULT_FONT: Uses the account's default font.• FIXED_WIDTH_FONT: Uses fixed width font.
includeInGlobalAddressList	A Boolean indicating if group is included in the Global Address List .

Role attributes

The following table lists the Role attributes:

Attributes	Description
objectID	roleID of role.
roleName	Name of the role.
roleDescription	Description of the role.
rolePrivileges	Privileges of the role. A multivalued attribute. Values: GROUPS_ALL, APP_ADMIN
isSystemRole	System role.
isSuperAdminRole	Super Admin role.

Provisioning Policy attributes

This section lists the provisioning policy attributes for the following:

- Create/Update/Delete Account
- Create/Update Group
- Create/Update Role

Note: In this section all the attributes marked with the * sign indicate that the attributes are mandatory.

In this section all the attributes marked with # sign indicate that the attributes are complex attributes and for more information, see “Complex Provisioning Policy attributes” on page 127.

Create Account

The following table lists the provisioning policy attributes for Create Account:

Attribute	Description
familyName*	The user's last name.

Provisioning Policy attributes

givenName*	The user's first name.
password*	Password for the user account.
primaryEmail*	Primary E-mail address of the user
suspended	A Boolean indicating if the user is suspended.
changePasswordAtNextLogin	Indicates if the user is forced to change their password at next login.
hashFunction	Hash function name for password. Values: MD5, SHA-1 and crypt.
includeInGlobalAddressList	A Boolean indicating if the user's profile is visible in the Global Address List.
ipWhitelisted	A Boolean indicating if user's IP address is white listed.
organizationUnit	Full path of the parent organization of the user. Root organization is represented as forward slash (/).
customSchemas	<p>Custom schema attribute assigned to the user. Attribute must be present in schema before adding in provisioning policy.</p> <p>Note: The customSchema attribute is a placeholder for actual custom schema attribute name. Use the custom schema attribute name while defining provisioning policy. For more information, see "Managing Custom Schema attributes" on page 127.</p>
Address Attributes	
country	Country
countryCode	Country code
extendedAddress	Extended addresses, such as an address that includes a sub-region.
locality	The town or city of the address.
poBox	The post office box, if present.
postalCode	The ZIP or postal code, if applicable.
primaryAddress	A Boolean indicating if this is primary address of the user
region	The abbreviated province or state.
sourcesStructured	A Boolean indicating if the user-supplied address was formatted. Formatted addresses are not currently supported.
streetAddress	The street address. Whitespace within the string is ignored.
addressType	The address type. Values: custom, home, other, work
addressCustomType	Custom address type.
Email Address Attributes	
emailAddress	User's primary E-mail address or an alias.
ifPrimary	A Boolean indicating if this is the user's primary E-mail.
emailtype	The type of the E-mail account. Values: custom, home, other, work

emailCustomType	Custom E-mail type.
External IDs Attributes	
externalIdsType	The type of the ID. Values: account, custom, customer, network, organization
externalIdsCustomType	Custom external ID type.
externalIdsValue	The value of the ID.
Messenger Attributes	
IMType	IM Type. Values: custom, home, other, work
IMCustomType	Custom IM type.
IMID	The user's IM network ID.
primaryIM	A Boolean indicating if this is the user's primary IM.
IMProtocol	An IM protocol identifies the IM network. The value can be a custom network or the standard network. Values: <ul style="list-style-type: none"> • custom_protocol: A custom IM network protocol • aim: AOL Instant Messenger protocol • gtalk: Google Talk protocol • icq: ICQ protocol • jabber: Jabber protocol • msn: MSN Messenger protocol • net_meeting: Net Meeting protocol • qq: QQ protocol • skype: Skype protocol • yahoo: Yahoo Messenger protocol
Organization Unit Details	
organizationName	Name of the organization.
costCenter	The cost center of the user's organization.
department	Specifies the department within the organization
description	Description of the organization.
domain	Domain the organization belongs to.
organizationLocation	Physical location of the organization.
primaryOrganization	A string indicating if this is the user's primary organization.
organizationSymbol	Text string symbol of the organization.
organizationTitle	User's title within the organization
organizationType	Type of organization. Values: unknown, school, work, domain_only
organizationCustomType	Custom organization type.
Phone Attributes	
primaryPhone	A Boolean indicating if this is user's primary phone number.

Provisioning Policy attributes

phoneNumber	Phone number.
phoneType	The type of phone number. Values: custom, home, work, other, home_fax, work_fax, mobile, pager, other_fax, compain_main, assistant, car, radio, isdn, callback, telex, tty_tdd, work_mobile, work_pager, main, grand_central
phoneCustomType	Custom phone type.
Relation Attributes	
relation	The name of the person the user is related to.
relationType	The type of relation. Values: custom, spouse, child, mother, father, parent, brother, sister, friend, relative, domestic_partner, manager, assistant, referred_by, partner
relationCustomType	Custom relation type.

Update Account

The following table lists the provisioning policy attributes for Update Account:

Attribute	Description
delegatedAdmins	The user's delegated administrators. Values are String or List of E-mail IDs.
name	<p>Full Name of the user. The value of the attribute must consist values for the following attributes:</p> <ul style="list-style-type: none"> • givenName • familyName <p>The value of the name attribute must be in the JSON format. For example, {"givenName": "abc", "familyName": "xyz"}</p>
aliases	The user's aliases. Values are String or List of E-mail IDs.
organizationUnit	Full path of the organization unit the user should be moved to. For example, test.com/Marketing/Manager where Manager is the organization unit the user should be moved to.
hashFunction	Hash function name for password. Values: MD5, SHA-1 and crypt
includeInGlobalAddressList	A Boolean indicating if the user's profile is visible in the Global Address List.
ipWhitelisted	A Boolean indicating if user's IP address is white listed.
primaryEmail	The user's primary E-mail address.
addresses#	<p>A list of the user's addresses.</p> <p>Each address can consist of values for the following attributes: type, customType, sourceIsStructured, formatted, poBox, extendedAddress, streetAddress, locality, region, postalCode, country, primary, countryCode</p>

emails#	A list of the user's E-mail addresses. Each email can consist of values for the following attributes: address, type, customType, primary
externalIds#	A list of external IDs for the user, such as an employee or network ID. Each externalId can consist of values for the following attributes: value, type, customType
ims#	A list of user's Instant Messenger (IM) accounts. Each ims can consist of values for the following attributes: type, customType, protocol, customProtocol, im, primary
organizations#	A list of organizations the user belongs to. Each organization can consist of values for the following attributes: name, title, primary, type, customType, department, symbol, location, description, domain, costCenter
customSchemas	Custom schema attribute assigned to the user. Attribute must be present in schema before adding in provisioning policy. Note: The customSchema attribute is a placeholder for actual custom schema attribute name. Use the custom schema attribute name while defining provisioning policy. For more information, see "Managing Custom Schema attributes" on page 127.
phones#	A list of the user's phone numbers. Each phone can consist of values for the following attributes: value, primary, type, customType
relations#	A list of the user's relationships to other users. Each relation can consist of values for the following attributes: value, type, customType

Delete Account

The following table lists the provisioning policy attributes for Delete Account:

Attribute	Description
newOwner	Email/object of account to which data must be transferred.

Create Group

The following table lists the provisioning policy attributes for Create Group:

Attribute	Description
email*	The group's E-mail address.
name	The group's name.
description	Description of the group.

Update Group

The following table lists the provisioning policy attributes for Update Group:

Provisioning Policy attributes

Attribute	Description
name	Name of the group.
description	Description of the group.
MANAGERS	Managers of a group.
OWNERS	Owners of a group.
email	(Required when creating a group) The group's email address. If your account has multiple domains, select the appropriate domain for the email address. The email must be unique.
whoCanJoin	Permissions to join the group. Values: ALL_IN_DOMAIN_CAN_JOIN, ANYONE_CAN_JOIN, CAN_REQUEST_TO_JOIN, INVITED_CAN_JOIN
whoCanViewMembership	Permissions to view membership. Values: ALL_IN_DOMAIN_CAN_VIEW, ALL_MANAGERS_CAN_VIEW, ALL_MEMBERS_CAN_VIEW
whoCanViewGroup	Permissions to view group. Values: ALL_IN_DOMAIN_CAN_VIEW, ANYONE_CAN_VIEW, ALL_MANAGERS_CAN_VIEW, ALL_MEMBERS_CAN_VIEW,
whoCanInvite	Permissions to invite members. Values: ALL_MANAGERS_CAN_INVITE, ALL_MEMBERS_CAN_INVITE
allowExternalMembers	A Boolean indicating if Google Apps users external to your account can view or become members of this group
whoCanPostMessage	Permissions to post messages to the group. Values: ALL_IN_DOMAIN_CAN_POST, ALL_MANAGERS_CAN_POST, ALL_MEMBERS_CAN_POST, ANYONE_CAN_POST, NONE_CAN_POST
allowWebPosting	A Boolean indicating if members are allowed posting to the group web forum.
primaryLanguage	Language tag of the primary language for the group
maxMessageBytes	Maximum size of a message, which, by default, is 1Mb.
isArchived	A Boolean indicating if the contents of the group are archived.
archiveOnly	A Boolean indicating if the group is only archived.
messageModerationLevel	Moderation level for messages. Values: MODERATE_ALL_MESSAGES, MODERATE_NEW_MEMBERS, MODERATE_NONE, MODERATE_NON_MEMBERS.
spamModerationLevel	Sets moderation levels for messages detected as spam. Values: ALLOW, MODERATE, SILENTLY_MODERATE, REJECT

replyTo	The default reply to a message is set here. Values: REPLY_TO_CUSTOM, REPLY_TO_IGNORE, REPLY_TO_LIST, REPLY_TO_MANAGERS, REPLY_TO_OWNER, REPLY_TO_SENDER
customReplyTo	If ReplyTo is REPLY_TO_CUSTOM, the customReplyTo must hold a custom E-mail address
sendMessageDenyNotification	A Boolean indicating if the member is notified if his message is denied by owner.
defaultMessageDenyNotificationText	Text for the rejection notification sent to the message's author.
showInGroupDirectory	A Boolean indicating if the group should be listed in the Groups directory.
allowGoogleCommunication	A Boolean allowing Google to contact group administrators.
membersCanPostAsTheGroup	A Boolean indicating if group members can post messages using the group's email address instead of the member's own email address.
messageDisplayFont	Default message's display font. Values: DEFAULT_FONT, FIXED_WIDTH_FONT
includeInGlobalAddressList	A Boolean indicating if the group needs to be included in the Global Address List .

Create Role

The following table lists the provisioning policy attributes for Create Role:

Attribute	Description
roleName*	Name of the role.
rolePrivileges *	List of Role Privileges which must be assigned. The value of the attribute must consist values for the following attributes: <ul style="list-style-type: none"> • privilegeName • serviceId The value of the rolePrivileges attribute must be in the JSON format. For example, { "privilegeName": "APP_ADMIN", "serviceId": "039kk8xu49mji9t" }
roleDescription	Description of the role.

In create role operation, API Console privileges and Admin Console privileges can be assigned for those whose **serviceId** and **privilegeName** are provided.

Note: Only those privileges would get assigned which are entered during Create/Update Role operation.

Update Role

The following table lists the provisioning policy attributes for Update Role:

Additional information

Attribute	Description
roleName	Name of the role.
rolePrivileges #	<p>List of Role Privileges which must be provisioned. The value of the attribute must consist values for the following attributes:</p> <ul style="list-style-type: none">• privilegeName• serviceId: unique Id associated with privilegeName <p>The value of the rolePrivileges attribute must be in the JSON format.</p> <p>For example,</p> <pre>{"privilegeName": "APP_ADMIN", "serviceId": "039kk8xu49mji9t"}</pre> <p>Ensure that at least one rolePrivilege must be present.</p>
roleDescription	Description of the role.

Note: - Every privilegeName must have one serviceId associated with it.
- Ensure that you enter an appropriate serviceId with respect to privilegeName.

Values for **rolePrivileges** attributes must be in JSON format and it must set values for all or some of the sub-attributes of respective attribute mentioned in the above table.

For example, the following request updates **rolePrivileges** attribute of a Role:

```
<AttributeRequest name="rolePrivileges" op="Add">
  <Value>
    <List>
      <String>{"privilegeName": "APP_ADMIN", "serviceId": "039kk8xu49mji9t"}</String>
      <String>{"privilegeName": "USERS_UPDATE", "serviceId": "00haapch16hlysv"}</String>
    </List>
  </Value>
</AttributeRequest>
```

Additional information

This section describes the additional information related to the Google Apps Connector.

Performance improvement

(Optional) For improving the account and group aggregation performance of Google Apps, perform the following:

- **Account Aggregation:** Perform the following:
 - a. Delete the **delegatedAdmins** attribute from Account schema.
 - b. Add the following entry in application debug page:

```
<entry key="isSkipAlias" value="true"/>
```

In case number of groups are greater or equal to the number of users in your environment, add the following entry in the application debug page to skip the pre-loading of group memberships:

```
<entry key="skipPreloadingMemberships" value="true"/>
```

- **Group Aggregation:** Delete the following attributes together from Group schema:
 - OWNERS
 - MANAGERS

Managing Custom Schema attributes

The connector supports for managing custom schema attributes defined in Google Apps by adding the custom attribute names in IdentityIQ application's account schema in the following format:

```
customSchemas.<SCHEMANAME>.<FIELDNAME>
```

For example, If the custom schema in Google Apps is as follows:

```
customSchemas = {SSO {"ADID":"101", "SAPID":"102"}, Address {"Postal_Code":"411018"}}
```

Add following attributes in IdentityIQ application's account schema:

```
customSchemas.SSO.ADID
```

```
customSchemas.SSO.SAPID
```

```
customSchemas.Address.Postal_Code
```

To support provisioning custom attributes, along with adding the custom attributes in account schema, add custom attribute names in the provisioning policy in the same format as described above.

Note: Connector does not support managing multi-level custom attributes (that is schema defined inside the custom schema) defined in Google Apps.

Complex Provisioning Policy attributes

Values for attributes marked with # must be in JSON format. It should set values for all or some of the sub-attributes of respective attribute mentioned in the tables of “Update Account” and “Update Role”.

For example,

- Following request updates emails attribute of an Account:

```
<AttributeRequest name="emails" op="Add">
  <Value>
    <List>
      <String>{"address": "abc1@test.com", "type": "work", "primary": "true"}</String>
      <String>{"address": "abc2@test.com", "type": "work"}</String>
    </List>
  </Value>
</AttributeRequest>
```

- Following request updates rolePrivileges attribute of a Role:

```
<AttributeRequest name="rolePrivileges" op="Add">
  <Value>
    <List>
      <String>{"privilegeName": "APP_ADMIN", "serviceId": "039kk8xu49mji9t"}</String>
    </List>
  </Value>
</AttributeRequest>
```

Troubleshooting

```
<String>{"privilegeName":"USERS_UPDATE","serviceId":"00haapch16h1ysv"}  
</String>  
</List>  
</Value>  
</AttributeRequest>
```

Troubleshooting

1 - Account Aggregation fails with error

The account aggregation may fail due to insufficient permissions or in case required services are not turned on.

Resolution: To test a GET (read) user using Admin SDK Directory API from browser, use the following URL replacing the userEmail and accessToken values:

https://www.googleapis.com/admin/directory/v1/users/userEmail?access_token=ya29.AHES6ZQRwqu6I9-EUNeSAQS8HoI2YKsLLriUKxAh5-UmlGsh-NEhgOA

2 - Group aggregation fails with error

The group aggregation fails due to insufficient permissions or in case required services are not turned on.

Resolution: To test a GET (read) group using Admin SDK Directory API from browser, use the following URL replacing the groupEmail and accessToken values:

https://www.googleapis.com/admin/directory/v1/groups/groupEmail?access_token=ya29.1.AADtN_UOeNHQjnEDkjoMTkiswabM7fzSeWWpknyy_qOLSQevSAb0HEK2djECyy&token_type=Bearer

To test a GET (read) group using Google Groups Settings API (Read Group Details – Y) from browser, use the following URL after replacing the groupEmail and accessToken values:

https://www.googleapis.com/groups/v1/groups/groupEmail?access_token=ya29.1.AADtN_W2DdsN1YkvRc7meVgQ6XDKIOqgZsbA-Nt9O9zNWEcoF7TLVUGXKDwkwcnmzyY7H4&token_type=Bearer&alt=json

3 - Error message appears for some corrupt objects

During aggregation, error messages appear for some corrupt objects.

Resolution: Corrupt objects can be skipped at the time of aggregation by setting the `isContinueOnError` attribute to true. By default, the value of `isContinueOnError` attribute is false. This value can be set to true in the Application XML as follows:

```
<entry key="isContinueOnError" value="true"/>
```

4 - Internal Server and Service Unavailable Error appears

The Internal Server and Service Unavailable error messages are sent by the Google Server.

Resolution: To retry the request, use `maxReadRetryCount` attribute. The retry count is set to 5 by default. Increase the retry count by adding the following entry to the Application XML and set the desired value:

```
<entry key="maxReadRetryCount" value="10"/>
```

5 - SocketTimeoutException error

The SocketTimeoutException error message appears.

Resolution: Increase the timeout interval by adding the `maxReadTimeout` attribute to the application debug. By default the value of `maxReadTimeout` attribute is 180 seconds. To increase the timeout, add the following entry to the Application XML and set the desired value:

```
<entry key="maxReadTimeout" value="240"/>
```

6 - Provisioning (Create Account) fails with an error message

Provisioning (Create Account) fails with the following error message:

Resource Not Found: domain

Resolution: Verify the domain name of primary email as the domain name of email id must be a valid domain and must be accessible for service account provided in application configuration.

7 - While updating the accounts from Integration Console, attributes having list values are not deleted from the account

While updating the accounts from Integration Console, attributes having list values are not deleted from the account.

Resolution: While deleting the attributes from Integration Console, it must be consider that those attributes are present in Google Apps managed system for that account.

If you are setting the "primary=false" in plan, then Google Apps does not consider that attribute. Hence while deleting any attribute (Organizations, phones and so on.) "primary" attribute type must not be present in the plan.

For example,

```
<AttributeRequest name="phones" op="Add">
  <Value>
    <List>
      <String>{"value":"345678","customType":"","type":"custom","primary":"false"}</String>
    </Value>
  </AttributeRequest>

<AttributeRequest name="phones" op="Remove">
  <Value>
    <List>
      <String>{"value":"345678","customType":"","type":"custom"}</String>
    </Value>
  </AttributeRequest>
```

In some attributes like IMS, some required attribute appear as empty value if those attributes are not passed while adding those attributes. Hence add those attribute value as empty in the delete plan.

For example,

```
<AttributeRequest name="ims" op="Add">
```

Troubleshooting

```
<Value>
<List>
    <String>{"im":"test1@dev.sailpoint.com","type":"work"}</String>
</List>
</Value>
</AttributeRequest>

<AttributeRequest name="ims" op="Remove">
<Value>
<List>
    <String>{"im":"test1@dev.sailpoint.com","type":"work",
"customProtocol":""}</String>
</List>
</Value>
</AttributeRequest>
```

8 - Role Provisioning fails with an error

One of the following error messages may occur during role provisioning:

- openconnector.ConnectorException: Exception occurred.
Error message - Required parameter: [resource.privileges[0].service_id]
Resolution: Ensure that the **serviceld** is correct.
- openconnector.ConnectorException: Exception occurred.
Error message - Invalid Role privileges
Resolution: Ensure that the **privilegeName** is correct.
- openconnector.ConnectorException: Exception occurred.
Error message - Unexpected character * at position *.
Resolution: Ensure that the value of **rolePrivileges** is in appropriate JSON format.

9 - Delete operation fails with an error

Delete operation fails with the following error message:

```
openconnector.ConnectorException: Exception occurred.
Error message - Role assignment exists: RolesDeleteRequest.resource_id
```

Resolution: Ensure that the role does not have any user attached to it.

10 - Provisioning Role operation fails with an error message

```
openconnector.ConnectorException: Exception occurred.
Error message - Operation not allowed
```

Resolution: Ensure that the selected role is not a system role.

Chapter 13: SailPoint GoToMeeting Connector

The following topics are discussed in this chapter:

Overview.....	131
Supported features	131
Pre-requisites	131
Administrator permissions	132
Configuration parameter.....	132
Schema attributes	132
Account attributes	132
Group attributes.....	132
Provisioning Policy attributes	133

Overview

SailPoint GoToMeeting Connector manages GoToMeeting organizers. It supports read and write to GoToMeeting to create, retrieve, update, delete users, and retrieve groups.

Supported features

SailPoint GoToMeeting supports the following features:

- Account Management
 - Manage GoToMeeting Users (excludes invited accounts) as Account
 - Aggregate, Refresh Accounts
 - Create, Delete
 - Enable, Disable
- Account - Group Management
 - Manage GoToMeeting Groups as Account - Groups
 - Aggregate, Refresh Groups

Pre-requisites

Note: If GoToMeeting Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

The user must go through the OAuth2 flow to generate the access token. The connector uses this Access Token to make calls to any GoToMeeting REST API.

Administrator permissions

Role of the user must be an Administrator.

Configuration parameter

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Access Token: A valid Access Token for the user is required which enables your application to access the user's information and take actions on their behalf. The application and user are verified with each API call by passing an access token along with each request.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 1—Account attributes](#)):

Table 1—Account attributes

Attributes	Description
OrganizerKey	A unique key associated with each organizer.
FirstName	First Name of the organizer.
LastName	Last Name of the organizer.
Email	Email id of the organizer
Status	The status of the organizer (Active, Invited or Suspended).
GroupKey	A unique key associated with a group of which the organizer is a part of.
Groups	The entitlements of the organizer.
MaximumAttendeesAllowed	The maximum number of attendees allowed for the organizer.

Group attributes

The following table lists the group attributes ([Table 2—Group attributes](#)):

Table 2—Group attributes

Attributes	Description
GroupName	The name of the group.
GroupKey	A unique key associated with the group.
ParentKey	The Parent Key of the group.
GroupStatus	The status of the group.
NumberOfOrganizers	The number of organizers in a group.

Provisioning Policy attributes

This following table lists the provisioning policy attributes for create ([Table 3—Provisioning Policy attributes](#)):

Table 3—Provisioning Policy attributes

Attributes	Description
OrganizerEmail	A valid email id is required to whom a GoToMeeting invite should be sent.

Note: If multiple groups are specified at the time of account creation, the group to which the user is attached would be selected at random, as GoToMeeting supports only one group per user.

Provisioning Policy attributes

Chapter 14: SailPoint IBM i Connector

The following topics are discussed in this chapter:

Overview.....	135
Supported features	135
Supported Managed Systems	136
Pre-requisites	136
Administrator permissions	137
Configuration parameters.....	137
Schema Attributes	138
Provisioning policy attributes	140
Additional information	140
Direct Permissions	140
Upgrade Consideration	141
Create TLS 1.2 Communication between IdentityIQ and IBM i system.....	141
Aggregation of Group Profile as part of Account Aggregation.....	142

Overview

The SailPoint IBM i Connector is a Read/Write connector that uses the User Profiles on IBM i computer for account provisioning. For group provisioning, user profiles which have Group ID are used. The IBM i Connector can be configured to use any of the attributes of user/group which are supported by IBM i commands.

Supported features

SailPoint IBM i Connector supports the following features:

- Account Management
 - Manage IBM i Users as Account
 - Aggregate, Partitioning Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add /Remove Entitlement
- Account - Group Management
 - Manage IBM i Groups as Account - Groups
 - Aggregate, Refresh Groups
 - Create, Update, Delete

Overview

- Permissions Management
 - Application reads permissions directly assigned to accounts and groups as direct permissions during accounts and groups aggregations respectively.
 - The connector supports automated revocation of the aggregated permissions for accounts and groups.
- Authorization list
 - Application reads Authorization list assigned to accounts and groups as AUTL attribute during accounts and groups aggregations respectively.
 - The connector supports
 - Aggregating Authorization Lists associated with users
 - Aggregating Authorization Lists associated with groups
 - ADD/Remove Authorization Lists

Note: USER DEFINED authority (Authority other than *ALL,*CHANGE,*EXCLUDE) in authorization list is not supported for provisioning operations. If user tries to add or remove authorization list having USER DEFINED authority, then connector displays the following error:

```
AUT(USER DEFINED) USER DEFINED is not a valid parameter
```

For example, if user tries to add an authorization list having user defined authority, following error message is displayed:

```
Failed to execute Command: ADDAUTL USER(<User Name>) AUTL(<AUTL Name>)  
AUT(USER DEFINED) USER DEFINED is not a valid parameter
```

References

- “Direct Permissions” on page 140
- Appendix B: Partitioning Aggregation
- Appendix C: Before and After Provisioning Action

Supported Managed Systems

SailPoint IBM i Connector supports the following versions of IBM i:

- IBM i V7R3
- IBM i V7R2
- IBM i V7R1

Pre-requisites

As we use IBM Toolbox for Java (JTOpen) to manage IBM i system, in order to connect to the IBM i system and access its data and services, the IBM Toolbox for Java (JTOpen) requires TC1 Licensed Program (TCP/IP Connectivity Utilities for IBM i, Host Server option of IBM i to be installed and configured on the system. It also utilizes following services running on the remote port:

Service Name	Port	SSL Port
as-signon	8476	9476
as-rmtcmd	8475	9475
as-svrmap	449	NA

Administrator permissions

The SailPoint IBM i Connector requires Security officer/User Profile with required permission to accomplish provisioning tasks. The administrator user that is, user configured for the IBM i Connector must be assigned sufficient privileges on IBM i Connector to create/update user profile and group profile.

For example, User profile can be configured as an administrator with any one of the following user classes and special authorities as mentioned in the following table:

Note: Depending on the System security level (QSECURITY) the special authorities assigned to the user differs. Ensure that the created user profiles have the following mentioned special authorities.

USER CLASS (USRCLS)	SPECIAL AUTHORITY (SPCAUT)
*USER (user)	*SECADM, *AUDIT, *ALLOBJ
*SECADM (security administrator)	*SECADM, *AUDIT, *ALLOBJ
*SECOFR (security officer)	*NONE
*SYSOPR (system operator)	*SECADM, *AUDIT, *ALLOBJ

Note: For configuration of User profile as an Administrator, you can specify different options of user class and special authority in various ways as desired.

The above table displays few examples which mention the different ways in which you can configure administrator.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SailPoint IBM i Connector uses the following connection attributes:

Attribute	Description
IBM i Host	Host Name/IP Address of IBM i Host.
User Profile	This user is used for get/set operations on IBM i Host.
Password	This field contains the password for user specified as User Profile in application.
Proxy Server	Proxy server host and port details to be used to connect to IBM i Host.

Schema Attributes

Attribute	Description
Use TLS	Select this option if using TLS certificates.
Partitioning Statements	Criteria to specify the range of users to be downloaded. For example, if the range is specified as A-M, then this specifies that all the Users who are between A and M (including A and M) would be treated as one partition and downloaded. For more information, see Appendix B: Partitioning Aggregation.

Schema Attributes

This section describes the different schema attributes.

Account and Account - Group attributes

The following table lists the account and account - group attributes:

Attributes	Description
USRPRF	User profile.
AUDLVL	Audit Level.
PWDEXP	Set password to expired.
STATUS	Status
USRCLS	User class
ASTLVL	Assistance level.
CURLIB	Current library.
INLPGM	Initial program to call.
INLMNU	Initial menu.
LMTCPB	Limit capabilities.
TEXT	Text description
SPCAUT	Special authority
SPCENV	Special environment
DSPSGNINF	Display sign-on information
PWDEXPITV	Password expiration interval
LCLPWDMGMT	Local password management
LMTDEVSSN	Limit device sessions
KBDBUF	Keyboard buffering
MAXSTG	Maximum allowed storage
PTYLMT	Highest schedule priority

Attributes	Description
JOBD	Job description
GRPPRF	Group profile
OWNER	Owner
GRPAUT	Group authority
GRPAUTTYP	Group authority type
SUPGRPPRF	Supplemental groups
ACGCDE	Accounting code
OBJAUD	Object Auditing
MSGQ	Message queue
DLVRY	Delivery
SEV	Severity code filter
PRTDEV	Print device
OUTQ	Output queue
ATNPGM	Attention program
SRTSEQ	Sort sequence
LANGID	Language ID
CNTRYID	Country or Region ID
CCSID	Coded character set ID
CHRIDCTL	Character Identifier control
SETOBJATTR	Local job attributes
LOCALE	Locale
USROPT	User options
UID	User ID number
GID	Group ID number (Only for Groups)
HOMEDIR	Home directory.
AUTL	Authorization Lists
<i>If required user must add the following attributes manually to IBM i Connector schema after upgrading to IdentityIQ version 7.3</i>	
PWDLASTCHG	Date and time when the sign on password was changed
PREVSIGNON	Date and time of previous sign on
PWDEXPDATE	Date when password will expire
INVSIGNON	Number of unsuccessful sign on attempt
USREXPACT	User Expiration Action
USREXPDATE	User Expiration Date

Provisioning policy attributes

Attributes	Description
USREXPITV	User Expiration Interval
LSTUSEDATE	Last Used Date
CHGDATE	Date and time of the last change to the user profile
CTRBYUSER	Created by user

Provisioning policy attributes

This section lists the provisioning policy attributes of SailPoint IBM i Connector that allows to select the type of user, login, or group.

Attribute name	Description
Create Account	
USRPRF	User profile.
USRCLS	User class.
UID	User ID.
GRPPRF	Group profile.
PASSWORD	Password
PWDEXP	Set password to expired.
Create Group	
USRPRF	User profile.
USRCLS	User class.
GID	Group ID.
PWDEXP	Set password to expired.
Update Group	
USRCLS	User class.
GID	Group ID.

Additional information

This section describes the additional information related to the IBM i Connector.

Direct Permissions

The SailPoint IBM i Connector supports reading direct permissions assigned to user profiles for different resources on IBM i system. The **Include Permissions** check box of account or group schema must be selected to get direct permissions during Aggregation.

The SailPoint IBM i Connector also supports revoking account direct permissions through certifications. The **directPermissionObjectType** configuration parameter contains list of various object types owned by the user. By default the following direct permission object types are configured:

- *LIB
- *MSGQ
- *FILE
- *PGM
- *CMD
- *MENU
- *AUTL
- *JOBQ

This above list can be modified as required for different object types. For example:

```
<entry key="directPermission" value="true" />
<entry key="directPermissionObjectType">
    <value>
        <List>
            <String>*LIB</String>
            <String>*MSGQ</String>
        </List>
    </value>
</entry>
```

Perform the following procedure to display direct permissions in IBM i system:

1. Execute DSPOBJAUT.
2. Enter the **Object** and **Object Type**.
3. Press **F11** twice.

Upgrade Consideration

(Optional) To manage the authorization list, add the following attributes in the mentioned schema level with appropriate properties while upgrading to IdentityIQ version 7.3:

- **Account schema:** AUTL with type string and properties as Managed, Entitlement, Multi-Valued
- **Group schema:** AUTL with type string and properties as Entitlement, Multi-Valued

Create TLS 1.2 Communication between IdentityIQ and IBM i system

Perform the following to enable TLS communication between IdentityIQ and IBM i server, for securing TLS connection for IBM i system

Note: For a Java client to connect using SSL and self-signed certificates, install the certificate into the JVM keystore.

1. Export server certificate and copy the exported .cacrt file to the host running IdentityIQ.
2. At the client computer execute the following command from the /jre/lib/security path:
`keytool -import -alias aliasName -keystore cacerts -trustcacerts -file <absolute path of certificate>`

Additional information

In the preceding command line, *aliasName* is the name of the alias.

3. Login to IdentityIQ.
4. Create the application for IBM i by selecting **Use SSL** option and provide all the required values.
5. Click on Test Connection and save the application.

Aggregation of Group Profile as part of Account Aggregation

To aggregate Group profiles as part of account aggregation enter the following entry in application xml:

```
<entry key="includeUserInfo" value="ALL"/>
```

Chapter 15: SailPoint JDBC Connector

The following topics are discussed in this chapter:

Overview.....	143
Supported features	143
Supported Managed Systems	144
Pre-requisites	144
Administrator permissions	144
Configuration parameters.....	144
Additional configuration parameters	147
Schema Attributes	147
Troubleshooting.....	148

Overview

The SailPoint JDBC Connector is used for Read/Write operations on the data of JDBC- enabled database engines. This connector supports flat table data. To handle complex, multi-table data, you need to define a rule and a more complex SQL statement.

This connector can be configured to enable the automatic discovery of schema attributes. See “Schema Attributes” on page 147.

Supported features

SailPoint JDBC Connector supports the following features:

- Account Management
 - Manage JDBC Users as Account
 - Aggregate, Delta Aggregation, Partitioning Aggregation, Refresh Accounts, Discover Schema
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Group Management
 - Manage JDBC Groups
 - Aggregate, Delta Aggregation, Refresh Groups
 - Create, Update, Delete

Note: For Group schema, multiple-group objects are supported.

Configuration parameters

SailPoint supports the following additional JDBC Connector features:

- Ability to provide the SQL statement or stored procedure during application configuration for automatic discovery of group schema attributes from same or different database used for the account schema.
- Ability to define provisioning rule(s) called for each row in the data file to provision account and group attributes.
- Ability to define separate provisioning rule for specific operation called for each row in the data file to provision account and group attributes. Operation that include are Enable, Disable, Unlock, Delete, Create, and Modify.

Note: An example of a provisioning rule is located in `examplerules.xml` file.

References

- Appendix A: Delta Aggregation
- Appendix B: Partitioning Aggregation

Supported Managed Systems

SailPoint JDBC Connector supports the following Managed System:

- Any database having JDBC Driver. For example, MySQL, Oracle, DB2, SQL Server and Sybase

JDBC connector now supports managing database hosted on AWS RDS and connecting through JDBC driver.

Pre-requisites

An appropriate JDBC driver for the database.

Administrator permissions

The JDBC application will use the database user context to support the aggregation and other provisioning operations. The database user mentioned in the application configuration must have appropriate rights to fetch and set data related to the entities and attributes mentioned in the JDBC SQL query that is, Account, Group, Entitlements/Direct Permission and so on.

Configuration parameters

This section contains the information that the connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The JDBC connector uses the following connection attributes under different tabs (Settings, Merging, Iteration Partitioning and Delta Aggregation):

Settings

Attributes	Description
JDBC Connection Settings	
Connection User*	The user with which to connect to the host.

Attributes	Description	
Connection Password	The password associated with the specified user.	
Database URL*	The URL with which to connect to the database.	
JDBC Driver*	The JDBC driver class path.	
Query Settings		
Test Connection SQL	The SQL Statement for Test Connection.	
	Use stored procedure	Select this option to use stored procedure instead of normal sql statement.
SQL Statement*	The SQL Statement attribute can be used to customize the selected statement that is generated when iterating over objects. You can specify the exact SQL Statement that is executed if you want to filter out objects or only want to select a few objects from a table. Additionally, if you need to perform joins between more than one table, it's impossible to describe with the schema alone.	
	Use stored procedure	Select this option to use stored procedure instead of normal sql statement.
getObjectSQL	The object SQL statement.	
	Use stored procedure	Select this option to use stored procedure instead of normal sql statement.
useExecuteQuery	Use Statement.executeQuery() instead of the default Statement.execute() Note: During Delta Aggregation with useExecuteQuery option enabled, ensure that the driver used supports the Execute Query feature.	
Direct Permission Execute Query	Enable this option to execute the query for direct permission.	
Get Direct Perm Object SQL	Direct Permission Execute Query is used to pull the direct permission data from permission table. Permission table must contain at least Identity attribute column. The permission data is pulled by referring the identity attribute in the column at the time of aggregation via main SQL query in which the identity attribute is mentioned. Note: Query must be written in such a way that ResultSet data must contain first column as Target , second column as Permission and third column as annotation (optional) . For example, SELECT column4 AS TARGET,column5 AS PERMISSION FROM Permission p WHERE CONCAT(TRIM(CONCAT(p.column1,' ')), TRIM(p.column2)) = '\${identity}'; Here table name is Permission and \$(identity) is Identity attribute.	
	Use stored procedure	Select this option to use stored procedure instead of normal sql statement.

Configuration parameters

Attributes	Description
usePrepareCall	<p>This parameter must be set from the application debug page when a stored procedure is called from Query Settings in JDBC connector.</p> <p>The entry key must be in the following format:</p> <pre><entry key="usePrepareCall"> <value> <Boolean>true</Boolean> </value> </entry></pre>

Merging

Attributes	Description
Data needs to be merged	<p>Select this option if the data for a single object spans multiple lines.</p> <p>This option enables the connector to verify the order of the data returned from the database when merging to prevent data loss. When merging, it is very important to have the ORDER BY clause in your SQL statement to prevent out of order errors.</p>
Index Column	Name of the index column that will be used when finding like objects in the dataset.
Which columns should be merged?	Names of the columns from the file from which values must be merged.
Note: User must discover the schema to get the suggested column values in index and merge columns for selection. Discover schema populates the values in multi-suggest attribute drop-down of index and merge columns which have the auto complete facility.	

Iteration Partitioning

Attributes	Description
Partitioning Enabled	Select this checkbox to configure and enable partitioning.
Partitioning Statements	<p>Enter the list of sql/stored procedure statements that must be executed when partitioning. The statements must include all of the rows and each line/statement so it can be proceeded in separate threads and/or multiple hosts.</p> <p>For more information, see Appendix B: Partitioning Aggregation.</p>
Use stored procedure	Select this option to use stored procedure instead of normal sql statement.

Delta Aggregation

Attributes	Description
Delta Aggregation Enabled	<p>To use Delta Aggregation feature, the Delta Aggregation Enabled field must be selected.</p> <p>The Database Table Containing Delta Changes attribute must be provided with the value of table name in which delta changes are captured. This table must have read and write permissions. This field is not mandatory.</p> <p>The Delta Aggregation SQL can be used when the alias is used in main SQL.</p> <p>For more information, see Appendix A: Delta Aggregation.</p>
Use stored procedure	Select this option to use stored procedure instead of normal sql statement.

Additional configuration parameters

The following parameters are not displayed on UI but are used to tuning jdbc pooling:

Parameters	Description
pool.maxWait	Maximum time to wait for connection to become available in millisecond. Default: <entry key="pool.maxWait" value="60000"/>
pool.evictRuns	Wait time between closing idle connections in milliseconds. Default: <entry key="pool.evictRuns" value="300000"/>
pool.maxActive	Maximum number of connections. Default: <entry key="pool.maxActive" value="10"/>
pool.setMaxIdle	Maximum number of idle connections Default: <entry key="pool.setMaxIdle" value="10"/>
pool.minEvictIdle	Minimum idle time to close connection in milliseconds. Default: <entry key="pool.minEvictIdle" value="10"/>

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface and supports multiple types of objects, account and any number of group application object types. Account objects are used when building identities Link objects. Additional schema definitions can be used when building AccountGroup objects which are used to hold entitlements shared across identities.

The JDBC connector's most important attribute is the SQL Statement. In many cases this is a stored procedure. (call mystoredProcedure). In other cases it is select from a table with any number of joins included. If this

Troubleshooting

connector is configured to use the automatic discovery function, it connects to the database and executes the statement provided and then uses the meta-data returned from the result to build the column names.

Troubleshooting

1 - Connection getting locked up

By default the JDBC Connector works in pooling connection mode. The connection gets locked up for already configured application account when password of the account changes dynamically from managed system without displaying any warning message.

Resolution: Add the following xml in the application.xml file:

```
<entry key="pool.disablePooling">
    <value>
        <Boolean>true</Boolean>
    </value>
</entry>
```

2 - JDBC Connector fails to discover schema with correct alias names in case of MYSQL Database

JDBC Connector fails to discover schema with correct alias names in case of MYSQL Database.

Resolution: Add **useOldAliasMetadataBehavior** parameter at the end of the database URL (after the SID (Database Name)) and set it to true.

For example, if the database url in the UI is URL = "jdbc:mysql://localhost:3306/mydb" then, prefix a ? to the **useOldAliasMetadataBehavior** parameter and add this parameter at the end after setting it to true.

The URL would be as follows:

```
URL = "jdbc:mysql://localhost:3306/mydb?useOldAliasMetadataBehavior=true"
```

3 - During aggregation an error message appears

The following error message appears during aggregation:

```
Exception during aggregation. Reason: java.lang.RuntimeException:  
sailpoint.connector.ConnectorException: Data out of order exception. Data should be  
sorted in ascending order. Last identifier '7' current '15'
```

Resolution: Add the **disableOrderingCheck** attribute as follows to the application debug page:

```
<entry key="disableOrderingCheck">
    <value>
        <Boolean>true</Boolean>
    </value>
</entry>
```

4 - During aggregation for large datasets an error messages appears

During aggregation for large datasets the following error message appears:

Out of memory

Resolution: For large datasets add the following entries to the application debug page:

```
<entry key="resultSetFetchSize" value="1000"/>
<entry key="statementFetchSize" value="1000"/>
<entry key="resultSetType" value="TYPE_FORWARD_ONLY"/>
```

SailPoint recommends that the values of **resultSetFetchSize** and **statementFetchSize** keys are same.

The **resultSetType** key with value as TYPE_FORWARD_ONLY, ensures that the dataset is read in forward manner only.

Troubleshooting

Chapter 16: SailPoint Jive Connector

The following topics are discussed in this chapter:

Overview.....	151
Supported features	151
Pre-requisites	152
Administrator permission	152
Configuration parameters.....	152
Schema attributes	152
Account attributes	152
Group attributes.....	154
Provisioning Policy attributes.....	154
Create account attributes	154
Create group attributes	155
Additional information	155
Troubleshooting	156

Overview

SailPoint Jive Connector was developed to manage Jive User Accounts and Security Groups (Security Groups may have permissions such as Full Access, Manage Community, Manage System, Moderate Content, Manage Users, Manage Groups).

Supported features

SailPoint Jive Connector supports the following features:

- Account Management
 - Manage Jive Person as Account
 - Aggregate, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable
 - Add/Remove Entitlements
- Account - Group Management
 - Manage Jive SecurityGroups as Account - Groups
 - Aggregate, Refresh Groups
 - Create, Update, Delete

Pre-requisites

1. Jive connector requires the following libraries in class path:
 - httpcore-4.2.1.jar
 - httpclient-4.2.1.jar
 - httpclient-cache-4.2.1.jar
 - commons-logging-1.1.1.jar
 - commons-codec-1.6.jar
 - gson-2.1.jar
2. Jive software should be up and running.
3. Administrator should be configured to have proper access rights for modifying Jive users.

Administrator permission

Administrator should be configured to have proper access rights for modifying Jive users.

Configuration parameters

The following table lists the configuration parameters of Jive Connector:

Parameters	Description
Jive URL	URL for accessing Jive (For example, http://jive1.jivedev.com)
User Name	User Name used for logging into Jive with sufficient rights (administrator credentials.)
Password	User's password.
Page Size	The maximum size of each data set when querying over large number of objects. Default and max value is 100.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
Id	Identifier (unique within an object type and Jive instance) of the object.
username	The login username for the person. This field is required for person creation, but cannot be changed on an update.
name	Name components (familyName, givenName and formatted) for the person.

Attributes	Description
location	Geographic location of the person.
type	The object type of the object (person).
displayName	Formatted full name of this person, suitable for use in UI presentation. If the person has privacy settings that do not allow you to see his or her name, this will be the Jive username instead.
published	Date and time when the person was originally created. Visible only to Jive administrators or on your person object.
thumbnailurl	URL of the thumbnail (avatar) image for the person.
followingCount	Number of people the person is following.
followerCount	Number of people following the object.
jive	Jive extensions to OpenSocial person object (it is a complex JSON object).
phoneNumbers	Phone numbers belonging to the person, with standard types: fax, home, mobile, other, pager, work.
Phone Number	Primary phone number of the user.
Mobile Phone Number	Mobile phone number of the user.
updated	Date and time the person was most recently updated.
addresses	Postal addresses belonging to the person, with standard types home, other, pobox, work and value type of address.
work_country	Name of the country where person is working.
work_locality	Name of the city where person is working.
work_street_1	Street 1 name where person is working.
work_street_2	Street 2 name where person is working.
work_postalCode	Zip/Postal code where person is working.
work_region	Name of State or Province where person is working.
home_country	Name of the country where person lives.
home_locality	Name of the city where person lives.
home_postalCode	Zip/Postal code where person lives.
home_region	Name of state or province where person lives.
home_street_1	Name of street 1 where person lives.
home_street_2	Name of street 2 where person lives.
givenName	First name of the person.
familyName	Last name of the person.
Department	The person's department name.
Company	The person's company name.
Title	The person's job title name.
Biography	The person's biography.

Provisioning Policy attributes

Attributes	Description
Expertise	The person's expertise.
Email	This attribute will be used for updating primary email address of a person.
emails	Email addresses belonging to this person, with standard types home, other, work and value type of string.
Alternate Email	Alternate email address of the user.

Group attributes

The following table lists the group attributes:

Attributes	Description
id	Identifier (unique within an object type and Jive instance) of the object.
name	Name of the security group.
type	Object type of this object ("securityGroup").
description	Description of this security group.
administratorCount	Number of administrative members in the security group.
memberCount	Number of regular members of the security group.
published	Date and time the security group was initially created.
updated	Date and time the security group was last updated.
federated	Flag indicating that the membership of the group is federated with an external directory service.
Admins	List of administrative members in the security group.
Members	List of regular members of the security group.

Provisioning Policy attributes

This section lists the different policy attributes of Jive Connector.

Note: All the attributes marked with * sign are the mandatory attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
username*	The login username for this person. This field is required for person creation, but cannot be changed on an update.
givenName*	First name of the person.

Attributes	Description
familyName*	Last name of the person.
Email*	Primary email address of a person.
password*	Password of a person.

To provision **phoneNumbers** and **emails** in Jive, add the following attributes in provisioning policy:

Attributes	Description
Primary.PhoneNumber	Primary phone number of a person.
Mobile.PhoneNumber	Mobile number of a person.
AlternateEmail	Alternate email address of a person.

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

Attributes	Description
name*	Name of the group.
description	Text describing the user.

Additional information

This section describes the additional information related to the Jive Connector.

Note: To enable logging, specify the logging

```
log4j.logger.openconnector.connector.JiveConnector in the log4j.properties
file. For example, logging
log4j.logger.openconnector.connector.JiveConnector=debug
```

Configuration settings

The **TimedCache** attribute is used to set the time period to hold the Account-Group memberships information in a cache. This is a configurable field and it has default value of 60 minutes.

To change the TimedCache timeout the user must add the following entry into the application by navigating to the debug page:

```
<entry key="timedCacheTimeout" value="90"/>
```

In the above entry **timedCacheTimeout** is **key** and **value** is the **value** of that key in minutes. For example, 90 minutes.

Upgrade considerations

When upgrading IdentityIQ to version 7.1 Patch 5,

- To provision phone numbers in Jive, add the following attributes in schema manually if they are not existing:
 - **phoneNumbers**
 - **Phone Number** (for provisioning primary phone number)
 - **Mobile Phone Number** (for provisioning mobile phone number)
- To provision emails in Jive, add the following attributes in schema manually if they are not existing:
 - **emails**
 - **Email** (for provisioning primary email)
 - **Alternate Email** (for provisioning alternate email)

Troubleshooting

1 - Inconsistent UI and API results.

The Jive software has a system property called `jive.pageCached.enabled` which is used to enable/disable the page caching on Jive. By default the value of this system property is **True**. This issue appears when user is performing add/remove entitlements operation.

Resolution: Set the `jive.pageCached.enabled` system property value to **False**.

Chapter 17: SailPoint LDAP Connector

The following topics are discussed in this chapter:

Overview.....	157
Supported features	157
Supported Managed Systems	158
Pre-requisites	160
Administrator permissions	160
Configuration parameters.....	160
Additional configuration parameter.....	161
Configuring Account Search Scope	161
Configuring Group Search Scope.....	163
Schema attributes	163
Account attributes	164
Group attributes.....	167
posixgroup and nisNetgroup Attributes.....	168
Group Membership attribute	169
Group Entitlement attribute	169
Provisioning Policy attributes	170
Additional information	171
Adding additional group types	171
Using Novell eDirectory as a Pass-through Authentication Source	173
Troubleshooting.....	173

Overview

This connector was developed using the LDAP RFC. The LDAP Connector must plug into almost any LDAP server with no customization. The LDAP Connector now supports provisioning of users and entitlements along with the retrieval of LDAP account and group object classes.

Supported features

SailPoint LDAP Connector supports the following features:

- Account Management
 - Manage LDAP Users as Account
 - Delta Aggregation (SunOne -Direct, IBM Tivoli DS- Direct, ADAM- Direct)
 - Aggregate, Refresh Accounts, Partitioning Aggregation, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements

Overview

Note: The Enable and Disable feature is supported only for ADAM-Direct, Novell eDirectory, Oracle Internet Directory and SunOne - Direct managed systems.

Note: The Unlock feature is supported only for Novell eDirectory, Oracle Internet Directory and SunOne - Direct managed systems.

- Account - Group Management

- Manage LDAP Groups as Account - Groups
- Delta Aggregation (SunOne -Direct, IBM Tivoli DS - Direct, ADAM- Direct)
- Aggregate, Refresh Group
- Create, Update, Delete

References

- “Appendix A: Delta Aggregation”
- “Appendix B: Partitioning Aggregation”

Supported Managed Systems

SailPoint LDAP Connector supports the following Managed Systems:

- Microsoft ADAM 2016, 2012 R2, 2012
- OpenLDAP version 2.4, 2.3
- ODSEE 11g
- IBM Tivoli Directory Server version 6.4, 6.3
- Novell eDirectory version 9.0, 8.8
- Oracle Internet Directory version 12c and 11gR2

For each of the supported managed systems we have application types already existing in the connector registry as mentioned in the following section.

LDAP Connector Application Types

In order to speed up the process of building an application quickly, this connector comes with the following default LDAP application types to manage the corresponding directory server:

Application Types	Description
ADAM - Direct	Manages ADAM directory server
SunOne - Direct	Manages SunOne directory server
IBM Tivoli DS - Direct	Manages Tivoli directory server
Novell eDirectory - Direct	Manages Novell eDirectory server
Oracle Internet Directory - Direct	Manages Oracle Internet directory server
OpenLDAP - Direct	Manages OpenLDAP directory server

Object Types Managed

Each of the application types has been pre-configured to manage commonly used object classes and their attributes. For instance, the application schema of ADAM directory server has been configured for user and

group object classes. Table 1 and Table 2 displays the mapping for various application types. Custom object classes may be mapped by modifying the corresponding application schema.

The following table displays the object classes mapped for each of the LDAP application types:

Table 1—Object classes mapped for each of the LDAP application types

Application types	Objectclass mapped for accounts	Objectclass mapped for groups	Group membership attributes	Group entitlement attribute
ADAM - Direct	user	group	member	groups
SunOne - Direct	inetOrgPerson	groupofUniqueNames	uniqueMember	groups
IBM Tivoli DS - Direct	inetOrgPerson	groupofUniqueNames	uniqueMember	groups
Novell eDirectory - Direct	inetOrgPerson	groupofUniqueNames	uniqueMember	groups
Oracle Internet Directory - Direct	inetOrgPerson	groupofUniqueNames	uniqueMember	groups
OpenLDAP - Direct	inetOrgPerson	groupofUniqueNames	uniqueMember	groups

The following table displays the object classes mapped for nisNetgroup and posixgroup for SunOne, Tivoli DS and OpenLDAP application types:

Table 2—Object classes mapped for nisNetgroup and posixgroup for LDAP application types

Application types	Objectclass mapped for groups	Group membership attributes	Group entitlement attribute
Sun One - Direct <i>OR</i> IBM Tivoli DS - Direct <i>OR</i> OpenLDAP - Direct	nisNetgroup	nisNetgroupTriple	nisNetgroups
	posixgroup	memberUid	posixgroups

Create TLS communication between IdentityIQ and LDAP Server

If you want secure TLS connection for LDAP, TLS communication needs to be enabled between IdentityIQ and LDAP Server. For a Java client to connect using TLS and self-signed certificates, you have to install the certificate into the JVM keystore.

To create TLS communication between IdentityIQ and LDAP Server, perform the following:

1. Export server certificate and copy the exported .cer file to the Java client computer (IdentityIQ computer).
2. At the client computer execute the following command from the bin directory of JDK:

```
keytool -importcerts -trustcacert -alias aliasName -file <absolute path of certificate> -keystore <JAVA_HOME>/jre/lib/security/cacerts
```

In the preceding command line, aliasName is the name of the alias.
3. Login to IdentityIQ.
4. Create the application for LDAP application type and provide all the required values after selecting the **Use TLS** checkbox.
5. Click on **Test Connection** and save the application.

Pre-requisites

SailPoint LDAP Connector requires that the directory server has the administrator credentials.

Ensure that the following pre-requisites are satisfied for the respective directory servers:

- (*For SunOne Directory Server*) Global Password Policy must have the **Require Password Change at First Login and After Reset** option selected.

Note: **For self change password, add the CURRENT_PASSWORD attribute to the featureString in the application schema.**

- (*For IBM Tivoli*)

- Password Policy must be enabled and assigned to the user.

- The **User can change Password** option must be selected.

- The **User must change Password after reset** option must be selected.

Note: **- Add the following attribute to the system configuration schema and set it to true:**

`<entry key="requireOldPasswordAtChange" value="true"/>`

- For self change password, add the CURRENT_PASSWORD attribute to the featureString in the application schema.

- (*For Novell eDirectory*)

- Universal Password must be configured.

- Enable Universal Password in Password Policy.

- The **Allow users to initiate password change** option must be selected.

- The **Do not expire the user's password when the administrator sets the password** option must not be selected.

- Grace logins can be applied according to password policy added to the application schema.

- (*For Oracle Internet Directory*) Assign password policy to user.

Note: **- Add the following attribute to the system configuration schema:**

`<entry key="requireOldPasswordAtChange" value="true"/>`

- For self change password, add the CURRENT_PASSWORD attribute to the featureString in the application schema.

Administrator permissions

SailPoint LDAP Connector must have the read /write privilege's over the directory information tree in order to manage the LDAP data.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The LDAP connector uses the following connection attributes:

Parameters	Description
useSSL	Specifies if the connection is over TLS.
authorizationType	The authorization type to use when connecting to the server.
user*	The user to connect as a DN string such as Administrator.
password	Password for the administrator account.
port*	Port the server is listening through.
host*	Host of the LDAP server.
pageSize	Number of objects to get, per page, when iterating over large numbers of objects. Default is 500.
authenticationSearchAttributes	Attributes used when authenticating the application using pass - through authentication.

Additional configuration parameter

Attributes	Description
expiredPasswordErrorMessages	List of possible error strings (full or partial) returned by the LDAP server that indicates the password expiration for the user.

Note: LDAP connector supports all jndi system properties. For more information, see <https://docs.oracle.com/javase/jndi/tutorial/ldap/connect/config.html>

Following are examples with sample values:

```
<entry key="com.sun.jndi.ldap.connect.pool.maxsize" value="10"/>
<entry key="com.sun.jndi.ldap.connect.pool.protocol" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.timeout" value="20000"/>
<entry key="com.sun.jndi.ldap.connect.pool.initsize" value="5"/>
<entry key="com.sun.jndi.ldap.connect.pool.authentication" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.debug" value="fine"/>
<entry key="com.sun.jndi.ldap.connect.pool" value="true"/>
<entry key="com.sun.jndi.ldap.read.timeout" value="120000"/>
```

Configuring Account Search Scope

The searchDNs define list of distinguished names of the containers along with other relevant attributes which defines scope for this application. Each of these searchDNs is considered as a partition for parallel aggregation. Accounts and Groups can have different set of searchDNs to define different scope for each of them. In case the scope is not defined for Groups, it follows Accounts scope. Defining one search DN to the minimum is required to successfully configure application.

Attributes to be defined for searchDNs are as follows:

Attributes	Description
searchDN*	Distinguished Name of the container.

Configuration parameters

Attributes	Description
iterateSearchFilter	LDAP filter that defines scope for accounts/groups from this container.
Filter String	Filter which can filter object after they have been returned from the underlying directory.
groupMembershipSearchScope	<p>(Applicable only for account search scope) List of map which represents scope and filter for group membership of each different group-objectType (for example, posixgroup, nisNetgroup).</p> <p>If group membership search scope is not defined then search DN will be considered as scope for fetching group membership for that specific group-object type</p> <ul style="list-style-type: none"> • objectType*: Group ObjectType name Note: This object type Name must match with objectType name of the respective group schema. • groupMembershipSearchDN: (Optional) Multivalued attribute to define scope for group memberships. • groupMemberFilterString: (Optional) LDAP filter for groups membership.

Note: The LDAP connector uses the **groupMembershipSearchDN** attribute as the starting point in the directory to start searching for ALL group memberships. LDAP does not store a user's group references on the user so the LDAP connector must always do a separate query to return a list of all of the user's groups.

Following is an example for the attributes of Account Search Scope with Group Membership Search Scope

```

<entry key="searchDNs">
    <value>
        <List>
            <Map>
                <entry key="groupMembershipSearchScope">
                    <value>
                        <List>
                            <Map>
                                <entry key="groupMemberFilterString"
value="(&amp; (objectClass=posixgroup) (cn*))"/>
                                <entry key="groupMembershipSearchDN">
                                    <value>
                                        <List>
                                            <String>ou=Sales,dc=org,dc=com</String>
                                            <String>ou=HR,dc=org,dc=com</String>
                                        </List>
                                    </value>
                                </entry>
                                <entry key="objectType" value="posixgroup"/>
                            </Map>
                        </List>
                    </value>
                </entry>
                <entry key="iterateSearchFilter"
value="(&amp; (objectClass=inetOrgPerson) (cn=a*))"/>
                    <entry key="searchDN" value="ou=Canada,dc=org,dc=com"/>
                    <entry key="searchScope" value="SUBTREE"/>
                </entry>
            </Map>
        </List>
    </value>
</entry>
```

```

        </Map>
        <Map>
            <entry key="iterateSearchFilter"
value="(&amp; (objectClass=inetOrgPerson) (cn=b*))" />
            <entry key="searchDN" value="ou=America,dc=org,dc=com"/>
            <entry key="searchScope" value="SUBTREE"/>
        </Map>
    </List>
</value>
</entry>

```

Configuring Group Search Scope

The <objectType>.searchDNs define list of distinguished names of the containers along with other relevant attributes which define scope and filter for group aggregation.

Attributes	Description
Search DN*	Defines scope of group aggregation for mentioned objectType.
Iterate Search Filter	LDAP filter for group aggregation.
Filter String	Filter which can filter object after they have been returned from the underlying directory.

Following is an example for attributes of Group Search Scope

```

<entry key="posixgroup.searchDNs">
    <value>
        <List>
            <Map>
                <entry key="iterateSearchFilter" value="(&amp; (objectclass=posixgroup) )" />
                <entry key="searchDN" value="ou=HR,dc=org,dc=com"/>
                <entry key="searchScope" value="SUBTREE"/>
            </Map>
            <Map>
                <entry key="iterateSearchFilter" value="(&amp; (objectclass=posixgroup) )" />
                <entry key="searchDN" value="ou=Sales,dc=org,dc=com"/>
                <entry key="searchScope" value="SUBTREE"/>
            </Map>
        </List>
    </value>
</entry>

```

Note: The name of the objectType must be same as the name of the objectType of respective group schema for which the scope is defined.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 3—LDAP Connector - Account Attributes

Name	Description
businessCategory	Types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: “engineering”, “finance”, and “sales”.
carLicense	License plate or vehicle registration number associated with the user.
cn	Names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: “Martin K Smith”, “Marty Smith” and “printer12”.
dn	Distinguished name by which the user is known. Note: The LDAP connector considers distinguished name as the default Native Identity attribute and is not customizable.
departmentNumber	Numerical designation for a department within your enterprise.
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: “Updates are done every Saturday, at 1am.”, and “distribution list for sales”.
destinationIndicator	Country and city strings associated with the object (the addressee) required to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: “AASD” as a destination indicator for Sydney, Australia. “GBLD” as a destination indicator for London, United Kingdom. Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.
displayName	Preferred name to be used for this person throughout the application.
employeeNumber	Numerical identification key for this person within you enterprise.
employeeType	Descriptive type for this user, for example, contractor, full time, or part time.
facsimileTelephoneNumber	Telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute.
givenName	Name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. Examples: “John”, “Sue”, and “David”.
groups	List of groups of which this person is a member. Example: “Sales” or “Engineering”

Table 3—LDAP Connector - Account Attributes (Continued)

Name	Description
posixgroups	(Applicable only for IBM Tivoli, OpenLDAP Direct, and SunOne Direct) List of posixgroups of which this person is a member. Example: "Sales" or "Engineering" For more information, see " posixgroup and nisNetgroup Attributes" on page 168.
nisNetgroups	(Applicable only for IBM Tivoli, OpenLDAP Direct, and SunOne Direct) List of nisNetgroup of which this person is a member. Example: "Sales" or "Engineering" For more information, see " posixgroup and nisNetgroup Attributes" on page 168.
homePhone	Employees home phone number.
homePostalAddress	Employees mailing address.
initials	Strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute. Examples: "J. A." and "J".
internationalISDNNumber	Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. Example: "0198 444 444".
l	Names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. Examples: "Austin", "Chicago", and "Brisbane".
mail	RFC822 mailbox for the user.
manager	Distinguished name of the manager to whom this person reports.
mobile	Mobile telephone number of this person.
o	Names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated.".
ou	Names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies".
pager	Telephone number of this persons pager.
physicalDeliveryOfficeName	Names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E".
postOfficeBox	Postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27".

Schema attributes

Table 3—LDAP Connector - Account Attributes (Continued)

Name	Description
postalAddress	Addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: “1111 Elm St.\$Austin\$Texas\$USA”.
postalCode	Codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: “78664”, to identify Pflugerville, TX, in the USA.
preferredDeliveryMethod	Indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: “mhs \$ telephone”.
preferredLanguage	Preferred written or spoken language of this person.
pwdReset*	(Applicable only for IBM Tivoli, OpenLDAP, and Oracle Internet Directory) Specifies whether the password has been reset by administrator. Note: Must be added manually to support password reset.
pwdLastSet*	(Applicable only for ADAM) User password last set time. Note: Must be added manually to support password reset.
passwordExpirationTime*	(Applicable only for Novell eDirectory and SunOne managed system) Users password expiration time. Note: Must be added manually to support password reset for Novell eDirectory.
registeredAddress	Postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: “Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA”
roomNumber	Room or office number or this persons normal work location.
secretary	Distinguished name of this persons secretary.
seeAlso	Distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object “cn=Elvis Presley,ou=employee,o=xyz\, Inc.” is related to the role objects “cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc.” and “cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.”. Since the role objects are related to the person object, the ‘seeAlso’ attribute will contain the distinguished name of each role object as separate values.
sn	Name strings for surnames, or family names. Each string is one value of this multi-valued attribute. Example: “Smith”
st	Full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”

Table 3—LDAP Connector - Account Attributes (Continued)

Name	Description
street	Site information from a postal address (i.e., the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”
telephoneNumber	Telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute.
teletexTerminalIdentifier	The withdrawal of recommendation F.200 has resulted in the withdrawal of this attribute.
telexNumber	Sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute
title	Persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”
uid	Computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”
objectClass	The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”.

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 4—LDAP Connector - Group Attributes

Name	Description
cn	Names of object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: “Martin K Smith”, “Marty Smith” and “printer12”
uniqueMember	Groups to which this person is a unique member.
dn	Directory path to the object.
o	Names of organization. Each name is one value of this multi-valued attribute. Examples: “xyz”, “xyz Technologies, Inc.”, and “xyz, Incorporated.”
ou	Names of organizational unit. Each name is one value of this multi-valued attribute. Examples: “Sales”, “Human Resources”, and “Information Technologies”
owner	Distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is “cn=All Employees, ou=Mailing List,o=xyz, Inc.”, is owned by the Human Resources Director. Therefore, the value of the ‘owner’ attribute within the mailing list object, would be the DN of the director (role): “cn=Human Resources Director,ou=employee,o=xyz, Inc.”.

Schema attributes

Table 4—LDAP Connector - Group Attributes (Continued)

Name	Description
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: “Updates are done every Saturday, at 1am.”, and “distribution list for sales”

posixgroup and nisNetgroup Attributes

Names	Description
nisNetgroup Attributes	
cn	Names of objects. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: Martin K Smith, Marty Smith and printer12
nisNetgroupTriple	Unique member of a nisNetgroup.
dn*	Directory path to the object.
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute.
posixgroup Attributes	
cn	Names of objects. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: Martin K Smith, Marty Smith and printer12
memberUid	Unique member of a posixgroup.
gidNumber*	Integer value that uniquely identifies a group in an administrative domain.
dn*	Directory path to the object.
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute.

Additional group schema configuration attributes

Entry Key	Description
*groupMemberAttribute	Name of Group Membership attribute of account. Example: posixgroup: memberUid nisNetgroup: nisNetgroupTriple groupOfUniqueNames: uniqueMember

Entry Key	Description
memberAttribute	<p>Attribute name or distinguished name which is stored as value of the groupMemberAttribute. Values: cn, uid or dn</p> <p>Example: groupMemberAttribute for posixgroup is memberUid against which the values of account can be cn or uid, so for posixgroup groupMemberAttribute name is memberUid and memberAttribute names are cn or uid or both.</p> <p>Similarly for groupOfUniqueNames, groupMemberAttribute is uniqueMember and memberAttribute name is dn.</p> <p>Note: One or more than one memberAttribute can be configured as given in the above example of sample schema for sudoRole under “ Adding additional group types” on page 171.</p>
memberPrefix	<p>Value for this field is required if we have any prefix string before the memberAttribute value.</p> <p>For example: (,user 1,)</p> <p>Here prefix is “(,”</p> <p>Note: The memberPrefix attribute is not required if the value of the member attribute is ‘dn’.</p>
memberSuffix	<p>Value for this field is required if we have any suffix string after the memberAttribute value.</p> <p>For example: (,user 1,)</p> <p>Here suffix is “,)”</p> <p>Note: The memberSuffix attribute is not required if the value of the member attribute is ‘dn’.</p>

Note: Value mentioned against “displayAttribute” and “identityAttribute” must be present in Group Schema attributes.

Group Membership attribute

The group membership attribute has been implicitly mapped for the various application types. This attribute and its value can be seen in the application page. Refer to Table 1 and Table 2 for the group membership attribute mapped for each application type. This attribute can be changed from the default to a membership attribute specific to the custom object class mapped. For instance, if the groupOfUniqueNames is the default object class that has been mapped in the application schema for managing groups, then the default group membership attribute can be changed from uniquemember to member if groupOfNames is mapped in the application schema to manage groups.

Group Entitlement attribute

By default, all application types have the groups attribute mapped as the default entitlement attribute. This attribute is simply a placeholder to contain user/group memberships. While creating a new group aggregation task for the application, you would need to specify the value groups in the group account attribute text box in the group aggregation task.

Provisioning Policy attributes

The following table lists the provisioning policy attributes:

Attribute	Description
Account creation	
dn	Distinguished name of the user to be created.
password	Password of the user to be created.
CN	Full name of the user to be created. For example, Martin K Smith , Marty Smith and printer12 .
givenName	First Name of the user to be created.
SN	Last name of the user to be created.
During creation of account ensure that the value of "Full Name" must be equal to value of "CN" in "dn".	
Example: If value of "dn" is CN=Jeff Smith,OU=Sales,DC=Fabrikam,DC=COM then value of "Full Name" must be "Jeff Smith"	
Create group (groupOfUniqueNames)	
dn	Distinguished name of the user to be created.
uniqueMember	The distinguished name for the member of a group.
description	Description of the group to be created.
Update group (groupOfUniqueNames)	
description	Description of the group to be created.
Create posixgroup	
dn	Distinguished name of the user to be created.
gidNumber	Contains an integer value that uniquely identifies a group in an administrative domain.
description	Description of the posixgroup to be created.
Update posixgroup	
description	Description of the posixgroup to be created.
Create nisNetgroup	
dn	Distinguished name of the user to be created.
description	Description of the nisNetgroup to be created.
Update nisNetgroup	
description	Description of the nisNetgroup to be created.

Configuring group provisioning policy for new group

1. Ensure that we have added new group schema in application configuration.
2. Identify required attributes for provisioning operation of newly added group.
3. If required add appropriate provisioning policy for create and update operation of that group.
4. Perform Provisioning operations.

Additional information

This section describes the additional information related to the LDAP Connector.

Adding additional group types

Following application types of LDAP support additional group types:

- SunOne - Direct
- IBM Tivoli DS - Direct
- OpenLDAP - Direct

Perform the following steps to configure additional group type:

1. After upgrading IdentityIQ to version 7.3, from UI navigate to account schema of the application and save the application.
2. Add required group schema.

Following is a sample schema for sudoRole:

```
<Schema aggregationType="group" created="" displayAttribute="cn"
featuresString="PROVISIONING" id="" identityAttribute="dn" instanceAttribute=""
modified="" nativeObjectType="sudoRole" objectType="sudoRole">
<AttributeDefinition name="cn" type="string">
<Description>common name(s) for which the entity is known by</Description>
</AttributeDefinition>
<AttributeDefinition name="dn" type="string">
<Description>Directory Path</Description>
</AttributeDefinition>
<AttributeDefinition name="ou" type="string">
<Description>organizational unit this object belongs to</Description>
</AttributeDefinition>
<AttributeDefinition name="description" type="string">
<Description>descriptive information</Description>
</AttributeDefinition>
<AttributeDefinition multi="true" name="sudoUser" type="string">
<Description>unique member of a sudoRole </Description>
</AttributeDefinition>
<Attributes>
<Map>
<entry key="groupMemberAttribute" value="sudoUser"/>
<entry key="memberAttribute">
```

Additional information

```
<value>
  <List>
    <String>cn</String>
    <String>uid</String>
  </List>
</value>
</entry>
</Map>
</Attributes>
</Schema>
```

3. Add entitlement attribute to account schema

- For newly added group schema add entitlement attribute in account schema from UI.
 - a. Ensure that the following steps are performed after creating entitlement attribute:
 - Change the type of entitlement attribute from String to newly added group schema object Type
 - Mark Entitlement attribute as Managed, Entitlement and Multivalued
- (*Applicable only if nisnet and posix groups are configured as entitlements*) To manage **nisnet** and **posix** groups on upgraded application as group, perform the following:
 - Add nisnet and posix group schema from Debug and Save application.

Note: **Name of schema objectType must match with** `objectType="nisNetgroup"` **or** `objectType="posixgroup"` **respectively.**

- From UI navigate to account schema of the application and change the type of entitlement attribute from **String** to **posixgroup** or **nisNetgroup** accordingly and save the application.

Note the following:

- Multigroup application supports static groups as follows:
 - `groupofUniqueNames`
 - `groupOfNames`
 - `nisNetgroup`
 - `posixgroup`
 - `sudoRole`
- For nisNetgroup memberships in **nisNetgroupTriple** attribute all types of braces are supported. By default curly braces '{' are supported.

For example, `{host1,user1,}` `(host1,user1,)`, `[host1,user1,]`, **is a supported format.** Whereas, `<host1,user1,>` are unsupported formats. For instance, if user1 has a nisnetgroup memberships which is in the format any other than the curly braces, round braces and square braces then this entitlement would not be retrieved.

Note: **For instance, if user wants to use "<" angular braces the format should be like - "<"**

- The nisNetgroup entitlement is added only with the user portion of the **nisNetgroupTriple** attribute value. The domain and host counterpart are not incorporated.

For example, on SunOne, Tivoli and Open LDAP `{,user1,}` is the value of the nisNetgroupTriple attribute after adding an entitlement for user1 on a nisNetgroup.

Using Novell eDirectory as a Pass-through Authentication Source

If using the Novell eDirectory - Direct application type as a pass-through authentication source, remove the dn entry from the Authentication Search Attributes. Using the DN is currently not supported.

Troubleshooting

1 - During account aggregation IdentityIQ displays two entries

During account aggregation, if distinguished names are having space difference, IdentityIQ displays two entries for account.

Resolution: Ensure that there are no space differences in distinguished names.

2 - Group aggregation displays multiple entries for a group with multiple object class

If any group is having multiple object class associated with it then after group aggregation multiple entries are displayed for that respective group.

3 - During account aggregation, nisNetgroup membership is not displayed

Latest version of OpenLDAP server on windows (2.4.26) is unable to perform an equality search on the **nisNetgroupTriple** attribute of **nisNetgroup** objectclass. As a result, the nisNetgroup membership is not displayed during account aggregation.

Resolution: Open `nis.schema` file of the OpenLDAP server installation and verify if the **nisNetgroupTriple** schema attribute definition is same as the following attribute type:

```
(1.3.6.1.1.1.14 NAME 'nisNetgroupTriple'
DESC 'Netgroup triple'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
```

If the attribute type is not same as the above, then take a back up of the existing `nis.schema` file and replace the existing `nisnetgrouptriple` definition with the above. Save the file and restart the OpenLDAP server. After performing aggregation, nisNetgroup membership must get fetched.

4 - After upgrading to IdentityIQ version 7.3, the nisNetgroup and posixgroup features are not working in the application configuration page

The version 6.0 patch 5 incorporated the support of **nisNetgroup** and **posixgroup** feature. According to which, **nisNetgroup** and **posixgroup** were mapped as native object types in the application schema. After upgrading to version 7.3, the functionality would not work for the existing applications.

Resolution: To ensure that the functionality works in the application configuration page, select the Enable Posix Groups check-box and enter appropriate value for the Map To Member Attribute field.

For instance, if the configuration attribute appeared as

- `<entry key="groupMembershipAttributeType" value="uid"/>`
- then it must be changed to
- ```
<entry key="posixgroup_Member_Attribute" value="uid"/>
```

## Troubleshooting

- <entry key="groupMembershipAttributeType" value="uid"/>  
then it must be changed to  
<entry key="nisNetGroupTriple\_Member\_Attribute" value="uid"/>

# Chapter 18: SailPoint LDIF Connector

---

The following topics are discussed in this chapter:

Overview.....	175
Configuration parameters.....	175
Schema Attributes .....	176

## Overview

---

The SailPoint LDIF Connector is a *read only* connector used to extract data from LDIF files. To help when the membership is not part of the account data there is an option that can be configured named **groupMembershipAttribute**. This configuration setting holds the name of the attribute from the group file which contains the list of its members. Add this attribute to account schema and mark it multi-valued.

The **groupMembershipAttribute** along with a group file must be configured for this feature to work. During account iteration the connector will read in the groups file to get the group => use mapping and adorn each account with their assigned groups as they are aggregated.

## Configuration parameters

---

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The LDIF connector uses the following configuration parameters:

Parameters	Description
filetransport	local, ftp, scp
host	The host of the server to which you are connecting.
transportUser	The user to use with ftp and scp. Not valid with local.
transportUserPassword	The password to use with of ftp and scp. Not valid with local.
file	The fully qualified path to the file.
fileEncoding	Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: <b><a href="http://www.iana.org/assignments/character-sets">http://www.iana.org/assignments/character-sets</a></b>  If this field is empty, the default encoding (the value of <code>file.encoding</code> specified by the jvm) is used.

## Schema Attributes

Parameters	Description
mapToResourceObjectRule	Rule that is called to override the transformation of the data from the Map<String, String> form into a ResourceObject.
filterString	Filter lines that match this string.
filterEmptyRecords	If activated, records that have no data are filtered.
preIterativeRule	The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values.  This rule is called after the file has been transferred, but before iteration over the objects in the file is started.  For validation this rule can use the existing statistics stored by the postIterationRule during the last aggregation. The rule can compare the stored values with the new values to check for problems
postIterativeRule	The post-iterate rule can store away the configuration object and rename/delete the file if desired.  This rule is called after aggregation has completed and ALL objects have been iterated.
groupMembershipAttribute	Holds the name of the attribute from the group file which contains the list of its members.

## Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group.

### Account attributes

Account objects are used when building identities Link objects.

**Table 1—LDIF Connector - Account Attributes**

Name	Description
businessCategory	The types of business performed by an organization. Each type is one value of this multi-valued attribute. Examples: “engineering”, “finance”, and “sales”.
carLicense	License plate or vehicle registration number associated with the user.
cn	Names of object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person’s full name. Examples: “Martin K Smith”, “Marty Smith” and “printer12”.
dn	Distinguished name by which the user is known.

**Table 1—LDIF Connector - Account Attributes (Continued)**

Name	Description
departmentNumber	Numerical designation for a department within your enterprise.
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: “Updates are done every Saturday, at 1am.”, and “distribution list for sales”.
destinationIndicator	Country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute. Examples: “AASD” as a destination indicator for Sydney, Australia. “GBLD” as a destination indicator for London, United Kingdom. <b>Note: The directory will not ensure that values of this attribute conform to the F.1 and F.31 CCITT Recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.</b>
displayName	Preferred name to be used for this person throughout the application.
employeeNumber	Numerical identification key for this person within you enterprise.
employeeType	Descriptive type for this user, for example, contractor, full time, or part time.
facsimileTelephoneNumber	Telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute.
givenName	Name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute. Examples: “John”, “Sue”, and “David”.
groups	List of groups of which this person is a member. Example: “Sales” or “Engineering”
homePhone	Employees home phone number.
homePostalAddress	Employees mailing address.
initials	Strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute. Examples: “J. A.” and “J”.
internationalISDNNumber	Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute. Example: “0198 444 444”.
l	Names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute. Examples: “Austin”, “Chicago”, and “Brisbane”.
mail	The RFC822 mailbox for the user.
manager	Distinguished name of the manager to whom this person reports.
mobile	Mobile telephone number of this person.

## Schema Attributes

**Table 1—LDIF Connector - Account Attributes (Continued)**

Name	Description
o	Names of an organization. Each name is one value of this multi-valued attribute. Examples: "xyz", "xyz Technologies, Inc.", and "xyz, Incorporated.".
ou	Names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: "Sales", "Human Resources", and "Information Technologies".
pager	Telephone number of this persons pager.
physicalDeliveryOfficeName	Names that a Postal Service uses to identify a specific post office. Examples: "Austin, Downtown Austin" and "Chicago, Finance Station E".
postOfficeBox	Postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute. Example: "Box 27".
postalAddress	Addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute. Example: "1111 Elm St.\$Austin\$Texas\$USA".
postalCode	Codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute. Example: "78664", to identify Pflugerville, TX, in the USA.
preferredDeliveryMethod	An indication of the preferred method of getting a message to the object. Example: If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs \$ telephone".
preferredLanguage	Preferred written or spoken language of this person.
registeredAddress	Postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute. Example: "Receptionist\$xyz Technologies\$6034 Courtyard Dr. \$Austin, TX\$USA".
roomNumber	Room, office number or this persons normal work location.
secretary	Distinguished name of this persons secretary.
seeAlso	Distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute. Example: The person object "cn=Elvis Presley,ou=employee,o=xyz\, Inc." is related to the role objects "cn=Bowling Team Captain,ou=sponsored activities,o=xyz\, Inc." and "cn=Dart Team,ou=sponsored activities,o=xyz\, Inc.". Since the role objects are related to the person object, the 'seeAlso' attribute will contain the distinguished name of each role object as separate values.
sn	Name strings for surnames or family names. Each string is one value of this multi-valued attribute. Example: "Smith".

**Table 1—LDIF Connector - Account Attributes (Continued)**

Name	Description
st	Full names of states or provinces. Each name is one value of this multi-valued attribute. Example: “Texas”.
street	Site information from a postal address (that is, the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute. Example: “15 Main St.”.
telephoneNumber	Telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute.
teletexTerminalIdentifier	The withdrawal of recommendation F.200 has resulted in the withdrawal of this attribute.
telexNumber	Sets of strings that are a telex number, country code, and answerback code of a telex terminal. Each set is one value of this multi-valued attribute
title	Persons job title. Each title is one value of this multi-valued attribute. Examples: “Vice President”, “Software Engineer”, and “CEO”.
uid	Computer system login names associated with the object. Each name is one value of this multi-valued attribute. Examples: “s9709015”, “admin”, and “Administrator”.
objectClass	The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either “top” or “alias”.

## Group attributes

---

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

**Table 2—LDIF Connector - Group Attributes**

Name	Description
cn	Names of object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name. Examples: “Martin K Smith”, “Marty Smith” and “printer12”.
uniqueMember	Groups to which this person is a unique member.
dn	Directory path to the object.
o	Names of an organization. Each name is one value of this multi-valued attribute. Examples: “xyz”, “xyz Technologies, Inc.”, and “xyz, Incorporated.”.
ou	Names of an organizational unit. Each name is one value of this multi-valued attribute. Examples: “Sales”, “Human Resources”, and “Information Technologies”.

## Schema Attributes

**Table 2—LDIF Connector - Group Attributes**

Name	Description
owner	Distinguished names of objects that have ownership responsibility for the object that is owned. Each owner's name is one value of this multi-valued attribute. Example: The mailing list object, whose DN is “cn=All Employees, ou=Mailing List,o=xyz, Inc.”, is owned by the Human Resources Director. Therefore, the value of the 'owner' attribute within the mailing list object, would be the DN of the director (role): “cn=Human Resources Director,ou=employee,o=xyz, Inc.”.
description	Human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute. Examples: “Updates are done every Saturday, at 1am.”, and “distribution list for sales”.

# Chapter 19: SailPoint Logical Connector

---

The following topics are discussed in this chapter:

Overview.....	181
Configuration parameters.....	181
Schema attributes .....	181
Additional information .....	182
Logical Connector - Tiers Tab .....	182
Defining Logical Connectors .....	184
Logical Application Filtering.....	184

## Overview

---

The SailPoint Logical Connector is a *read only* connector developed to create objects that function like applications, but that are actually formed based on the detection of accounts from other, or tier, applications in existing identity cubes.

For example, you might have one logical application that represents three other accounts on tier applications, an Oracle database, an LDAP authorization application, and a custom application for internal authentication. The logical application scans identities and creates an account on the logical application each time it detects the three required accounts on a single identity.

You can then use the single, representative account instead of the three separate accounts from which it is comprised for certification, reporting, and monitoring.

## Configuration parameters

---

This section contains the information that this connector uses to connect and interact with the application. Logical applications do not have connection attributes, by default. If you have defined custom logical connectors there might be connection attributes on this tab.

Use this tab to test your logical application connection.

## Schema attributes

---

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Logical applications enable you to pull schema attribute information from the tier applications from which it is compiled. When you use this feature the schema attribute information is automatically added to the attributes table and you can edit it as needed.

## Additional information

Click **New Tier Attribute** to display the Select Source Attribute dialog and select the tier application and attribute to pull into the logical application.

# Additional information

---

This section describes the additional information related to the Logical Connector.

## Logical Connector - Tiers Tab

---

This section contains the information that this connector uses to build the relationships between the tier applications that make up a logical application. For an identity to have an account on a logical application they must have the required, matching accounts on all tier applications. For example an identity, Lori Ferguson, might be represented by the attribute `dbid` on one tier and `username` on another. You must correlate those attributes, either manually or with a correlation rule, to create accounts on the logical application.

### Add Tiers to a Logical Application

You must define the tier applications that are contained within the logical application and identify the application to be used as the primary tier application.

To add a tier application, select the application from the Select an Application drop-down list and click **Add Tier**. Click the arrow to right of the field to display all applications configured to work with IdentityIQ or type the first few letters of an application name to display a list of applications with names containing that letter string. You can add as many applications as required.

Specify the primary tier application by selecting it in the Primary Tier column. The primary tier application is the application containing all of the attributes to which the attributes on the other tiers will correlate. Every account on the logical application must have an account on the primary tier application. In some instances this might be a human resources application containing all of the identities. A logical application can only have one primary tier application.

To remove tier applications, select the application using the selection boxes in the left-most column and click **Remove Selected**.

### Correlate Tier Application Attributes

Use the logical application tier attribute mapping, or correlation, panel to either manually map attributes for correlation or assign an existing correlation rule. For an identity to have an account on a logical application they must have the required, matching, accounts on all tier applications. Map the attributes on each application that should have matching values.

To manually map attributes on the tier applications do the following:

1. Select a non-primary tier application in the application list. The selected application is highlighted and any mapped correlation attributes are displayed in the attribute correlation panel.  
If you select the primary tier application a note is displayed stating that no correlation is required on the primary tier.
2. Click **Add Attribute** to display a row in which to add the new attribute.
3. Click on the row to activate either the **Tier Attribute** or **Primary Tier Attribute** field.
4. Select an attribute from the drop-down list in both columns.

5. Click **Save Changes** or continue mapping attributes for the applications.

To use an existing correlation rule, open the Use Correlation Rule panel and select a rule from the **Correlation Rule** drop-down list. The rule should contain all of the attribute mapping required for this logical application.

The Tiers tab contains the following information:

**Table 1—Logical Connector - Tier Applications**

Attribute	Description
Account Rule	<p>Select an existing account rule from the drop-down list.</p> <p>The logical application rule defines the requirements that must be met before an identity is assigned an account on this logical application.</p> <p><b>Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.</b></p>
Provisioning Rule	<p>Select an existing provisioning rule from the drop-down list.</p> <p>The logical provisioning rule defines how provision requests for the logical application account or any of the accounts with which it is comprised are handled.</p> <p><b>Note: Click the “...” icon to launch the Rule Editor to make changes to your rules if needed.</b></p>
Application	The tier applications that make up the logical application.
Primary Tier	<p>Designate one tier application as the primary tier application. The primary tier application is the application containing all of the attributes to which the attributes on the other tiers will correlate. Every account on the logical application must have an account on the primary tier application. In some instances this might be a human resources application containing all of the identities in IdentityIQ.</p> <p><b>Note: A logical application can only have one primary tier application.</b></p>
Tier Attribute	Attributes from the selected tier application whose values must match the values of the associated attributes from the primary tier application.
Primary Tier Attribute	Attributes on the primary tier application to which the attribute values from the tier applications must match.

**Table 1—Logical Connector - Tier Applications**

Attribute	Description
Account Matching	<p>Use account matching to select attributes and permissions from existing application tiers as the parameters for your logical application. This panel contains the following:</p> <p><b>Application Items</b> — Click <b>Add Attribute</b> to include application attributes in your account matching parameters. Click <b>Add Permission</b> to include application permissions in your account matching parameters.</p> <p><b>Operation</b> — choose the AND / OR operator to include multiple attributes / permissions</p> <p><b>Type</b> — indicates either Attribute or Permission</p> <p><b>Application</b> — indicates the application from which the attribute or permission is being matched</p> <p><b>Name</b> — select an attribute from the drop-down list or input the permission name into the field</p> <p><b>Value</b> — input the value of the attribute or permission</p> <p><b>Group/Ungroup/Delete Selected</b> — use the check box to select line items on which to perform the respective action</p>

## Defining Logical Connectors

---

Use the following procedure to define a logical connector.

1. Define all tier applications.
2. Perform the following tasks on each tiered application:
  - a. Run aggregation task.
  - b. Run entitlement correlation task.
  - c. Scan for missing entitlements or define new managed entitlements.
3. Define the logical application
  - a. Define application tiers
  - b. Discover schema attributes from selected tier applications for editing.
  - c. Scan for missing entitlements using the filters from the selected tiered applications for editing.
4. Run aggregation task on your newly defined logical application.
5. (*Optional*) Run Account-Group Aggregation task on the newly defined logical application.  
This will update the logical application entitlements to have the configured display value for the respective groups. The tier application information used to update the entitlement is based upon the logical applications configured “Group Attribute” from its Account Schema.

## Logical Application Filtering

---

Logical applications use the **Find missing entitlement** scan on the Managed Entitlements tab as filtering action using the Account Matching criteria provided on the Tiers tab. This gives a more focused starting point instead of using all of the entitlement values from the selected application tiers.

For example, a new logical application uses the “memberOf” attribute in Active Directory. There are likely thousands of values that are assigned in an enterprise. With specific criteria defined in Account Matching, only the values you are interested in appear for easier editing.



## **Additional information**

# Chapter 20: SailPoint Lotus Domino Connector

---

The following topics are discussed in this chapter:

Overview.....	187
Supported features .....	187
Supported Managed Systems .....	188
Pre-requisites .....	188
Administrator permissions .....	189
Configuration parameters.....	189
Schema attributes .....	190
Account attributes .....	190
Group attributes.....	193
Provisioning policy attributes.....	193
Create account attributes .....	193
Create group attributes .....	196
Update policies.....	196
Additional information .....	198
ID Vault functionalities.....	198
Password management .....	198
Troubleshooting.....	199

## Overview

---

SailPoint Lotus Domino Connector manages the accounts and groups contained in a Notes database.

### Supported features

---

SailPoint Lotus Domino Connector supports the following features:

- Account Management
  - Manage Lotus Domino Users as Account
  - Aggregate, Delta Aggregation, Refresh Accounts, Pass Through Authentication (uses HTTP password)
  - Create, Update, Delete (Update attributes, Rename, Re-certify, Move user to a different certifier)
  - Enable, Disable, Unlock, Change Password (HTTP (default) and ID file)
 

**Note:** For Self Service HTTP Password Change without specifying the current password, user has to remove the CURRENT\_PASSWORD feature string from the application xml file. For ID File Password Change, without specifying the CURRENT\_PASSWORD feature string add the IDFfileCurrentPassword attribute in change password provisioning.

**Note:** Lotus Domino Connector supports attaching ID file to mail file at the time of change password. For more information, see "Attach ID File to Mail File" on page 198.
  - ID Vault functionalities: Reset Password, Extract ID from vault, Upload ID to vault, Sync ID file

## Overview

- Account - Group Management
  - Manage Lotus Domino Groups as Account-Groups
  - Aggregate, Refresh Groups, Delta Aggregation
  - Create, Update, Delete

## References

- “Password management” on page 198
- “Appendix A: Delta Aggregation”
- “Appendix D: IQService”

## Supported Managed Systems

---

- Domino Server 9.0.x (for “ ID Vault functionalities”)
- Domino Server 8.5.x (for “ ID Vault functionalities”)

## Pre-requisites

---

- The computer must have the `NCSO.jar` file in the classpath.
  - Ensure that Domino server `notes.ini` file contains the following line:  
`ServerTasks=<any other tasks>, DIIOP, HTTP`  
HTTP task is required to be mentioned only if the DIIOP port is not a part of the HostName in the Application attributes.
  - In the Domino server, select **Server => Full Access Administrators** should have the name of the user which is being used to open a session with the server.
  - Domino Server should be reachable from the host computer.
  - The IQService is a native Windows service that enables this connector to participate in a Windows environment and access information only available through Windows APIs.  
IQService must be installed before performing the following operations:
    - Sync ID file
    - Delta Aggregation
    - Upload ID file to vault
    - Get ID file from vault
    - Reset password of an ID file stored in an ID Vault
    - ID File password change through self-service
    - Helpdesk HTTP (Internet) Password change.
- Note:** For more information on IQService, see “Appendix D: IQService”.
- Note:** Lotus Domino Connector requires Microsoft Visual Studio C++ 2015 Redistributable 14.0 (32-bit) installed on the computer where IQService is installed.

## Administrator permissions

---

The Administrator user should have Manager Access to the following databases on Domino Server:

- Public Address Book (PAB) Database (default name is `names.nsf`)
- Administration Requests Database (default name is `admin4.nsf`)
- Certification Log Database (default name is `certlog.nsf`)

## Configuration parameters

---

**Note:** All paths are with respect to the Domino Server computer. For example, ID file path, mail file path and so on. These paths must be accessible from the Domino Server computer.

The following table lists the configuration parameters of Lotus Domino Connector:

Parameters	Description
IQService Host	Host Name of the computer on which IQService is installed.
IQService Port	IQService port number.
Admin ID File Path	Administrator ID file path required by IQService.
Host Name	Fully qualified host name of the Domino Server. The DIIOP port should be a part of HostName if the HTTP task is not mentioned in the Server Tasks (refer Pre-requisites section). In this case, the HostName should be <code>fullyQualifiedHostName:DIIOPPortNumber</code> . For example, <code>sailpoint.server.com:63148</code>
Admin Name	Name of the Database Administrator which must be in the format <code>Administrator/CertifierName</code> .
Admin Password	HTTP password of administrator account.
Database Name	Name of the database to be managed. For example, <code>names.nsf</code>
Server Name	Name of the server to be managed. For example, <code>Lotus/IBM</code>
Search formula - Accounts	<p>Search formula to be used during Account Aggregation.</p> <p>The Search formula follows the following format:</p> <p><code>Domino @Formula language</code></p> <p>For example,</p> <ul style="list-style-type: none"> <li>• For non-indexed database search: <code>SELECT @UpperCase (Type) = "PERSON"</code></li> <li>• For indexed database search: <code>[Type] = "Person"</code></li> </ul>
Search formula - Groups	Search formula to be used during Group Aggregation.

## Schema attributes

Parameters	Description
Indexed database	<p>Specifies if the database is indexed or not.</p> <ul style="list-style-type: none"><li>• Y - The database is indexed</li><li>• N - The database is not indexed</li></ul> <p><b>Note:</b> A maximum of 5,000 documents will be returned by default. The <b>FT_MAX_SEARCH_RESULTS</b> variable in <code>Notes.ini</code> file overrides this limit for indexed databases or databases that are not indexed but that are running an agent on the client. For a database that is not indexed and is running in an agent on the server, set the <b>TEMP_INDEX_MAX_DOC</b> variable in the <code>Notes.ini</code> file. The absolute maximum value is 2147483647.</p>
Indexed search interval in seconds	<p>(Applicable only when <b>Indexed Database</b> attribute is set to <b>Y</b>) In case of an exception for indexed search, the number of seconds to wait until an indexed database search query should be fired again.</p> <p>Add the attribute to application xml with key as <b>idxInterval</b>. For example, <code>&lt;entry key="idxInterval" value="15"/&gt;</code></p>

## Additional configuration parameters

Parameters	Description
groupMembersSize	<p>Specifies maximum size in bytes of the group membership attribute value for any group. The connector returns error if the size of membership attribute value is exceeding the limit at the time of adding user to a group.</p> <p>For example, if groupMembersSize is 23000 and the size of group g1's 'Members' attribute is 23400 bytes, then an exception will be thrown when an add user to group g1 will be performed.</p>

## Schema attributes

This section describes the different schema attributes.

**Note:** For an attribute to be multivalued on managed system side, change the attribute type to multivalued in schema.

## Account attributes

The following table lists the account attributes:

Attributes	Description
NOTEID	NOTEID of the user.
UserName	The user full name.
Type	The type of the document.

Attributes	Description
Owner	The owner of the document.
MailSystem	The type of mail system.
InternetAddress	Mail internet address.
JobTitle	Job title of the user.
CompanyName	Company name of the user.
Department	Department of the user.
EmployeeID	EmployeeID of the user.
Location	Location of the user.
Manager	Manager of the user.
OfficePhoneNumber	Office phone number of the user.
OfficeFAXPhoneNumber	Office fax phone number of the user.
CellPhoneNumber	Cell phone number of the user
PhoneNumber_6	Phone number_6 of the user.
Assistant	Assistant of the user.
OfficeStreetAddress	Office street address of the user.
OfficeCity	Office city of the user.
OfficeState	Office state of the user.
OfficeZIP	Office ZIP/Postal of the user.
OfficeCountry	Office country of the user.
OfficeNumber	Office number of the user.
StreetAddress	Street address of the user.
City	City of the user.
State	State of the user.
Zip	Zip/Postal code of the user.
Country	Country of the user.
PhoneNumber	Phone number of the user.
HomeFAXPhoneNumber	Home fax phone number of the user.
Spouse	Spouse of the user.
Children	Children of the user.
PersonalID	PersonalID of the user.
Comment	Office number of the user.
WebSite	Address of the user Web Page.
PhotoURL	Photo URL of the user.
LocalAdmin	Local Admin of the user.

## Schema attributes

Attributes	Description
CheckPassword	Check password of the user.
PasswordChangeInterval	Password change interval of the user.
PasswordGracePeriod	Password grace period of the user.
PasswordDigest	Password digest of the user.
Policy	Policy of the user.
Profiles	Profiles of the user.
ClientType	Type of the client.
PostalAddress	Postal address of the user.
HomePostalAddress	Home postal address of the user.
Street	Street of the user.
BusinessCategory	Business category of the user.
CarLicense	Car license of the user.
DepartmentNumber	Department number of the user.
EmployeeNumber	Employee number of the user.
EmployeeType	Employee type of the user.
FirstName	First name of the user.
MiddleInitial	Middle initials of the user.
LastName	Last name of the user.
FullName	Full name of the user.  <b>Note:</b> To aggregate the account, account's FullName must not be null or empty and must be unique.
ShortName	Short name of the user.
MailDomain	Mail domain of the user.
MailServer	Mail server of the user.
MailFile	Mail file of the user.
PasswordChangeDate	Password change date of the user.
HTTPPasswordChangeDate	HTTP password change date of the user.
SametimeServer	Home sametime server of the user.
\$UpdatedBy	Name of the user who last updated the user document.
Groups	A list of groups of which the user is a member of.

**Note:** If the FullName/UserName attributes are to be updated, the attribute name to be used in the policy must be 'UserName'.

## Group attributes

---

The following table lists the group attributes:

Attributes	Description
GroupType	Type of the group.
ListDescription	Description of the group.
MailDomain	Mail domain of the group.
InternetAddress	Internet address of the group.
Comments	Comments about the group.
ListOwner	Owner of the group.
LocalAdmin	Local administrator of the group.
ListName	Name of the group.
\$UpdatedBy	Name of the user who last updated the group document.

## Provisioning policy attributes

---

This section lists the different policy attributes of Lotus Domino Connector.

**Note:** For Create and Update Provisioning Policy all the existing schema attributes mentioned in “Account attributes” on page 190 are supported.

**Note:** All paths are with respect to the Domino Server computer. For example, ID file path, mail file path and so on. These paths must be accessible from the Domino Server computer.

## Create account attributes

---

The following table lists the provisioning policy attributes for Create Accounts:

Attributes	Description
ServerName*	The name of the server on which the account should be created.
CertifierIDfile	The ID file path of the certifier. For example, c:\id\cert.id
CertifierPassword	The password of the certifier ID file.
FirstName	First name of the user.
MiddleInitial	Middle name initial of the user.
LastName*	Last name of the user.
FullName*	Full name of the user. The format is as follows: FirstName LastName/CertifierName.
IDFilePath*	ID file path of the user. For example, c:\id\user.id
UserIDFilePassword*	ID file password of the user.

## Provisioning policy attributes

Attributes	Description
IDType*	Type of the ID file. Following are the permissible values for the keyword: <ul style="list-style-type: none"><li>• FLAT</li><li>• HIERARCHICAL</li><li>• CERTIFIER</li></ul>
MinimumPasswordLength	Minimum length of the ID file password.
IDFileIsNorthAmerican	Indicates whether the id file is North American or not. Following are the permissible values for the keyword: <ul style="list-style-type: none"><li>• Y: indicates that the ID file is North American</li><li>• N: indicates that the ID file is not North American</li></ul>
StoreIDInAddressbook	Indicates whether the ID file should be stored in the address book or not. Following are the permissible values for the keyword: <ul style="list-style-type: none"><li>• Y: indicates that the ID file should be stored in the address book</li><li>• N: indicates that the ID file should not be stored in the address book.</li></ul>
MailServer	Server on which the mail file should be created.
MailSystem	Specifies the type of the mail system. Following are the permissible values for the keyword: <ul style="list-style-type: none"><li>• NOTES</li><li>• POP</li><li>• IMAP</li><li>• INOTES</li><li>• INTERNET</li><li>• OTHER</li><li>• NONE</li></ul>
MailInternetAddress	Internet address for the mail.
MailTemplateName	Name of the mail template.
MailForwardingAddress	Forwarding address for the mail.
MailFileName	Name of the mail file. For example, mail/mailfilename.nsf
MailReplicaServer	The names of the servers on which the mail file replicas should be created. Applies only to clustered servers. Should be multi-valued.
MailACLManger	A name that is assigned to the Manager for accessing the mail database ACL. The format should be as, FirstName LastName/CertifierName.
MailOwnerAccess	The mail database ACL setting for the owner. Allowed values are MANAGER, EDITOR and DESIGNER
MailQuotaSizeLimit	The maximum size of the user's mail database, in megabytes.
MailQuotaWarningThreshold	The size in megabytes, at which the user's mail database issues a warning that it is getting too large.
ShortName	The short name of the user.

Attributes	Description
CreateMailDatabase	<p>Indicates whether the mail database should be created or not. Following are the permissible values for the keyword:</p> <ul style="list-style-type: none"> <li>• <b>Y:</b> indicates that a mail database should be created for the user</li> <li>• <b>N:</b> indicates that a mail database should not be created for the user; it will be created during setup.</li> </ul>
StoreIDInMailFile	<p>Indicates whether the ID should be stored in mail file or not. Following are the permissible values for the keyword:</p> <ul style="list-style-type: none"> <li>• <b>Y:</b> indicates that the ID file should be stored in mail file.</li> <li>• <b>N:</b> indicates that the ID file should not be stored in mail file.</li> </ul>
SynchInternetPassword	<p>Indicates whether the ID password and internet password should be in synchronization. Following are the permissible values for the keyword:</p> <ul style="list-style-type: none"> <li>• <b>Y:</b> indicates that the ID file password and internet password should be in synchronization</li> <li>• <b>N:</b> indicates that the ID file password and internet password should not be in synchronization</li> </ul>
ExpirationPeriod	<p>The expiration period in years. For example, if 20 is specified and the current year is 2013, the expiration period will be 2033.</p>
RegistrationLog	<p>No logging occurs if this parameter is null. If this parameter has a value other than null, logging goes to the <b>certlog.nsf</b> file in the Domino data directory on the registration server.</p>
EnforceUniqueShortName	<p>Indicates whether a unique short name should be used. Following are the permissible values for the keyword:</p> <ul style="list-style-type: none"> <li>• <b>Y:</b> indicates that the short name should be unique</li> <li>• <b>N:</b> indicates that the short name may or may not be unique</li> </ul>
Policy	<p>Name of the explicit policy.</p>
RoamingUser	<p>Indicates whether a user is roaming or not. Following are the permissible values for the keyword:</p> <ul style="list-style-type: none"> <li>• <b>Y:</b> indicates that the user is roaming</li> <li>• <b>N:</b> indicates that the user is not roaming</li> </ul>
UseCAProcess	<p>Specifies if CA process should be used at the time of user creation. Values: Y, N</p>
CertifierName	<p>(Required if UseCAProcess attribute is enabled) Name of the certifier of format /ABC/rootCert.</p>
RoamingCleanupSetting	<p>Indicates the clean-up process for data on Notes clients set up for roaming users. The values are as follows:</p> <ul style="list-style-type: none"> <li>• NEVER CLEANUP</li> <li>• CLEANUP EVERY N DAYS</li> <li>• CLEANUP AT SHUTDOWN</li> <li>• CLEANUP PROMPT</li> </ul>

## Provisioning policy attributes

Attributes	Description
RoamingCleanupPeriod	(The RoamingCleanupSetting attribute must be CLEANUP EVERY N DAYS). The interval in days for cleaning up data on Notes clients set up for roaming users.
RoamingServer	The server on which the user's roaming data is stored.
RoamingSubdir	The subdirectory that contains the user's roaming data. For example, roaming\TestUser

**Note:** All attributes should be of type 'String'.

In the above table the attributes marked with the \* sign indicate that the attributes are mandatory.

## Create group attributes

The following table lists the provisioning policy attributes for Create Group:

Attributes	Description
ListName	Name of the group to be created.

**Note:** Only the name of the group is required at the time of group creation. Even if the other attributes are specified they will not be set. After creation, the group will be of type 'Multi-Purpose'.

## Update policies

**Note:** In update policies for account/group, the attribute names must be the same as their corresponding names in the document properties of account/group on Lotus Notes. For updating the FullName of an account, the name of the attribute should be UserName.

The following table lists the attributes for different update policies:

Attributes	Description
<b>Rename a user</b>	
AC_Operation*	The value of this attribute should be <b>Rename User</b> .
AC_certifierFilePath*	The location of the Certifier ID file of the user. For example, c:\id\cert.id
AC_certifierPassword*	The Certifier ID file password.
AC_lastName	New last name of the user.
AC_middleInitial	New middle name initial of the user
AC(firstName	New first name of the user.
AC_orgUnit	New organizational unit of the user.
AC_altOrgUnit	New alternate organizational unit of the user.
AC_altLanguage	New alternate language of the user.
AC_renameNotesUser*	If you want to rename Notes User or not. Values: True or False.

Attributes	Description
<b>Recertify a user</b>	
AC_Operation*	The value of this attribute should be <b>Recertify User</b> .
AC_certifierFilePath*	The location of the Certifier ID file of the user. For example, c:\id\cert.id
AC_certifierPassword*	The Certifier ID file password.
<b>Move a user</b>	
AC_Operation*	The value of this attribute should be <b>Move User</b> .
AC_currentCertifierFilePath*	The location of the Certifier ID file of the user. For example, c:\id\cert.id
AC_currentCertifierPassword*	The Certifier ID file password.
AC_targetCertifierFilePath*	The location of the Certifier ID file of the user. For example, c:\id\target.id
AC_targetCertifierPassword*	The Certifier ID file password.
AC_targetCertifierName*	The name of the target certifier.
<b>Change/Reset password of a user</b>	
HTTP_PASSWORD_CHANGE	Should be set to <b>Yes</b> to change the HTTP (Internet) Password of a user. Values: <b>Yes</b> (Default) and <b>No</b> .
IDFileCurrentPassword	(Used only when CURRENT_PASSWORD feature string is not present) The current password of the ID file.
IDFilePath	The location where the user ID file is stored. For example, c:\id\user.id should be provided to change the ID file password of a user.
RESET_PASSWORD	Should be set to <b>Yes</b> to reset the password of an ID file stored in the vault. Values: <b>Yes</b> and <b>No</b> (Default).
<b>Sync ID File</b>	
Operation*	The value of this attribute should be <b>Sync ID File</b> .
IDFilePath*	The location of the User ID file of the user. For example, c:\id\user.id
IDFilePassword*	The User ID file password.
<b>Get ID File</b>	
Operation*	The value of this attribute should be <b>Get ID File</b> .
IDFilePath*	The location where the User ID file should be stored. For example, c:\id\user.id
IDFilePassword*	The User ID file password.
<b>Upload ID File</b>	
Operation*	The value of this attribute should be <b>Upload ID File</b> .
IDFilePath*	The location where the User ID file is stored. For example, c:\id\user.id

## Additional information

Attributes	Description
IDFilePath*	The User ID file password.
<b>Attach ID File to Mail File</b>	
ATTACH_ID_TO_MAIL	Specifies if ID File should be attached to Mail File or not. Values: Yes and No (Default).
MailServer	Mail server of the users mail file.
MailFile	Mail File path. For example, mail\userMail.nsf

**Note:** No default value needs to be assigned for optional attributes if those need not to be set. In the above table the attributes marked with the \* sign indicate that the attributes are mandatory.

## Additional information

This section describes the additional information related to the Lotus Domino Connector.

### ID Vault functionalities

ID Vault is a feature introduced by IBM in Domino version 8.5. The following functionalities are a part of ID Vault which are supported through IQService:

- **Reset Password:** Allows the Help Desk Personnel to reset the password of the ID file for a vaulted user. This requires application Administrator to have password reset authority.
- **Extract ID file from vault:** A vault administrator assigned to the Auditor role in the vault database ACL can extract an ID from a vault to gain access to a user's encrypted data. A copy of the ID remains in the vault after extraction.
- **Upload ID file to vault:** Upload an ID file that has not yet been uploaded to the vault.
- **Sync ID file:** Synchronizes the local ID file with the copy in ID vault.

Check the provisioning policy for each of the above transactions.

### Password management

Administrator password reset (Change password for others):

- HTTP Password
- Password of the vaulted ID file (Reset password)

Self-service password change:

- HTTP Password
- ID File Password
- Password of the ID file which is vaulted (Reset password)

**Note:** Ensure that self-service change password with the IBM Lotus Domino connector is successful after adding the 'ValidateHttpPassword' attribute, even if the current passwords for HTTP Password and ID File are different.

# Troubleshooting

---

## 1 - Could not get IOR from Domino Server

**Resolution:** Perform the following:

1. Check if the Domino Server is accessible from a computer which is using the Fully Qualified Internet Host Name. The ping should be successful using the Fully Qualified Internet Host Name of the Domino Server.
2. Check if DIIOP is present in the ServerTasks of `notes.ini` file.
3. If HTTP is not added to `notes.ini` file ServerTasks, the HostName in the Application Parameter should include the port number of the DIIOP Server in the following format:  
`fullyQualifiedInternetHostName:DIIOPPortNumber`  
For example, `LOTUS-AME.SAILPOINT.COM:63148`
4. Check if `NCSO.jar` file is present in the CLASSPATH environment variable.

## 2 - Could not open the ID file

**Resolution:** Perform the following:

1. All paths in the connector are with respect to the Domino Server. Verify if the ID file path you have provided is accessible from the Domino Server computer.
2. The ID files will be read from and created on a path with respect to the Domino Server.

## 3 - Add Account gives Object does not exist exception

**Resolution:** Perform the following:

1. If the name of the user you created is Derek Stevens and the name of the certifier under which the user was created is /USA then the following attributes should be populated:
  - FirstName: Derek
  - LastName: Stevens
  - FullName: Derek Stevens/USA
Add account searches a user based on the FullName of the user, hence it is important that it is provided correctly.

## 4 - IQService - Unable to load DLL 'SPLotusNotesWrapper.dll': The specified module could not be found or the IQService stops responding

**Resolution:** Perform the following:

1. Verify if the PATH system variable contains the Notes data folder. For example, `C:\Program Files\IBM\Notes` must be present in the PATH system variable.
2. Verify if you have restarted the computer after modifying the PATH system variable.
3. Close the Notes Administrator/Client and restart the IQService.
4. Copy the IQService installation files in the Notes folder of IBM Lotus Notes Client. For example:  
`C:\Program Files\IBM\Notes`

## 5 - Attempt to load a program with an incorrect format

An attempt to load a program with an incorrect format, displays the following error message:

Unable to create iterator: sailpoint.connector.ConnectorException: Errors returned from IQService.

**Resolution:** Ensure that all the pre-requisites mentioned in the “Pre-requisite for Lotus Domino Connector” on page 554 are performed.

## 6 - ID file could not be opened

When creating an account from IdentityIQ, the following error message appears:

sailpoint.connector.ConnectorException: Notes error: Could not open the ID file

**Resolution:** The cert.id file must be present on the Server computer.

## 7 - During Delta Aggregation an exception error is displayed

While performing Delta Aggregation, the following error message is displayed:

Exception during aggregation. Reason: sailpoint.connector.ConnectorException: Errors returned from IQService. Object reference not set to an instance of an object.

**Resolution:**

1. While performing Delta aggregation or resetting the password, ensure that the Domino Client is closed.
2. If Domino Client is open and you perform Delta Aggregation, an exception error message is displayed.  
In such scenario, if you close the Domino Client and perform the same operation, the transaction appears to be in pending or idle state.
3. Restart the Domino Server and terminate / cancel such request in IdentityIQ.

## 8 - Test connection fails with Invalid user name/password

Lotus Notes test connection fails with invalid user name/password.

**Resolution:** Ensure that the administrator password is a HTTP password and not ID file password.

# Chapter 21: SailPoint Linux Connector

---

The following topics are discussed in this chapter:

Overview.....	201
Supported features .....	201
Supported Managed Systems .....	202
Pre-requisites .....	202
Administrator permissions .....	203
Configuration parameters.....	204
Additional configuration parameters for SSH configuration .....	204
Public key authentication configuration.....	205
Schema attributes .....	205
Account attributes .....	206
Group attributes.....	206
Provisioning policy attributes.....	206
Account attributes .....	207
Group attributes.....	207
Additional information .....	208
Unstructured Target Collector .....	208
Troubleshooting .....	209

## Overview

---

The SailPoint Linux Connector was developed to enable user managing their Linux Account, Groups and resources. The Linux system data will be aggregated and user would be able to edit entities and their attributes.

### Supported features

---

The SailPoint Linux Connector supports the following features:

- Account Management
  - Manage Linux Users as Account
  - Aggregate, Refresh Accounts
  - Create, Update, Delete
  - Enable, Disable, Unlock, Change Password
  - Add/Remove Entitlements
- Account Group Management
  - Manage Linux Groups as Account-Groups
  - Aggregate, Refresh Groups
  - Create, Update, Delete

## Overview

- Permissions Management
  - Application can be configured to read permissions directly assigned to accounts and groups using Unstructured Target Collector.
  - The connector supports automated revocation of the aggregated permissions for accounts and groups.

**Note:** Linux connector supports MD5, SHA-1, and SHA-2 cryptographic hash functions.

## References

- “Appendix C: Before and After Provisioning Action”
- “Additional information” on page 208

## Supported Managed Systems

---

The Linux connector supports the following versions of the operating system:

- Ubuntu 16.04 LTS
- Red Hat Enterprise Linux versions 7.5, 7.4, 7.3, 7.2, 7.1, 7.0, 6.9, 6.8 and 6.7
- SUSE Linux Enterprise Server 12 and 11

**Note:** For any issues related to SUSE Linux, see “Troubleshooting” on page 209 section.

## Pre-requisites

---

- SSH should be installed on Linux computer
- The **sshj-0.23.0.jar** and **ganymed-ssh2-build209-1.0.jar** files must be present in \WEB-INF\lib folder
- For Sudo users and permissions
  - The administrator user must have rights to execute /usr/bin/awk command.  
Update /etc/sudoers file entry for the administrator user with /usr/bin/awk command.
  - User and group schema must add new multi valued schema attribute as **sudoCommands** which would collect all the necessary user commands and store it as a part of this attribute.
  - If end user wants to aggregate the sudo commands from multiple sudo files then user must provide list of files as a separate configuration attribute.

For example, `<entry key="sudoCmdFiles" value="/etc/sudoers.d/special_user.conf,/etc/sudoers.d/special_group.conf"/>`

**Note:** The default command which would collect the sudo commands is as follows:

```
awk '/#[^#]/' /etc/sudoers.
```

In the above command, the commented lines are skipped and the remaining content of /etc/sudoers file are aggregated in to a temporary file on Linux computer.

The temporary file of Linux computer would get copied to local IdentityIQ computer and process all the sudo user and group commands.

If the end user wants to provide new command for aggregating file data, then it can be configured as a part of application xml file.

For example: key: sudoUserCommand and value : awk '/#[^#]/' /etc/sudoers

## Administrator permissions

---

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

**Rights to execute the following commands with root privilege:**

```
/bin/chmod, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel,
/usr/sbin/groupadd, /usr/sbin/groupmod, /usr/sbin/groupdel, /usr/bin/passwd,
/usr/bin/faillog, /usr/bin/groups, /bin/rm -f spt_tmp_*, /bin/echo,
/usr/bin/chage, /usr/bin/find, /bin/cat /etc/shadow, /bin/cat /etc/passwd,
/bin/cat /etc/group, /bin/cat /etc/pam.d/system-auth, /usr/bin/getent, /bin/grep,
/usr/bin/awk, /usr/bin/id, /usr/bin/lastlog, /usr/sbin/pam_tally2,
/sbin/pam_tally2, /bin/cat /etc/pam.d/password-auth, /bin/cat
/etc/pam.d/common-account, /bin/cat /etc/pam.d/common-auth, /usr/bin/printf
```

**An entry in /etc/sudoers file must look similar to the following:**

```
username ALL = (root) PASSWD: /bin/chmod, /usr/sbin/useradd, /usr/sbin/usermod,
/usr/sbin/userdel, /usr/sbin/groupadd, /usr/sbin/groupmod,
/usr/sbin/groupdel,/usr/bin/passwd, /usr/bin/faillog, /usr/bin/groups, /bin/rm -f
spt_tmp_*, /bin/echo, /usr/bin/chage, /usr/bin/find, /bin/cat /etc/shadow,
/bin/cat /etc/passwd, /bin/cat /etc/group, /bin/cat /etc/pam.d/system-auth,
/usr/bin/getent, /bin/grep, /usr/bin/awk, /usr/bin/id, /usr/bin/lastlog,
/usr/sbin/pam_tally2, /sbin/pam_tally2, /bin/cat /etc/pam.d/password-auth,
/bin/cat /etc/pam.d/common-account, /bin/cat /etc/pam.d/common-auth,
/usr/bin/printf
```

**Note:** All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on Linux computers as they might differ from the values mentioned here.

**Note:** If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

**Note:** With this release of IdentityIQ, Linux Connector does not cause 'NumberFormatException' with Multiple "deny=" Lines in /etc/pam.d/system-auth file while performing Preview accounts and account aggregation task.

## Read Only permissions

If you want to use sudo user to perform read only operations, the sudo user must be configured with the following rights and permissions:

- **For Account Aggregation only**

**Rights to execute the following commands with root privilege:**

```
/bin/echo, /bin/cat /etc/group, /bin/rm -f spt_tmp_*, /bin/cat /etc/passwd,
/bin/grep, /bin/cat /etc/shadow, /bin/cat /etc/pam.d/system-auth, /bin/cat
/etc/pam.d/password-auth, /usr/bin/faillog,
/usr/sbin/pam_tally2, /sbin/pam_tally2, /usr/bin/lastlog, /usr/bin/awk
```

**An entry in /etc/sudoers file must look similar to the following:**

```
username ALL = (root) PASSWD: /bin/echo, /bin/cat /etc/group, /bin/rm -f spt_tmp_*, /bin/cat
/etc/passwd, /bin/grep, /bin/cat /etc/shadow, /bin/cat
```

## Configuration parameters

```
/etc/pam.d/system-auth, /bin/cat /etc/pam.d/password-auth, /usr/bin/faillog,
/usr/sbin/pam_tally2, /sbin/pam_tally2, /usr/bin/lastlog, /usr/bin/awk
```

- **For Group Aggregation only**

Rights to execute the following commands with root privilege:

```
/bin/echo, /bin/ cat /etc/group, /bin/rm -f spt_tmp_*, /bin/grep
```

An entry in /etc/sudoers file must look similar to the following:

```
username ALL = (root) PASSWD: /bin/echo, /bin/ cat /etc/group, /bin/rm -f
spt_tmp_*, /bin/grep
```

**Note:** If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry must be performed. Verify the command paths on Linux computers as they might differ from the values mentioned here.

## Supported Authentication methods

The Linux Connector supports the following authentication methods for root and sudo user:

- publickey
- username and password

# Configuration parameters

---

The following table lists the configuration parameters of Linux Connector:

Parameters	Description
Unix Server Host	Host Name/IP address of the computer.  <b>Note: For IdentityIQ version 6.4 Patch 4 and above, the format of the application XML has been changed from</b> <code>&lt;entry key="UnixServerHost" value="<hostname>" /&gt;</hostname></code> <b>to</b> <code>&lt;entry key="host" value="<hostname>" /&gt;</hostname></code>
SSH Port	SSH port configured. Default value: 22
User Name	User ID on the computer that you want to use for connector operations.
User Password	Password of the target system user account that you want to use for connector operations.
Not a 'root' user	If User ID specified is not root, check this parameter.
Private Key File Path	Path to Private Key File. Private/Public key authentication will have precedence over password authentication.
Passphrase For Private Key	Passphrase provided for creating Private Key.

## Additional configuration parameters for SSH configuration

---

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

**Note:** These additional configuration parameters must be added in the Application/Target Source debug page.

1. Following is the default command for setting shell prompt on UNIX computer:

```
<entry key="SetPrompt" value="PS1='SAILPOINT'"/>
```

In the above command, "SetPrompt" is the application/target source attribute and PS1='SAILPOINT' is the value of the application/target source attribute.

If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

For example: For tcsh shell, the entry value would be:

```
<entry key="SetPrompt" value="set prompt='SAILPOINT'"/>
```

2. For executing the commands, verify that the default shell is present on your system.

If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

```
<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>
```

## Public key authentication configuration

---

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

1. Generate Private and Public key's. For more information of the standard steps, see "8 - Test connection fails for SUSE Linux" on page 212.
2. Append contents of public key file to `~/.ssh/authorized_keys` as shown below.  
`cat <public key file> >> ~/.ssh/authorized_keys`
3. The `~/.ssh/authorized_keys` file must have the read, write, and execute permissions in the `-rw-r--r--` format. Enter the following command to achieve the `-rw-r--r--` format permissions:  
`chmod 0644 ~/.ssh/authorized_keys`
4. Copy private key file to a location which is accessible by the server.
5. Provide path of private key file in application configuration.

**Note:** When generating public keys, if permission related issue occurs use the following command from user home directory (this overrides selinux policies):

```
chcon -t ssh_home_t .ssh
```

## Schema attributes

---

This section describes the different schema attributes.

## Provisioning policy attributes

### Account attributes

---

The following table lists the account attributes:

Attributes	Description
username	It is used when user logs in.
uid	Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
home	The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
pwdlastchg	Days since Jan 1, 1970 that password was last changed.
pwdmin	The minimum number of days required between password changes that is, the number of days left before the user is allowed to change his/her password.
pwdmax	The maximum number of days the password is valid (after that user is forced to change his/her password).
pwdwarn	The number of days before password is to expire that user is warned that his/her password must be changed.
comment	Description
expiration	Days since Jan 1, 1970 that account is disabled that is, an absolute date specifying when the login may no longer be used.
inactive	The number of days after password expires that account is disabled.
lastlogin	Last login date and time of the Account.
primgrp	Name of primary group of the user.
groups	Secondary groups of user.
shell	User's shell.

### Group attributes

---

The following table lists the group attributes:

Attributes	Description
groupid	GID. Each user must be assigned a group ID. You can see this number in your /etc/group file.
name	It is the name of group. If you run <b>ls -l</b> command, you will see this name printed in the group field.

## Provisioning policy attributes

---

This section lists the different policy attributes of Linux Connector.

## Account attributes

---

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
User Name	It is used when user logs in. It should be between 1 and 32 characters in length.
User ID	Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
Home Directory	The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
Min password change days	The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password.
Max password validity	The maximum number of days the password is valid (after that user is forced to change his/her password).
Password change warning time	The number of days before password is to expire that user is warned that his/her password must be changed.
Comment	Description
Account expire duration	Days since Jan 1, 1970 that account is disabled that is, an absolute date specifying when the login may no longer be used.
Account inactivity time	The number of days after password expires that account is disabled.
Do not add to last login	Whether to add to last login log file.
Shell	User's shell.
Allow duplicate UID	Allow creation of account with a duplicate (non-unique) UID.
Create Home Directory	Whether to create home directory for new user.
Primary Group name	Specify primary group name.
Password	Initial password for newly created user account.
Force password change on next login	Specify if user has to be forced to change password on next logon.

## Group attributes

---

The following table lists the provisioning policy attributes for Create Group:

## Additional information

Attributes	Description
Group ID	GID. Each user must be assigned a group ID. You can see this number in your /etc/passwd file.
Group Name	It is the name of group. If you run ls -l command, you will see this name printed in the group field.
Allow duplicate GID	Duplicate GID of Group.

## Additional information

---

This section describes the additional information related to the Linux Connector.

**Note:** To enable logging, specify the logging

log4j.logger.openconnector.connector.unix.UnixConnector **and**  
log4j.logger.openconnector.connector.unix.LinuxConnector **in the**  
log4j.properties **file.**  
**For example,** log4j.logger.openconnector.connector.unix.UnixConnector=debug  
log4j.logger.openconnector.connector.unix.LinuxConnector=debug.

## Unstructured Target Collector

---

Linux uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group identityAttribute for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint User’s Guide*.

For Linux target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED\_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

Linux Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated to the Users and Groups. For UNIX platforms direct access means ownership of file or directory.

Attributes	Description	Possible values
Unix File System Path(s)*	Absolute path(s) which are to be scanned for resources.	Multiple paths can be mentioned with comma separated values. For example, /etc, /tmp
Application Name*	Name of the application with which Unstructured Target will be correlated.	

**Note:** Attributes marked with \* sign are the mandatory attributes.

**Note:** If Unstructured Configuration is configured before upgrading to version 7.3 from version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

## Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

**Correlation Rule:** The rule used to determine how to correlate account and group information from the application with identity cubes in IdentityIQ.

**Note:** For version 6.2 onwards, the default schema does not have correlation keys defined. Update correlation rule in Unstructured Target Configuration accordingly.

## Provisioning related parameters

Select the settings for provisioning to the box.

- **Override Default Provisioning:** Overrides the default provisioning action for the collector.
- **Provisioning Action:** The overriding provisioning action for the collector.

# Troubleshooting

---

## 1 - Test connection failed on SUSE computer with an error message

Test connection failed on SUSE computer with the following error message:

```
[InvalidResponseException] [Possible Suggestion]Make sure standalone command works
with the UNIX terminal. The standalone command is - echo 'TestConnection'[Error
details] Unexpected output captured. Host:xxx.xx.xx.xxx. Output: sword sudo:
pam_authenticate: Module is unknown SAILPOINT> Password Sh: Password: command not
found.
```

**Resolution:** When the test connection fails on SUSE computer, the following setting must be changed in /etc/ssh/sshd\_config file:

```
PasswordAuthentication yes
```

Enter the following command to restart the sshd after updating the sshd\_config file:

```
/etc/init.d/sshd restart
```

## 2 - Password command failed with an error message

Password command failed with the following error message:

```
sailpoint.connector.InvalidConfigurationException: [InvalidConfigurationException]
[Possible suggestions] a) Make sure the provided password is correct as per the
password policy defined on the UNIX machine. b) Make sure application configuration
attribute 'PasswdPrompts' is set correctly. c) Tune the parameter 'sshWaitTime'.
[Error details] Password prompt mismatch. Check the shell output for more details.
```

Password command fails if password prompts are not matching.

**Resolution:** Verify the password command on Linux computer for password prompts and if the required prompts are present in your application.

## Troubleshooting

For example, `passwd Person2`

## Changing password for Person2.

**New Password:** New Password is the prompt, so if this prompt is not present in your application, add/update it as follows:

For example,

## 3 - Aggregation/test connection fails with timeout error

Aggregation/test connection fails with the following timeout error:

Exception during aggregation of Object Type account on Application <application name>  
Reason: Unable to create iterator sailpoint.connector.TimeoutException:  
[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout occurred while reading output stream for the executed command.

Test Connection fails with following timeout error:

[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout occurred while reading output stream for the executed command.

**Resolution:** Change the value of the **sshWaitTime (in millisecond)** application attribute as per your requirement in the debug page of the application:

```
<entry key="sshWaitTime" value="500"/>
```

#### 4 - After target aggregation resources are not getting correlated with Account Groups

After target aggregation the resources are not getting correlated with Account Groups.

**Resolution:** Ensure that your correlation rule populates "Correlator.RULE\_RETURN\_GROUP\_ATTRIBUTE" as follows:

```
....
 if (isGroup) {
 returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE, "nativeIdentity");
 returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);
 }
....
```

#### 5 - Test connection fails for key based authentication with an error

Test connection fails for key based authentication with the following error.

```
Login failed. Error while connecting to host:<hostname>. Cannot read key file.
```

**Resolution:** Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using openssl. This method can be used for any version of SSH.
  - a. Create private key using the following command:
 

```
openssl <gendsa/genrsa> -des3 -out <private_key> 1024
```
  - b. Change the permission on the <private\_key> file as follows:
 

```
chmod 0600 <private_key>
```
  - c. Create public key from private\_key
 

```
ssh-keygen -y -f <private_key> > <public_key>
```
  - d. Use the <private\_key> and <public\_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
  - a. Create private and public key using the following command
 

```
ssh-keygen -t <dsa/rsa> -b 1024
```

 By default files with name `id_dsa/id_rsa` and `id_dsa.pub/id_rsa.pub` will be created.
  - b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:
 

```
openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3
```
  - c. Change the permission on the <new\_private\_key> file as follows:
 

```
chmod 0600 <new_private_key>
```
  - d. Create public key file using the new private key as follows:
 

```
ssh-keygen -y -f <new_private_key> > <new_public_key>
```
  - e. Use the <new\_private\_key> and <new\_public\_key> files for authentication.

### 6 - Test connection fails with one of the following error when sudo user is configured for public key authentication

- Test connection fails with the following error when sudo user is configured for public key authentication:

Test SSH communication failed over host: xxxxxxxx. Error while executing command: sudo -p %SAILPOINTSUDO echo TestConnection over host: xxxxxxxx. Invalid sudo user password.

**Resolution:** On managed system,

- if Sudoers file is having Sudo user with **PASSWD** attribute assigned, then the sudo user's password specified in application configuration, password must be correct for certificate based authentication.
- if Sudoers file is having Sudo user with **NOPASSWD** attribute assigned, then the sudo user's password specified in application configuration, password can be incorrect/or any value. Certificate based authentication must still work.

**Note: Password is mandatory field on application UI.**

- [InvalidConfigurationException] [Possible suggestions] a) Verify the private key file is correct for specified user. b) Verify the private key Passphrase is correct for specified user. c) Verify the private/ public key file permissions are correct on the given unix host. [Error details] Failed to authenticate the ssh credentials for the user: <user> to the host: XXX.XX.XX.XXX.

**Resolution:** Verify pam\_tally2 counter and reset it to 0 (zero) and perform the operations again.

### 7 - Enable user failed with an error

Enable user failed with the following error:

sailpoint.connector.InvalidResponseException: [InvalidResponseException] [Possible suggestions] Make sure standalone command works with the UNIX terminal. The standalone command is - passwd -u "<user>" [Error details] Command failed. Status: 254 , Output: Unlocking password for user <user>. passwd: Warning: unlocked password would be empty. passwd: Unsafe operation (use -f to force). passwd: Unsafe operation (use -f to force)

**Resolution:** Update the following entry in connector registry/debug application configuration as:

```
<entry key="enable.account" value="passwd -u -f"/>
```

### 8 - Test connection fails for SUSE Linux

Test connection fails on SUSE Linux as Password Authentication was not enabled.

[InvalidConfigurationException] [Possible suggestions] Provide either publickey or password as an authentication method for the user. [Error details] No supported authentication methods found on the host XXX.XX.XX.XXX for user <user>

**Resolution:** Perform the following steps to enable Password Authentication:

1. Change the value of Password Authentication from No to Yes in /etc/ssh/sshd\_config file as follows:  
PasswordAuthentication no  
To  
PasswordAuthentication yes
2. Restart the server using the following command:  
/etc/init.d/sshd restart

## 9 - Account Aggregation and Account provisioning displays an error for Lock/Unlock status

**Resolution:** Perform the following:

### For RHEL 6.x and above

1. Specify the maximum allowed failed login attempts before the account is locked by the system. Edit the configuration file pointed by registry key:

```
<entry key="get.loginsyslimit" value="cat /etc/pam.d/system-auth"/>
```

**Default value:** /etc/pam.d/system-auth or /etc/pam.d/password-auth

Specify maximum allowed failed login using "deny=".

For example, add the following lines in /etc/pam.d/system-auth or /etc/pam.d/password-auth:

```
auth required pam_tally2.so onerr=fail deny=5
account required pam_tally2.so
```

2. Ensure that pam\_tally2 command, as required in the following registry key works correctly:

```
<entry key="aggregation.lockstatus" value="pam_tally2 | awk '{print $1} {print $2}'>/>
```

3. Ensure that the following command to get failed login works on the system:

```
<entry key="get.userfailedlogin" value="pam_tally2"/>
```

4. Verify if unlock command specified in the registry correctly resets the failed login counter:

**Default settings:** <entry key="unlock.account" value="pam\_tally2 -u"/>

**Note:** For RHEL version's below 6.0 where pam\_tally2 module not installed, replace pam\_tally2 with faillog in above commands.

## **Troubleshooting**

# Chapter 22: SailPoint Mainframe Connector

---

The following topics are discussed in this chapter:

Overview.....	215
Configuration parameters.....	215
Schema attributes .....	216

## Overview

---

The SailPoint Mainframe Connector is a *read only* connector which uses a technique called screen scraping and each deployment must write Rules to drive the login/logout/fetch accounts. The connector parses the screens and emulates the user during the interaction. On some legacy systems screen scraping is the only way to get to the data needed by IdentityIQ. Each Mainframe connector requires a lot of hands on configuration, because the Rules that drive this connector are very specific to the application on which the connector is running.

The Mainframe connector is designed for TN3270 applications and built on the IBM Host Access API libraries. You must have the IBM Host Access API libraries before working with this connector. You can purchase these libraries from IBM.

## Configuration parameters

---

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Mainframe connector uses the following connection attributes:

**Table 1—Mainframe Connector - Configuration parameters**

Parameters	Descriptions
host	The host of the server to which you are connecting.
port	The port the server is listening through.
user	The valid user name with which to connect to the server.
password	The password associated with the connection user.
logonRule	The rule used to log on to the application
logoffRule	The rule called to log off of the application
userIterateRegularExpression	The regular expression that should be used when fetching/iterating accounts. This expression breaks the screens into records that can be manipulated by the script.
userTransformRule	The rule called for each record delineated by the regular expression. This rule takes the text from the screens and converts it to a ResourceObject.

## Schema attributes

**Table 1—Mainframe Connector - Configuration parameters**

Parameters	Descriptions
userIterateCommand	The command used to natively iterate over all users
defaultTimeout	The length of time scripts should wait for data to be returned during command execution.
defaultIdleTimeout	The length of time the screen should be idle before timing out.
morePrompt	The prompt scripts should expect to receive to indicate there is more data on the screen
readyPrompt	The prompt scripts should expect to receive to indicate the mainframe is ready

## Schema attributes

---

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group.

## Account attributes

---

Account objects are used when building identities Link objects.

**Table 2—Mainframe Connector - Account Attributes**

Name	Description
USER	The user ID or login ID of the user.
NAME	The user's name.
DEFAULT-GROUP	The default group to which the owner of the attribute belongs.
OWNER	The owner of the profile, or object.
SECURITY-LABEL	The security label assigned to the data being collected as defined by the Open Systems Interconnection Security Architecture.
ATTRIBUTES	The attributes assigned to the user.
GROUP	Group ID for the owner group.

# Chapter 23: SailPoint Microsoft SQL Server

---

The following topics are discussed in this chapter:

Overview.....	217
Supported features .....	217
Supported Managed Systems .....	218
Pre-requisites .....	218
Administrator permissions .....	218
Configuration parameters.....	220
Schema attributes .....	221
Account attributes .....	221
Group attributes.....	222
Provisioning Policy attributes .....	222
Additional information .....	223
Upgrade considerations.....	223
Direct permission.....	224
Amazon Web Services Relational Database Service (AWS RDS).....	224
Troubleshooting.....	225

## Overview

---

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).

## Supported features

---

SailPoint Microsoft SQL Server Connector supports the following features:

- Account Management
  - Manage Microsoft SQL Server Login Users as accounts
  - Manage Microsoft SQL Server Database Users connected to Login Users as attributes of Server login User account
  - Aggregate, Refresh Accounts
  - Create, Update
  - Delete
    - **Server Login Accounts:** The deletion of any server login account would trigger the deletion of associated Database accounts provided that database is not excluded.
    - **Database Accounts:** For deletion of the Database account, all the database roles corresponding to the database account must be removed.

## Overview

- Enable, Disable, Change Password
- Add/Remove Entitlements
- Account - Group Management
  - Aggregate Microsoft SQL Server Database Roles and Server Roles as Account-Groups
  - Aggregate, Refresh Groups
  - Create and Delete only Database Roles
- Permissions Management
  - Permissions directly assigned to accounts and groups as direct permissions during accounts and groups aggregations respectively.
  - Revocation of the aggregated permissions for accounts and groups.

For more information on Direct Permissions, see “Direct permission” on page 224.

## Supported Managed Systems

---

- Following versions of Microsoft SQL Server is supported by the SailPoint Microsoft SQL Server Connector:
  - Microsoft SQL Server 2017
  - Microsoft SQL Server 2016
  - Microsoft SQL Server 2014
  - Microsoft SQL Server 2012
- SailPoint Microsoft SQL Server Connector now supports managing SQL Server hosted on Amazon Web Services Relational Database Service (AWS RDS).  
For more information on AWS RDS, see “Amazon Web Services Relational Database Service (AWS RDS)” on page 224.

## Pre-requisites

---

The compatible JDBC drivers must be used in the classpath of IdentityIQ for connecting to Microsoft SQL Server. For example, `sqljdbc4.jar`.

### Prerequisites for using the AlwaysOn availability groups feature

- All the Server Login accounts must be replicated across all the primary and secondary nodes of the Microsoft SQL Servers which are configured for availability group. Although this is not provided out of the box by Microsoft, there are [MSDN articles](#) on how to achieve it.
- Only the user databases that are in the availability databases of the availability group must be part of the databases managed by the Microsoft SQL Server Application in IdentityIQ.

**Note:** **The Microsoft SQL Connector currently does not support the ability to read from secondary databases that are available for read-only access.**

## Administrator permissions

---

Login using administrator credentials and create a new user on managed system using the following command:

```
CREATE LOGIN <USER> WITH PASSWORD = '<PASSWORD>'
```

Following are the minimum permissions required for Microsoft SQL Server user based on the operations:

Operation	Permissions	
Test Connection	<pre>grant Connect SQL to [user]</pre> <p><b>Note: In order to access databases, service account must have user mapping on the databases with public role defined.</b></p>	
Aggregation	<pre>use [master] GO grant Connect SQL to [user] grant view any database to [user] grant view any definition to [user]</pre> <ul style="list-style-type: none"> <li>• <b>For Microsoft SQL Server version 2014 and above:</b> <pre>grant connect any database to [user]</pre> </li> <li>• <b>For versions below Microsoft SQL Server 2014:</b> <pre>use [database name] GO exec sp_addrolemember 'db_datareader', '[User]' GO</pre> </li> </ul>	
Enable/Disable User	<pre>grant alter any login to [user]</pre>	
Change Password	<pre>grant alter any login to [user]</pre>	
Delete	Account	Role
	<pre>grant alter any login to [user]</pre>	<pre>use [database name] create user [username] for login [server login name] grant alter any role to [username]</pre>

## Configuration parameters

Operation	Permissions	
Create	Account	Role
	<pre>grant alter any login to [user]</pre> <p>To perform any operation on a database, the service account must have database user on the specific database.</p> <pre>use [database name] create user [username] for login [server login name] exec sp_addrolemember db_owner, [username]</pre> <p><b>Note: User must have proper server role assignments to assign the same role to another user.</b></p> <p><b>For example, if administrator has granted Role1, Role2, Role3 roles to user A then, user A can grant only Role1, Role2, Role3 to any other user. User A cannot assign other roles apart from the roles assigned to it.</b></p> <p>Above mentioned permissions are required for adding and removing entitlements.</p>	<pre>use [database name] create user [username] for login [server login name] grant alter any role to [username]</pre>

## Configuration parameters

---

The following table lists the configuration parameters of SailPoint Microsoft SQL Server Connector:

Parameters	Description
URL*	<p>A valid URL of Microsoft SQL Server with the following format:</p> $jdbc:sqlserver://[\text{serverName}[\backslash\text{instanceName}] [:portNumber]]$ <ul style="list-style-type: none"> <li>• <b>jdbc:sqlserver://</b>: (<i>Required</i>) is known as the sub-protocol and is constant</li> <li>• <b>serverName</b>: is the address of the server to connect to. This could be a DNS, IP address, localhost, or 127.0.0.1 for the local computer.</li> <li>• <b>instanceName</b>: is the instance to connect to <i>serverName</i>.</li> <li>• <b>portNumber</b>: is the port to connect to <i>serverName</i>. The default is 1433.</li> </ul>
User*	Administrative Account to connect to Microsoft SQL Server.

Parameters	Description
Password*	Administrative Account password.
Driver*	The name of the Driver class supported by JDBC com.microsoft.sqlserver.jdbc.SQLServerDriver
Included Databases	List of comma separated databases names to be included in the aggregation operation.
Excluded Databases	List of comma separated databases name to be excluded in the aggregation operation.  <b>Note:</b> If the <b>Include Database</b> parameter is populated, the <b>Exclude Database</b> parameter would be ignored.

**Note:** All the parameters in the above table marked with the \* sign are mandatory parameters.

## Schema attributes

---

This section describes the different schema attributes.

### Account attributes

---

The following table lists the account attributes:

Attribute name	Description
native_identity	Native identity represented by default as loginName.
server_login	Server login associated to account.
name	Account name.
principal_id	ID of database principal.
type	Type of the login.
type_desc	Description type of the login.
create_date	Creation date of the login.
modify_date	Last modification date of the login.
sid	SID of the login.
server_name	Server name.
is_fixed_role	If the value is 1, then this row represents an entry for one of the fixed database roles.
owning_principal_id	ID of the principal that owns this database principal.
roles	Server and database roles assigned to the login and its mapped database users.
DBUser	Database users which are associated to the login

## Provisioning Policy attributes

**Note:** It is recommended that, on Managed System the server name and database name must not be the same.

## Group attributes

---

The following table lists the group attributes:

Attribute name	Description
native_identity	Native identity represented by default as groupName.
name	Group name.
database_name	Database name in which group exists.
database_id	Database ID in which group exists.
principal_id	ID of database principal.
roles	Roles assigned to the group.
server_name	Server name of the group.
type_desc	Description type of the group.

## Provisioning Policy attributes

---

The following table lists the provisioning policy attributes for Create Account and Create Group:

Attribute name	Description
<b>Create Account</b>	
Login name*	Server login name.  <b>Note:</b> <ul style="list-style-type: none"><li>With this release of IdentityIQ, the login name would not require the account with suffix <b>@servername</b>.</li><li>After upgrading IdentityIQ to version 7.3, for older application login name must have suffix as <b>@servername</b>.</li></ul>
<b>Create Group</b>	
Group name*	Group name.
Group Type*	Type of the Group. Default: Database Role

# Additional information

---

This section describes the additional information related to the Microsoft SQL Server Connector.

## Upgrade considerations

---

**Note:** After upgrading IdentityIQ to version 7.3, the existing Microsoft SQL Server application would continue to work as it was. SailPoint recommends to create new application to use the new data model.

When upgrading IdentityIQ to version 7.3:

- Microsoft SQL Server Connector would now be having only a single account which is the Server login account. All the associated Database Accounts and Roles along with the direct permissions would be visible on the IdentityIQ UI under the single account as attributes of that account.
- **Aggregation:** All the Database accounts associated with the Server login account would be listed as an attribute along with the respective Database roles. In addition the direct permissions associated with the Server login and Database accounts are aggregated:
  - account aggregation would fetch server accounts and roles without @servername
  - Group aggregation would fetch the server role without @servername

**Note:** Public roles and Application roles would not be aggregated through aggregation.

- For the existing Microsoft SQL Server Connector application to work as per the new data model, set the **useEnhancedConnector** flag to **true** in the application debug page.

Group provisioning policy must be modified as follows:

- a. Navigate to Group Provisioning Policy.
- b. Delete **Group Password** item.
- c. Navigate to edit the **Group Type** and click on value setting.
- d. Delete the **APPLICATION\_ROLE** value.
- e. Click on **Apply** and **Save** button.
- f. Save the Application.

**Note:** The ‘useEnhancedConnector’ attribute that is set to true, must not be deleted or reverted back to false, else the earlier data would be lost or would have to be cleared.

- (*Applicable only if useEnhancedConnector flag is set to true*) For the existing Microsoft SQL Server Connector application to work as per the new data model and to remove the use of @servername for native\_identity, account name, and group name, set the value of **appendHostName** to **false** in the application debug page as follows:

```
<entry key="appendHostName">
 <value>
 <Boolean>false</Boolean>
 </value>
</entry>
```

**Note:** SailPoint recommends the use of the default setting of ‘appendHostName’ that is false.

## Additional information

- To minimize the impact of redesign, customer can continue the option of using the suffix **@hostname** in server login and server role (unless the customer is planning to use High Availability feature). The following changes would be observed if the customer continues the use of **@hostname** and the **useEnhancedConnector** attribute:
  - Database account links would be removed from identities. Correlation rule may need to be modified according to this change.
  - Server login account would still be aggregated as earlier but there are few attribute changes as per new design.
  - There would not be any change in the way Server and Database Roles would be aggregated, hence no changes would be required wherever they would be used. For example, AccessProfile, Roles and so on.

## Direct permission

---

**Note:** With this release of IdentityIQ, for new Microsoft SQL Server application the Direct Permission check box is unchecked by default.

Following targets are supported:

- DATABASE
- SERVER
- PROCEDURE
- ASYMMETRIC\_KEY
- SYMMETRIC\_KEYS
- USER
- CERTIFICATE\_MAPPED\_USER
- DATABASE\_ROLE

For example,

```
GRANT CONNECT on 'databaselogin@databasename:DATABASE'
GRANT CONNECT SQL on 'serverloginname:SERVER'
```

## Amazon Web Services Relational Database Service (AWS RDS)

---

AWS RDS is a managed service provided by Amazon and there are certain restrictions on what an administrative account can perform on the MSSQL instance on AWS RDS. As a result of these restriction, SailPoint Microsoft SQL Server Connector supports Microsoft SQL Server on AWS RDS with the following mentioned limitations:

- Application configuration attributes are the same.

**Note:** The end point URL of AWS RDS instance would have to be obtained from AWS Management Console and it must be accessible from the IdentityIQ Server. AWS Security groups must be updated for this.

- Operations which are prohibited for **master** user of the AWS RDS Microsoft SQL instance would also not be allowed for the service account user for SailPoint Microsoft SQL Server Connector.
- The model system database cannot be managed using SailPoint's Microsoft SQL Server Connector and has to be added to the excluded databases list configuration. It is recommended to exclude the other system databases (namely master, tempdb and msdb) as all operations (that is, provisioning) are not supported for these databases.

- The service account has restricted permissions to the following server roles:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

**Note:** It is recommended that these server roles must be made non requestable in IdentityIQ. Apart from the above listed server roles, for provisioning any custom server roles with service user, the following permission must be assigned to the service user:

```
grant alter any server role to [user]
```

- If SailPoint Microsoft SQL Server Connector is to be used as a read only connector, there is a limitation from AWS RDS for assigning the following permission (in reference to the permissions mentioned under “Aggregation” on page 219):

```
grant connect any database to [user]
```

As a workaround a database account must be associated with the service account for each database that needs to be managed.

**Note:** Support for Windows authentication is yet to be validated for AWS RDS.

## Troubleshooting

---

### 1 - Aggregation fails

When a login user is created in Microsoft SQL Server and is granted permission only on some of the Databases present on the server and if aggregation task is run for that application, Aggregation fails as the user is not able to access other databases.

**Resolution:** In application Configuration page under the “Include Databases” section, provide the complete list of databases (comma separated list) for which the login user have accesses.

This completes the aggregation successfully, and only details of the users present in the list of included database will be fetched.

## **Troubleshooting**

# Chapter 24: SailPoint Microsoft SharePoint Server Connector

---

The following topics are discussed in this chapter:

Overview.....	227
Supported features .....	227
Supported Managed system .....	228
Pre-requisites .....	228
Application Account permissions .....	229
Configuration parameters.....	229
Schema attributes .....	230
Account attributes .....	231
Group attributes.....	231
Provisioning Policy attributes .....	232
Additional information .....	232
Certifications.....	232
Performance improvement.....	232
Troubleshooting.....	233

## Overview

---

SharePoint is central platform used for content management and provisioning of variety of business applications. SharePoint integrates intranet, content management, and document management. It is mostly used by midsize businesses and large departments.

The Microsoft SharePoint Server Connector is designed to manage SharePoint users and groups from all the SharePoint Site Collections present on the SharePoint Server via remote execution of the Microsoft SharePoint Server PowerShell commands.

## Supported features

---

The Microsoft SharePoint Server Connector supports the following features:

- Account Management
  - Manage SharePoint Users as Accounts.
  - Account Aggregation
  - Create
  - Add/Remove Entitlements (Groups from Sites and Subsites)

**Note:** Active Directory Domain Groups are modeled as users in SharePoint. To avoid creating Identities for such groups, connector skips these domain groups during account aggregation. The membership of domain groups to SharePoint groups can be managed from the groups properties post group aggregation.

## Overview

- Account - Group Management
  - Manage SharePoint Groups as Account-Group
  - Aggregate Groups (From Sites and Subsites)
  - Add/remove Active Directory groups from SharePoint groups
- Support native Before/After scripts for provisioning requests.

## References

- “Troubleshooting” on page 233
- “IQService Before/After Scripts” on page 557”
- “Appendix D: IQService”

## Supported Managed system

---

Microsoft SharePoint Server Connector supports following Microsoft SharePoint servers:

- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2010

## Pre-requisites

---

- Before you can use any of the features of the connector, the IQService must be installed on the computer having the same domain as that of SharePoint Server. For more information about installing IQService see, “Appendix D: IQService”.
- Install PowerShell version 3.0 or later on SharePoint Server.
- To enable the Connector to remotely communicate with SharePoint Server using PowerShell commands, perform the following on SharePoint Server computer:
  - a. Ensure that **WinRM** service is running on SharePoint Server and on IQService system.
  - b. To enable PowerShell Remoting, execute the following command on the system:

```
Enable-PSRemoting -Force
```

Configure Trust in SharePoint Server system and IQService system by running the following command on SharePoint Server:

```
Set-Item wsman:\localhost\client\trustedhosts "<IQService Host>"
```

- c. When IQService establishes remote PowerShell session with SharePoint Server, by default it uses **CredSSP** authentication mechanism. Execute the following command to set **CredSSP** as the authentication type for remote PowerShell session to work on respective computers:

- (*On SharePoint Server*): Enable-WSManCredSSP -Role Server
- (*On IQService*): Enable-WSManCredSSP -Role client -DelegateComputer "<SharePoint Server System Name>"

If one wants to use Authentication mechanism as **Default**, add the following entry in the Application debug page using IdentityIQ.

```
<entry key="Authentication" value="default"/>
```

**Note:** **CredSSP and Default are the only two authentication mechanisms that are supported.**

- d. On SharePoint Server and IQService system, restart the **WinRM** service for the new settings to take effect:

```
Restart-Service WinRM
```

## Hardware and Software requirements

- Windows Server 2012 R2
- Windows Server 2012

## Application Account permissions

---

To perform various operations on different SharePoint Site Collections, provide the following permissions to a SharePoint Server user that must be configured as Application User in IdentityIQ application:

- User must be a part of the following groups on SharePoint Server system:
  - Remote Desktop Users
  - WinRMRemoteWMIUsers
  - WSS\_ADMIN\_WPG
- User must have **SPShellAdmin** access role on all the content databases from the SharePoint Server that this connector must manage. This allows connector to execute SharePoint cmdlets. Execute the following command on SharePoint Server to give the **SPShellAdmin** role to the application user:

```
Add-SPShellAdmin -UserName <DOMAIN\UserName> -Database (Get-SPContentDatabase -Identity "WSS_Content")
```

To grant access to all content databases use the following command:

```
Get-SPDatabase | Add-SPShellAdmin DOMAIN\UserName
```

- On SharePoint Server and IQService host, the Application User must have **Read and Execute** permission for **Microsoft.PowerShell32** and **Microsoft.PowerShell**. Execute the following command on SharePoint Server and IQService host systems to allow that permission:

```
Set-PSSessionConfiguration -Name "Microsoft.PowerShell32" -ShowSecurityDescriptorUI
Set-PSSessionConfiguration -Name "Microsoft.PowerShell" -ShowSecurityDescriptorUI
```

- On SharePoint Server and IQService system, restart the **WinRM** service for the new settings to take effect:

```
Restart-Service WinRM
```

- The Application User must have access to all SharePoint Web Applications that must be managed by the connector. Create PowerShell script as follows and execute on SharePoint Management Shell:

```
$webApp = Get-SPWebApplication -Identity "Web App Url"
$webApp.GrantAccessToProcessIdentity("Domain\ User Name ")
Add above lines for each web application.
```

## Configuration parameters

---

This section contains the information that this connector uses to connect and interact with the application.

The Microsoft SharePoint Server Connector uses the connection attributes listed in the following table:

## Schema attributes

Parameters	Description
SharePoint Server	SharePoint Server host name.
Username*	Application account user with permissions as mentioned in “Application Account permissions” in Domain\User form.
Password*	Password of the user.
IQService Host*	Name of the host computer where IQService is deployed and running.
IQService Port*	Port of communication of IQService.
Page Size	Number of objects fetched per page, when iterating over large numbers of objects. (Default: 500).
Site Collections	Site collections to be managed.  For example, <a href="http://sp-server:4208/sites/site_collection1">http://sp-server:4208/sites/site_collection1</a> .  If not specified, manages all site collections from the server.

## Additional configuration parameter

Parameter	Description
setAttributeLevelResult	Enables setting results for every attribute request.  <b>Note:</b> Enabling this parameter will result in delay as every attribute request is committed in Active Directory instead of bulk commit.
skipDomainGroups	Skips Active Directory Domain Groups during account aggregation.  Default: True
manageSubsites	(Optional) Manages provisioning operations of groups which are residing inside subsite of a site collection when set to true as follows in the application xml file as follows:  <code>&lt;entry key="manageSubsites" value="true" /&gt;</code>  <b>Note:</b> After setting this parameter to true, it is recommended to perform account and group aggregation. It would set URL of groups which are inside subsite appropriately which is used for provisioning operations of such groups.  Default: False (Connector would work as is but provisioning of groups which are inside subsite of a site collection would not work as expected).
manageDomainGroups	Manages Active Directory Domain Groups during group aggregation.  Default: True

## Schema attributes

This section describes the different schema attributes.

## Account attributes

---

The following table lists the account attributes ([Table 1—Account attributes](#)):

**Table 1—Account attributes**

Attributes	Description
AccountName	Account name of the user.
DisplayName	Display name of the user.
Email	E-mail address of the user.
Groups	Collection of groups of which the user is a member.
OwnedGroups	Groups that the user owns.
SiteCollections	SharePoint site collections associated with the user.
SID	Unique security ID for the network account of the user.
IsDomainGroup	If user is a domain group.
Notes	Notes in the user properties.
UserName	sAMAccountName of the user.

## Group attributes

---

The following table lists the group attributes ([Table 2—Group attributes](#)):

**Table 2—Group attributes**

Attributes	Description
GroupUrl	Identity attribute of group in <b>ParentWeb.Url\GroupName</b> format.
GroupName	Name of the group.
LoginName	Login name of the group.
ID	Identifier (ID) for the group.
ParentWeb	Parent web of the group.
Description	Description for the group.
AllowMembersEditMembership	Who can edit the membership.
OnlyAllowMembersViewMembership	Who can view the membership of the group.
AutoAcceptRequestToJoinLeave	Whether membership requests are automatically accepted.
AllowRequestToJoinLeave	Whether to allow users to request for membership of the group.
RequestToJoinLeaveEmailSetting	Membership requests to this e-mail address.
Owner	Name of the group owner.

## Provisioning Policy attributes

**Table 2—Group attributes**

Attributes	Description
SiteAdmin	Name of site collection administrator.
ADGroups	List of Active Directory groups which are member of this SharePoint Group.

# Provisioning Policy attributes

---

This following table lists the provisioning policy attributes ([Table 3—Provisioning Policy attributes](#)):

**Table 3—Provisioning Policy attributes**

Attributes	Description
<b>Provisioning policy attributes for Account creation</b>	
AccountName*	Login name of the user. The user must exist in the configured SharePoint user store (for example, Active Directory). For Windows Claim based authentication, the user name must be in encoding format. For example, i:0#w contoso\jim
sitecollectionurl	To create user without assigning entitlements, value of this attribute should be list of SharePoint site collections this user must be added to. This attribute is not applicable when entitlements are requested in create request.
<b>Provisioning policy attributes for Group update</b>	
ADGroups	List of domain group that must be added or removed from the SharePoint group.

**Note:** Attributes marked with \* sign are the mandatory attributes.

## Additional information

---

This section describes the additional information related to the Microsoft SharePoint Server Connector.

## Certifications

---

During group aggregation SharePoint Site Collection administrator can be promoted as owner of the entitlement. This facilitates in assigning certification for each SharePoint Site Collection to respective SharePoint Site Collection Administrator using **Entitlement Owner** certification. To configure this, import **Promote SiteAdmin as Entitlement Owner** rule from `examplerules.xml` file and set this rule as Group Aggregation Refresh Rule in task before running the Account-Group aggregation.

## Performance improvement

---

(Optional) For improving the group aggregation performance, add the following entry in the application debug page:

```
<entry key="manageDomainGroups" value="false"/>
```

## Troubleshooting

---

### 1 - Exception while creating new PowerShell Session

The following error message appears in many scenarios:

```
Exception while creating new PowerShell Session
```

**Resolution:** Perform the following:

1. Ensure that IQService and SharePoint Server are in the same domain.
2. Verify if maximum number of PowerShell users allowed for Application User are not exceeding. Verify by executing the following command:  
`Get-Item WSMAN:\localhost\Shell\MaxShellsPerUser`  
 If number is not adequate then increase the number. For example,  
`Set-Item WSMAN:\localhost\Shell\MaxShellsPerUser 50`
3. Application Account has enough privileges as described in “Application Account permissions”.
4. Allocate enough memory for PowerShell session. Verify memory space using the following command:  
`Get-Item WSMAN:\localhost\Shell\MaxMemoryPerShellMB`  
 If number is not adequate then increase the number. For example,  
`Get-Item WSMAN:\localhost\Shell\MaxMemoryPerShellMB 256`

### 2 - The WS-Management service cannot process the request

The following error message appears when the user has exceeded the maximum number of concurrent shells:

The WS-Management service cannot process the request. This user is allowed a maximum number of 5 concurrent shells, which has been exceeded. Close the existing shells or raise the quota for this user.

**Resolution:** Verify if maximum number of PowerShell users allowed for Application User are not exceeding by executing the following command:

```
Get-Item WSMAN:\localhost\Shell\MaxShellsPerUser
```

If number is not adequate increase it. For example,

```
Set-Item WSMAN:\localhost\Shell\MaxShellsPerUser 50
```

### 3 - Access is denied

**Resolution:** Verify the following:

- if username and password are correctly entered.
- if this application user has enough access on SharePoint Server.

### 4 - Failed to connect to the SharePoint Server

**Resolution:** Verify if SharePoint Server is accessible through IdentityIQ server via IQService. Verify if IP or hostname are resolving correctly.

## 5 - Account aggregation fails

Account aggregation fails with the following error message in IQService logs:

Site Collection Administrator for Site collection: is =>

**Resolution:** Run the following command to get the user added as claim user:

```
$webapp=get-spwebapp -Identity "SITE_URL"
$webapp.grantaccesstoprocessidentity(''DOMAIN\LOGON_USER'')
```

## 5 - Exception while adding account into group

The following error message appears while adding <account> into group - <group\_name>:

The specified group does not exist.

**Resolution:** Ensure that the following points are satisfied in SharePoint Server environment:

1. Verify if group (<group\_name>) exists in SharePoint Server.
2. If <group\_name> exists, verify if it is residing in a subsite of a site collection and check prefix URL of <group\_name> in entitlement catalog.
3. If group is residing in a subsite of site collection then set the following parameter in application debug page:  
`<entry key="manageSubsites" value="true"/>`

**Note:** After setting the manageSubsites parameter to true, it is recommended to perform account/group aggregation before performing any provisioning operation. For more information, see “Additional configuration parameter” on page 230.

# Chapter 25: SailPoint Microsoft SharePoint Online Connector

---

The following topics are discussed in this chapter:

Overview.....	235
Supported features .....	235
Prerequisites .....	236
Administrator permissions .....	236
Configuration parameters.....	236
Schema attributes .....	237
Account attributes .....	237
Group attributes.....	238
Provisioning Policy attributes .....	238

## Overview

---

SailPoint Microsoft SharePoint Online Connector manages SharePoint users and groups from all site collections present on the Microsoft SharePoint Online directory.

The support for multiple site collections and subsites are available under this Connector.

## Supported features

---

SailPoint Microsoft SharePoint Online Connector supports the following features:

- Account Management
  - Manage Microsoft SharePoint Online Users as Accounts
  - Account Aggregation, Refresh Account
  - Create
  - Add/Remove Entitlements (Groups from Sites and Subsites)
- Account - Group Management
  - Manage Microsoft SharePoint Online Groups as Account-Groups
  - Aggregate Groups (From Sites and Subsites)
  - Create, Update, Delete
  - Add/Remove Azure Active Directory Groups from SharePoint Groups
- Support native Before/After scripts for provisioning requests

This feature requires installation and registration of IQService. For more information, see “Installing and registering IQService” on page 555.

## Prerequisites

---

The SharePoint Online Connector application must be registered on SharePoint Online with tenant level access for configured Site Collections.

## Administrator permissions

---

For the SharePoint Online Connector application to be registered on SharePoint Online, perform the following procedure:

1. Login to the following URL with account having global administrator role and generate **Client Id** and **Client Secret**:  
[https://<sitename>.SharePoint.com/\\_layouts/15/appregnew.aspx](https://<sitename>.SharePoint.com/_layouts/15/appregnew.aspx)  
Enter the respective values for the following fields:

Fields	Values
Title	Add-In
AppDomain	localhost
RedirectUrl	https://localhost

2. Click **Create** button, which registers the **Add-In** and returns the success message with created information.  
Grant permissions to **Add-In** to access the SharePoint data.

**Note:** Provide ‘Full Control’ permission level to the tenant scope, to enable read, write and mange the Site Collections information.

3. Navigate to the SharePoint site and enter the following URL to redirect to Grant permission page:  
[https://<sitename>-admin.sharepoint.com/\\_layouts/15/appinv.aspx](https://<sitename>-admin.sharepoint.com/_layouts/15/appinv.aspx) in the browser
4. Enter the **Client Id** created in Step 1 in **AppId** textbox and click the **Lookup** button.  
This would populate the value to other textboxes in **Title**, **App Domain** and **Redirect URL**.
5. Enter the following permission request in XML format:  

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
 <AppPermissionRequest Scope="http://sharepoint/content/tenant"
 Right="FullControl" />
</AppPermissionRequests>
```
6. Click **Create** button.  
This redirects you to a page where you have to click on **Trust**, the add-in proceed further.

## Configuration parameters

---

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Microsoft SharePoint Online Connector uses the configuration parameters listed in the following table ([Table 1—Configuration parameters](#)):

**Table 1— Configuration parameters**

Parameters	Description
Client Id*	Enter the Client Id of the application created on SharePoint Online.
Client Secret*	Enter the Client Secret of the application created on SharePoint Online.
SharePoint Online Domain Name*	Enter the SharePoint Online Domain. For example, contoso.sharepoint.com
Manage All Site Collections	Manages the users and groups from all site collections.  By default the check box is selected. If unselected, user must provide list of site collections that are to be managed under <b>Include Site Collections</b> .
Include Site Collections	List all the site collections that are to be managed.
Exclude Site Collections	List all the site collections that are to be excluded.
Page Size	The number of objects to fetch in a single page for a Site Collection.  Default: 500

\* Indicates the mandatory attributes to create the application.

## Schema attributes

---

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

**Account:** Account objects are used when building identities Link objects.

**Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

## Account attributes

---

The following table lists the account attributes ([Table 2—Account attributes](#)):

**Table 2—Account attributes**

Attributes	Description
LoginName	Login name of the user.
Name	Display name of user.
Email	E-mail address of the user.
FirstName	First name of the user.
LastName	Last name of the user.
ID	Member ID of the user.

## Provisioning Policy attributes

**Table 2—Account attributes (Continued)**

Attributes	Description
Groups	Specifies all the groups to which user belongs to.
SiteCollections	Specifies all the site collections to which user belongs to.

## Group attributes

The following table lists the group attributes ([Table 3—Group attributes](#)):

**Table 3—Group attributes**

Attributes	Description
ID	Identifier (ID) for the group.
Group URL	URL to group in a site collection.
Name	Display name of the group.
Description	Description of the group.
OwnerTitle	Owner title of the group.
Azure AD Groups	Security DomainGroups for SharePoint Online.
AutoAcceptRequestToJoinLeave	Gets or sets a value that indicates whether the request to join or leave the group can be accepted automatically.
AllowMembersEditMembership	Gets or sets a value that indicates whether the group members can edit membership in the group.
RequestToJoinLeaveEmailSetting	Gets or sets the email address to which the requests of the membership are sent.
OnlyAllowMembersViewMembership	Gets or sets a value that indicates whether only group members are allowed to view the membership of the group.
AllowRequestToJoinLeave	Gets or sets a value that indicates whether to allow users to request membership in the group and request to leave the group.

## Provisioning Policy attributes

[Table 4—Provisioning Policy attributes](#) lists the provisioning policy attributes for create account and group and update group respectively.

**Table 4—Provisioning Policy attributes**

Attributes	Description
<b>Create Account</b>	
UserName*	Enter the user name of Office 365. For example, <a href="mailto:johny@contoso.com">johny@contoso.com</a> .
<b>Create Group</b>	

**Table 4—Provisioning Policy attributes (Continued)**

Attributes	Description
Title*	Name of the group.
SiteCollection*	Site collection URL.
<b>Update Group</b>	
OnlyAllowMembersViewMembership	Members who can view the membership list.
AllowMembersEditMembership	Members who can edit the members.
AllowRequestToJoinLeave	Allow request for join or leave from the group.
AutoAcceptRequestToJoinLeave	Membership requests are automatically accepted.
Azure AD Groups	Domain security groups.

**Note:** User cannot add Azure AD Groups while creating SharePoint Online Group due to API limitation.

## **Provisioning Policy attributes**

# Chapter 26: SailPoint Okta Connector

---

The following topics are discussed in this chapter:

Overview.....	241
Supported features .....	241
Pre-requisite .....	242
Configuration parameters.....	242
Schema attributes .....	243
Account attributes .....	243
Group attributes.....	244
Application attributes .....	245
Provisioning Policy attributes.....	245
Create Account Policy .....	245
Create Group Policy .....	246
Disable Policy .....	246
Additional information .....	247
Supported Aggregation Filters .....	247
Account Status Mapping .....	248
Delta Aggregation.....	248
Troubleshooting.....	249

## Overview

---

SailPoint Okta Connector is an enterprise-level solution for centrally storing and managing user profiles and identity data. SailPoint Okta Connector enables single sign-on authentication across multiple applications and devices - even when they are behind firewalls or in the cloud and makes it easier for IT personnel to access essential employee information.

SailPoint Okta Connector manages Users, Groups, Roles and Application using Rest API provided by Okta. In IdentityIQ Okta users are managed as accounts and groups, roles and applications are managed as entitlement.

## Supported features

---

The Okta connector supports the following features:

- Account Management
  - Manage Okta Person as Account
  - Create, Update, Delete
  - Enable, Disable, Unlock
  - Change Password, Refresh Accounts
  - Aggregation, Delta Aggregation, Filter condition for Aggregation

## Configuration parameters

**Note:** For more information on Delta Aggregation, see "Delta Aggregation" on page 248.

- Add/Remove Groups, Roles and Applications
- Discover Schema, Pass Through Authentication
- Account-Group Management
  - Aggregation, Refresh Group
  - Create, Update, Delete Groups
  - Manages Okta Groups and Applications as Account-Groups
- Custom Attributes
  - Support for aggregation
  - Support for provisioning

**Note:** For more information on provisioning, see "Create Account Policy" on page 245.

## Pre-requisite

---

An administrative user must be granted an Okta API token for authentication purposes.

To generate an Okta API token, perform the steps mentioned in the following URL:

[https://developer.okta.com/docs/api/getting\\_started/getting\\_a\\_token.html](https://developer.okta.com/docs/api/getting_started/getting_a_token.html)

**Note:** Okta API tokens generated from the above URL are valid for 30 days and automatically refresh with each API call. Tokens that are not used for 30 days would expire.

## Configuration parameters

---

This section contains the information that the connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Okta connector uses the following connection parameters:

Attributes	Description
<b>Okta Connection Settings</b>	
URL*	The host URL of Okta instance.
API Token*	SSWS API token required for Okta authentication.
Page Size	The maximum size of each data set when querying large number of objects. Minimum value is 1 and maximum value is 200. Default: 200
<b>Aggregation Filter Settings</b>	
Filter Condition for Accounts	Optional condition to bring subset of Accounts during aggregation. For example, status eq "ACTIVE"  <b>Note:</b> For more information on the Aggregation Filters supported on the managed system, see "Supported Aggregation Filters" on page 247.

Attributes	Description
Filter Condition for Groups	<p>Optional condition to bring subset of Groups during aggregation.</p> <p>By default the value is set as follows:</p> <pre>type eq "BUILT_IN" or type eq "OKTA_GROUP"</pre> <p><b>Note:</b> For more information on the Group Aggregation Filters supported on the managed system, see "Supported Aggregation Filters" on page 247.</p>

**Note:** Attributes marked with \* sign are the mandatory attributes.

## Schema attributes

---

This section describes the different schema attributes.

### Account attributes

---

The following table lists the account attributes:

Name	Description
id	Unique key for user.
login	Unique identifier for the user (username).
email	Primary address of the user.
secondEmail	Secondary email address of user typically used for account recovery.
firstName	First name of the user.
lastName	Last name of the user.
middleName	Middle name(s) of the user(s).
displayName	Name of the user, suitable for display to end users.
nickName	Casual way to address the user in real life.
title	User's title, such as 'Vice President'
honorificPrefix	Honorific prefix(es) of the user, or title in most western languages.
honorificSuffix	Honorific suffix(es) of the user.
profileUrl	URL of user's Online profile. For example, a web page.
primaryPhone	Primary phone number of user such as home number.
mobilePhone	Mobile phone number of user.
streetAddress	Full street address component of user's address
city	City or locality component of user's address.
state	State or region component of user's address.
zipCode	Zipcode or postal code component of user's address.

## Schema attributes

Name	Description
countryCode	Country name component of user's address.
postalAddress	Mailing address component of user's address.
preferredLanguage	User's preferred written or spoken languages.
locale	User's default location for purposes of localizing items such as currency, date time format, numerical representations, and so on.
timezone	User's time zone.
userType	Used to identify the organization to user relationship such as 'Employee' or 'Contractor'.
employeeNumber	Organization or company assigned unique identifier for the user.
costCenter	Name of cost center assigned to the user.
organization	Name of user's organization.
division	Name of user's division.
department	Name of user's department.
manager	Display name of the user's manager.
managerId	ID of a user's manager.
status	Status of the user. For example, ACTIVE, PROVISIONED, DEPROVISIONED and so on.
created	Timestamp of user creation.
activated	Timestamp when transition to ACTIVE status completed.
statusChanged	Timestamp when status last changed.
lastUpdated	Timestamp when user was last updated.
lastLogin	Timestamp of last login.
passwordChanged	Timestamp when password was last changed.
groups	Groups assigned to the user.
applications	Applications assigned to the user.
roles	Administrator roles assigned to the user.

**Note:** Custom attributes of User profile from Okta can be populated using the Discover Schema functionality.

## Group attributes

The following table lists the group attributes:

Name	Description
groupId	Unique key for group.
name	Name of the group.

Name	Description
created	Timestamp when group was created.
description	Description of the group.
lastMembershipUpdated	Timestamp when group's memberships were last updated.
type	Determines how a group's profile and memberships are managed.
lastUpdated	Timestamp when group's profile was last updated.
objectClass	Determines the group's profile.
applications	Applications assigned to group.

## Application attributes

---

The following table lists the application attributes:

Name	Description
applicationId	Unique key for application.
name	Unique key for application definition.
label	Unique user-defined display name for application.
created	Timestamp when application was created.
status	Status of application.
signOnMode	Authentication mode of application.
features	Enabled application features.
lastUpdated	Timestamp when application was last updated.

## Provisioning Policy attributes

---

This section lists different policy attributes for Okta Connector.

**Note:** The attributes marked with \* sign are required attributes.

### Create Account Policy

---

Following table describes various attributes in the create account policy.

Attribute	Description
FirstName*	First name of the user.
LastName*	Last name of the user.
Email*	Primary address of the user.

## Provisioning Policy attributes

Attribute	Description
Login*	Must be an email.
Activate	Checked to set the status as provisioned, unchecked to set status as staged.
Password	Login password for the user.

To provision custom attributes, add a similar attribute in the provisioning policy.

For example, if there are custom attributes **customAttr1** and **customAttr2** on Okta and provisioning is required for them, in that case the provisioning plan must contain attributes with the same names that is, **customAttr1** and **customAttr2**.

Following table describes the status of created account according to different parameters provided in create account policy above.

Activate checkbox	Password	Okta Status	IdentityIQ Status
Unchecked	Provided/Not Provided	STAGED	Disabled
Checked	Not Provided	PROVISIONED	Enabled
Checked	Provided	PASSWORD_RESET	Enabled

## Create Group Policy

Following table describes various attributes in the create group policy.

Attribute	Description
Group Name*	Name of the group.
Group Description	Description of the group.

## Disable Policy

Following table describes various attributes in the disable policy.

Attribute	Value	Description
Status*	Suspend	Temporarily disables the account.
	Deprovision	Delete all the applications and deactivate the account.

For more information on the various mapped status of Okta and IdentityIQ, see "Account Status Mapping" on page 248.

# Additional information

---

This section describes the additional information related to the Okta Connector.

## Supported Aggregation Filters

---

Based on added Accounts/Groups Aggregation Filter, only respective accounts/groups are aggregated in IdentityIQ.

### Account Aggregation Filters

For example, if the added filter is **status eq "STAGED"** then only staged account must be aggregated in IdentityIQ.

Aggregation Filters	Description
status eq "STAGED"	Users that have a status of STAGED.
status eq "PROVISIONED"	Users that have a status of PROVISIONED.
status eq "ACTIVE"	Users that have a status of ACTIVE.
status eq "RECOVERY"	Users that have a status of RECOVERY.
status eq "PASSWORD_EXPIRED"	Users that have a status of PASSWORD_EXPIRED.
status eq "LOCKED_OUT"	Users that have a status of LOCKED_OUT.
status eq "DEPROVISIONED"	Users that have a status of DEPROVISIONED.
lastUpdated lt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Users last updated before a specific timestamp.
lastUpdated eq "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Users last updated at a specific timestamp.
lastUpdated gt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Users last updated after a specific timestamp.
id eq "00u1ero7vZFVEIYLWPBN"	Users with a specified id.
profile.login eq "login@example.com"	Users with a specified login.
profile.email eq "email@example.com"	Users with a specified email*.
profile.firstName eq "John"	Users with a specified firstName*.
profile.lastName eq "Smith"	Users with a specified lastName*.

### Group Aggregation Filters

By default group Aggregation filter would aggregate **type eq "OKTA\_GROUP"** or **type eq "BUILD\_IN"** groups.

Aggregation Filters	Description
type eq "OKTA_GROUP"	Groups that have a type of OKTA_GROUP
type eq "APP_GROUP"	Groups that have a type of APP_GROUP
type eq "BUILT_IN"	Groups that have a type of BUILT_IN

## Additional information

Aggregation Filters	Description
lastUpdated lt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with profile last updated before a specific timestamp
lastUpdated eq "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with profile last updated at a specific timestamp
lastUpdated gt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with profile last updated after a specific timestamp
lastMembershipUpdated lt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with memberships last updated before a specific timestamp
lastMembershipUpdated eq "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with memberships last updated at a specific timestamp
lastMembershipUpdated gt "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Groups with memberships last updated after a specific timestamp
id eq "00g1emaKYZTWRYYRRTSK"	Group with a specified id

## Account Status Mapping

Following table lists the mapped status of Okta on IdentityIQ:

Okta Account Status	IdentityIQ Account Status
PROVISIONED	Active
PASSWORD_EXPIRED	
ACTIVE	
RECOVERY	
DEPROVISIONED	Disable
SUSPEND	
STAGED	
LOCKED_OUT	Locked

## Delta Aggregation

The Okta Connector supports only account delta aggregation. On Full Aggregation, the respective timestamp of account aggregation is stored in the Application object which is used by Delta Aggregation to retrieve the changed data into IdentityIQ. This timestamp is updated after each account delta aggregation.

In account delta aggregation changes to user profile attributes and their entitlements are populated.

The delta for Okta user profile attributes are populated from the users API by comparing the timestamp of last successful account aggregation against the last updated attribute.

To detect entitlement changes and deleted users, the data is populated from events API by comparing the timestamp of last successful account aggregation against the published attribute.

**Note:** Log data older than 90 days is not returned, in accordance with Okta's Data Retention Policy. This means, that any data change done prior to 90 days of the last successful account aggregation timestamp, would not be captured and a full account aggregation would be required.

## Troubleshooting

---

### 1 - Issues in Delta Aggregation

Following are the various scenarios where data is not populated in Delta Aggregation:

- Roles assigned to account from API are not populated in delta aggregation
- Custom applications created on Okta managed system and assigned from API are not populated in delta aggregation
- Some type of applications which are removed from users are not aggregated in account delta aggregation  
**Resolution:** Perform full account aggregation task.
- Groups which are deleted from managed system are not captured in account delta aggregation  
**Resolution:** Perform full group aggregation to refresh entitlements.
- For account created with only Default group **Everyone** event system logs are not captured.  
**Resolution:** To see the Default group assigned to account, perform full account aggregation or create account with other group so that in delta aggregation for account, both groups details (default and other group) are aggregated.

### 2 - For Unlocked Okta account from IdentityIQ, account details page does not display correct status

In IdentityIQ, managed account refresh action only affects the status of the account in IdentityIQ. Account Details are not changed and **Status** is one of the account attribute.

**Resolution:** To get the correct account details and value of the **Account status**, execute account aggregation task.

### 3 - Create Account fails for account created with group assigned as 'Everyone'

By default, on Okta managed system the group 'Everyone' gets assigned to every account created. Create account would fail with following error message is displayed:

```
sailpoint.connector.ConnectorException: [ConnectorException] [Error details] Request execution failed. HTTP Error code: 501, Okta Error code: E0000060, errorSummary: Unsupported operation., errorCauses:[].
```

**Resolution:** While performing create account with **Manage user access** select the group type other than **Everyone**.

## **Troubleshooting**

# Chapter 27: SailPoint Oracle Connector

---

The following topics are discussed in this chapter:

Overview.....	251
Supported features .....	251
Supported Managed Systems .....	252
Pre-requisites .....	252
Administrator permissions .....	252
Configuration parameters.....	254
Additional configuration parameter.....	255
Schema attributes .....	255
Account attributes .....	255
Group attributes.....	256
Provisioning policy attributes .....	257
Additional information .....	257
Amazon Web Services Relational Database Service (AWS RDS).....	257
Troubleshooting.....	258

## Overview

---

The Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system (ORDBMS).

SailPoint Oracle Server Connector is a connector to Oracle database server that allows full user administration with provisioning and password management capabilities of Oracle server. Oracle Server Connector manages the following entities of Oracle server:

- User
- Role

## Supported features

---

SailPoint Oracle Connector supports the following features:

- Account Management
  - Manages Oracle users
  - Aggregation, Refresh Accounts, Pass Through Authentication, Discover Schema
  - Create, Update, Delete
  - Enable, Disable, Change Password
  - Add/Remove Entitlements

**Note:** The Oracle Connector respects the case sensitivity of the oracle user name. Users having mixed case character must enclose the name in double quotes for login into the system.

## Overview

- Account - Group Management
  - Manages Oracle groups as ROLE
  - Aggregation, Refresh Groups
  - Create, Update, Delete
- Permission Management
  - Permissions directly assigned to accounts and groups as direct permissions during account and group aggregation.
  - Automated revocation of the aggregated permissions.

## Supported Managed Systems

---

- SailPoint Oracle Connector supports the following versions of Oracle Server:
  - Oracle Database 12cR2
  - Oracle Database 12c
- Note: **The connector manages users and groups of an Oracle 12c and 12cR2 pluggable database (PDB). The users and groups of an Oracle 12c and 12cR2 container database (CDB) are not supported.**
- SailPoint Oracle Connector now supports managing Oracle Database hosted on Amazon Web Services Relational Database Service (AWS RDS).  
For more information on AWS RDS, see “Amazon Web Services Relational Database Service (AWS RDS)” on page 257.

## Pre-requisites

---

The compatible JDBC drivers must be used in the classpath of IdentityIQ for connecting to Oracle Server. For example, ojdbc6.jar.

## Administrator permissions

---

The Oracle administrator must have all the permissions mentioned below for performing the provisioning operations.

Login with administrator credentials and execute the following command to create a new user:

```
CREATE USER ${UserName} IDENTIFIED BY ${Password};
```

The following table lists the required permissions for the specific operations mentioned below in this section:

**Table 1— Operation specific required permissions**

Operation	Required permissions
Test Connection	Test Connection
Account Aggregation	Test Connection and Account Aggregation
Group Aggregation	Test Connection and Group Aggregation
CREATE Account	Test Connection, Account Aggregation and CREATE Account

**Table 1— Operation specific required permissions**

Operation	Required permissions
DELETE/DROP Account	Test Connection, Account Aggregation and DELETE/DROP Account
UPDATE/MODIFY Account	Test Connection, Account Aggregation and UPDATE/MODIFY Account

## Test Connection

SELECT Permission
GRANT create session TO \${UserName};

## Account Aggregation

SELECT Permission
GRANT SELECT ON dba_users TO \${UserName}; <b>(By this command discovering schema is possible)</b>
GRANT SELECT ON dba_sys_privs TO \${UserName};
GRANT SELECT ON dba_role_privs TO \${UserName};

**Note:** To view Sysdba privileges: SELECT \* FROM V\$PWFILE\_USERS

## Group Aggregation

SELECT Permission
GRANT SELECT ON dba_roles TO \${UserName}; <b>(By this command discovering schema is possible)</b>
GRANT SELECT ON dba_tab_privs TO \${UserName};
GRANT SELECT ON dba_col_privs TO \${UserName};
GRANT SELECT ON dba_sys_privs TO \${UserName};
GRANT SELECT ON system_privilege_map TO \${UserName};
GRANT SELECT ON V_\$version TO \${UserName};
GRANT SELECT on V_\$PWFILE_USERS to \${UserName};
GRANT SELECT ON dba_role_privs TO \${UserName};

**Note:** To view Sysdba privileges: SELECT \* FROM V\$PWFILE\_USERS

## Configuration parameters

### CREATE Account

Account	SELECT Permission	
	Role	Profile
GRANT CONNECT TO \${UserName};		
GRANT CREATE USER TO \${UserName};		
GRANT GRANT ANY ROLE TO \${UserName};		
GRANT GRANT ANY PRIVILEGE TO \${UserName};		

### DELETE/DROP Account

Account	SELECT Permission (Role)
GRANT DROP USER TO \${UserName};	GRANT DROP ANY ROLE TO \${UserName};

### UPDATE/MODIFY Account

Account	SELECT Permission	
	Role	Profile
GRANT ALTER USER TO \${UserName};	GRANT ALTER ANY ROLE TO \${UserName};	GRANT ALTER PROFILE TO \${UserName};

## Configuration parameters

---

The following table lists the configuration parameters of SailPoint Oracle Connector:

Parameters	Description
Administrative Username*	Name of the administrative account which has all the privileges to perform the CRUD (Create, Read, Update, and Delete) operations. The default administrator of Oracle Server is <b>system</b> .
Password*	The password of Administrative account.
Driver class*	Name of the type4 driver to use when making connection with oracle server. By default this connector uses <code>oracle.jdbc.driver.OracleDriver</code>

Parameters	Description
Url*	<p>The url to connect to the database. The format is <code>jdbc:oracle:thin:@&lt;HOST&gt;:&lt;PORT&gt;:&lt;SID&gt;</code></p> <p>For example <code>jdbc:oracle:thin:@xxx.xx.xx.xx:1521:ORCL</code> url consist of</p> <ul style="list-style-type: none"> <li>• <b>jdbc:oracle:thin:@</b>: This is common part which states that the connection is made using thin driver.</li> <li>• <b>xxx.xx.xx.xx</b>: server Name or IP of the oracle server</li> <li>• <b>1521</b>: The port number of the oracle server. This port number should be known by the oracle server administrator.</li> <li>• <b>ORCL</b>: The SID of the oracle server.</li> </ul> <p><b>Note:</b> To connect to PDB of Oracle Database 12c, use the service name instead of SID in the URL as follows:</p> <pre>jdbc:oracle:thin:@&lt;HOST&gt;:&lt;PORT&gt;/&lt;SERVICE_NAME&gt;</pre> <p>For example,</p> <pre>jdbc:oracle:thin:@xxx.xx.xx.xxx:1522/orcl.16.23.200</pre>

**Note:** All the parameters marked with the \* sign in the above table are the mandatory parameters.

## Additional configuration parameter

---

Parameter	Description
deleteUserOnCascade	<p>An Oracle database user can have associated objects. Specify <code>deleteUserOnCascade</code> with the value set to <code>true</code> as a configuration attribute to drop all objects in the user's schema before dropping the user:</p> <pre>&lt;entry key="deleteUserOnCascade"&gt;   &lt;value&gt;     &lt;Boolean&gt;true&lt;/Boolean&gt;   &lt;/value&gt; &lt;/entry&gt;</pre>

## Schema attributes

---

This section describes the different schema attributes.

## Account attributes

---

The following table lists the account attributes:

## Schema attributes

Attribute name	Description
USERNAME	User name.
USER_ID	User ID.
ACCOUNT_STATUS	Account status.
DEFAULT_TABLESPACE	Default tablespace.
ROLES	Roles assigned to the user.
PROFILES	Profiles assigned to the user.
TEMP_TABLESPACE	Temporary tablespace.
SYSTEM_PRIVILEGES	System privileges assigned to the user.  <b>Note:</b> SYSDBA and SYSOPER permissions are visible under SYSTEM_PRIVILEGES.
AUTHENTICATION_TYPE	Authentication type.  <b>Values:</b> <ul style="list-style-type: none"><li>• NONE (When role is updated)</li><li>• EXTERNAL</li><li>• GLOBAL</li><li>• PASSWORD</li></ul>

## Group attributes

---

The following table lists the group attributes:

Attribute name	Description
ROLE	Role name.
AUTHENTICATION_TYPE	Authentication type.  <b>Values:</b> <ul style="list-style-type: none"><li>• NONE (When role is updated)</li><li>• EXTERNAL</li><li>• GLOBAL</li><li>• PASSWORD</li></ul>
SYSTEM_PRIVILEGES	System privileges assigned to the role.
ROLES	Roles assigned to the role.
PASSWORD_REQUIRED	Password is required or not.

## Provisioning policy attributes

---

The following table lists the default provisioning policy attributes for Create Account and Create Group:

Attribute name	Description
<b>Create Account</b>	
Username*	Username
Password*	Password of the user.
DEFAULT_TABLESPACE	Default tablespace.
TEMP_TABLESPACE	Temporary tablespace.
PROFILE	Profile
<b>Create Group</b>	
Role*	Role name.
Password	Password for role when authentication type is set as PASSWORD.
Granted Role	Role to assign for new role.
System Privileges	System privileges to assign for new role.

**Note:** All the parameters marked with the \* sign in the above table are the mandatory parameters.

## Additional information

---

This section describes the additional information related to the Oracle Connector.

### Amazon Web Services Relational Database Service (AWS RDS)

---

AWS RDS Oracle is a managed service provided by Amazon and there are certain restrictions on what an administrative account can perform on the ORACLE Database instance on AWS RDS. As a result of these restriction, SailPoint Oracle Connector supports ORACLE on AWS RDS with the following mentioned limitations:

- Application configuration attributes are the same.

**Note:** The end point URL of AWS RDS instance would have to be obtained from AWS Management Console and it must be accessible from the IdentityIQ Server. AWS Security groups must be updated for this.

- By default, Oracle Connector aggregates all the roles, system privileges and profiles as entitlements where Profiles are not requestable. Among the aggregated roles/system privileges, **master** user does not have all the permissions for provisioning on AWS RDS instance.
- For creation of the service account, the following permissions (in reference to the permissions mentioned under “Administrator permissions” on page 252) cannot be assigned to the service account by **master** user:

```
GRANT GRANT ANY ROLE TO ${UserName};
GRANT GRANT ANY PRIVILEGE TO ${UserName};
```

## Troubleshooting

```
GRANT SELECT ON dba_col_privs TO ${UserName};
GRANT SELECT ON V_$version TO ${UserName};
GRANT SELECT on V_$PWFILE_USERS to ${UserName};
```

- In order to enable the service account to grant role (which is allowed on RDS) to a user, the following permission must be given to the service account through the **master** user:

```
GRANT <RoleName> TO ${UserName} WITH ADMIN OPTION
```

Or

For assigning the service user permissions, use the SQL script located at the following location:

<https://community.sailpoint.com/docs/DOC-11323>

This script which would grant all the permissions available with master user **WITH ADMIN OPTION=YES** to the service user.

- In case of custom roles create by master user or any other user which is equivalent to **master**, the owner of the role must assign the role to the service user **WITH ADMIN OPTION**.
- In case of table/system privileges, the revoke option is available with the user who has granted them.

## Troubleshooting

---

### 1 - Provision of SYSDBA and SYSOPER fails

When provisioning SYSDBA and SYSOPER, the following error message is displayed:

"User entitlements not modified sailpoint.connector.ConnectorException: ORA-01031: insufficient privileges"

**Resolution:** If above error message is displayed, the user must connect with the user as "<username> as sysdba" and the <username> must have Sysdba privileges.

### 2 - Error messages appear in access requests when Service user does not have the required permissions

The following error message appears in access requests, when the service user does not have the required permissions available, which would be granted to other users:

[ConnectorException] [Error details] NEW The server encountered an unexpected error while contacting target system. Please check the logs. User entitlements not modified sailpoint.connector.ConnectorException: ORA-01924: role 'AUDIT\_VIEWER' not granted or does not exist

[ConnectorException] [Error details] NEW The server encountered an unexpected error while contacting target system. Please check the logs. User entitlements not modified sailpoint.connector.ConnectorException: ORA-00604: error occurred at recursive SQL level 1 ORA-20997: "ALTER SYSTEM" grants not allowed ORA-06512: at "RDSADMIN.RDSADMIN", line 79 ORA-06512: at line 2

**Resolution:** Ensure that the roles/system privileges being granted are available with service user **WITH ADMIN OPTION=YES**.

# Chapter 28: SailPoint Oracle HRMS Connector

---

The following topics are discussed in this chapter:

Overview.....	259
Supported features .....	259
Supported Managed Systems .....	259
Pre-requisites .....	259
Administrator permissions .....	260
Configuration parameters.....	261
Schema attributes .....	262
Account attributes .....	262

## Overview

---

The Oracle HRMS connector aggregates the HR Person details from Oracle HRMS system. Details would include the Personal and organization information.

### Supported features

---

SailPoint Oracle HRMS Connector supports the following features:

- Account Management
  - Manages Oracle HRMS Users as Accounts
  - Aggregation, Refresh Accounts
  - Update Email address and Work-Telephone

### Supported Managed Systems

---

Following versions of Oracle HRMS are supported by the connector:

- Oracle E-Business Suite 12.2
- Oracle E-Business Suite 12.1

### Pre-requisites

---

The compatible JDBC drivers must be used in the classpath of IdentityIQ for connecting to Oracle Server. For example, ojdbc6.jar.

## Administrator permissions

---

1. Enter the following command to find the rights present on the Package:

```
SELECT dbo.object_name,
(DECODE(SIGN(bitand(options,16)),1,'INVOKER','DEFINER')) "authid"
FROM dba_objects dbo, sys.PROCEDURE$ p
WHERE p.obj# = dbo.object_id
AND dbo.object_type = 'PACKAGE'
AND dbo.object_name = 'xxx'
AND dbo.owner = 'APPS'
```

Where xxx package is **HR\_PERSON\_API** and **HR\_PHONE\_API**.

For example, enter the following command to find the rights present on the **HR\_PERSON\_API**:

```
SELECT dbo.object_name,
(DECODE(SIGN(bitand(options,16)),1,'INVOKER','DEFINER')) "authid"
FROM dba_objects dbo, sys.PROCEDURE$ p
WHERE p.obj# = dbo.object_id
AND dbo.object_type = 'PACKAGE'
AND dbo.object_name = 'HR_PERSON_API'
AND dbo.owner = 'APPS'
```

2. If xxx Package has Invoker rights, perform the following:

Copy the Package scripts from

identityiq\intergation\OracleHRMS\iiqIntegartion-OracleHRMS.zip directory to the OracleHome\bin directory and rename the type of scripts from .txt to .sql.

Using SQL\*Plus, log in to the Oracle database as **APPS** and run the following scripts:

- Run the @SP\_UPDATE\_EMAIL\_API.sql
- Run the @SP\_UPDATE\_EMAIL\_API\_BODY.sql
- Run the @SP\_CREATE\_OR\_UPDATE\_PHONE.sql
- Run the @SP\_CREATE\_OR\_UPDATE\_PHONE\_BODY.sql

3. Log in to the Oracle database as database administrator for creating the new administrator user account using SQL\*Plus as follows:

```
create role ${new role};
create user ${new user} identified by ${password};
grant create session to ${new user};
grant create synonym to ${new user};
grant ${new role} to ${new user};
```

**Grant permissions to the new role created in the above step (\${new role}):**

```
GRANT SELECT ON HR.HR_ALL_ORGANIZATION_UNITS_TL TO ${new role};
GRANT SELECT ON HR.HR_ALL_POSITIONS_F_TL TO ${new role};
GRANT SELECT ON HR.PER_ALL_PEOPLE_F TO ${new role};
GRANT SELECT ON HR.PER_ALL_ASSIGNMENTS_F TO ${new role};
GRANT SELECT ON HR.PER_ASSIGNMENT_STATUS_TYPES_TL TO ${new role};
GRANT SELECT ON HR.PER_PERSON_TYPES_TL TO ${new role};
GRANT SELECT ON HR.PER_JOB_GROUPS TO ${new role};
```

- ```
GRANT SELECT ON HR.PER_JOBS TO ${new role};
GRANT SELECT ON HR.PER_PHONES TO ${new role}
• If xxx package has Definer rights, perform the following:
  GRANT EXECUTE ON APPS.HR_PERSON_API TO ${new role};
  GRANT EXECUTE ON APPS.HR_PHONE_API TO ${new role};
• If xxx package has Invoker rights, perform the following:
  GRANT EXECUTE ON APPS.SP_UPDATE_EMAIL_API TO ${new role};
  GRANT EXECUTE ON APPS.SP_CREATE_OR_UPDATE_PHONE TO ${new role};
4. Login by the new user name ${new user} and create the following synonym:
```

```
CREATE OR REPLACE SYNONYM HR_ALL_ORGANIZATION_UNITS_TL FOR
HR.HR_ALL_ORGANIZATION_UNITS_TL;
CREATE OR REPLACE SYNONYM HR_ALL_POSITIONS_F_TL FOR HR.HR_ALL_POSITIONS_F_TL;
CREATE OR REPLACE SYNONYM PER_ALL_ASSIGNMENTS_F FOR HR.PER_ALL_ASSIGNMENTS_F;
CREATE OR REPLACE SYNONYM PER_ALL_PEOPLE_F FOR HR.PER_ALL_PEOPLE_F;
CREATE OR REPLACE SYNONYM PER_ASSIGNMENT_STATUS_TYPES_TL FOR
HR.PER_ASSIGNMENT_STATUS_TYPES_TL;
CREATE OR REPLACE SYNONYM PER_JOBS FOR HR.PER_JOBS;
CREATE OR REPLACE SYNONYM PER_JOB_GROUPS FOR HR.PER_JOB_GROUPS;
CREATE OR REPLACE SYNONYM PER_PERSON_TYPES_TL FOR HR.PER_PERSON_TYPES_TL;
CREATE OR REPLACE SYNONYM PER_PHONES FOR HR.PER_PHONES;
• If xxx package has Definer rights, perform the following:
  CREATE OR REPLACE SYNONYM HR_PERSON_API FOR APPS.HR_PERSON_API;
  CREATE OR REPLACE SYNONYM HR_PHONE_API FOR APPS.HR_PHONE_API;
• If xxx package has Invoker rights, perform the following:
  CREATE OR REPLACE SYNONYM HR_PERSON_API for APPS.SP_UPDATE_EMAIL_API;
  CREATE OR REPLACE SYNONYM HR_PHONE_API for APPS.SP_CREATE_OR_UPDATE_PHONE;
```

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection. The following table lists the configuration parameters of Oracle HRMS Connector:

Note: Attributes marked with * sign are the mandatory attributes.

| Attributes | Type |
|------------|---|
| URL | URL for server which directly interacts with the Managed system.
For example, jdbc:oracle:thin:@xxx.xx.xx.xxx:xxxx:VIS |
| User | Oracle user minimum permissions mentioned in “Administrator permissions” section.
For example, “apps” |
| Password | Authentication details of login. |

Schema attributes

| Attributes | Type |
|------------|--|
| Driver | Name of the Driver class supported by JDBC Type 4.

For example, "oracle.jdbc.driver.OracleDriver" |

Note: The Oracle HRMS Connector by default aggregates only organizational data with active persons and assignments. If there is a requirement to aggregate personal data, navigate to debug page to access the application created for Oracle HRMS Connector and modify the following entry by setting the value to false:

```
<entry key="aggregateOnlyOrganisationData" value="true"/>
```

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
FULL_NAME	Full name of the employee.
GENDER	Gender of an employee. For example, M-male, F- female
EMAIL_ADDRESS	Email ID of the person. For updating email address the default mode is "UPDATE" and the effective date is considered as "Sysdate".
MARITAL_STATUS	Marital status - M: Married - S: Single - D: Divorced
PERSON_ID	Unique ID from which details of person can be fetched.
SUPERVISOR_ID	Person ID of Supervisor.
BUSINESS_GROUP	Business group from which employee is.
DATE_OF_BIRTH	Date of Birth of Employee.
PERSON_TYPE	Person Type will give the current status. For example, Employee, applicant
SUPERVISOR_NAME	Name of supervisor/mentor/manager of an employee.
EMPLOYEE_NUMBER	If person is an employee then employee number will be displayed at this attribute.
JOB	Job details of an employee.
START_DATE	Start date of the person.
END_DATE	End date of the person.

Attributes	Description
ORGANIZATION	Organization name in which employee is working.
POSITION	Current position or job title of an employee.
WORK_TELEPHONE	Work telephone of an employee. For updating phone number of type Work , the default mode is "Correction". Note: When upgrading from any older version to version 7.3, ensure that you add WORK_TELEPHONE attribute manually.

Schema attributes

Chapter 29: SailPoint PeopleSoft HCM Database Connector

The following topics are discussed in this chapter:

Overview.....	265
Supported features	265
Supported Managed Systems	266
Pre-requisites	266
Administrator permission	266
Configuration parameters.....	267
Schema attributes	269
Account attributes	269
Additional information	270
Configuring Component Interface Security	270
Creating PeopleSoft HRMS Jar File	271
Troubleshooting.....	272

Overview

The SailPoint PeopleSoft HCM Database Connector aggregates and provisions the Personal and Job related data of Person records from PeopleSoft HCM Database system. The connector makes use of database connection for all aggregation related operations. The provisioning operations are handled using the PeopleSoft Component Interfaces.

Supported features

SailPoint PeopleSoft HCM Database Connector supports the following features:

- Account Management
 - Manages PeopleSoft HRMS Person as Account
 - Aggregation, Refresh Accounts
 - Provisioning

IdentityIQ supports the following additional PeopleSoft HCM Database Connector provisioning features in version 7.0 and above:

- Ability to define Global Provisioning Rule
- Ability to define separate provisioning rule for specific operation (operations that include are Enable, Disable, Unlock, Delete, Create, and Modify).

For more information, see “Customization Rule” on page 270.

Supported Managed Systems

SailPoint PeopleSoft HCM Database Connector supports the following PeopleSoft version running on Oracle or Microsoft SQL Server database:

- Supported PeopleSoft versions
 - PeopleSoft HCM versions 9.2 and 9.1
 - PeopleTools version 8.55, 8.54 and 8.53

Pre-requisites

- Using the PeopleSoft Application Designer verify if the following Component interfaces related to person's personal data and job data are present:
 - CI_PERSONAL_DATA
 - CI_JOB_DATA
 - The following jar files must be present on the configured IdentityIQ Application Server:
 - psjoa.jar (found on PeopleSoft server at %PS_HOME%\classes where %PS_HOME% is the location of PeopleSoft installation server directory, referred to as PS_HOME)
 - PeopleSoftHRMS.jar (For more information, see “Creating PeopleSoft HRMS Jar File” on page 271)
- Note: The PeopleSoft jar files can be located in WEB-INF\lib directory.**
- Visit the Oracle website and download an appropriate Oracle JDBC driver and compatible JDK version for IdentityIQ.
- Or**
- Visit the Microsoft SQL Server website and download an appropriate SQL Server JDBC driver and compatible JDK version for IdentityIQ.

Administrator permission

This section describes the Database and Component Interface related permissions.

Database related permissions

The PeopleSoft Application Server will use the database user context to support the aggregation operations. The database user mentioned in the application configuration should have appropriate rights to fetch data related to the entities and attributes mentioned in the SQL query related to Person.

Creating Administrator Account in Oracle database for PeopleSoft HCM

1. Log in to the Oracle database as database administrator for creating the new administrator user account using SQL*Plus as follows:

```
create user ${new user} identified by password;
grant create session to ${new user};
grant create synonym to ${new user};
```

Grant permissions to the new user created from the above step (\${new user}):

```

grant select on ${Administrator}.PS_PERSONAL_DATA to ${new user}
grant select on ${Administrator}.PS_PERSONAL_PHONE to ${new user}
grant select on ${Administrator}.PS_EMAIL_ADDRESSES to ${new user}
grant select on ${Administrator}.PS_JOB to ${new user}

```

2. Login by the new user name (`${new user}`) and create the following synonym:

```

create synonym PS_PERSONAL_DATA for ${Administrator}.PS_PERSONAL_DATA;
create synonym PS_PERSONAL_PHONE for ${Administrator}.PS_PERSONAL_PHONE;
create synonym PS_EMAIL_ADDRESSES for ${Administrator}.PS_EMAIL_ADDRESSES;
create synonym PS_JOB for Admin.PS_JOB;

```

Component interface related permissions

For provisioning operations, the PeopleSoft user who acts as an administrator must have proper access to the HRMS related Component Interfaces. For more information, see “Configuring Component Interface Security” on page 270.

Creating Administrator Account in Microsoft SQL Server database for PeopleSoft HCM

Log in to the Microsoft SQL Server for creating the administrator user account using the following:

```

CREATE LOGIN <LOGINUSER> WITH PASSWORD = '<PASSWORD>'

use [master]

grant Connect SQL to [user]

```

Create corresponding database user and grant permissions to the new user created from the above step (`${new user}`):

```

use [HCM92]

create user [dbusername] for login [LOGINUSER]

grant select on PS_PERSONAL_PHONE to [dbusername];
grant select on PS_EMAIL_ADDRESSES to [dbusername];
grant select on PS_PERSONAL_DATA to [dbusername];
grant select on PS_JOB to [dbusername];

```

Configuration parameters

This section contains the information that IdentityIQ uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Note: Attributes marked with * sign are the mandatory attributes.

The PeopleSoft HCM Database connector uses the following connection attributes:

Attribute	Description
JDBC Connection Settings	

Configuration parameters

Attribute	Description
URL*	The URL with which to connect to the database.
User*	The user with which to connect to the host of the database.
Password*	The password associated with the specified user.
Driver class*	The Java JDBC class to use for the connection.
SQL Query*	<p>Is used to provide a SQL statement which provides a criteria to fetch record of person from PeopleSoft HRMS underlying tables. This query would be used in aggregation task.</p> <p>Default: Query is provided to fetch personal record which also includes email and phone from PeopleSoft HRMS system.</p>
getObjectSQL	The getObjectSQL query will be similar to SQL query, but will fetch details of only one Person record at a time. It is recommended that this query must be updated when the SQL Query is updated.
Person Sub Query	Used to provide a SQL statement which will give a criteria to fetch record of Job from PeopleSoft HRMS underlying tables.
BuildMap Rule	The rule called for each row returned by the database after the SQL has been executed. The rule uses ResultSet and builds a Map out of it to be consumed by IdentityIQ.
PeopleSoft Connection Settings	
Host	Peoplesoft Server host.
Port	JOLT Port.
User	Administrator User of PeopleSoft Server.
Password	Password for the administrator user.
Domain Connection Password Enabled	Determines if Domain connection Password is configured.
Domain Connection Password*	Password is required if Domain Connection Password Enabled attribute is selected.
Jar location	<p>If there are more than one PeopleSoft application of different PeopleTools versions running under the same instance of JVM, the location specified would be added in the classpath. (The <code>psjoa.jar</code> and <code>PeopleSoftHRMS.jar</code> files). For more information, see Pre-requisites).</p> <p>Note: For single PeopleSoft application, the peoplesoft jars can be located in <code>WEB-INF\lib</code> directory.</p>

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
EMPLID	ID of the employee.
COUNTRY	Country of the employee.
CITY	City of the employee.
STATE	State of the employee.
PER_ORG	Organization of the employee.
ADDRESS1	Postal address1 of the employee.
ADDRESS2	Postal address2 of the employee.
ADDRESS3	Postal address3 of the employee.
BIRTHDATE	Birth date of the employee.
FIRST_NAME	First name of the employee.
HR_RESPONSIBLE_ID	HR Responsible ID of the employee.
LAST_NAME	Last name of the employee.
MIDDLE_NAME	Middle name of the employee.
NAME	Name of the employee.
NAME_PREFIX	Prefix of the employee.
NAME_SUFFIX	Suffix of the employee.
NAME_TITLE	Title of the employee.
POSTAL	Postal pin code of the employee.
PREF_FIRST_NAME	Preferred first name of the employee.
EMPL_RCD	Job employee record.
EFFDT	Effective date of the job.
DEPTID	Department ID of the job.
JOBCODE	Job code of the job.
POSITION_NBR	Position number of the job.
SUPERVISOR_ID	Supervisor ID of the job.
HR_STATUS	HR status of the job.
EMPL_STATUS	Employee status of the employee.

Additional information

Attributes	Description
ACTION	Action code of the job.
ACTION_REASON	Action reason of the job.
LOCATION	Location of the job.
FULL_PART_TIME	Full part time of the job.
COMPANY	Company of the employee for current job.
EMPL_TYPE	Employee type of the job.
OFFICER_CD	Officer code of the job.
EMPL_CLASS	Employee class of the job.
ACCT_CD	Account code of the job.
BUSINESS_UNIT	Business unit of the job.
REPORTS_TO	Reporting to of the employee.
HIRE_DT	Hiring date of the employee.
TERMINATION_DT	Termination date of the employee.
EMAIL_ADDR	Email address of the employee.
PHONE	Personal phone of the employee.

Customization Rule

- **Modify Rule:** The rule name is defined as **Example PeopleSoft HRMS Modify Rule**. This is a sample rule to update Collection Attribute (E-mail, Phone and samAccountName) and Non Collection Attribute (BirthPlace).
- **buildMap Rule:** The rule name is **Example PeopleSoft HRMS BuildMap Rule**. This rule is required to fetch additional attributes which are not defined in the account schema.

Additional information

This section describes the additional information related to the PeopleSoft HCM Database Connector.

Configuring Component Interface Security

Before using the connector, allow the PeopleSoft user that the connector is configured with to access the generated component interfaces.

Perform the following to set security for the PeopleTools project:

1. Log into the PeopleSoft web interface.
2. Navigate to **PeopleTools => Security => Permissions & Roles => Permission Lists**.
3. Click **Add a New Value** to create a new permission list. Enter **IIQ_HRMS_PERM** as the name of the permission list, and click **Add**.
4. Click the **Component Interfaces** tab, which will be used in provision rule. For example, **CI_PERSONAL_DATA**.

5. For each Component Interface used in provisioning rule provide an appropriate access based on operation performed.
6. Click **Save** to save the new permission list.
7. Navigate to **PeopleTools => Security => Permissions & Roles => Roles**.
8. Click **Add a New Value** to create a new role. Enter **IIQ_HRMS_ROLE** as the name of the Role, and click **Add**.
9. Enter **Allows access to the IIQ HRMS component interfaces** as the description.
10. Click the **Permission Lists** tab and add the **IIQ_HRMS_PERM** permission list. Click **Save** to save the role.
11. Navigate to **PeopleTools => Security => User Profiles**, and select the user that is being used in the connector.
12. Click the **Roles** tab and add the **IIQ_HRMS_ROLE** role. Click **Save** to add the role to the user.

Creating PeopleSoft HRMS Jar File

Perform the following steps to create the `PeopleSoftHRMS.jar` file:

1. Login to PeopleSoft Application Designer in two tier mode.
2. Open the component interface which will be used in provisioning rule. For example, **CI_PERSONAL_DATA**
3. From the menu select **Build => PeopleSoft APIs**.
4. From the Build PeopleSoft API Bindings window, select the JAVA classes Build checkbox and deselect the **COM Type Library** and **C Header Files Build** check boxes.
5. In the JAVA Classes frame check Build and select the appropriate Component Interfaces from the drop down menu. Select the following options from the drop down menu:
 - `Comlntfc.ComlntfcPropertyInfo`
 - `Comlntfc.ComlntfcPropertyInfoCollection`
 - `PeopleSoft*`
 - `Comlntfc.CI_PERSONAL_DATA*`
 - `Comlntfc.CI_JOB_DATA*`

Specify the appropriate file path for the JAVA files. The Component Interface JAVA files are generated in the `PeopleSoft\Generated\CompIntfc` directory that is created in the specified location.

For example, if you specify `C:\CI` as the file path, then the Component Interface Java files are generated in `C:\CI\PeopleSoft\Generated\CompIntfc`.

6. Compile the JAVA files by performing the following steps:
 - a. Open the command prompt and change directories to the folder where the generated JAVA files are located. For example, `C:\CI`.
 - b. Navigate to `PeopleSoft\Generated\CompIntfc\` directory.
 - c. Run the following command:

```
javac -classpath %PS_HOME%\class\psjja.jar *.java
```

Where `%PS_HOME%` is the location that PeopleSoft is installed.

Note: Ensure that the JAVA compiler used for compiling the generated JAVA files is compatible with the JAVA provided with the PeopleSoft installation that needs to be managed.

- d. (*Optional*) You can delete all the generated java files from the existing directory, except the `.class` files.

Troubleshooting

7. Perform the following steps to package the compiled files as the PeopleSoftHRMS.jar file:
 - a. Open the Command prompt and change directories to the folder where the generated JAVA files are located. For example, if the java files are generated in C:\CI\PeopleSoft\Generated\CompIntfc folder, then run the command from cd C:\CI
 - b. Run the command: jar -cvf PeopleSoftHRMS.jar *
8. Copy the generated PeopleSoftHRMS.jar and %PS_HOME%\class\psjoa.jar files to the computer where IdentityIQ is running.
9. The location of the above jar file must be specified for the Jar location configuration attribute during connector configuration.

Troubleshooting

1 - Cannot find Component Interface

- The Cannot find Component Interface {CI_PERSONAL_DATA} (91,2) ERROR (0,0) : Failed to execute PSSession request error message appears when the Component Interface is not present on the PeopleSoft HRMS Application server
Resolution: Ensure that the Component Interface is present on the PeopleSoft HRMS Application Server.
- The jar file is created but inflated structure is not properly extracted.
Resolution: Extract the jar file and ensure that the inflated structure is present in the PeopleSoft/Generated/CompIntfc directory. If inflated structure is not properly created, recreate it as mentioned in the “Creating PeopleSoft HRMS Jar File” on page 271 section.

2 - While performing test connection, when the supported platform version is Java 1.6 an error message appears

When the supported platform version is Java version 1.6, the following error message appears:

```
java.lang.UnsupportedClassVersionError: psft/pt8/joa/API: Unsupported major.minor version 51.0 (unable to load class psft.pt8.joa.API)
```

Resolution: Ensure that the supported platform version is Java 1.7.

3 - While performing the test connection or aggregation the SQL Query error may appear

The following error message appears if the user does not have minimum permission:

```
Error in SQL Query: User might not have permission to access the table mention in the query
```

Resolution: Ensure that the administrator permissions mentioned in the “Administrator permission” on page 266 section are provided.

4 - While performing provisioning, error message appears if proper access rights are not provided for Component Interface

The following error message appears if minimum permissions are not provided for Component Interface:

```
Operation failed. Please Check logs for more details. InvocationTargetException:  
Method save for Class: PeopleSoft.Generated.CompIntfc.CIPersonalData not able to invoke
```

Resolution: Ensure that the steps mentioned in “Configuring Component Interface Security” on page 270 are performed.

Troubleshooting

Chapter 30: SailPoint RACF Connector

The following topics are discussed in this chapter:

Overview.....	275
Supported features	275
Configuration parameters.....	275
Schema Attributes	277
Account attributes	277
Group attributes.....	281

Overview

The SailPoint RACF Connector is a *read only* connector to read the file produced by the RACF unload utility.

Note: The RACF Connector supports the provisioning operations. For more information, see *SailPoint Connector for RACF Administration Guide* and *SailPoint Quick Reference Guide for Gateway Connectors*.

Supported features

SailPoint RACF Connector supports the following features:

- Account Management
 - Manages RACF users as Accounts
 - Aggregation, Discover Schema
- Account - Group Management
 - Manages RACF groups as Account-Groups
 - Aggregation
- Permission Management
 - Application reads permissions directly assigned to accounts and groups as direct permissions during account and group aggregation.
 - The connector does not support automated revocation of the aggregated permissions and creates work item for such requests

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Configuration parameters

The RACF connector uses the following configuration parameters:

Table 1—RACF Connector - Configuration parameters

Parameters	Description
filetransport	local, ftp, scp
host	The host of the server to which you are connecting.
transportUser	The user to use with ftp and scp. Not valid with local.
transportUserPassword	The password to use with of ftp and scp. Not valid with local.
file	The fully qualified path to the file.
fileEncoding	Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: http://www.iana.org/assignments/character-sets If this field is empty, the default encoding (the value of <code>file.encoding</code> specified by the jvm) is used.
mapToResourceObjectRule	Rule that is called to override the transformation of the data from the <code>Map<String, String></code> form into a <code>ResourceObject</code> .
filterString	Filter lines that match this string.
filterEmptyRecords	If activated, records that have no data are filtered.
preiterativeRule	The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values. This rule is called after the file has been transferred, but before iteration over the objects in the file is started. For validation this rule can use the existing statistics stored by the <code>postIterationRule</code> during the last aggregation. The rule can compare the stored values with the new values to check for problems
postIterativeRule	The post-iterate rule can store away the configuration object and rename/delete the file if desired. This rule is called after aggregation has completed and ALL objects have been iterated.
RACF Attribute Customization Rule	The rule used to extend the parsing capabilities to customer records or redefine existing record configurations. The RACF attribute customization rule creates a map of <code>LineRecord</code> objects that hold the record ID and other field definitions.

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 2—RACF Connector - Account Attributes

Attribute	Description
CLASSES	
CATEGORIES	Defines the categories associated with a general resource. There is one record per general resource/category combination.
KERB_NAME	RACF user name as taken from the profile.
KERB_MAXLIFE	Maximum ticket life.
KERB_KEY_VER	Current key version.
KERB_ENCRYPT_DES	Is key encryption using DES enabled?
KERB_ENCRYPT_DES3	Is key encryption using DES3 enabled?
KERB_ENCRYPT_DESD	Is key encryption using DES with derivation enabled?
KERB_ENCRYPT_A128	Is key encryption using AES128 enabled?
KERB_ENCRYPT_A256	Is key encryption using AES256 enabled?
KERB_KEY_FROM	Key source. Valid values are PASSWORD or PHRASE.
NAME	User ID as taken from the profile name.
CREATE_DATE	The date that the profile was created.
OWNER_ID	The user ID or group name that owns the profile.
ADSP	Does the user have the ADSP attribute?
SPECIAL	Does the user have the SPECIAL attribute?
OPER	Does the user have the OPERATIONS attribute?
REVOKE	Is the user REVOKEd?
GRPACC	Does the user have the GRPACC attribute?
PWD_INTERVAL	The number of days that the user's password can be used.
PWD_DATE	The date that the password was last changed.
PROGRAMMER	The name associated with the user ID.
DEFGRP_ID	The default group associated with the user.
LASTJOB_TIME	The time that the user last entered the system.
LASTJOB_DATE	The date that the user last entered the system.

Schema Attributes

Table 2—RACF Connector - Account Attributes (Continued)

Attribute	Description
INSTALL_DATA	Installation-defined data.
UAUDIT	Do all RACHECK and RACDEF SVCs cause logging?
AUDITOR	Specifies if the user has the auditor attribute.
NOPWD	YES - indicates that this user ID can logon without a password using OID card. NO - indicates that this user must specify a password. PRO - indicates a protected user ID. PHR - indicates that the user has a password phrase.
OIDCARD	Specifies if this user has the OIDCARD data.
PWD_GEN	The current password generation number.
REVOKE_CNT	The number of unsuccessful logon attempts.
MODEL	The data set model profile name.
SECLEVEL	The user's security level.
REVOKE_DATE	The date that the user will be revoked.
RESUME_DATE	The date that the user will be resumed.
ACCESS_SUN	Can the user access the system on Sunday?
ACCESS_MON	Can the user access the system on Monday?
ACCESS_TUE	Can the user access the system on Tuesday?
ACCESS_WED	Can the user access the system on Wednesday?
ACCESS_THU	Can the user access the system on Thursday?
ACCESS_FRI	Can the user access the system on Friday?
ACCESS_SAT	Can the user access the system on Saturday?
START_TIME	After what time can the user logon?
END_TIME	After what time can the user not logon?
SEC_LABELS	The user's default security label.
ATTRIBS	Other user attributes (RSTD for users with RESTRICTED attribute).
PWDENV_EXISTS	Has a PKCS#7 envelope been created for the user's current password?
PWD_ASIS	Should the password be evaluated in the case entered?
PHR_DATE	The date the password phrase was last changed.
PHR_GEN	The current password phrase generation number.
CERT_SEQN	Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered.
PPHENV_EXISTS	Has the user's current password phrase been PKCS#7 enveloped for possible retrieval?

Table 2—RACF Connector - Account Attributes (Continued)

Attribute	Description
ASSOCIATED_MAPPING	Defines the certificate name filter in the DIGTNMAP class associated with this user ID.
CSDATA_CUSTOM	Record type of the User CICS Data record
LNOTES_SHORTNAME	User ID as taken from the profile name.
CICS_OP_CLASSES	The class associated with the CICS operator.
GROUPS	
OVM_UID	User identifier (UID) associated with the user name from the profile.
OVM_HOME_PATH	Home path associated with the user identifier (UID).
OVM_PROGRAM	Default program associated with the user identifier (UID).
OVM_FSROOT	File system root for this user.
PRIMARY_LANGUAGE	The primary language for the user.
SECONDARY_LANGUAGE	The secondary language for the user.
CICS_RSL_KEY	Defines the resource security level (RSL) keys associated with a CICS user. There is one record per combination of user and CICS RSL key.
LDAP_HOST	LDAP server URL.
LDAP_BIND_DN	LDAP BIND distinguished name.
NETVIEW_IC	Command list processed at logon.
NETVIEW_CONSOLE_NAME	Default console name.
NETVIEW_CTL	CTL value: GENERAL, GLOBAL, or SPECIFIC.
NETVIEW_MSGRECVR	Eligible to receive unsolicited messages?
NETVIEW_NGMFADMN	Authorized to NetView graphic monitoring facility?
NETVIEW_NGMFVSPN	Value of view span options.
NDS_UNAME	NDS user name associated with the user ID.
CICS_OPIDENT	The CICS operator identifier.
CICS_OPPRTY	The CICS operator priority.
CICS_NOFORCE	Is the extended recovery facility (XRF) NOFORCE option in effect?
CICS_TIMEOUT	The terminal time-out value. Expressed in hh:mm.
DCE_UUID	DCE UUID associated with the user name from the profile.
DCE_NAME	DCE principal name associated with this user.
DCE_HOMECELL	Home cell name.
DCE_HOMEUUID	Home cell UUID.
DCE_AUTOLOGIN	Is this user eligible for an automatic DCE login?
CERTIFICATE	Defines the names of the certificate profiles in the DIGTCERT class that are associated with this user ID.

Schema Attributes

Table 2—RACF Connector - Account Attributes (Continued)

Attribute	Description
CICS_TSL_KEY	Defines the transaction security level (TSL) keys for a CICS user. There is one record per combination of user and CICS TSL key.
TSO_ACCOUNT_NAME	User ID as taken from the profile name.
TSO_COMMAND	The command issued at LOGON.
TSO_DEST	The default destination identifier.
TSO_HOLD_CLASS	The default hold class.
TSO_JOB_CLASS	The default job class.
TSO_LOGIN_PROC	The default logon procedure.
TSO_LOGIN_SIZE	The default logon region size.
TSO_MSG_CLASS	The default message class.
TSO_LOGON_MAX	The maximum logon region size.
TSO_PERF_GROUP	The performance group associated with the user.
TSO_SYSOUT_CLASS	The default sysout class.
TSO_USER_DATA	The TSO user data, in hexadecimal in the form X<cccc>.
TSO_UNIT_NAME	The default SYSDA device.
TSO_SECLABEL	The default logon security label.
DFP_DATA_RECORDS	Defines the information required by the System Managed Storage facility of the Data Facility Product (DFP).
AREA_NAME	Area for delivery for the user.
BUILDING	Building for delivery.
DEPARTMENT	Department for delivery.
ROOM	Room for delivery.
ADDRESS1	Address line 1
ADDRESS2	Address line 2
ADDRESS3	Address line 3
ADDRESS4	Address line 4
ACCOUNT_NUMBER	User account number for delivery.
MVS_UID	z/OS UNIX user identifier (UID) associated with the user name from the profile.
MVS_HOME_PATH	HOME PATH associated with the z/OS UNIX user identifier (UID).
MVS_PROGRAM	Default Program associated with the z/OS UNIX user identifier (UID).
MVS_MAX_CPUTIME	Maximum CPU time associated with the UID.
MVS_MAX_ASSSIZE	Maximum address space size associated with the UID.
MVS_MAX_FILEPROC	Maximum active or open files associated with the UID.

Table 2—RACF Connector - Account Attributes (Continued)

Attribute	Description
MVS_MAX_PROC	Maximum number of processes associated with the UID.
MVS_MAX_THREADS	Maximum number of threads associated with the UID.
MVS_MAX_MAP_STORAGE	Maximum mappable storage amount associated with the UID.
MVS_MEM_LIMIT	Maximum size of non-shared memory.
MVS_SHMEM_LIMIT	Maximum size of shared memory.
NETVIEW_OPCLASS	OPCLASS value from 1 to 2040.
EIM_LDAPPROFILE	EIM LDAPBIND profile name.

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 3—RACF Connector - Group Attributes

Attribute	Description
SUBGROUPNAME	The name of a subgroup within the group.
MVS_GID	OMVS z/OS UNIX group identifier (GID) associated with the group name from the profile.
CSDATA_CUSTOM	Defines the custom fields associated with a group. There is one record per combination of group and CSDATA custom fields.
MEMBERS	A user ID within the group.
NAME	Group name as taken from the profile name.
SUPERIOR_GROUP	Name of the superior group to this group.
CREATE_DATE	Date that the group was defined.
OWNER_ID	The user ID or group name which owns the profile.
UACC	The default universal access. Valid values are NONE for all groups other than the IBM-defined VSAMDSET group which has CREATE.
NOTERMUACC	Indicates if the group must be specifically authorized to use a particular terminal through the use of the PERMIT command.
INSTALL_DATA	Installation-defined data.
GROUP_MODEL	Data set profile that is used as a model for this group.
UNIVERSAL	Indicates if the group has the UNIVERSAL attribute.
OVM_GID	OMVS z/OS UNIX group identifier (GID) associated with the group name from the profile.
TME_ROLE	Role profile name.

Schema Attributes

Chapter 31: SailPoint Remedyforce Connector

The following topics are discussed in this chapter:

Overview.....	283
Supported features	283
Configuration parameters.....	284
Additional configuration parameters	285
Schema attributes	285
Account attributes	286
Additional account attributes for Remedyforce connector.....	287
Profile attributes.....	288
Provisioning Policy attributes	288
Troubleshooting.....	289

Overview

The SailPoint Remedyforce Connector supports reading and provisioning of Remedyforce accounts, profiles as account groups and implement the **sailpoint.connector** interface.

This connector is written using the `partner.wsdl` and underlying soap interface. The connector uses SOAP stub generated from a wsdl that was available at the time of development. The stubs are generated using axis 1.2. We do not have to generate the stubs once already done. Partner API is easy to use and we can add custom attributes in the schema without generating the stubs. Partner API is generic and have the same java implementation for Remedyforce connector.

The API is fairly rich for SOAP based API and has the concept of login which requires us to login just once for each operation. It has formal models around the user and profile objects and they are generated as part of the stubs. Remedyforce extends account schema of remedyforce to accommodate extra attributes related to BMC Remedy systems.

Supported features

SailPoint Remedyforce Connector supports the following features:

- Account Management
 - Manages Remedyforce users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update,
 - Enable, Disable, Change Password
 - Add entitlement (Account-Groups and User Roles)
 - Add and Remove entitlements (PermissionSet, PublicGroups)

Configuration parameters

- Account - Group Management
 - Manages Remedyforce Profiles as Account-Groups
 - Aggregation, Refresh Groups
- Permission Management
 - Application reads permissions directly assigned to groups as direct permissions during group aggregation.
 - The connector does not support automated revocation of the aggregated permissions and creates work item for such requests

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Remedyforce connector uses the following connection parameters:

Table 1—Configuration parameters

Parameters	Description
Salesforce URL*	Enter the fully qualified url to the root of the remedyforce server. For example, <code>http://login.salesforce.com/services/Soap/u/26.0/</code> Note: Even if the version of the URL is mentioned in the above example, connector would always use version from stub jar of Salesforce which is 33.0. Note: To figure out the url of your site, login to remedyforce.com. Click Develop under the Application heading toward bottom. Next, click API > Generate Partner WSDL, and click Generate. The URL is located under the SalesforceService service name.
Username*	Display name attribute. It's typically in an email in email type format. For example, <code>denise.hunt@demoexample.com</code>
Password*	Defines the password which is used for logging in the managed System.
Manage Active Accounts	Retrieves the active accounts during account aggregation. Otherwise it retrieves all the accounts which are enabled/disabled while account aggregation.

Table 1—Configuration parameters

Parameters	Description
Search Query for User/Profile	<p>Helps to scope the User/Profile that are retrieved during Account or Account-Group aggregation.</p> <p>For instance, specifying the following search query, retrieves only Active Users during Account Aggregation:</p> <pre>select Id from User where IsActive = true</pre> <p>Users or Profiles retrieved during aggregation can be scoped using custom attributes in the where clause as follows:</p> <pre>select Id from User where EMP_DEPARTMENT__c= 'tester'</pre> <p>Note: Only the where clause of the search query can be modified as per the customers requirement.</p> <p>While configuring Remedyforce application if where clause in Search Query For User/Profile field contains apostrophe(') then use backslash(\) prefix to apostrophe. For example,</p> <pre>Find Id,username from user where lastname is buru'4</pre> <p>Expected Query: Select Id,username from user WHERE lastname ='buru\'4'</p>
Connection Timeout	Timeout value in minutes. Default: One minute

Note: In the above table all the attributes marked with * sign are mandatory attributes.

Additional configuration parameters

- To enable the session and getObject feature, update the following parameters in the application debug page:
 - **sessionEnabled**: Set this parameter to **true** to persist the session information. Default: **false**
 - **disableGetObject**: Set this parameter to **true** to perform the 'getObject' operation after creating an account on the managed system. Default: **false**
- To enable the delete operation, set the value of the **deleteToDisable** parameter to **true** as follows in the application debug page:


```
<ProvisioningConfig deleteToDisable="true"/>
```

Schema attributes

This section provides the different attributes of the Account attributes and Profile attributes for Remedyforce connector.

Account attributes

The Remedyforce connector returns several attributes falling into two categories. The first are general attributes: name, city, state, and so on. Additionally, there are entitlement attributes that specifies user level access granted to Remedyforce:

Attributes	Description
UserName	By default, this attribute is the connectors default nativeldentity AND display name attributes. It's typically in an email type format. For example, denise.hunt@demoexample.com
Id	This attribute is the connector default nativeldentity and internal salesforce id like "005A00000014ySylXX".
Name	User's fullname
FirstName	User's firstname
LastName	User's lastname
Alias	User's assigned alias
City	User's city
CommunityNickname	DisplayNames for user's online communities
CallCenterId	User's call center
CompanyName	User's company name
Country	User's country
Department	User's department
Email	User's Email address
Division	User's division
EmployeeNumber	User's employee number
Extension	User's telephone extension
Fax	User's fax number
IsActive	Flag that indicates if the user is active in sf. False would indicate disabled.
EmailEncodingKey	Encoding that should be used during email communications
ProfileId	ID of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected.
ProfileName	Name of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected.
UserRoleId	User Role's Id.
UserRoleName	User Role's name.

Attributes	Description
PublicGroups	Public groups are the entitlements for user.
UserPermissionsMarketingUser	Maps to the Marketing User Flag.
UserPermissionsMobileUser	Maps to the Mobile User Flag.
UserPermissionsOfflineUser	Maps to the Offline user Flag.
Phone	User's phone number.
ReceivesAdminInfoEmails	Receive the remedyforce.com administrator newsletter.
UserType	Type of the user.
UserPermissionsSFContentUser	Maps to Sales Anywhere User.
ReceivesInfoEmails	Receive theremedyforce.com newsletter.
State	User's state.
Title	User's title.

Note: The ProfileId and UserRoleid fields are required in the schema to fetch the ProfileName and userRoleName respectively. If the ProfileId or UserRoleid is removed then profile name and user role name will not be fetched.

Additional account attributes for Remedyforce connector

- In addition to the above account attributes, following are the additional custom attributes which are required for configuration to connect to Remedyforce connector:

Attributes	Description
BMCServicedesk__IsStaffUser__c	Maps to BMC ServiceDesk Staff
BMCServicedesk__Remedyforce_Knowledge_User__c	Maps to Remedyforce Knowledge User
BMCServicedesk__Account_Name__c	Maps to Account Name
BMCServicedesk__remarks__c	Maps to Remarks
BMCServicedesk__IsOutOfOffice__c	Maps to Out of Office
BMCServicedesk__ContactId__c	Maps to Contact Id
BMCServicedesk__Account_ID__c	Maps to Account ID
BMCServicedesk__FPLoginID__c	Maps to FootPrints Login ID
BMCServicedesk__Room__c	Maps to Room attribute

- (For attributes present on user object only) Remedyforce Connector supports aggregating custom attributes present on user object. The custom attributes added in the Remedyforce user schema must have "__c" appended to the attribute names created in Remedyforce.

For example, if you added **EMP_DEPARTMENT** in the user entity on the Remedyforce system, you would add **EMP_DEPARTMENT__c** in the Remedyforce user schema.

Provisioning Policy attributes

Note: The custom attributes can also be updated.

- In addition to the above account attributes, following is the additional schema attribute which is required for configuration to connect to Remedyforce connector:

Note: The attribute in the following table must be added manually when upgrading from any version below 7.0 to version 7.3.

Attribute	Description
PermissionSet	Entitlement and Managed System property must be enabled. It is the PermissionSet assigned to a user. PermissionSet contains settings and permissions which control users action. The available Permission depends on which user license is selected. User can have multiple permission sets.

Profile attributes

Profiles are aggregated during account group aggregation, below are the attributes returned by the group aggregation process.

Attributes	Description
Id	The internal id for this group. For example, 00eA000000OoP6IAK.
Name	The friendly name assigned to the profile. For example, Force.com - Free User, it also has to be unique so is used as the identity and display attribute by default.
UserType	This is the type of profile even though the attribute name would indicate a user.
Description	Description for the profiles.
UserLicense	User's license.

DirectPermissions: The connector reads the permissions assigned to a profile using the remedyforce.com api. To get the permissions, the connector queries the service to describe the profile object. In the returned attribute all of the permissions contained by a group are prefixed with **Permissions**, and camel cases the permission such that right and target are separated by camel case convention. For example, **PermissionsEditTask** or **PermissionsTransferAnyEntity**. We break these down into a Permission attribute per prefixed-attribute.

Provisioning Policy attributes

IdentityIQ has a default Provisioning Policy defined which allows for the creation of accounts. The provisioning policy can be edited to fit specific customer environments.

Most of the fields on the Remedyforce connector default provisioning policy are generated and all fields are marked review required. The provisioning policy attributes must be customized based on specific customer requirements.

Attributes	Description
Create User Policy	

Attributes	Description
Alias	8 character alias, which is required. By default, it generates a value based on lastname and firstname in the field's inline script. It takes first 7 chars from last name and prefixes it with the first character of the first name.
IsActive	Defaults to true.
Username	Defaults to the identity's email address.
Email	Defaults to the identity's email address.
FirstName	Defaults to the identity's first name.
Lastname	Defaults to the identity's last name
CommunityNickname	Defaults to identity's full name.
TimeZoneSidKey	Defaults to America/Los_Angeles. The timezone of the user, it uses a display name defined by sales force. Only a few timezones are defined in the policy drop down and this will need to be customized for each customer.
LocaleSidKey	Defaults to UTF-8. This is the user's locale.
Email EncodingKey	Defaults to UTF-8 and there are several selections to choose from in from the web interface. They can be customized by the customer.
LanguageLocaleKey	Defaults to en_US. There are several selections to choose from in from the web interface. They can be customized by the customer.

Troubleshooting

- The Community nickname must be unique.
- Do not add profileId/userRoleId attribute in create /update user policy as the code would automatically handle when the customer is selecting profile name and userrolename from **Entitlement** section.

Troubleshooting

Chapter 32: SailPoint RSA Authentication Manager Connector

The following topics are discussed in this chapter:

Overview.....	291
Supported features	292
Supported Managed Systems	292
Pre-requisites	292
Administrator permissions	293
Configuration parameters.....	294
Schema attributes	295
Account attributes	295
Group attributes.....	296
Provisioning Policy attributes	296
Additional information	297

Overview

SailPoint RSA Authentication Manager Connector manages the Users, Groups and Access Tokens in the RSA Authentication Manager. The connector manages the following entities:

- Users
- Groups
- Administrative Roles
- Secure ID tokens

The RSA groups are considered as the account-group with support for provisioning (Create, Update, and Delete) while Administrative Roles are additional entitlements which can only be assigned or removed.

Supported features

SailPoint RSA Authentication Manager Connector supports the following features:

- Account Management
 - Manages RSA Users as Accounts
 - Aggregation, Refresh Accounts, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password, PIN Reset
 - Add/Remove entitlements
- Account - Group Management
 - Manages RSA User Groups as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete
- Permission Management
 - Application reads RSA tokens directly assigned to accounts as direct permissions during account aggregation.
 - The connector supports automated revocation of the aggregated permissions.

Note: To aggregate direct permissions while aggregation, select the ‘Include Permission’ check box from account or group schema.

References

- “RSA Token PIN Reset” on page 294

Supported Managed Systems

SailPoint RSA Authentication Manager Connector supports the following versions of RSA Authentication Manager:

- RSA Authentication Manager 8.3
- RSA Authentication Manager 8.2
- RSA Authentication Manager 8.1 SP1

Pre-requisites

• Configuring the Trust Store

Server root certificate should be imported into the keystore for the remote API calls. Ensure to add the following Java option to the application server for SSL SOAP connections:

```
-Djavax.net.ssl.trustStore = <Path of the imported root certificate>
```

• Importing the Server Root Certificate (Java)

When RSA Authentication Manager is installed, the system creates a self-signed root certificate and stores it in RSA_AM_HOME/server/security/server_name.jks directory. This certificate must be exported from the server, and import it into the keystore for remote API clients. Use the Java keytool, as described in the following sections to export and import the certificate into Java clients.

- To export the server root certificate:
 - a. Change directories to `RSA_AM_HOME/appserver/` and enter the following:


```
jdk/jre/bin/keytool -export -keystore
RSA_AM_HOME/server/security/server_name.jks -file am_root.cer -alias
rsa_am_ca
```
 - b. At the prompt for **keystore_password**, press **Enter** without the password.

Note: **Ignore the warning message that appears as the server root certificate will still be exported.**

 - c. The Java keytool outputs the certificate file to the `RSA_AM_HOME/appserver/` directory.
- To import the server root certificate (Java):

Note: **Provide your cacerts keystore password to import the server root certificate. The Java default is "changeit".**

 - a. Locate the server root certificate file that you exported from Authentication Manager, and copy it to the target host.
 - b. Import the certificate to the local cacerts keystore. Change directories to `JAVA_HOME/jre/bin`, and enter the following:

```
keytool -import -keystore SDK_HOME/lib/java/trust.jks -storepass
cacerts_keystore_password -file am_root.cer -alias rsa_am_ca -trustcacerts
```

 - c. The Java keytool displays a confirmation that the certificate was added to the keystore.

Note: **In the above directory paths,**

 - **RSA_AM_HOME** directory is a generic placeholder for `/opt/rsa/am` path
 - **server_name.jks** is a placeholder for `caStore.jks` keystore

Administrator permissions

The RSA Authentication Manager Connector administrator must have enough rights to execute the requested operation.

To assign the rights to the administrator:

- Assign the default administrative roles present on the RSA. For most of the operations **Auth Mgr Realm Admin** administrative role must be assigned.

Note: **For supported versions of RSA Authentication Manager, the TrustedRealmAdminRole administrative role must be assigned.**

- Create new administrative roles with relevant permissions and assigning it a scope and then assign it to the administrator. The scope of an administrative role determines in what security domains an administrator may manage objects and from what identity sources an administrator may manage users. Below are the permissions which can be assigned to the administrative role.
 - **All** grants an administrator permission to perform any administrative action on the object.
 - **Delete** grants an administrator permission to delete an object.
 - **Add** grants an administrator permission to add an object.
 - **Edit** grants an administrator permission to view and edit an object, but not the ability to add or delete.
 - **View** grants an administrator permission to view an object, but not to add, edit, or delete.

RSA Token PIN Reset

The RSA Authentication Manager Connector supports updating the PIN of assigned RSA tokens to an identity.

Configuring RSA Token PIN Reset feature

1. Open IdentityIQ console and import `IIQHOME\WEB-INF\config\workflow_RSA_PIN_Reset.xml` file. The `workflow_RSA_PIN_Reset.xml` file creates the **Update My RSA Token PIN** quick link on the dashboard and adds the Update RSA Token PIN workflow.
2. The **Update My RSA Token PIN** quick link must be visible when logged into IdentityIQ.

Configuration parameters

The following table lists the configuration parameters of RSA Authentication Manager Connector:

Parameters	Description
Host	The RSA Authentication Manager to connect to.
Port	The port to use to connect to RSA Authentication Manager. Default: 7002.
Authentication Manager Version	Select the version of the Authentication Manager that must be managed by the application.
Administrator	The account that has permission to connect to the RSA Authentication Manager resource remotely. This account should have permission to manage this resource.
Password	Password of the Administrator account.
Command Client User*	<p>The command client user name. On installation of RSA Authentication Manager, the system creates a command client user name and password for secure connections to the command server. This user name and password are randomly generated on creation, and are unique to each deployment.</p> <p>Note: For obtaining the command client user name, see “ Obtaining the command client user name and password from Authentication Manager”.</p>
Command Client Password*	<p>Command Client Password corresponding to the Command Client User. Note: For obtaining the command client password, see “ Obtaining the command client user name and password from Authentication Manager”.</p>
Realm	Name of the Realm to manage.
Identity Source	Identity Source name linked to the Realm.
Security Domain	Name of the security domain to manage.
Search Subdomain	Whether or not to manage the subdomain, when the parent security domain is specified for Security Domain field.
Page Size	Limit to fetch number of accounts or groups per iteration through RSA Authentication Manager. Default: 500.

Obtaining the command client user name and password from Authentication Manager

1. From a command prompt on Authentication Manager host, change directories to RSA_AM_HOME/utils.
2. Enter the following:

```
rsautl manage-secrets --action list
```
3. When prompted, enter your master password.
The system displays the list of internal system passwords.
4. Locate the values for command client user name and password.
For example:

```
Command Client User Name.....: CmdClient_vKr9aLK9
Command Client User Password.....: e9SHbK0W4i
```

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following types of objects:

- **Account:** Account objects are used when building identities Link objects.
- **Group:** The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

Attributes	Data type	Description
Guid	String	Guid of the entity (Native Identity).
userID	String	Unique name by which the entity is known by. (Display Attribute).
firstName	String	First name of the entity.
middleName	String	Middle name of the entity.
lastName	String	Last name for which the entity is known by.
Notes	String	Notes or description for the entity.
Email	String	Email of the entity.
certificateDN	String	Certificate DN of the entity.
securityDomain	String	Security Domain Name to which entity belongs.
identitySource	String	Identity Source Name to which entity belongs.
lastModifiedBy	String	Administrator or user who modified the entity last time.
Groups	Multivalued String	Groups Membership (marked as "groupAttribute").

Provisioning Policy attributes

Attributes	Data type	Description
Roles	Multivalued String	Administrative roles assigned to the entity.
lastModifiedOn	String	Last time when the entity was modified.
accountStartDate	String	Time when the entity was created.
accountExpireDate	String	Time when the entity will get expired.
lastLogin	String	Last time when the entity was logged in.
forceChangePassword	Boolean	Whether or not user need to change the password during next logon.
Mobile Number	String	Mobile number of the entity.

Group attributes

The following table lists the group attributes:

Attributes	Data type	Description
guid	String	Guid of the entity (Native identity).
groupName	String	Name by which the entity is known by (Display Attribute).
Notes	String	Notes or description for the entity.
securityDomain	String	Security Domain Name to which entity belongs.
identitySource	String	Identity Source Name to which entity belongs.

Provisioning Policy attributes

The following table lists the provisioning policy attributes for Create Account of RSA Authentication Manager Connector.

Note: The attributes marked with * sign are the required attributes.

Attributes	Description
Provisioning policy attributes for Create Account	
userID*	Unique name by which the entity is known by.
password*	Password for RSA user.
lastName*	Last Name of the user.
firstName	First name of the user.
email	Email of the user.
forceChangePassword	Whether or not user need to change the password during next logon.

Attributes	Description
nextAvailableToken	Select to assign the next available SecureID token to the user.
Provisioning policy attributes for UpdateGroup	
groupName	Name of the group.
notes	Notes or description of the group.
securityDomain	Security Domain Name to which group belongs (Read only attribute).
identitySource	Identity Source Name to which group belongs (Read only attribute).

Additional information

This section describes the additional information related to the RSA Connector.

Active Directory configured as an identity source

When configuring Active Directory as an identity source, it is recommended to verify the default LDAP policy on the active directory server and check for **MaxPageSize** that limits the number of objects that the server will return. The default value is 1000.

Perform the following steps to verify the quotas on the Active Directory server:

1. Open the ADSI Edit page.
2. In the Configuration partition window, navigate to **Services ==> Windows NT ==> Directory Service ==> Query Policies**.
3. In the left pane, click on the **Query Policies** container, then right-click on the **Default Query Policy** object in the right pane, and select **Properties**.
4. Double-click on the **IDAPAdminLimits** attribute and select the **MaxPageSize** attribute.
5. Click Remove and modify the value in the **Value to add** field to add the new value (for example, **MaxPageSize=2000**) and click **Add**.
6. Click **OK** twice.

Note: LDAP policy can also be modified using `Ntdsutil.exe`, follow instructions mentioned in <https://support.microsoft.com/en-us/kb/315071> to view and set LDAP policy on Active Directory server.

Additional information

Chapter 33: SailPoint Salesforce Connector

The following topics are discussed in this chapter:

Overview.....	299
Supported features	300
Administrator permissions	300
Pre-requisites	301
Configuration parameters.....	301
Additional configuration parameters	302
Schema attributes	303
Account attributes	303
Profile attributes.....	306
Group attributes.....	306
Provisioning Policy attributes	307
Upgrade Considerations	307
Query filter for PermissionSet.....	308
Additional information	307
Troubleshooting.....	309

Overview

The SailPoint Salesforce Connector supports reading and provisioning of Salesforce accounts, profiles as account groups and implement the **sailpoint.connector** interface.

This connector is written using the `partner.wsdl` and underlying soap interface. The connector uses SOAP stub generated from a wsdl that was available at the time of development. The stubs are generated using axis 1.2. We do not have to generate the stubs once already done. Partner API is easy to use and we can add custom attributes in the schema without generating the stubs. Partner API is generic and have the same Java implementation for Salesforce connector.

The API is fairly rich for SOAP based API and has the concept of login which requires us to login just once for each operation. It has formal models around the user and profile objects and they are generated as part of the stubs. Earlier Salesforce Connector internally used Enterprise WSDL which was complex to use (regenerating STUB classes by customer) and not much flexible on custom attributes. With Partner WSDL approach, those limitations will be removed.

Note: As Salesforce is in the process of decommissioning the support for TLS version 1.0, SailPoint Salesforce Connector would use TLS version 1.2 for connection purpose.

Supported features

SailPoint Salesforce Connector supports the following features:

- Account Management
 - Manages Salesforce users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add entitlement (Account-Groups, User Roles and Remove Role)
 - Add and Remove entitlements (PermissionSet, PublicGroups)

Note: **Administrator Reset Password operation does not set password provided for the user account. Salesforce sends Email Notification with temporary password to the user for these operations.**

- Account - Group Management
 - Manages Salesforce Profiles as Account-Groups
 - Aggregation, Refresh Groups
 - Aggregate PermissionSet attribute as group object
- Permission Management
 - Application reads permissions directly assigned to groups as direct permissions during group aggregation.
 - The connector does not support automated revocation of the aggregated permissions and creates work item for such requests

Administrator permissions

For user provisioning, it is required that the administrator must have the appropriate rights on the Salesforce Account.

The System Administrator Profile can configure and customize the application.

- Has access to all functionality that does not require an additional license. Can create, edit, and delete custom profiles. Can reset password of multiple user accounts.
- Can add multiple user accounts.
- Has access to all User Accounts, Profile Permissions
- Enable /Disable User Accounts

Profile Access to User Accounts

A user Profile determines what a user can do in the system. By default, the System Administrator Profile can do the most; the Read Only Profile can do the least. For most users, the Standard User Profile is a good choice: it lets people create and edit most records, as well as access and run reports.

(The following assumes that you are a System Administrator for your organization's instance of Salesforce.com.)

Users, Roles and Profiles are all configured within the Setup area. To access these settings when logged in to Salesforce, click on your name in the upper right corner, then choose Setup from the drop-down menu. The Users, Roles and Profiles settings are all available under Manage Users in the lower left Administration Setup menu.

User Licenses create access

Most of your users will need a standard Salesforce user license. This license gives the user full access to Salesforce's CRM features and applications, including Chatter. Other user license options limit user access.

Pre-requisites

For OAuth2 Authentication support, Salesforce connector supports UserName Password flow. For this support User must create connected application on the Salesforce system.

For creating the connected application, see

https://help.salesforce.com/articleView?id=connected_app_create.htm

After creating the connected application, user gets the **Consumer Key** and **Consumer Secret** from the connected application.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Salesforce connector uses the following connection parameters:

Table 1—Configuration parameters

Parameters	Description
Salesforce URL *	(Not applicable if Enable OAuth2 API is selected) Enter the fully qualified url to the root of the salesforce server. For example, http://login.salesforce.com/services/Soap/u/26.0/ Note: Even if the version of the URL is mentioned in the above example, connector would always use version from stub jar of Salesforce which is 33.0. Note: To figure out the url of your site, login to salesforce.com. Click Develop under the Application heading toward bottom. Next, click API > Generate Partner WSDL, and click Generate. The URL is located under the SalesforceService service name.
Username*	Display name attribute. It's typically in an email in email type format. For example, denise.hunt@demoexample.com
Password*	Defines the password which is used for logging in the managed System. Note: This is API user's Salesforce.com password. If the client's IP address has not been whitelisted in your organization, add a security token to your password for OAuth2 authentication. Password must be 'Account Password + Security Token'.

Configuration parameters

Table 1—Configuration parameters

Parameters	Description
Manage Active Accounts	Retrieves the active accounts during account aggregation. Otherwise it retrieves all the accounts which are enabled/disabled while account aggregation.
Exclude Frozen Accounts	Salesforce connector now supports exclusion of the Frozen users during aggregation.
Search Query for User/Profile	<p>Helps to scope the User/Profile that are retrieved during Account or Account-Group aggregation.</p> <p>For instance, specifying the following search query, retrieves only Active Users during Account Aggregation:</p> <pre>select Id from User where IsActive = true</pre> <p>Users or Profiles retrieved during aggregation can be scoped using custom attributes in the where clause as follows:</p> <pre>select Id from User where EMP_DEPARTMENT__c= 'tester'</pre> <p>Note: Only the where clause of the search query can be modified as per the customers requirement.</p> <p>While configuring Salesforce application if where clause in Search Query For User/Profile field contains apostrophe(') then use backslash(\) prefix to apostrophe. For example,</p> <pre>Find Id,username from user where lastname is buru'4</pre> <p>Expected Query: Select Id,username from user WHERE lastname = 'buru\'4'</p> <p>Note: If Manage Active Accounts is selected then it would add the following condition in the where clause of Search Query for User: AND user.IsActive = true</p>
Connection Timeout	Timeout value in minutes. Default: One minute
OAuth2 Client API Configuration	
Enable OAuth2 API	Select this option when Connected Application is configured to support OAuth2 authentication. Default: false
Client ID*	(Applicable if Enable OAuth2 API is selected) The Client ID for OAuth2 authentication (Consumer key from Salesforce connected application).
Client Secret*	(Applicable if Enable OAuth2 API is selected) The Client Secret for OAuth2 authentication (Consumer Secret from Salesforce connected application).

Note: In the above table all the attributes marked with * sign are mandatory attributes.

Additional configuration parameters

- To enable the session and getObject feature, update the following parameters in the application debug page:
 - **sessionEnabled**: Set this parameter to **true** to persist the session information. Default: **false**

Note: Not applicable for OAuth2 Authentication.

- **disableGetObject**: Set this parameter to **true** to perform the 'getObject' operation after creating an account on the managed system. Default: **false**
- To enable the delete operation, set the value of the **deleteToDisable** parameter to **true** as follows in the application debug page:
`<ProvisioningConfig deleteToDisable="true"/>`
- **OAuth2TokenURL**: This parameter can be used in future if there is any change in OAuth2 Token URL. Add the following parameter in the application debug page:
`<entry key="OAuth2TokenURL" value="https://login.salesforce.com/services/oauth2/token"/>`
- To change the query filter for fetching Roles, add the **SearchStringForRole** parameter in the application debug page as follows:
`<entry key="SearchStringForRole" value="where name like 'CF%' "/>`

Schema attributes

This section provides the different attributes of the Account attributes and Profile attributes for Salesforce connector.

Note: The Identity Attribute for account and group has been changed to 'Id'. For applications created in prior to version 6.4, IdentityIQ would display the Identity Attribute for account as 'Username' and for groups the Identity Attribute would be 'Name'.

Account attributes

The Salesforce connector returns several attributes falling into two categories. The first are general attributes: name, city, state, and so on. Additionally, there are entitlement attributes that specifies user level access granted to Salesforce:

Attributes	Description
UserName	By default, this attribute is the connectors default nativ entityId AND display name attributes. It's typically in an email type format. For example, denise.hunt@demoexample.com
Id	This attribute is the connector default nativ entityId and internal salesforce id like "005A00000014ySylXX".
Name	Users fullname.
FirstName	Users firstname.
LastName	Users lastname.
Alias	Users assigned alias.
City	Users city.
CommunityNickname	Display Names for user's online communities
CallCenterId	Users call center.
CompanyName	Users company name.

Schema attributes

Attributes	Description
Country	Users country.
Department	Users department.
Email	Users Email address.
Division	Users division.
EmployeeNumber	Users employee number.
Extension	Users telephone extension.
Street	Name of the street.
Fax	Users fax number
FederationIdentifier	A Federation ID is an identifier that is unique within a salesforce Organization.
IsActive	Flag that indicates if the user is active in sf. False would indicate disabled.
EmailEncodingKey	Encoding that should be used during email communications
ProfileId	ID of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected.
ProfileName	Name of the profile assigned to a user. Profiles contain settings and permissions, which control what users can do. The available profiles depend on which user license is selected.
Role	The role assigned to a user.
PublicGroups	Public groups are the entitlements for user.
UserPermissionsMarketingUser	Maps to the Marketing User Flag.
UserPermissionsMobileUser	Maps to the Mobile User Flag.
UserPermissionsOfflineUser	Maps to the Offline user Flag.
Phone	Users phone number.
ReceivesAdminInfoEmails	Receive the salesforce.com administrator newsletter.
UserType	Type of the user.
UserPermissionsSFContentUser	Maps to Sales Anywhere User.
ReceivesInfoEmails	Receive the salesforce.com newsletter.

Attributes	Description
PermissionSet	PermissionSet assigned to a user. PermissionSet contain settings and permissions which control users action. The available Permission depends on which user license is selected. User can have multiple permission sets. Note: For more information on aggregating PermissionSet attribute as group object, see "Upgrade Considerations" on page 307.
UserLicense	User's license
State	Users state.
Title	Users title.
TimeZoneSidKey	Defaults to America/Los_Angeles. The timezone of the user, it uses a display name defined by sales force. Only a few timezones are defined in the policy drop down and this will need to be customized for each customer.
LocaleSidKey	Defaults to UTF-8. This is the user's locale.
Email EncodingKey	Defaults to UTF-8 and there are several selections to choose from in from the web interface. They can be customized by the customer.
LanguageLocaleKey	Defaults to en_US. There are several selections to choose from in from the web interface. They can be customized by the customer.

Note: The ProfileId and UserRoleid fields are required in the schema to fetch the ProfileName and userRoleName respectively. If the ProfileId or UserRoleid is removed then profile name and user role name will not be fetched.

Additional account attributes for Salesforce connector

- (For attributes present on user object only) Salesforce Connector supports aggregating custom attributes present on user object. The custom attributes added in the Salesforce user schema must have "__c" appended to the attribute names created in Salesforce.

For example, if you added **EMP_DEPARTMENT** in the user entity on the Salesforce system, you would add **EMP_DEPARTMENT__c** in the Salesforce user schema.

Note: The custom attributes can also be updated.

- Support for custom attributes which are not present in Salesforce system but are required internally in IdentityIQ (may be for a process in a correlation rule):

SailPoint Salesforce Connector respects the attribute only if it starts with _#.

For example _#Emp_company_history.

Role attributes

When aggregating **Group** attribute as Role object, following are the attributes returned by the role aggregation process.

Schema attributes

Attributes	Description
Id	Role Id.
Name	Role name

Aggregating Group as role object would be able to:

- delete Role associated to the user while doing provisioning
- assign new Role to user while doing provisioning

Group attributes

When aggregating **PermissionSet** attribute as group object, following are the attributes returned by the group aggregation process.

Attributes	Description
Name	Internal ID of this group.
Label	Friendly name assigned to the group.
Description	Description for the profiles.

Aggregating PermissionSet as group object would be able to:

- to verify the aggregated permissions set which are not even associated with users currently
- assign new permissions set to user while doing provisioning

Profile attributes

Profiles are aggregated during account group aggregation, below are the attributes returned by the group aggregation process.

Attributes	Description
Id	The internal id for this group. For example, 00eA000000OoP6IAK.
Name	The friendly name assigned to the profile. For example, Force.com - Free User, it also has to be unique so that it can be used as the identity and display attribute by default.
UserType	This is the type of profile even though the attribute name would indicate a user.
Description	Description for the profiles.
UserLicense	User's license. Note: This attribute must be added manually when upgrading from any version below 6.4 to version 7.3.

DirectPermissions: The connector reads the permissions assigned to a profile using the salesforce.com api. To get the permissions, the connector queries the service to describe the profile object. In the returned attribute all of the permissions contained by a group are prefixed with **Permissions**, and camel cases the permission such that

right and target are separated by camel case convention. For example, **PermissionsEditTask** or **PermissionsTransferAnyEntity**. We break these down into a Permission attribute per prefixed-attribute.

Provisioning Policy attributes

IdentityIQ has a default Provisioning Policy defined which allows for the creation of accounts. The provisioning policy can be edited to fit specific customer environments.

Most of the fields on the Salesforce connector default provisioning policy are generated and all fields are marked review required. The provisioning policy attributes must be customized based on specific customer requirements.

Attributes	Description
Create User Policy	
Alias	8 character alias, which is required. By default, it generates a value based on lastname and firstname in the field's inline script. It takes first 7 chars from last name and prefixes it with the first character of the first name.
IsActive	Defaults to true.
Username	Defaults to the identity's user name.
Email	Defaults to the identity's email address.
FirstName	Defaults to the identity's first name.
Lastname	Defaults to the identity's last name
CommunityNickname	Defaults to identity's full name.
TimeZoneSidKey	Defaults to America/Los_Angeles. The timezone of the user, it uses a display name defined by sales force. Only a few timezones are defined in the policy drop down and this will need to be customized for each customer.
LocaleSidKey	Defaults to UTF-8. This is the user's locale.
Email EncodingKey	Defaults to UTF-8 and there are several selections to choose from in from the web interface. They can be customized by the customer.
LanguageLocaleKey	Defaults to en_US. There are several selections to choose from in from the web interface. They can be customized by the customer.
Federation Identifier	A Federation ID is an identifier that is unique within a salesforce Organization.

Additional information

This section describes the additional information related to the Salesforce Connector.

Upgrade Considerations

- When upgrading IdentityIQ to version 7.3, add the **IsFrozen** schema attribute through the application debug page as follows:

```
<AttributeDefinition name="IsFrozen" type="boolean">
```

Additional information

```
<Description>Flag that indicates if the user is frozen in  
Salesforce.</Description>  
</AttributeDefinition>
```

- When upgrading IdentityIQ from any version below 6.4 to IdentityIQ version 7.3, add the following attributes in account schema, which are required for configuration to connect to Salesforce connector:

Attributes	Description
PermissionSet	PermissionSet assigned to a user. PermissionSet contain settings and permissions which control users action. The available Permission depends on which user license is selected. User can have multiple permission sets.
UserLicense	User's license.

- Aggregating **PermissionSet** as group object:

When upgrading IdentityIQ to version 7.3, add the following group object schema in the existing application manually to aggregate **PermissionSet** as group object:

```
<Schema displayAttribute="Name" identityAttribute="Name"  
nativeObjectType="PermissionSet" objectType="PermissionSet">  
    <AttributeDefinition name="Name" type="string">  
        <Description>The internal id for this group.</Description>  
    </AttributeDefinition>  
    <AttributeDefinition name="Label" type="string">  
        <Description>The friendly name assigned to the profile.</Description>  
    </AttributeDefinition>  
    <AttributeDefinition name="Description" type="string">  
        <Description>Description for the profiles.</Description>  
    </AttributeDefinition>  
</Schema>
```

- Update the account schema attribute **PermissionSet** to add the **schemaObjectType="PermissionSet"** as follows:

```
<AttributeDefinition entitlement="true" managed="true" multi="true"  
name="PermissionSet" type="string" schemaObjectType="PermissionSet">  
    <Description>PermissionSet assigned to a user.</Description>  
</AttributeDefinition>
```

- When upgrading IdentityIQ to version 7.3, up-gradation of existing application is not recommended with Role as a group object due to data discrepancy.

Query filter for PermissionSet

To change the query filter for **PermissionSet** fetch query, add the attribute as follows in the application attribute map:

```
<entry key="SearchStringForPermissionSet" value="where IsOwnedByProfile=false"/>
```

Troubleshooting

- ProfileName is required to select in **Entitlement** section as it is mandatory to Salesforce, else it will not create account in Salesforce system.
- If Duplicate User error is displayed, it means that the email Id of that user is already utilized in Salesforce system. You have to create new account with different email id. Email id is the username of the user which is mandatory.
- If any of the create User policy attributes are not filed up, it will display an error and it may happen in rare case that the email id is used by Salesforce system and you have to create new account with different email Id's.
- The Community nickname must be unique.
- Do not add profileId/userRoleId attribute in create /update user policy as the code will automatically handle when the customer is selecting profile name and userrolename from **Entitlement** section.
- If the total active account limit on Salesforce exceeds, the connector fails to create new account by displaying an error. You can increase the account limit on Salesforce or disable the existing active accounts in Salesforce system before initiating create request.
- In Salesforce, Email notification is not sent to the user if the `disableUserCreationEmail` flag is set to true as follows:

```
<entry key="disableUserCreationEmail">
    <value>
        <Boolean>true</Boolean>
    </value>
</entry>
```

- If the Salesforce url has version mismatch with the existing stubs version, and if any of the Aggregation task fails, then generate new stubs of that specific version.

For more information, see “ Configuration of Stubs generation” section below.

Configuration of Stubs generation

Salesforce.com allows each customer to extend the schemas for objects. Customers add new attributes specific to the data kept by a company as the connector uses SOAP. Each time a customization is made to the salesforce.com data model by the customer a new wsdl file is retrieved and integrated into the IdentityIQ environment. To integrate changes from salesforce.com to IdentityIQ perform the following:

1. Download the new version of the partner wsdl from Salesforce.com
After your changes to the salesforce.com model you must download the newest wsdl file from salesforce.com which will include the customized attributes as part of the SOAP stub model. This involves logging into Salesforce and requesting the new wsdl file, the wsdl file is generated using the customer's data model defined in salesforce when the wsdl is generated.
In salesforce, the navigation is: **Setup => App Setup => Develop => API => Generate Partner WSDL**.
2. Generate stubs using the downloaded wsdl file and AXIS.
After generating and downloading the new version, run them through AXIS classes that will generate java stubs. Use the `iijq` command to configure the class path to AXIS and then point to the new wsdl file.
For example, `$iijq org.apache.axis.wsdl.WSDL2Java -p sailpoint.connector.salesforce.webservices.partner partner.wsdl`
Note: The `-p` flag tells the generator which package to place the stubs under. In this case the class files will be placed into a directory named **SailPoint**
3. Compile generated stubs using JAVA.

Troubleshooting

The stubs that are generated by axis are java class files and reference only the axis libraries. Building the java files is very simple and below are some instructions on how this can be done.

The stubs must be compiled using axis 1.4 (the old version of axis not axis2) along with any supported jdk.

The compile procedure appears as the following example:

- a. Create a directory called salesforcestubs/src.
- b. Copy files from the sailpoint directory generated from the wsdl file into salesforcestubs/src directory.
- c. In the salesforcestubs directory create a file called build.xml and copy the sample build.xml (listed in the last section) into that file.

Directory structure of salesforcestubs should look like this:

```
build.xml  
src/sailpoint/connector/salesforce/webservices/partner  
(this is where all the .java files will be)  
d. Edit the pathToAxis1_4 property to point at the axis 1.4 distribution (where axis.jar exists). If this is being done on a local box where IdentityIQ is expanded, you can alternately just specify the iiqHome and it will find the jar file relative to that directory.  
e. Run ant command. This will compile all of the Java files into .class files which are placed into a directory called build which has the typical web-application directory structure. (For example, WEB-INF/classes/).
```

4. Copy newly compiled stubs into WEB-INF/classes/sailpoint/connector/salesforce/webservices/partner directory.

Copy the new class files manually by copying the contents produced in the build directory to the installation directory.

If you are doing this locally on the same machine AND using the example build.xml alternatively, you can use the 'copy' target (ant copy). This target will copy the new salesforce.com stub class files from the build area into to the iiqHome directory specified at the top of the build.xml file.

Once the files have been copied, the java work is done then configure IdentityIQ to fetch and write the custom attributes

5. Restart the application server.

After the new class files have successfully been copied over to the IdentityIQ directory you must reboot the application server and close any open consoles.

In order for the salesforce connector to read the new version, change the old version to new version in salesforce URL. Additionally, if you are provisioning salesforce accounts, define new attributes in your provisioning policies.

Sample build.xml file

Using the following build.xml file along with ant makes building these stubs simple.

```
<!-- (c) Copyright 2008 SailPoint Technologies, Inc., All Rights Reserved. -->  
<project name="salesforceStubs" default="build" basedir=".">>  
    <description>  
        Build file for to help compiling stubs for sales force.  
    </description>  
    <!-- ***** -->  
    <!-- Global properties -->  
    <!-- ***** -->  
  
    <property name="src" location="src" />
```

```

<!-- Only necessary if you want to copy the class files using ant -->
<property name="iiqHome" location="c:/home/SystemID/work/trunk/build/" />

<!-- This needs to point at a directory with the axis 1.4 distribution. -->
<!-- By default relative to the iiq instalation, but doesn't have to be -->
<!-- if on a different machine -->
<property name="pathToAxis1_4"
          location="${iiqHome}/WEB-INF/lib/" />

<property name="build" location="build" />

<!-- **** -->
<!-- Target: build -->
<!-- **** -->

<target name="build" >
    <mkdir dir="${build}/WEB-INF/classes" />

    <!-- build the salesforce axis stubs -->
    <javac destdir="${build}/WEB-INF/classes"
           debug="true" fork="true" memoryMaximumSize="512m"
           includeantruntime="false">
        <src path="${src}" />
        <classpath>
            <fileset dir="${pathToAxis1_4}">
                <include name="**/*.jar" />
            </fileset>
        </classpath>
    </javac>

</target>

<!-- **** -->
<!-- Target: clean -->
<!-- **** -->
<!-- Clean build area -->
<!-- **** -->
<!-- **** -->

<target name="clean">
    <delete dir="${build}" />
</target>

<!-- **** -->
<!-- Target: copy -->
<!-- **** -->
<!-- Copy stub class files into the IIQ installation -->
<!-- Target: copy -->
<!-- **** -->

```

Troubleshooting

```
<target name="copy" depends="build">
  <copy todir="${iiqHome}">
    <fileset dir="${build}">
      <exclude name="**/*.java" />
    </fileset>
  </copy>
</target>

</project>
```

Chapter 34: SailPoint SAP HR/HCM Connector

The following topics are discussed in this chapter:

Overview.....	313
Supported features	313
Supported Managed Systems	314
Pre-requisites	314
Administrator permissions	314
Configuration parameters.....	317
Schema Attributes	319
Account attributes	319
Additional information	325
Troubleshooting.....	330

Overview

The SailPoint SAP HR/HCM Connector aggregates and provisions the employee information from the SAP HR/HCM system.

SAP HR/HCM Connector supports the following SAP Info Types:

- Action (0000)
- Organizational Assignment (0001)
- Personal Data (0002)
- Addresses (0006)
- Communication (0105)

In addition to aggregate any specific Info Type the build map rule can be used.

Supported features

SailPoint SAP HR/HCM Connector supports the following features:

- Account Management
 - Manages SAP HR/HCM employees as Accounts (Active, Terminated and Future Hires)
 - Aggregation, Partitioning Aggregation, Delta Aggregation, Refresh Accounts

For more information on Delta and Partitioning Aggregation, see “Delta Aggregation” on page 324 and “Partitioning” on page 324.

- Provisioning (with the help of rule) support added for IdentityIQ version 7.0 and above
 - Ability to define separate provisioning rule for specific operation (operations that include are Enable, Disable, Unlock, Delete, Create, and Modify).

An example of modify provisioning rule is located in WEB-INF/config/examplerules.xml file. For more information, see “Customization Rule” on page 325 section.

Supported Managed Systems

Following versions of SAP HR/HCM system are supported by the SAP HR/HCM connector:

- SAP ECC 6.0 on SAP NetWeaver 7.5, 7.4, 7.3, 7.2, 7.1 and 7.0

Pre-requisites

SAP JCO version 3.0.x libraries, along with `sapjco3.dll` (on Microsoft Windows) or `libsapjco3.so` (on UNIX), must be present in the `java.library.path` directory on the IdentityIQ host. The JCO libraries (JCO Release 3.0.x) must be downloaded from the SAP website by navigating to the customer service marketplace.

Administrator permissions

The following table lists the required permissions for the specific operations mentioned below in this section:

Table 1— Operation specific required permissions

Operation	Required permissions
Test Connection	Test Connection
Account Aggregation	Test Connection and Account Aggregation
Delta Aggregation	Test Connection, Account Aggregation and Delta Aggregation
Provisioning Rule	Test Connection, Account Aggregation and Provisioning Rule

The role assigned to the SAP Administrative user must have the following Authorization Objects as mentioned in the tables below.

Test Connection

Authorization Objects	Field name	Field description	Field value
S_RFC	ACTVT	Activity	16 -Execute
	RFC_NAME	Name of RFC object	RFCPING
	RFC_TYPE	Type of RFC object	FUGR, FUNC

Account Aggregation

Authorization Objects	Field name	Field description	Field value
S_RFC	RFC_NAME	Name of RFC object	OPBAPI0105,BAPI_ADDRESSSEMPGETDETAILEDLIST, BAPI_EMPLCOMM_GETDETAILEDLIST, BAPI_EMPLOYEE_GETDATA, BAPI_EMPLOYEE_GETLIST, BAPI_PERSDATA_GETLIST, MSS_GET_SY_DATE_TIME, BAPI_PERSDATA_GETDETAIL, SDTX, SMSSDATA1, PERS,PADR,RFC_GET_FUNCTION_INTERFACE, DDIF_FIELDINFO_GET, BAPI_COMPANYCODE_GETDETAIL and 0002
S_TABU_DIS	ACTVT	Activity	03 Display
	DICBERCLS	DICBERCLS	FC01
S_TABU_NAM	ACTVT	Activity	03 Display
	TABLE Name	TABLE	<p>HRP1001, HRP1000, PA0000, PA0001, PA0002</p> <p>Note: T530T, T529T (Add these tables name only if you want to populate Infotype0000 account schema attribute)</p> <p>Note: PA0006, PA0105 (Add these tables if Delta Aggregation Enabled checkbox is selected)</p>

Authorization Objects	Field name	Field description	Field value
P_Orgin	AUTHC	Authorization Level	R
	INFETY	INFOTYPE	0001, 0002, 0003, 0006, 0032, 0105
	PERSA	Personal area	(Depending on the organizational area you have assigned to user while creating)
	PERSG	Employee group	(Depending on the organizational area you have assigned to user while creating)
	SUBTYPE	SUBTY	<ol style="list-style-type: none"> 1. '*' - If you want to get data for all the subtypes 2. If you want to get data for specific info types then add specific subtypes as per data in your environment) <p>For example: For Addresses (Infotype 0006) add subtype - 1,2,3,4,5,6</p> <p>For Communication (Infotype 0105) add subtype - 0001,0010, 0020, 0030</p>
	PERSK	Employee sub group	(Depending on the organizational area you have assigned to user while creating)
	VDSK1	Organization Key	(Depending on the organizational area you have assigned to user while creating)

Provisioning Rule

The administrator permissions mentioned in the following table are applicable only to the provisioning operations specified in the **Example SAP HRMS Modify Rule** (E-mail, Phone number and System user name) rule. This rule is specified in the `examplerules.xml` file located in `WEB-INF/config`/directory.

Authorization Objects	Field name	Field description	Field value
S_RFC	RFC_NAME	Name of RFC object	RFC1, 1065, BAPI_EMPLOYEE_ENQUEU, SYSU, SYSTEM_RESET_RFC_SERVER, SDIFRUNTIME
P_Orgin	AUTHC	Authorization Level	E,S,W
	INFETY	INFOTYPE	0007

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SAP HR/HCM connector uses the following connection attributes:

Table 2—SAP HM/HCM Connector - Configuration parameters

Parameters	Description
SAP JCO Connection Settings	
SAP Host*	Host on which the SAP Server is running
System Number*	2-digit SAP system number. Default: 00
ClientNumber*	3-digit SAP client number. Default: 001
ClientLanguage*	2-letter SAP client language. Default: EN
Username*	SAP service account which has permissions mentioned in “Administrator permissions” on page 314.
Password*	SAP service account password.
Action Type(s)	Enter the comma separated value of Action Type(s) to be aggregated for each SAP HR person. For example: 01, 20, 21 In the above example, 01 stands for Hiring, 20 for Termination and 21 for Re-Hire.
Aggregate Inactive Employees	Select this checkbox to aggregate inactive employees. Inactive employees refers to the employees having the following values of STAT2 as follows: <ul style="list-style-type: none"> • STAT2 = 0 refers to Action Type Termination • STAT2 = 1 refers to Action Type Leave of absence • STAT2 = 1 refers to Action Type Retirement
Inactive Employees Offset	Enter the number of past days to aggregate inactive employees. The Inactive Employees Offset can have the following values: <ul style="list-style-type: none"> • 0: aggregates only the active employees • Blank: aggregates all the inactive employees • Any positive value: indicates the number of days in past since when the terminations must be aggregated Default value is 30.

Configuration parameters

Table 2—SAP HM/HCM Connector - Configuration parameters

Parameters	Description
Future Dated Hires Offset	<p>Indicates the number of days to aggregate the future hires. The Future Dated Hires Offset can have the following values:</p> <ul style="list-style-type: none"> • 0: aggregates no future hires. • Blank: aggregates all future hires until 9999-12-31, • A positive value: aggregates Future Hires within the specified number of days. <p>Default value is 30.</p>
JCO RFC Trace	<p>To enable JCO RFC client trace. If enabled, the JCO traces are written to one or multiple .trc files.</p> <p>Note: .trc files are generated on the application server where IdentityIQ is running.</p>
BuildMap Rule	<p>A rule is used to modify the default object or add custom attributes to each object returned from SAP HR server. (A rule that is called for each object returned from SAP.)</p>
Manager Configuration	
Manager Relationship Model	<p>Select one of the following model to determine the manager of an employee:</p> <ul style="list-style-type: none"> • Organization Chief Manager Model (O-O-S-P): OOSP (Organization(O)-Organization(O)-Position(S)-Person(P)) model deals with the chief of organization model in which employee's organization is detected with the relationship code 012. between position of employee and organization unit. • Supervisory Model (S-S): SS (position(S)-position(S)) model deals with the position-position manager relationship model in which the two positions are connected with the relationship code 002. • Custom: Custom rule to determine the manager of the SAP HR employee. An example of custom rule is located in WEB-INF/config/examplerules.xml file. For more information on Manager Rule, see “Customization Rule” on page 325.
SNC Configuration	
SNC Mode	<p>Represents Secure Network Connection which also internally signifies jco.client.snc_mode in SAP. SNC will be enabled if the mode is selected as ON whose value is 1. If SNC is off, the value will be 0.</p>
SNC Level of Security	<p>Represents the quality of protection level (QOP) which is defined from 1 to 9. In SAP, it relates to jco.client.snc_qop.</p> <p>Default: 1</p>

Table 2—SAP HM/HCM Connector - Configuration parameters

Parameters	Description
SNC Partner Name	Represents SNC partner. For example, provide input as follows in SAP: <code>p:CN=R3, O=XYZ-INC, C=EN</code> If SNC is configured, it relates to <code>jco.client.snc_partnername</code>
SNC Name	Represent SNC name which internally signifies <code>jco.client.snc_myname</code> . It overrides default SNC Partner Name.
SNC Library	Path to library which provides SNC service. It internally signifies <code>jco.client.snc_lib</code> . For example, the value to be passed: <ul style="list-style-type: none"> on Microsoft Windows: <code>C:/sapcrypto/lib/sapcrypto.dll</code> (the location of the cryptographic library) on UNIX: <code>/opt/sailpoint/lib/custom/libsapcrypto.so</code> (the location of the cryptographic library)

Note: Attributes marked with * sign are the mandatory attributes.

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. Account objects are used when building identities Link objects.

Account attributes

Table 3—SAP HR/HCM Connector - Account Attributes

Name	Description
Academic Grade	Academic grade attained by the person
Address	Address of the employee
Address Type	Address type of employee; home, work
Address Type Code	Address type code of employee
Admin Group	Administrative group to which the employee belongs
Aristocratic Title	Aristocratic title that apply to this person
Birth Date	Date of birth of employee
Birth Name	Name given to the employee at time of birth
Birth Place	Name or location of birth place of employee

Schema Attributes

Table 3—SAP HR/HCM Connector - Account Attributes (Continued)

Name	Description
Business Area	Business area
City	City in which the employee is located
Co Area	Corporate area
Comp Code	Company code
Company Name	Name of the company by which they are employed
Contract	Work contract: yes, no
Cost Center	Cost center with which they are associated
Country	Country in which employee is located
Country Code	Country code
Country of Birth	Country in which they were born
Country of Birth Code	Country code for country in which they were born
District	District in which they are located or report
E Group	E-mail groups to which they belong
Email	E-mail address
Employee Number	Employee number
FirstName	First name
Form of Address	Form of address; Miss, Mrs., Sir
FullName	Full, legal name
Fund	Fund name.
Funds Center	Funds center name.
Effective Dates	Stores the Effective Dates for manager change and last name change in the following format: {<lastname schema attr =lastname>#<effective date =yyyy-MM-dd>, <supervisor-id schema attr = supervisor-id>#<effective date= yyyy-MM-dd >} For example: LastName=Jones#effectiveDate=1999-12-04, Supervisor=00000067#effectiveDate=2016-05-23
Gender	Gender
Gender Code	Gender code
Id Number	Identification number
Initials	Initials

Table 3—SAP HR/HCM Connector - Account Attributes (Continued)

Name	Description
Infotype0000	<p>Information about Actions Info Type.</p> <p>The Infotype0000 attribute contains action data in the following format:</p> <p>ActionTypeCode=Code of Action to Use(a numeric value), ActionTypeName=Name of action (a text), ReasonCode=code of Reason to use(a numeric value), ReasonTypeText=Name of reason(a text), ActionStartDate=yyyy-MM-dd, ActionEndDate=yyyy-MM-DD</p> <p>Note: Applicable only when upgrading IdentityIQ from version 7.0 Patch 2 to IdentityIQ version 7.3. With this release of IdentityIQ version 7.3, SailPoint recommends the use of newly added 'Infotype0000JSON' schema attribute.</p>
Job	Title
Job Description	Description of their function
Known As	Nickname or preferred name
Language	Primary language
Language Code	Language code
Language ISO	Primary language ISO code
LastName	Surname
LegPerson	Legal Person
Marital Status Code	Code associated with the marital status of this person
Marital Status Since	Time period since the last change in marital status
MaritalStatus	Marital status
MiddleName	Middle name
Name	Full name
Name Format Indicator	The formats used for name formatting
Nationality	Nationality
Name State of Birth	Name of the state of birth
Name Third Nationality	Name of the third nationality
Nationality Code	Nationality code
Number of Children	Number of children
Org Key	Organizational key
Org Unit	Organizational unit
Organization Description	Organization description
P subArea	Personal sub area.

Schema Attributes

Table 3—SAP HR/HCM Connector - Account Attributes (Continued)

Name	Description
Payarea	The area from which their pay is received
Payroll Admin	The payroll administrator associated with this person
Personal Admin	The personal administrator associated with this person
Personal Area	The personal area to which employee report
Personal Number	Their personal number
Position	Title
Position Description	Description of job function
Reason Code	Name of the reason code.
Religion	Religion
Religion Code	Religion code
Second Academic Grade	Secondary academic grade associated with this person
Second Address Line	Second line of address
Second Name Prefix	Secondary name prefix
Second Nationality	Secondary nationality
Second Nationality Code	Secondary nationality code
SecondName	Second name
State	State in which they are located
State Abbreviation	State abbreviation of the state in which they are located
State of Birth	State of birth
STAT2_Current	Current employment status
STAT2_Next	Future employment status
STAT2_Next_Start_Date	Effective start date of future employment status
Sub E Group	Sub E group name.
Supervisor	Supervisor area
Surname Prefix	Last name prefix
System user name (SY-UNAME)	User ID
Telephone	Telephone number and dialing code
Third Nationality	Third nationality
Time Admin	Time Administrator name
Title	Function
Zip Code	Zip code for this person

Table 3—SAP HR/HCM Connector - Account Attributes (Continued)

Name	Description
Infotype0000JSON	<p>Information about Actions InfoType.</p> <p>The Infotype0000 attribute contains action data in JSON format:</p> <pre>Infotype0000JSON : { "Actions": [{ "ActionTypeCode" : "<Value>","ActionTypeName": "<Value>","ReasonCode" : "<Value>","ReasonTypeText": "<Value>","ActionStartDate" : "<Value>","ActionEndDate" : "<Value>" }] }</pre> <p>The attributes in the above format represent the following:</p> <ul style="list-style-type: none"> • ActionTypeCode = Code of action to use (a numeric value) • ActionTypeName = Name of action (a text), • ReasonCode = Code of reason to use (a numeric value) • ReasonTypeText = Name of reason (a text) • ActionStartDate = YYYY-MM-DD • ActionEndDate = YYYY-MM-DD <p>Note:</p> <ul style="list-style-type: none"> - In case there are no values in infotype0000JSON, then the attribute is displayed as Infotype0000JSON: { "Actions": [] } - To fetch all action data in JSON format for an user after upgrading IdentityIQ to version 7.3, add the Infotype0000JSON attribute to the application with: <p>Property: Multi-Valued Data Type: string</p>

Support for Future Data

SAP HR/HCM connector supports aggregation for the future data for all the accounts.

Schema changes

Following new parameters are added in the IdentityIQ version 7.3 for the future data:

Attribute	Description	Value
STAT2_Current	Current employment status	Inactive: For terminated employee Active: For active employee
STAT2_Next	Future employment status	Inactive: For terminated employee Active: For active employee
STAT2_Next_Start_Date	Effective start date of future employment status	Date with format yyyy-MM-dd

Schema Attributes

Attribute	Description	Value
Effective Dates	Stores the Effective Dates for manager change and last name change in the following format: {<lastname schema attr =lastname>#<effective date =yyyy-MM-dd>, <supervisor-id schema attr = supervisor-id>#<effective date= supervisor-id#<effective date= yyyy-MM-dd >}	{<lastname schema attr =lastname>#<effective date =yyyy-MM-dd>, <supervisor-id schema attr = supervisor-id>#<effective date= yyyy-MM-dd >}

Note: By default the above attributes will be a part of the schema attributes for the new application. If required for the upgraded application the above attributes must be added manually. If attribute is not present, then no valid event exists for that attribute.

Delta Aggregation

In IdentityIQ version 7.3, delta aggregation approach is updated. With this approach no additional configuration is required for SAP HR/HCM system.

To enable delta aggregation, user must select the **Enable Delta Aggregation** flag in Account aggregation task.

The supported features for delta aggregation are:

- Any future hires that have been added since the date of last full aggregation.

Note: These are the Future Hires within the 'offset'.

- All changes in the **Effective Dates** attribute of the employee are already aggregated in IdentityIQ.
- Any changes for the employment status (STAT2) of the employee already aggregated in IdentityIQ.
- Any changes in following employee attributes:
 - Organization data
 - Communication details
 - Personal data
 - Address information data
 - Supervisor Changes

Note: To improve the performance, it is recommended to select the 'Enable Delta Aggregation' flag while running the account aggregation task.

Upgrade Consideration

Users upgrading to IdentityIQ version 7.3 must update the service account with permissions specified in the "Administrator permissions" section.

Partitioning

To perform partitioning aggregation, you must enable the **Partition Enabled** option in account aggregation task definition user interface page. The SAP HR/HCM connector itself will determine the number of optimal partitions to be made based on the total number of account and IdentityIQ system wide settings.

Note: If you have enabled ‘Partition Enabled’ option, there is a second configuration option presented in the task definition UI -- Objects per partition -- which supports setting the number of accounts to include in each partition. This parameter will not be considered in case of SAP HR/HCM.

Customization Rule

- **Modify Rule:** The rule name is defined as Example “SAP HRMS Modify Rule”. This is a sample rule to assign and update the E-mail, Phone number and System user name.
- **Manager Rule:** The rule name is defined as Example “SAP HR Custom Manager Model Rule”. This example rule can be used as reference to populate the supervisor of the employee.

Additional information

This section describes the additional information related to the SAP HR/HCM Connector.

Upgraded Application

Note: For any operation other than Test Connection, the attributes of old application will be updated to new one (upgraded to 7.3). Test Connection would still display the same attributes present in old application.

For the application upgrading to IdentityIQ version 7.3, following changes would be performed:

- Include Terminated Employees Flag and Termination offset

Include Terminated Employees	Termination Offset	Aggregate Inactive Employees	Inactive Employees Offset
Checked		Checked	Blank
Checked	-1	Checked	Blank
Checked	0	Checked	Blank
Not checked	-1	Not checked	Not Applicable

- SearchAdditionalField and SearchAdditionalValue

searchAdditionalField	searchAdditionalStrings	Future Dated Hires Offset
DATE or SEARCH_DATE	+n (any positive integer)	n
DATE or SEARCH_DATE	Any date (for example, 31-12-999)	120
Any value other than DATE or SEARCH_DATE	Not considered	

Note: For the upgraded application (IdentityIQ 7.3) partitioning Aggregation will not respect the partitioning string which was present in old application, for more information, see “Partitioning” on page 324.

Support for Custom BAPI Invocation

SAP HR/HCM connector provides support to invoke Custom BAPI's which can be used for the following purposes:

- Filter the Employees
- To aggregate the additional attributes

Pre-requisite

- Based on your requirement write a custom BAPI code which follows the Input and Output parameters standard specified by SailPoint Technologies.

For more information on the input and output parameters, see “Input and Output parameters for BAPI” on page 327.

- Add the required entry key in the application debug page as follows:

- a. For applying filter, use entry key as `<entry key="aggregateFilterBAPI" value="name of the BAPI"/>`

The Aggregate Filter BAPI has the following input attributes, whose values would be determined from the values specified in the SAP HR/HCM connector:

- Status
- Termination offset
- Future offset
- Mode of aggregation

In addition to the attributes mentioned above, additional input parameters can be specified using the following entry key:

```
<entry key="customParameterForAggregationFilterBAPI">
  <value>
    <Map>
      <entry key="" value ="<value of attribute>"/>
    </Map>
  </value>
</entry>
```

For example,

```
<entry key="customParameterForAggregationFilterBAPI">
  <value>
    <Map>
      <entry key="Dept" value ="QA"/>
    </Map>
  </value>
</entry>
```

- b. For the BAPI which will be used for the aggregating additional attributes, use entry key as `<entry key="additionalAttrBAPI" value="name of the BAPI"/>`

All additional attributes must be added manually in the SAP HR/HCM application.

To add additional input parameters, use **AdditionalAttrBAPI** entry key as follows (having input parameter as PERNR):

```
<entry key="customParameterForAdditionalAttrBAPI">
  <value>
```

```

<Map>
  <entry key="CustAttrName" value ="<value>"/>
</Map>
</value>
</entry>
```

For additional attribute from schema, use the **AdditionalAttrBAPI** entry key value as follows:

```

<entry key="customParameterForAdditionalAttrBAPI">
  <value>
    <Map>
      <entry key="schema.AttrName" value ="<value>"/>
    </Map>
  </value>
</entry>
```

Note the following:

- The connector would process all the returned additional attributes as single-valued only.
- If any attribute returned from this BAPI matches with the out of the box schema attribute then the value of attribute returned from BAPI would be overwritten.

Input and Output parameters for BAPI

This section describes the parameters required for the BAPIs used for aggregation filtering and fetching additional attributes.

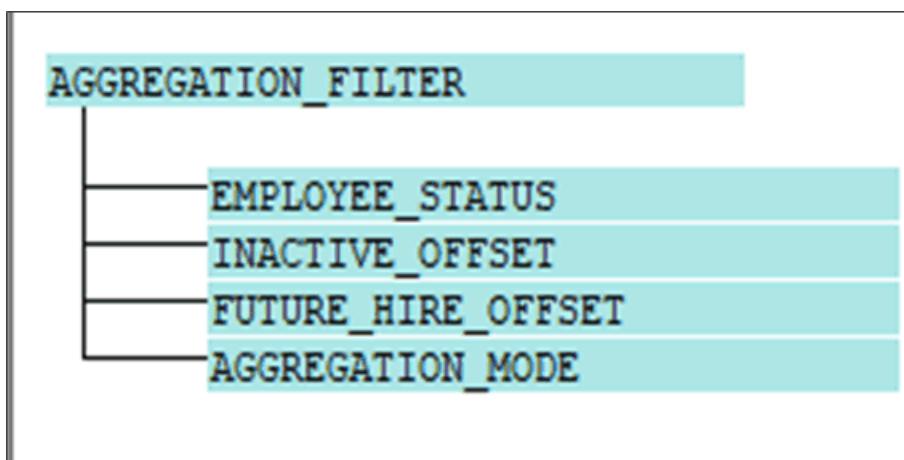
BAPI for Aggregation filter

This section describes the input and output of aggregation filter BAPI.

Input of aggregation filter BAPI

Input of this BAPI must be a structure.

1. Parameter Name of the structure: **AGGREGATION_FILTER**



Additional information

2. This structure must have the following components:

- **EMPLOYEE_STATUS**: Aggregation based on the various status of the employee.

Possible values: ACTIVE / FUTURE / INACTIVE / RETIREE / WITHDRAWN

Note: If the 'Aggregate Inactive Employees' checkbox is not selected then the EMPLOYEE_STATUS value is set as ACTIVE and FUTURE.

Note: If the 'Aggregate Inactive Employees' checkbox is selected then the EMPLOYEE_STATUS value is set as ACTIVE, FUTURE, INACTIVE, RETIREE, and WITHDRAWN.

- **INACTIVE_OFFSET**: Indicates the number of past days to aggregate inactive (INACTIVE, RETIREE and WITHDRAWN) employees. Inactive offset value is referred to from the attribute **terminationOffset** and is effective only if **Aggregate Inactive Employees** checkbox is selected.

The following table gives example of this offset:

Value from configuration	Description	Value passed on to BAPI
0	No Inactive employees must be aggregated	0
Positive Value	The number of days past from when the inactive employees must be aggregated	For example, 30
No input provided (BLANK)	Aggregation of all the employees based on status (INACTIVE/RETIREE/WITHDRAWN)	-1

- **FUTURE_HIRE_OFFSET**: Indicates the number of days to aggregate the future hires. The future offset values are referred to from the attribute **Future Dated Hires Offset**. These values are read from the configuration parameter **futureOffset**.

The following table gives example of this offset:

Value from configuration	Description	Value passed on to BAPI
0	Aggregates no future hires	0
Positive Value	Aggregates future hires within the specified number of days.	For example, 30
No input provided (BLANK)	Aggregates all future hires until 9999-12-31	-1

Note: With every EMPLOYEE_STATUS, offset values would be passed as specified in configuration parameters.

- **AGGREGATION_MODE**: The values for AGGREGATION_MODE can be FULL or DELTA.

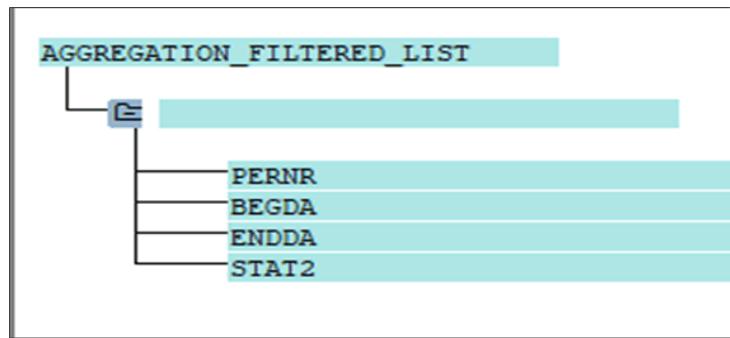
Note: By default, AGGREGATION_MODE is FULL. If the 'Enable Delta Aggregation' checkbox is selected on the Source UI, the mode would be DELTA.

Output of aggregation filter BAPI

The output of this value must be in form of a JCOTable.

JCOTable is an interface that describes internal tables used in the parameter lists of function modules. It is a data container, which can have multiple rows of the same fields.

The table name is AGGREGATION_FILTERED_LIST. The Output table must be a structure as described below:



The table must have the following components:

- PERNR
- BEGDA
- ENDDA
- STAT2

For example,

PERNR	BEGDA	ENDDA	STAT2
00000001	20170505	99991231	3
00000002	20170506	99991231	2
00000003	20170508	20181231	1

Note: PERNR - Personnel Number, BEGDA - Beginning Date, ENDDA - End Date, STAT2 - Employment Status

Note: The SAP-HR connector expects only one PERNR for one employee. If multiple person IDs are returned, the connector creates two separate identities for the person IDs as the default native identity is Employee Number.

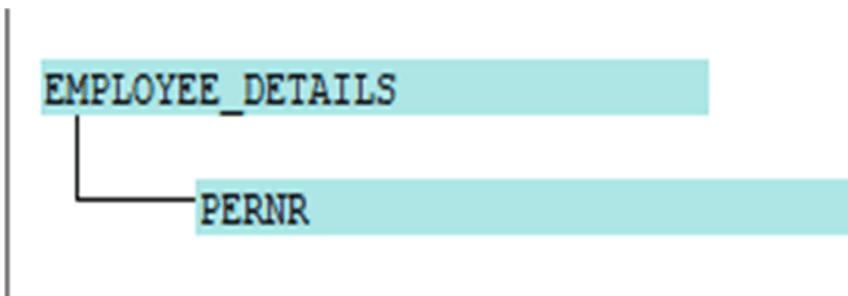
BAPI for fetching additional attributes

This section describes the input and output of BAPI to fetch additional attributes for a given employee number.

Input of BAPI for fetching additional attributes

Input of this BAPI must be a structure.

Parameter name of the structure: EMPLOYEE_DETAILS



This structure must have the PERNR field.

Output of BAPI for fetching additional attributes

Output of this BAPI should be a JCOTable

Table Name: ADDITIONAL_ATTRIBUTE_LIST

The table must have all the additional attributes for given PERNR.

For example,

MANAGER	REGION	LAST_DAY_WORK	GLOBAL_ID
John	Pune	2018-5-1	11111

In the above example, MANAGER, REGION, LAST_DAY_WORK, and GLOBAL_ID are additional attributes.

Error Handling

BAPIs used for aggregation filtering and fetching additional attributes must return an additional JCOTable.

- Table Name: RETURN
- Table must have the following components:
 - TYPE: Possible values are as follows:
 - S (success)
 - E (error)
 - W (warning)
 - MESSAGE

Troubleshooting

1 - Unable to load class files

The following error appears, when unable to load class files:

```
java.lang.UnsatisfiedLinkError: no sapjco3 in java.library.path*
at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1682)
```

Resolution:

Install the missing DLLs by performing the following steps:

1. Navigate to the following link:
<http://www.microsoft.com/technet/security/bulletin/MS09-035.mspx>
2. Under the selected security update, scroll to the **Affected Software** section on the right-hand side and click on the **Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package (KB973544)** link.
3. Select the required language and download the platform-specific files.
4. Execute the downloaded file and follow the installation instructions.

For more information, see <https://service.sap.com/sap/support/notes/684106>

Note: The URL requires SAP support (SAP username and password).

2 - When the supported platform version is Java 1.7 or 1.8 an error message appears

When the supported platform version is Java version 1.7 or 1.8, the following error message appears:

```
getting the version of the native layer: java.lang.UnsatisfiedLinkError: no sapjco3
in java.library.path
```

Resolution: Download the latest SAPJCO.jar and SAPJCO.dll files from SAP Marketplace and use that SAPJCO.jar file with the latest downloaded SAPJCO.dll file.

3 - Provisioning rules not working for upgraded application

Provisioning rules not working for upgraded application.

Resolution: For applications upgraded from IdentityIQ version 6.4 patch 3 and below to IdentityIQ version 7.0 Patch 3, add the following attributes in feature string:

```
PROVISIONING and SYNC_PROVISIONING
```

4 - For SAP HR/HCM upgraded application, action data for an employee is not fetched in IdentityIQ

Action data for an user is not getting fetched in IdentityIQ for SAP HR upgraded application.

Resolution: To fetch all action data for an user after upgrading IdentityIQ to version 7.0 Patch 2 or above, add the **Infotype0000** attribute to the application with:

- Property: **Multi-Valued**
- Data Type: **string**.

5 - While performing Test Connection through SAP router strings, an error message is displayed

While performing Test Connection through SAP router strings, the following error message is displayed:

```
ERROR SPSAPROUTER: route permission denied (xxx.xx.X.X to xxx.xxxx.Y.Y, 3300.)
```

Resolution: Depending on the error message, (for the above error) add the following entry in the **saprouttab** of your SAProuter:

```
P xxx.xx.X.X xxx.xxxx.Y.Y 3300
```

Troubleshooting

Activate the new saprouttab with the following command or restart the SAProuter:

```
saprouter -n
```

Chapter 35: SailPoint SAP HANA Connector

The following topics are discussed in this chapter:

Overview.....	333
Supported features	333
Supported Managed Systems	334
Pre-requisites	334
Administrator permissions	334
Configuration parameters.....	335
Schema Attributes	337
Provisioning Policy attributes	338
Additional information	339
Enabling SSL connection to SAP HANA database through IdentityIQ	339
Delete provision request	339
Troubleshooting.....	340

Overview

SailPoint SAP HANA Connector manages the users, roles and privileges for the SAP HANA database system.

Supported features

SailPoint SAP HANA Connector supports the following features:

- Account Management
 - Manages SAP HANA database users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
 - Manages System Privileges and Application privileges of user
- Account - Group Management
 - Supports multiple group functionality
 - Manage SAP HANA Catalog Roles as Account group CATALOG_ROLE
 - Aggregation
 - Manage SAP HANA Repository Roles as Account group REPOSITORY_ROLE
 - Aggregation

Overview

Note: SAP HANA connector supports single and multi tenant SAP HANA system. User must create separate applications based on the type of Database to be configured. It can be a System database or a Tenant database. For more information, see “Configuration parameters” on page 335.

Supported Managed Systems

SailPoint SAP HANA Connector supports the following SAP System:

- SAP HANA SPS12
- SAP HANA SPS11

Pre-requisites

SAP HANA JDBC driver is required for proper functioning of SailPoint SAP HANA connector. The `ngdbc.jar` file must be copied in the `..\identityiq\WEB-INF\lib` directory of IdentityIQ installation.

Administrator permissions

Following are the minimum required permissions for SAP HANA Administrative account for the listed operation:

Operation	Minimum required permissions
Test Connection	CATALOG role as public
Account Aggregation	System Privileges <ul style="list-style-type: none">• CATALOG Read• ROLE ADMIN• USER ADMIN
Group Aggregation	<ul style="list-style-type: none">• CATALOG Read• ROLE ADMIN
Removing the role assigned to the user	Revoking the role can be achieved by the same user who has granted it.

Operation	Minimum required permissions		
	CATALOG role and Application privilege	Repository roles	System Privileges
Creating a user by assigning the respective privileges and roles	CATALOG role and Application privilege System privileges <ul style="list-style-type: none"> • CATALOG Read • ROLE ADMIN 	System Privileges <ul style="list-style-type: none"> • CATALOG Read • ROLE ADMIN • USER ADMIN Object Privileges <p>Execute Privilege on the following objects:</p> <ul style="list-style-type: none"> • _SYS_REPO.GRANT_ACTIVATED_ROLE • _SYS_REPO.REVOKE_ACTIVATED_ROLE • _SYS_REPO.GRANT_APPLICATION_PRIVILEGE • _SYS_REPO.REVOKE_APPLICATION_PRIVILEGE 	Service account must have the system privilege assigned to it with Grantable to other user and groups flag as checked.

Note: For Enable/Disable/Lock/Unlock Account/Delete Account/change password only System privileges USER ADMIN is required.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SAP HANA Connector uses the following connection attributes:

Attributes	Description
User Name*	Enter name of the user.
Password*	Enter the password to connect to SAP HANA Database.

Configuration parameters

Attributes	Description
Database URL*	<p>Enter the URL of the database. The specified default URL is provided for multi-tenant HANA Database system. For a single tenant HANA Database system, attribute database name is not applicable.</p> <p>Default: <code>jdbc:sap://<HOST>:<PORT>/?databaseName=<dbName></code></p> <p>where:</p> <ul style="list-style-type: none"> • <HOST>: server name where SAP HANA database is installed • <PORT>: port on which database is configured • <dbName>: name of the database to connect to <p>Note: Enter URL in the following format to connect to the SAP HANA SSL enabled database:</p> <pre>jdbc:sap://<HOST>:<PORT>/?databaseName=<dbName>&encrypt=true</pre> <p>For more information on enabling SSL Connection to SAP HANA database through Identity IQ, see “Enabling SSL connection to SAP HANA database through IdentityIQ” on page 339.</p>
Driver Class*	<p>The driver class used for connecting to SAP HANA Database.</p> <p>Default: <code>com.sap.db.jdbc.Driver</code></p>
Page Size	<p>The number of objects to fetch in a single page when iterating over large data sets.</p> <p>Default: 1000</p>

Additional Configuration parameters

The following table lists the additional configuration parameters to be added in the application debug page:

Attributes	Description
endDateonDelete	<p>To drop a user when delete operation is performed, set the value of the endDateonDelete parameter to false as follows:</p> <pre><entry key="endDateonDelete"> <value> <Boolean>false</Boolean> </value> </entry></pre> <p>Default: True (when Delete operation is performed, the account would be only marked end dated)</p>

Attributes	Description
useRestrictOnDrop	<p>(Applicable only when endDateonDelete is set to false) To drop all the associated objects with this user, set the value of useRestrictOnDrop parameter to false as follows:</p> <pre><entry key="useRestrictOnDrop"> <value> <Boolean>false</Boolean> </value> </entry></pre> <p>Default: True</p> <p>For more information, see “Delete provision request” on page 339.</p>

Schema Attributes

The application schema is used to configure the objects returned from a connector. When application is configured, the schema is supplied to the methods on the connector interface and supports multiple types of objects, account and any number of group application object types. Account objects are used when building identities Link objects. Additional schema definitions can be used when building AccountGroup objects which are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

Attributes	Description
USER_ID	ID of the user
USER_NAME	Name of the user
USER_MODE	Mode of the user: <ul style="list-style-type: none"> • LOCAL • GLOBAL • EXTERNAL
CREATOR	Creator of the user
VALID_FROM	Specify a date time from which the user account is valid
VALID_UNTIL	Specify a date time until which the user account is valid
IS_RESTRICTED	Specifies if the user is a restricted user
IS_CLIENT_CONNECT_ENABLED	Specifies if the user is able to connect to client
HAS_REMOTE_USERS	Specifies if there is a database user in another tenant database as the remote identity of the database user.
SESSION_CLIENT	Specifies the client whose data can be accessed by user
EMAIL_ADDRESS	Email address of the user.

Provisioning Policy attributes

Attributes	Description
TIME_ZONE	Time zone of the user.
AUTHENTICATION_TYPE	Different Authentication methods supported by user.
SYSTEM_PRIVILEGES	System Privileges assigned to the user.
APPLICATION_PRIVILEGES	Application Privileges assigned to the user.
CATALOG_ROLES	Catalog Roles assigned to the user.
REPOSITORY_ROLES	Repository Roles assigned to the user.
FORCE_PWD_CHG_ON_NEXT_LOGON	Specifies whether user must change the password on next logon in the SAP HANA database.

Group attributes

The following table lists the group attributes for REPOSITORY_ROLE and CATALOG_ROLE:

Attributes	Description
ROLE_NAME	Role name
ROLE_ID	Role ID
ROLE_MODE	Mode of the role: LOCAL
GLOBAL_IDENTITY	Identity specified for role with ROLE_MODE GLOBAL
CREATOR	Name of the user who created the role
GRANTED_ROLES	The roles which are assigned to the current role

Provisioning Policy attributes

This section lists the different policy attributes of SAP HANA Connector.

Note: The attributes marked with * sign are the required attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
User Name	User name of SAP HANA database.
Password	Password of the user.

Attributes	Description
Force Password Change On Next Logon	<p>Set the value to Yes, if user must be requested to change his password on next login.</p> <p>Note: If 'Force Password Change On Next Logon' flag is not specified in the provisioning plan, then password mode (permanent or temporarily) would be depended on SAP HANA security policy.</p>
Restricted	Select true if user must be created in Restricted mode.

Additional information

This section describes the additional information related to the SAP HANA Connector.

Enabling SSL connection to SAP HANA database through IdentityIQ

Perform the following steps to enable SSL Connection to SAP HANA database through IdentityIQ:

1. Import the SSL certificates from the SAP HANA Server to IdentityIQ system.
2. Add the certificates to the IdentityIQ keystore present in the <JAVA_HOME>/jre/lib/security/cacerts directory.
3. Restart the WEB Server.
4. Login to IdentityIQ.
5. For database, use the following URL:
`jdbc:sap://<HOST>:<PORT>/?databaseName=<dbName>&encrypt=true`

Delete provision request

- (*Applicable only when endDateOnDelete is set to false*) Set the value of **useRestrictOnDrop** parameter to **false** to drop all the associated objects with this user. In this case the CASCADE option would be used when dropping the user along with the following actions:

- the schema with the user's name and the schema's belonging to the user, together with all the objects stored in them (even if they are created by other users) are deleted
- objects owned by the user, even if they are part of another schema, are deleted
- objects that are dependent on deleted objects are deleted.
- public synonyms owned by the deleted user are deleted
- privileges on deleted objects are revoked. Privileges granted by the deleted user are revoked. Revoke privileges may cause further revokes if they had been granted further
- users created by the deleted user and roles created by him are not deleted
- audit policies created by the deleted user are not deleted.

Set the value of **useRestrictOnDrop** parameter to **true** for using the RESTRICT option when dropping the account. In this case, the account would not be deleted if it is the owner of any object other than the schema

Troubleshooting

with its name and other schema's created by it or in case there is an object stored in one of its schema's which was not created by it.

Troubleshooting

1 - An error message was displayed while removing the role assigned to the user

Following error message was displayed while removing the role assigned to the user:

Service account does not have permission to revoke <Role>

Resolution: Revoking the role can be achieved by the same user who has granted it.

2 - Test connection fails with an error message

Test connection fails with the following error message:

Please enter valid Driver Class

Resolution: Copy the latest ngdbc.jar file in \identityiq\WEB-INF\lib directory.

Chapter 36: SailPoint ServiceNow Connector

The following topics are discussed in this chapter:

Overview.....	341
Supported features	341
Supported Managed System.....	342
Pre-requisites	342
User permissions	343
Configuration parameters.....	343
Schema attributes	344
Account attributes	345
Group attributes.....	346
Role attributes	347
Provisioning Policy attributes	348
Additional information	349
Upgrade.....	349
Session management.....	349
Custom attributes.....	350
Troubleshooting.....	351

Overview

The SailPoint ServiceNow Connector manages ServiceNow accounts, groups, and roles. It supports read and write for ServiceNow accounts and groups.

Supported features

SailPoint ServiceNow Connector supports the following features:

- Account Management
 - Manages ServiceNow Users as Accounts
 - Aggregation, Partitioning Aggregation, Refresh Accounts
 - Delta Aggregation
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements (ServiceNow Groups and ServiceNow Roles)
- Account - Group Management
 - Manages ServiceNow Groups and Roles as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete (applicable for groups only)

Overview

- ServiceNow Connector supports configuration of multiple applications of different ServiceNow versions on same IdentityIQ.

Supported Managed System

SailPoint ServiceNow Connector supports the following ServiceNow versions:

- Kingston
- Jakarta
- Istanbul

ServiceNow Rest API supports Basic and OAuth2 methods of authentication.

Each client must perform the OAuth setup to participate in OAuth authorization. To configure OAuth in ServiceNow Connector, a Client ID, Client Secret and Refresh Token are required. The Client ID, Client Secret and Refresh Token are specific to the ServiceNow instance and configured while enabling the OAuth in ServiceNow instance. Contact your ServiceNow Administrator to obtain the Client ID, Client Secret and Refresh Token.

Refer to the following link for token generation:

http://wiki.servicenow.com/index.php?title=Generating_OAuth_Tokens

Pre-requisites

- ServiceNow must be up and running.
- Apply the ServiceNow Connector update set as follows:
 1. Copy the relevant update set from `identityiq-releaseVersion.zip\integration\servicenow\iiqIntegration-Servicenow.zip\ConnectorUpdateSet`
In the above directory, `releaseVersion` is the version of the current IdentityIQ release.
 2. Based on the required version of ServiceNow, copy the relevant update set from the following respective files:

ServiceNow version	Update Sets
Istanbul or later	<code>SailPointServiceNowConnector.v1.3.xml</code>

3. Import relevant update set in ServiceNow instance. For more information and guidelines on usage of the update set, refer to the following wiki link:

http://wiki.servicenow.com/index.php?title=Saving_Customizations_in_a_Single_XML_File#gsc.tab=0
For Service Now Istanbul or later instance

- a. Set system (target) table application access after import, preview and commit of update set.

For the connector to work smoothly, ensure that all the access (read, create, update, delete and allow access to the following tables via web services) has been provided.

- User [sys_user]
- Group [sys_user_group]
- Group Member [sys_user_grmember]
- User Role [sys_user_has_role]
- Group Role [sys_group_has_role]

Perform the following to provide application access:

- Ensure that Global scope is selected in ServiceNow.
 - Navigate to **System Definition => Tables**.
 - Search for the table using label or name.
 - Click on table and scroll down to **Application Access**.
 - Select Can read, Can create, Can update, Can delete and Allow Access to this table via web services.
 - Update or Save the table.
- b. To support unlock operation in ServiceNow Istanbul or later, create the following ACL in global scope and assign it to the `x_sapo_iiq_connect.admin` Role:

ACL	Type	Operation	Name	Attribute
<code>sys_user.locked_out</code>	record	read	User [sys_user]	Locked out

For more information on procedure for creating the ACL, see the following link:

http://wiki.servicenow.com/index.php?title=Using_Access_Control_Rules#Creating_ACL_Rules

- (*For Delta Aggregation*) To support delta aggregation in ServiceNow Istanbul or later, create the following ACL in global scope and assign it to the `x_sapo_iiq_connect.admin` Role:

ACL	Type	Operation	Name	Attribute
<code>sys_audit_delete</code>	record	read	Audit Deleted Record[sys_audit_delete]	None

In ServiceNow delta aggregation, deleted user's connection is read from `sys_audit_delete` table.

If this ACL is not created then deleted connections from the user would not be detected in delta aggregation.

Note: **Delta Aggregation does not detect the deleted accounts. SailPoint recommends performing full aggregation to detect the deleted accounts on ServiceNow.**

- To configure any custom field the import set and transform map must be updated with the custom field.
For more information, see “Custom attributes” on page 350.

User permissions

Assign the `x_sapo_iiq_connect.admin` role to the user when using ServiceNow instance.

Configuration parameters

This section contains the information that is used to connect and interact with the application. Each application type requires different information to create and maintain a connection.

Schema attributes

The ServiceNow Connector uses the connection attributes listed in the following table:

Parameters	Description
Authentication Type	<ul style="list-style-type: none">Basic: In case of Basic, Username/Password will be used for authentication.OAuth2: Select this option when ServiceNow is configured to support OAuth2 authentication. <p>Note: Each time the Authentication Type for ServiceNow Connector is changed, ensure that you perform Test Connection operation.</p>
UserName	Name of the user having the privileges mentioned in the "User permissions" section above.
Password	Password of the user having minimum privileges.
Client ID	The Client ID for OAuth2 authentication.
Client Secret	The Client Secret for OAuth2 authentication.
Refresh Token	The Refresh Token for OAuth2 authentication.
Page Size	The Page size specifies the maximum size of each data set when querying large number of objects. Its default value is set to 1000 and maximum value be 10000.
Filter Condition for Account	(Optional) To filter accounts during aggregation. For example, active=true^locked_out=false
Filter Condition for Group	(Optional) To filter groups during aggregation. For example, active=true
Filter Condition for Role	(Optional) To filter roles during aggregation. For example, sys_scope=Global

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. IdentityIQ currently supports the following types of objects:

- Account:** Account objects are used when building identities Link objects.
- Group:** The group schema is used when building Account-Group objects that are used to hold entitlements shared across identities.
- Role:** The role schema is used when building roles as Account-Group objects.

Note:

- For account aggregation, ServiceNow Role aggregation is supported only for roles having the following property:
inherited=false
- For Account-Group aggregation, ServiceNow Role aggregation is supported only for direct roles connected to the group.

Account attributes

The following table lists the account attributes ([Table 1—Account attributes](#)):

Table 1—Account attributes

Attributes	Description
first_name	First name of the user.
last_name	Last name of the user.
email	Email ID of the user.
user_name	Name of the user.
department	The user's department name.
title	Title (designation) of the user.
sys_id	Unique ID generated by system for user.
phone	Phone number of user.
calendar_integration	Determines whether change requests assigned to that user are sent to their Outlook calendar.
sys_class_name	Class name of the user.
company	The user's company.
cost_center	Cost center of the user.
sys_created_on	Date this user is created in ServiceNow.
sys_created_by	Administrator who created the user in ServiceNow.
groups	List of groups the user is part of.
roles	List of roles the user is part of.
active	Determines whether the user account has been staged for use.
building	The building of the user.
city	The city of the user.
country	The country of the user.
location	The location of the user.
manager	The manager of the user.
middle_name	Middle name of the user.
name	Name of the user.
password_needs_reset	Determines should the user be prompted to change password at next login.
default_perspective	Default perspective for the user.
sys_domain	Domain of the user.
employee_number	Employee number of the user.
failed_attempts	Number of login failed attempts.

Schema attributes

Table 1—Account attributes (Continued)

Attributes	Description
gender	Gender of the user.
home_phone	Home phone number of the user.
ldap_server	LDAP server the user has an account. Identifies which LDAP server authenticates the user when there are multiple LDAP servers.
preferred_language	Language spoken by the user.
last_login	Last login date of the user.
last_login_time	Time of the last login time for the user.
locked_out	Determines if user account is locked.
mobile_phone	Mobile number of the user.
notification	Determines if the user should be notified for any changes made on his account.
schedule	Schedule of the user.
state	The state for the user.
source	Identifies whether LDAP is used to validate a user. If the Source field starts with Idap , then the user is validated via LDAP. If the Source field does not start with Idap , then the password on the user record is used to validate the user upon login.
street	The street for the user.
time_format	Time format selected for user to display time fields.
time_zone	The timezone for the user.
sys_updated_on	Last updated time for the user.
sys_updated_by	The last update for the user occurred from.
sys_mod_count	Number of updates for the user.
vip	Determines if the user is treated as VIP.
zip	Zip for the user.

Group attributes

The following table lists the group attributes ([Table 2—Group attributes](#)):

Table 2—Group attributes

Attributes	Description
active	Determines whether the user account has been staged for use.
cost_center	Cost center of the user group.
sys_created_on	Date the user group is created in ServiceNow.
sys_created_by	Administrator who created the user in ServiceNow.

Table 2—Group attributes

Attributes	Description
default_assignee	Defaults assignee for the user group.
description	Description of the user group
exclude_manager	Determines if the manager should be excluded for the use group.
name	Name of the user group.
parent	Parent group of this user group.
roles	Roles the user group is having.
source	Source of the user group.
sys_id	Unique ID generated by system for user group.
type	Type of the user group
sys_updated_on	Last updated time for the user group.
sys_updated_by	The last update for the user group occurred from.
sys_mod_count	Number of updates for the user group.

Role attributes

The following table lists the role attributes ([Table 3—Role attributes](#)):

Table 3—Role attributes

Attributes	Description
sys_name	System name of the Role.
sys_updated_on	Last updated.
sys_id	Unique ID generated by system for role.
grantable	Can be granted independently.
sys_created_on	Created date and time.
suffix	Application scope.
sys_created_by	Created by.
can_delegate	Can be delegated.
sys_policy	Determines how application files are protected when downloaded or installed.
sys_updated_by	Updated by.
sys_tags	Tags
sys_package	Application name.
description	Description of the role.
name	Name of the role.
sys_class_name	Class name of the role.

Provisioning Policy attributes

Table 3—Role attributes

Attributes	Description
sys_update_name	System updated name.
elevated_privilege	This role is an elevated privilege.
sys_mod_count	Number of updates for the role.
sys_customer_update	Added or modified by customer.
sys_scope	Scope name.
includes_roles	Includes roles.
contains_roles	Contained roles.

Provisioning Policy attributes

This following table lists the provisioning policy attributes for create ([Table 4—Provisioning Policy attributes](#)):

Table 4—Provisioning Policy attributes

Attributes	Description
UserID	User ID for the user.
FirstName	First name of user.
LastName	Last name of the user.
Department	The user's department name.
Title	Title (designation) of the user
Password	Password for the user.
Password need reset	Determines if the user must be prompted to change the password at next login.
Locked Out	Determines if user account is locked.
Active	Determines whether the user account has been staged for use.
Notifications	Determines if the user should be notified for any changes made on his account.
Calender integration	Determines whether change requests assigned to that user are sent to their Outlook calendar.
Time Zone	The time zone for the user.
Email	Email of the user.
Mobile Phone	Mobile number of the user.
Business Phone	Official phone of the user.

Additional information

This section describes the additional information related to the ServiceNow Connector.

Note: To enable logging, specify the logging

```
log4j.logger.openconnector.connector.servicenow.ServiceNowConnector in the
log4j.properties file. For example,
log4j.logger.openconnector.connector.servicenow.ServiceNowConnector=debu
g.
```

- ServiceNow Connector uses the following Transform Maps for write operations:

- SailpointSysUserHasRole_D
- SailpointSysGroupHasRole_D
- SailPointSysUserHasRole
- SailPointSysGroupHasRole
- SailpointSysUserGrmmember_D
- SailpointSysUserGroup_D
- SailPointSysUserGrmmember
- SailpointSysUser_D
- SailPointSysUsers
- SailPointSysUserGroup

Note: “_D” suffix entities used for delete operation only.

- The user can configure the connector to use any of the attributes of ServiceNow User/Group/Role which are supported by ServiceNow Rest APIs.

Upgrade

For upgrading ServiceNow Connector from version 7.0 to version 7.3, ServiceNow Connector update set must be applied on ServiceNow instance.

Note: If the following error message is displayed while previewing the update set, then accept the remote update (that is, overwrite with the change in the update set) followed by committing it:

```
Preview problems for IdentityIQ ServiceNow Connector: 463 Errors | 0
Warnings. To commit this update set you must address all problems
```

For more information, see “Pre-requisites” on page 342.

Session management

A REST session is a Glide session established with a ServiceNow instance by any external REST client like SailPoint ServiceNow Connector. It was observed that for every request the connector would open one session which resulted in opening number of sessions on ServiceNow. With this release of SailPoint ServiceNow Connector, the

Additional information

connector would maintain a pool of sessions (using the following parameters) which would be reused for subsequent operations:

- **sessionPoolSize**: defines how many maximum sessions can be opened on the ServiceNow. This parameter can be set in application template using the debug page. Default: 10.
- **sessionRetryCounter**: defines how many times IdentityIQ should try to get free session from sessionPoolSize. This parameter can be set in application template using the debug page. Default: 10.

Behavioral change

Following are the behavioral changes observed with increase or decrease of the **sessionPoolSize** or **sessionRetryCounter** parameters:

- If the value of the **sessionPoolSize** or **sessionRetryCounter** parameter has been increased, to reflect the changes user has to perform test connection. The following examples would set the **sessionPoolSize** or **sessionRetryCounter** to 15.

```
<entry key="sessionPoolSize" value="15"/>
<entry key="sessionRetryCounter" value="15"/>
```

Note: If the value of **sessionPoolSize** parameter is increased ServiceNow Connector will open new set of sessions. The sessions which was opened in ServiceNow would be closed after timeout in ServiceNow.

- If the value of **sessionPoolSize** parameter is decreased (for example from 10 to 5) the value will be effective only after restarting the Server.
- If the value of the **sessionRetryCounter** parameter has been decreased, to reflect the changes user has to perform test connection.

Note: ServiceNow Connector creates a new set of sessions on Test Connection.

Note: (*Factors to be considered while defining the session pool size*) The number of session pool size required to handle the ServiceNow Connector requests depends on the number of clients requests at a time, load on IdentityIQ and ability of ServiceNow instance to work with those sessions.

Custom attributes

Perform the following procedure to add a custom attribute on ServiceNow **SailPointSysUser** import set table:

Note: Follow similar steps for other tables.

1. Navigate to **System Definition ==> Tables** and search for table with the name **sys_user**. Open the table and search for any additional field in the Table Columns embedded list which is not mapped to the **SailPointSysUser** import set table.
2. Navigate to **System Definition ==> Tables** and search for table with label **SailPointSysUser**. Open the table and in the Table Columns embedded list click on **New** button. Define the record by providing all the required values and click **Submit**.
3. Navigate to **SailPoint ServiceNow Connector ==> Import Set Tables ==> SailPointSysUser ==> Transform Maps ==> SailPointSysUser** and create new Field Maps in the Field Maps embedded list by clicking on the **New** button. Select the following:
 - Source field = office
 - Target field = officeConfigure additional mapping information and click **Submit**.

Note: The custom attribute must be added to the application account schema in IdentityIQ and ensure that the name must be same as that on ServiceNow of the custom attribute.

Troubleshooting

1 - If a record is not created or updated on ServiceNow and there are no errors displayed

Resolution: Verify the record in relevant import set table listed in “Additional information” on page 349.

2 - Account entity has ‘groups and roles’ field which displays all the groups and roles the user is a part of

The Account entity has groups and roles fields which display all the groups and roles the user is a part of; the Account aggregation only displays the sys id of those groups and roles. After aggregation the account details display some alpha numeric strings in the **groups** and **roles** field.

Workaround: In order to get the group name, the account-group aggregation must be executed, which will replace the sys id from the corresponding group name in the **groups** field and role name in the **roles** field.

3 - Test Connection failed with message: "null". Please check the connection details

- The error is displayed when trying to create ServiceNow application for ServiceNow Calgary or Dublin instance.

Resolution: Test connection fails as supported versions of ServiceNow are Istanbul and later.

- When application created for ServiceNow on IdentityIQ version 7.0 contains url in the following format and after upgrading IdentityIQ to version 7.3 the above error message appears while performing test connection:

<https://demo.service-now.com/navpage.do> and after upgrading IdentityIQ to 7.3, above error occurs while performing test connection

Resolution: Change the url format to <https://demo.service-now.com> and perform test connection.

4 - Error message is displayed in log

- The error means that ServiceNow connector retried to get free session from the session pool but does not get it after number of retries mentioned by *sessionRetryCounter*. The following error message is displayed in the log when ServiceNow Connector tries to get free sessions from the session pool as mentioned in the *sessionRetryCounter*:

No session information available. Please increase session pool size.

Resolution: Due to high load the sessions are too busy and occupied hence increase the session pool size by using the following entry so that more concurrent operations can be performed.

```
<entry key="sessionPoolSize" value="15"/>
```

- Old sessions still valid and reused

For example:

Suppose S1 and S2 are the sessions opened by ServiceNow admin1 defined in ServiceNow Connector as administrator. User changes the username from admin1 to admin2. Now, the sessions S1 and S2 are still valid and can be used till it gets timeout on ServiceNow regardless of the user information is changed in IdentityIQ.

Troubleshooting

In this case for ServiceNow administrator admin1 is still performing operations from IdentityIQ whereas customer would expect admin2 to perform operations from IdentityIQ. The activities will be logged against admin1 whereas, IdentityIQ should use admin2 to perform the operations.

Resolution: Any kind of changes in application should be followed by the test connection operation, otherwise old sessions are still valid and will be reused.

- Reducing value of **sessionPoolSize** entry not been honored until the server is restarted.

Resolution: If the values are increased then user must save application and perform test connection so that the changes are effective.

Note: The recommended sessionPoolSize is 10 so that IdentityIQ can process 10 requests at a time.

```
<entry key="sessionPoolSize" value="10"/>
```

5 - openconnector.ConnectorException: Disable failed. HTTP/1.1 400 Bad Request

When update set is not present on ServiceNow instance, the following error message is displayed:

```
sailpoint.connector.InsufficientPermissionException:  
[InsufficientPermissionException]  
  
[Possible suggestions] Furnish appropriate permissions to the user.  
  
[Error details] Insufficient privileges detected. Status: 403, Output: "Operation Failed", Detail: "ACL Exception Insert Failed due to security constraints", Status: failure
```

Resolution: Upload the update set on ServiceNow instance and assign the appropriate role to the user. For more information, see "User permissions" on page 343.

6 - Unable to fetch next block of account. Exception occurred during account aggregation. Transaction canceled: maximum execution time exceeded

ServiceNow prevents inbound REST request running for longer than 60 seconds starting with Fuji release.

Resolution: On ServiceNow increase the value of **Maximum Duration** field on Transaction Quota Rule - REST request timeout using the following steps:

1. Navigate to **System Definition => Quota Rules**.
2. Select REST request timeout and increase value of **Maximum Duration** (seconds) field as per requirement.

7 - Error message appears in IdentityIQ when deleting the ServiceNow entitlement from Identity

The following error message appears in IdentityIQ when deleting the ServiceNow entitlement from Identity:

```
Remove role 'xxx' failed for user 'abc' with message: Exception in  
removeItem() with error message HTTP/1.1 403 Forbidden
```

Resolution: Perform the following:

1. Clear import set table records.

For example,

- If the error message is displayed during the addition of the entitlements for identity, then clear records from the **SailPointSysUserHasRole** import set table.
- If the error message is displayed during the deletion of the entitlements for identity, then clear records from the **SailPointSysUserHasRole_D** import set table.

2. If a performance degradation is observed in provisioning operation, then use the following ServiceNow link to troubleshoot and improve the performance of the import set jobs:
http://wiki.servicenow.com/index.php?title=Troubleshooting_Import_Set_Performance

8 - Create/Update account fails with an error message

Create/Update account fails with the following error message:

Create account failed with error message: null

Resolution: Assign the **import_set_loader** role to the ServiceNow Connector administrator user.

9 - Create account failed with an error message

Create account failed with the following error message:

HTTP/1.0 302 Found

This error code indicates that the resource requested has been temporarily moved to the URL given by the Location header.

Resolution: The above error message is displayed when **http** is configured in the ServiceNow URL. To resolve the issue use **https** in the ServiceNow URL.

10 - Unable to provision space " " to custom attribute for choice type list

Resolution: Include an **OnComplete** event script (which would run after all rows are already mapped from staging table to target table) on their respective ServiceNow instance and perform the following

- mention all the custom choice field type attribute which are required to clear
- explicitly set the custom field type attributes string to **NULL** when a string --**NONE**-- is provisioned through connector.

Perform the following procedure to reset custom choice field type to --**NONE**--:

1. On ServiceNow instance, configure custom attributes with Choice List and Select Type as string and Choice as **NONE** in Choice List specification section in **SailPointSysUser** table while configuring the custom attribute.
2. Navigate to **SailPoint ServiceNow Connector ==> Transform Maps ==> SailPointSysUser** and under **Transform Scripts** column section click on **New** and select **onComplete** script in **When** field section and add the script to reset Choice field attributes to --**NONE**-- in Script section.

For example, For the following created custom attribute with Choice field type, set the attributes to **NULL** whenever the user sends the string **NONE** in IdentityIQ provisioning for only choice field type (for normal text field, Simple space " " is enough to clear the target field):

- **u_team**
- **u_fruit**

The above example can be defined in the **onComplete** event script as follows:

```
(function runTransformScript(source, map, log, target /*undefined onStart*/ ) {
//Put here all the custom created choice fields as below
if(source.u_team=="NONE")
target.setValue('u_team','NULL');
if(source.u_fruit=="NONE")
target.setValue('u_fruit','NULL');
//target should be updated only once
```

Troubleshooting

```
target.update();  
})();(source, map, log, target);
```

Similarly add the required respective choice field attributes in the above script whenever applicable.

Chapter 37: SailPoint Solaris Connector

The following topics are discussed in this chapter:

Overview.....	355
Supported features	355
Supported Managed Systems	356
Pre-requisites	356
Administrator permissions	357
Configuration parameters.....	358
Additional configuration parameters for SSH configuration	358
Public key authentication configuration.....	359
Schema attributes	359
Account attributes	359
Group attributes.....	361
Provisioning policy attributes.....	361
Account attributes	361
Group attributes.....	362
Additional information	363
Unstructured Target Collector	363
Troubleshooting	364

Overview

In Solaris Connector, users on Solaris computer are used for account provisioning. For group provisioning, groups are used. You can configure the Connector to use any of the attributes of user/group which are supported by Solaris commands.

Supported features

SailPoint Solaris Connector supports the following features:

- Account Management
 - Manages Solaris Users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages Solaris groups as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete

Overview

- Permission Management
 - Solaris application can be configured to read file permissions directly assigned to accounts and groups using Unstructured Target Collector.
 - The connector also supports automated revocation of the aggregated permissions for accounts and groups.

Note: Solaris connector supports MD5, SHA-1, and SHA-2 cryptographic hash functions.

References

- “Unstructured Target Collector” on page 363
- Appendix C: Before and After Provisioning Action

Supported Managed Systems

The Solaris connector supports the following versions of the operating system:

- Solaris 11.3 SPARC x86
- Solaris 11.2 SPARC x86
- Solaris 11 SPARC x86
- Solaris 10 SPARC x86

Note: For any issues related to Solaris, see “Troubleshooting” on page 364 section.

Pre-requisites

- SSH should be installed on Solaris computer.
 - The **sshj-0.23.0.jar** and **ganymed-ssh2-build209-1.0.jar** files must be present in \WEB-INF\lib folder
 - For Sudo users and permissions
 - The administrator user must have rights to execute /usr/bin/awk command.
Update /etc/sudoers file entry for the administrator user with /usr/bin/awk command.
 - User and group schema must add new multi valued schema attribute as **sudoCommands** which would collect all the necessary user commands and store it as a part of this attribute.
 - If end user wants to aggregate the sudo commands from multiple sudo files then user must provide list of files as a separate configuration attribute.
For example, `<entry key="sudoCmdFiles" value="/etc/sudoers.d/special_user.conf,/etc/sudoers.d/special_group.conf"/>`
- Note:** The default command which would collect the sudo commands is as follows:
`awk '/[^#]/' /etc/sudoers`.
In the above command, the commented lines are skipped and the remaining content of /etc/sudoers file are aggregated in to a temporary file on Solaris computer.
The temporary file of Solaris computer would get copied to local IdentityIQ computer and process all the sudo user and group commands.
If the end user wants to provide new command for aggregating file data, then it can be configured as a part of application xml file.
For example: key: sudoUserCommand and value : `awk '/[^#]/' /etc/sudoers`

Administrator permissions

- You can use root user for managing your applications.
- If you want to use sudo user to perform the provisioning operations, the sudo user must be configured with the following rights and permissions:

Rights to execute the following commands with root privilege:

```
/bin/chmod, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel,
/usr/sbin/groupadd, /usr/sbin/groupmod, /usr/sbin/groupdel, /usr/bin/passwd,
/usr/bin/groups, /usr/bin/date, /bin/rm -f spt_tmp_*, /bin/echo, /usr/bin/find,
/bin/cat /etc/shadow, /bin/cat /etc/passwd, /bin/cat /etc/group, /bin/cat
/etc/user_attr, /usr/bin/getent, /bin/grep -i * /etc/default/login, /bin/grep -i
* /etc/security/policy.conf, /usr/bin/finger, /usr/bin/dispid, /usr/bin/awk
```

An entry in /etc/sudoers file should look similar to the following:

```
username ALL = (root) PASSWD: /bin/chmod, /usr/sbin/useradd,
/usr/sbin/usermod, /usr/sbin/userdel, /usr/sbin/groupadd, /usr/sbin/groupmod,
/usr/sbin/groupdel, /usr/bin/passwd, /usr/bin/groups, /usr/bin/date, /bin/rm -f
spt_tmp_*, /bin/echo, /usr/bin/find, /bin/cat /etc/shadow, /bin/cat /etc/passwd,
/bin/cat /etc/group, /bin/cat /etc/user_attr, /usr/bin/getent, /bin/grep -i *
/etc/default/login, /bin/grep -i * /etc/security/policy.conf, /usr/bin/finger,
/usr/bin/dispid, /usr/bin/awk
```

Note: All commands mentioned above are for default configuration. If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry should also be performed. Verify command paths on Solaris computers as they might differ from the values mentioned here.

Note: If you want to use sudo user to perform the provisioning operations ensure to configure home directory with proper write access for this sudo user. In case sudo user is using Guest home directory then ensure it has proper write access over this directory.

Read Only permissions

If you want to use sudo user to perform read only operations, the sudo user must be configured with the following rights and permissions:

- **For Account Aggregation only**

Rights to execute the following commands with root privilege:

```
/bin/echo, /bin/cat /etc/group, /bin/grep, /bin/rm -f spt_tmp_*, /bin/cat
/etc/passwd, /bin/cat /etc/shadow, /bin/cat /etc/user_attr, /usr/bin/date,
/bin/grep -i 'RETRIES=' /etc/default/login, /bin/grep -i 'Lock_After_Retries='
/etc/security/policy.conf
```

An entry in /etc/sudoers file must look similar to the following:

```
username ALL = (root) PASSWD: /bin/echo, /bin/cat /etc/group, /bin/grep, /bin/rm
-f spt_tmp_*, /bin/cat /etc/passwd, /bin/cat /etc/shadow, /bin/cat /etc/user_attr,
/usr/bin/date, /bin/grep -i 'RETRIES=' /etc/default/login, /bin/grep -i
'Lock_After_Retries=' /etc/security/policy.conf
```

- **For Group Aggregation only**

Rights to execute the following commands with root privilege:

```
/bin/echo, /bin/cat /etc/group, /bin/rm -f spt_tmp_*, /bin/grep
```

Configuration parameters

An entry in /etc/sudoers file must look similar to the following:

```
username ALL = (root) PASSWD: /bin/echo, /bin/cat /etc/group, /bin/rm -f spt_tmp_*,  
/bin/grep
```

Note: If any of the command is modified in application definition, then the respective changes in /etc/sudoers file entry must be performed. Verify the command paths on Solaris computers as they might differ from the values mentioned here.

Supported Authentication methods

The Solaris Connector supports the following authentication methods for root and sudo user:

- publickey
- username and password

Configuration parameters

The following table lists the configuration parameters of Solaris Connector:

Parameters	Description
UNIX Server Host	Host Name/IP address of Solaris computer. Note: For IdentityIQ version 6.4 Patch 4 and above, the format of the application XML has been changed from <code><entry key="UnixServerHost" value="<hostname" ><="" code=""> to <code><entry key="host" value="<hostname" ><="" code=""></hostname"></code></hostname"></code>
SSH Port	SSH port configured. Default value: 22
Not a 'root' user	If User ID specified is not root, check this parameter.
User Name	User ID on Solaris computer that you want to use for connector operations.
User Password	Password of the target system user account that you want to use for connector operations. Default value: sadmin
Private Key File Path	Path to Private Key File. Private/Public key authentication will have precedence over password authentication.
Passphrase For Private Key	Passphrase provided for creating Private Key.

Additional configuration parameters for SSH configuration

The following procedure provides the steps for adding the additional configuration parameters for SSH configuration in Application or Target Source debug page.

Note: These additional configuration parameters must be added in the Application/Target Source debug page.

1. Following is the default command for setting shell prompt on UNIX computer:

```
<entry key="SetPrompt" value="PS1='SAILPOINT>' "/>
```

In the above command, “SetPrompt” is the application/target source attribute and PS1='SAILPOINT' is the value of the application/target source attribute.

If the command for setting shell prompt is different than the default command, change the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

For example: For tcsh shell, the entry value would be:

```
<entry key="SetPrompt" value="set prompt='SAILPOINT'"/>
```

2. For executing the commands, verify that the default shell is present on your system.

If the default shell present on your UNIX system is different, modify the value of the application/target source attribute. If the application/target source attribute is not found, add the following new entry in the application/target source debug page:

```
<entry key="DEFAULT_SSH_SHELL" value="tcsh"/>
```

Public key authentication configuration

This is an alternative security method to using passwords. To use public key authentication, you must generate a public and a private key (that is, a key pair). The public key is stored on the remote hosts on which you have accounts. The private key is saved on the computer you use to connect to those remote hosts. This method allows you to log into those remote hosts, and transfer files to them, without using your account passwords.

Perform the following configuration steps to make the UNIX computer as the server and IdentityIQ computer as client:

1. Generate Private and Public key's. For more information of the standard steps, see “7 - Test connection fails for key based authentication with an error” on page 366.
2. Append contents of public key file to `~/.ssh/authorized_keys` as shown below.
`cat <public key file> >> ~/.ssh/authorized_keys`
3. Copy private key file to a location which is accessible by the server.
4. Provide path of private key file in application configuration.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
username	Name of user.
uid	Numeric ID for user.
primgrp	An existing group integer ID or character-string name. Without the -D option, it defines the new user primary group membership and defaults to the default group. You can reset this default value by invoking useradd -D -g group. GIDs 0-99 are reserved for allocation by the Solaris Operating System.

Schema attributes

Attributes	Description
groups	Secondary groups of user. List of groups assigned to user.
roles	Contains the list of roles for each user.
home	Home directory of user.
shell	Default shell of user.
comment	Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's /etc/passwd entry.
authorization	One or more comma separated authorizations defined in auth_attr(4). Only a user or role who has grant rights to the authorization can assign it to an account.
skel_dir	A directory that contains skeleton information (such as.profile) that can be copied into a new user's home directory. This directory must already exist. The system provides the /etc/skel directory that can be used for this purpose
project	Name of the project with which the added user is associated. See the projname field as defined in project(4).
expire	Specify the expiration date for a login. After this date, no user will be able to access this login. The expire option argument is a date entered using one of the date formats included in the template file /etc/datemsk. See getdate(3C). If the date format that you choose includes spaces, it must be quoted. For example, you can enter 10/6/90 or October 6, 1990. A null value (" ") defeats the status of the expired date. This option is useful for creating temporary logins.
inactive	The maximum number of days allowed between uses of a login ID before that ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status.
lock_after_retries	Specifies whether an account is locked after the count of failed logins for a user equals or exceeds the allowed number of retries as defined by RETRIES in /etc/default/login. Possible values are yes or no. The default is no. Account locking is applicable only to local accounts and accounts in the LDAP name service repository if configured with an enableShadowUpdate of true as specified in ldapclient(1M) .
limitpriv	The maximum set of privileges a user or any process started by the user, whether through su(1M) or any other means, can obtain. The system administrator must take ensure that when deleting the privileges from the limit set. Deleting any basic privilege has the ability of crippling all applications; deleting any other privilege can cause many or all applications requiring privileges to malfunction.
defaultpriv	The default set of privileges assigned to a user's inheritable set upon login.
profiles	Contains an ordered, comma-separated list of profile names selected from prof_attr(4). Profiles are enforced by the profile shells, pfcsh, pfksh, and pfsh. See pfsh(1). A default profile is assigned in /etc/security/policy.conf (see policy.conf(4)). If no profiles are assigned, the profile shells do not allow the user to execute any commands.

Attributes	Description
failedretries	Indicates if the user account is locked. Possible values include: <ul style="list-style-type: none"> true: The user account is locked. The values yes, true, and always are equivalent. The user is denied access to the system. false: The user account is not locked. The values no, false, and never are equivalent. The user is allowed access to the system. Default value.
pwdminage	The minimum number of days required between password changes for user. MINWEEKS is found in /etc/default/passwd and is set to NULL.
pwdmaxage	The maximum number of days the password is valid for user. MAXWEEKS is found in /etc/default/passwd and is set to NULL.
pwdwarn	The number of days relative to max before the password expires and the name are warned.
pwdlastchg	The date password was last changed for name. All password aging dates are determined using Greenwich Mean Time (Universal Time) and therefore can differ by as much as a day in other time zones.
audit_flags	Specifies per-user Audit pre selection flags as colon-separated always-audit-flags and never-audit-flags . For example, audit_flags=always-audit-flags:never-audit-flags.

Group attributes

The following table lists the group attributes:

Attributes	Description
groupname	Name of the account group
groupid	Numeric ID of account group

Provisioning policy attributes

This section lists the different policy attributes of Solaris Connector.

Account attributes

The following table lists the provisioning policy attributes for Create and Update Account:

Attributes	Description
Create Account	
username	Name of user.
uid	Numeric ID for user.
	Allow duplication of User ID

Provisioning policy attributes

Attributes	Description
primgrp	An existing group integer ID or character-string name. Without the -D option, it defines the new user primary group membership and defaults to the default group. You can reset this default value by invoking useradd -D -g group. GIDs 0-99 are reserved for allocation by the Solaris Operating System.
home	Home directory of user.
shell	Default shell of user.
comment	Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's /etc/passwd entry.
authorization	One or more comma separated authorizations defined in auth_attr(4). Only a user or role who has grant rights to the authorization can assign it to an account.
profiles	Contains an ordered, comma-separated list of profile names selected from prof_attr(4). Profiles are enforced by the profile shells, pfcsh, pfksh, and pfsh. See pfsh(1). A default profile is assigned in /etc/security/policy.conf (see policy.conf(4)). If no profiles are assigned, the profile shells do not allow the user to execute any commands.
project	Name of the project with which the added user is associated. See the projname field as defined in project(4).
expire	Specify the expiration date for a login. After this date, no user will be able to access this login. The expire option argument is a date entered using one of the date formats included in the template file /etc/datemsk. See getdate(3C). If the date format that you choose includes spaces, it must be quoted. For example, you can enter 10/6/90 or October 6, 1990. A null value (" ") defeats the status of the expired date. This option is useful for creating temporary logins.
inactive	The maximum number of days allowed between uses of a login ID before that ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status.
lock_after_retries	Specifies whether an account is locked after the count of failed logins for a user equals or exceeds the allowed number of retries as defined by RETRIES in /etc/default/login. Possible values are yes or no . The default is no. Account locking is applicable only to local accounts and accounts in the LDAP name service repository if configured with an enableShadowUpdate of true as specified in Idapclient(1M) .
pwdwarn	Warning period for user's password expiry.
pwdminage	Minimum period between user's password change.
forcepwdchange	If user has to be forced to change password on next logon.
pwdmaxage	Maximum period for which password is valid for user.
Password	Initial password for newly created user account.

Group attributes

The following table lists the provisioning policy attributes for Create and Update Group:

Attributes	Description
groupname	Name of the account group
groupid	Numeric ID of account group
dupgid	Allow duplication of groupid.
Update Group	
groupid	Numeric ID of account group.
dupgid	Allow duplication of groupid.

Additional information

This section describes the additional information related to the Solaris Connector.

Note: To enable logging, specify the logging

```
log4j.logger.openconnector.connector.unix.UnixConnector and
log4j.logger.openconnector.connector.unix.SolarisConnector in the
log4j.properties file.
For example, log4j.logger.openconnector.connector.unix.UnixConnector=debug
log4j.logger.openconnector.connector.unix.SolarisConnector=debug.
```

Unstructured Target Collector

Solaris uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with account **identityAttribute** for Accounts and group **identityAttribute** for Account Groups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint User’s Guide*.

For Solaris target permission, the Unstructured Targets functionality will be enabled if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Multiple target sources can be specified and configured for an application which supports unstructured targets. This will be useful for applications which want to fetch resource information from multiple target sources.

Solaris Target Collector support aggregation of file/directories under specified file system path(s). Only direct access permissions will be correlated to the Users and Groups. For UNIX platforms direct access means ownership of file or directory.

Attributes	Description	Possible values
Unix File System Path(s)*	Absolute path(s) which are to be scanned for resources.	Multiple paths can be mentioned with comma separated values. For example, /etc, /tmp
Application Name*	Name of the application with which Unstructured Target will be correlated.	

Note: Attributes marked with * sign are the mandatory attributes.

Troubleshooting

Note: If Unstructured Configuration is configured before upgrading to version 7.3 from version 6.0 Patch 5 or 6.0 Patch 6, then update the configuration and specify the Connector Application Name.

Rule configuration parameters

The rule configuration parameters are used to transform and correlate the targets.

Correlation Rule: The rule used to determine how to correlate account and group information from the application with identity cubes in IdentityIQ.

Note: For version 6.2 onwards, the default schema does not have correlation keys defined. Update correlation rule in Unstructured Target Configuration accordingly.

Provisioning related parameters

Select the settings for provisioning to the box.

- **Override Default Provisioning:** Overrides the default provisioning action for the collector.
- **Provisioning Action:** The overriding provisioning action for the collector.

Troubleshooting

1 - Test connection fails with an error.

The following error message appears when test connection fails:

java.io.IOException: Corrupt Mac on input

OR

Error: Login failed. Error while connecting to host: xxxxx. The message store has reached EOF

Resolution: Add Cipher **3des-cbc** or **blowfish-cbc** to the list of Cipher's in `/etc/ssh/sshd_config` file and restart `sshd`.

- **For X86:** include **3des-cbc** or **blowfish-cbc** in Ciphers list
For example, Ciphers aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc, arcfour, arcfour128, arcfour256, 3des-cbc, blowfish-cbc
- **For SPARC:** include **3des-cbc** in Ciphers list
For example, Ciphers aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc, arcfour, arcfour128, arcfour256, 3des-cbc

2 - Test connection fails with an error

The following error message appears when test connection fails:

[InvalidConfigurationException] [Possible suggestions] Provide right credentials.
[Error details] Failed to authenticate the ssh credentials for user: <user> to the host: XXX.XX.XX.XXX

Resolution: Update `/etc/ssh/sshd_config` file for the following entry and restart `sshd`:

`PasswordAuthentication yes`

3 - Aggregation fails with an error

The following error message appears when aggregation fails:

```
Exception during aggregation of Object Type account on Application Solaris. Reason:  
Unable to create iterator sailpoint.connector.InvalidResponseException:  
[InvalidResponseException] [Possible suggestions] Make sure standalone command works  
with the UNIX terminal. The standalone command is - cat /etc/group | grep -v '^+' |  
grep -v '^-' [Error details] Command failed. Status: 1, Output: sh:  
spt_tmp_groupsb257b857860c4518a5fcac11f789a133: cannot create [Permission denied]
```

Resolution: Create home directory for sudo user and run aggregation again. Ensure that the sudo user is able to create files in its home directory.

4 - Test connection fails with an error

The following error message appears when aggregation fails:

```
Fails with error Login failed. Failed to authenticate the ssh credentials for user:  
test to host: xxxxxxx
```

Resolution: The **ksh93** shell is the default shell `/usr/sbin/sh -> .../bin/i86/ksh93`. The **J2SSH** library does not work properly with this shell.

In default installation of Solaris 11, bash and tcsh are installed, use one of them for provisioning. Use application attribute **DEFAULT_SSH_SHELL**.

For more information on **DEFAULT_SSH_SHELL** parameter, see “Additional configuration parameters for SSH configuration” on page 358.

5 - Aggregation/test connection fails with timeout error

Aggregation/test connection fails with the following timeout error:

```
Exception during aggregation of Object Type account on Application Solaris. Reason:  
Unable to create iterator sailpoint.connector.TimeoutException: [TimeoutException ]  
[Possible suggestions] Tune the parameter <sshTimeOut>. [Error details] Timeout  
occurred while reading output stream for the executed command.
```

Test Connection fails with following timeout error:

```
[TimeoutException] [Possible suggestions] Tune the parameter <sshTimeOut>. [Error  
details] Timeout occurred while reading output stream for the executed command.
```

Resolution: Change the value of the **sshWaitTime (in millisecond)** application attribute as per your requirement in the debug page of the application:

```
<entry key="sshWaitTime" value="500"/>
```

6 - After target aggregation resources are not getting correlated with Account Groups

After target aggregation the resources are not getting correlated with Account Groups.

Resolution: Ensure that your correlation rule populates "Correlator.RULE_RETURN_GROUP_ATTRIBUTE" as follows:

```
....  
if (isGroup) {  
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE, "nativeIdentity");  
    returnMap.put(Correlator.RULE_RETURN_GROUP_ATTRIBUTE_VALUE, nativeId);
```

```
}
```

....

7 - Test connection fails for key based authentication with an error

Test connection fails for key based authentication with the following error.

```
Login failed. Error while connecting to host:<hostname>. Cannot read key file.
```

Resolution: Perform the following steps to generate/convert private/public keys in format which is supported by UNIX direct connectors.

- Generate keys using openssl. This method can be used for any version of SSH.
 - a. Create private key using the following command:

```
openssl <gendsa/genrsa> -des3 -out <private_key> 1024
```
 - b. Change the permission on the <private_key> file as follows:

```
chmod 0600 <private_key>
```
 - c. Create public key from private_key

```
ssh-keygen -y -f <private_key> > <public_key>
```
 - d. Use the <private_key> and <public_key> files for authentication.
- Generate keys using ssh-keygen. (OpenSSH 5.8 or above)
 - a. Create private and public key using the following command

```
ssh-keygen -t <dsa/rsa> -b 1024
```

By default files with name **id_dsa/id_rsa** and **id_dsa.pub/id_rsa.pub** will be created.
 - b. Convert <private key> to have DES-EDE3-CBC encryption algorithm by using the following command:

```
openssl <dsa/rsa> -in <private_key> -out <new_private_key> -des3
```
 - c. Change the permission on the <new_private_key> file as follows:

```
chmod 0600 <new_private_key>
```
 - d. Create public key file using the new private key as follows:

```
ssh-keygen -y -f <new_private_key> > <new_public_key>
```
 - e. Use the <new_private_key> and <new_public_key> files for authentication.

8 - Test connection fails with an error when sudo user is configured for public key authentication

Test connection fails with the following error when sudo user is configured for public key authentication:

```
[InvalidResponseException] [Possible Suggestion] Make sure standalone command works with the UNIX terminal. The standalone command is - echo 'TestConnection'[Error details] Unexpected output captured. Test SSH communication failed over host: xxxxxxxx. Error while executing command: sudo -p %SAILPOINTSUDO echo TestConnection over host: xxxxxxxx. Invalid sudo user password.
```

Resolution: On managed system,

- if Sudoers file is having Sudo user with **PASSWD** attribute assigned, then the sudo user's password specified in application configuration, password must be correct for certificate based authentication.
- if Sudoers file is having Sudo user with **NOPASSWD** attribute assigned, then the sudo user's password specified in application configuration, password can be incorrect/or any value. Certificate based authentication must still work.

Note: Password is mandatory field on application UI.

9 - Aggregation fails with an error for Solaris

The following error message appears when aggregation fails:

```
Exception during aggregation of Object Type account on Application Solaris trouble.
Reason: Unable to create iterator sailpoint.connector.InvalidResponseException:
[InvalidResponseException] [Possible suggestions] Make sure standalone command works
with the UNIX terminal. The standalone command is - date '+%m/%d/%Y %H:%M:%S' [Error
details] Command failed. Cannot collect systems current date. Status: SAILPOINT>echo
$? 0
```

Resolution: Add the following entry of application attribute in the debug page of the application:

```
<entry key="DEFAULT_SSH_SHELL" value="bash"/>
```

10 - For any Solaris user at a time, the maximum allowed addition of groups are 16

Solaris has restriction to select 16 group at a time to allocate any user, while IdentityIQ supports to select more than 16 through Console. In this case only first 16 selected groups are being allocated to user.

Note: This issue does not display any type of error from IdentityIQ and access request will also be marked as committed.

Resolution: To add more number of groups (exceeding 16), user must add the groups in batch of 16 at a time.

Troubleshooting

Chapter 38: SailPoint SuccessFactors Connector

The following topics are discussed in this chapter:

Overview.....	369
Supported features	369
Pre-requisites	370
Administrator permissions	371
Configuration parameters.....	371
Schema attributes	374
Account attributes	374
Additional Information	376

Overview

The SailPoint SuccessFactors Connector facilitates the aggregation of Employees and Contingent workers from SuccessFactors Employee Central cloud module. This connector aggregates information on basis of Personal, Job, Business and Employment objects.

Supported features

SailPoint SuccessFactors Connector supports the following features:

- Account Management
 - Manages SuccessFactors Employees and Contingent workers as Accounts (Active, Inactive and Future Hires)
 - Aggregation, Refresh Accounts
 - Provisioning (with the help of rule)
 - Ability to define separate provisioning rule for modify operation
 - An example of modify provisioning rule is located in **WEB-INF/config/examplerules.xml** file. as mentioned in the following Customization Rule section.

Customization Rule

Modify Rule: The rule name is defined as **Example Rule For Modifying Attributes In SuccessFactors**. This is a sample rule to assign and update the E-mail, Phone number and User Name.

Pre-requisites

Ensure that the following pre-requisites are performed on the SuccessFactors system:

- Enable the SOAP API
- Register a Client application

Enabling the SOAP API

Perform the following procedure to enable the SOAP API:

1. Log in to the **Provisioning Access Console** and select your company and navigate to **Company Settings** and perform the following:
 - a. Under Web Services enable the following checkboxes:
 - **SF API**
 - **Employee Central SOAP API**
 - b. For API to provide information about global assignments, dependents in general, dependents accompanying and Employee on a global assignment, enable the following checkboxes:
 - **Enable Global Assignment Management**
 - **Enable Dependents Management**

Register a Client application

Pre-installation for the SuccessFactors connector involves registering a client application with the target system so that the connector can access REST APIs.

The pre-installation step includes client application registration, certificate generation, and obtaining Client ID attributes as follows:

1. Register your client application with SuccessFactors by navigating to **Admin Centre ==> Company Settings ==> Manage OAuth2 Client Applications ==> Register Client Application**.
2. While creating an application, ensure that you provide information for the mandatory fields such as **Application Name**, **Description**, and **Application URL**.
For example, Application name value can be **SailPointApp** and URL can be **https://SailPointApp**
3. Click on **Generate X.509 Certificate** button and enter the values as required.
4. Click **Generate** and download a copy of the **X.509** certificate on your computer.
5. Open the **X.509** certificate file. The **X.509** certificate has the following parts
 - Private key
 - Certificate

Copy and paste the characters between **--BEGIN ENCRYPTED PRIVATE KEY--** and **--END ENCRYPTED PRIVATE KEY**. This Private key would be used as a configuration parameter for Test connection operation.

6. Click on **Register**.

Note: Save the generated API Key as that would be used as a 'Client ID' which is a configuration parameter for Test connection operation.

Administrator permissions

For specific operations, following are the required permissions for SuccessFactors Connector:

Operation	Required permissions
Test Connection	Test Connection
Aggregation	Test Connection and Aggregation
Provisioning	Test Connection, Aggregation and Provisioning

Test Connection

Category	Permission
General User Permission	SFAPI User Login
Employee Central API	Employee Central HRIS SOAP API
Manage Integration Tools	Admin access to OData API Manage OAuth2 Client Applications

Aggregation

Category	Permission
Manage User	Employee Export
Metadata Framework	Read/Write Permission on Metadata Framework Admin access to MDF OData API
Manage System Properties	Picklist Management and Picklists Mappings Set Up
Employee Central API	Employee Central Foundation OData API (read-only) Employee Central HRIS OData API (read-only)

Provisioning

Category	Permission
Manage User	Import Employee Data
Employee Central API	Employee Central HRIS OData API (editable)

Configuration parameters

This section contains the information that the connector uses to connect and interact with SuccessFactors system through the application. Each application type requires different information to create and maintain a connection.

Configuration parameters

The SuccessFactors connector uses the following connection parameters:

Table 1—Configuration parameters

Parameters	Description
Base Company URL*	Unique endpoint URL to connect SuccessFactors system through API. For example, https://<hostname>.successfactors.com:<port>
Company ID*	Enter the company ID for user provisioning. During licensing of SuccessFactors solution, a unique company ID is provided. The OData API uses the company ID attribute to validate your access token.
User Name*	Name of the user having the privileges mentioned in the "Administrator permissions" on page 371.
Password*	Password for the user account specified in the User Name. Note: This password is required for SOAP authentication
Client ID*	Enter the client identifier (a unique string) issued by the authorization server to your client application during the registration process. You obtained the client ID while performing the procedure specified in section "Register a Client application" on page 370. Note: Client ID information is required for OAuth2 authentication.
Private Key*	Extracted key from X.509 Certificate of SuccessFactors using OAuth2 client application. Note: Private key information is required for OAuth2 authentication for accessing SuccessFactors Odata API.
Aggregate Future Hires	Select this checkbox to aggregate to Future Hires. Note: By default the value of 'futureOffset' (indicates the number of days to aggregate the future hires) would be set to 30 in application debug page. The Future Dated Hires Offset would have the following values: <ul style="list-style-type: none"> • 0: aggregates no future hires. • A positive value: aggregates Future Hires within the specified number of days.
Aggregate Inactive Employees	Select this checkbox to aggregate inactive Employees. Note: By default the value of 'inactiveOffset' (indicates the number of past days to aggregate inactive Employees) would be set to 30 in application debug page. The Inactive Employees Offset would have the following values: <ul style="list-style-type: none"> • 0: aggregates only the active Employees • Any positive value: indicates the number of days in past since when the inactive accounts must be aggregated.

Note: In the above table all the attributes marked with * sign are mandatory attributes.

Note: Service account User Name must be equal to User Id.

Additional configuration parameter

Add the **odataEventOptionIdMap** attribute in the application debug page as follows:

- **For Future, Inactive and Rehire employees**

The odataEventOptionIdMap attribute can be used to aggregate employee based on the Picklist id events - option id values. By default the following values are set for the respective attributes:

- Hire as 3669
- Rehire as 3676
- SCWK as 30768

Any changes in these values, must be updated using the **odataEventOptionIdMap** entry key as follows:

```
<entry key="odataEventOptionIdMap">
  <value>
    <Map>
      <entry key="Hire" value="<optionid>"/>
      <entry key="Rehire" value="<optionid>"/>
      <entry key="SCWK" value="<optionid>"/>
    </Map>
  </value>
</entry>
```

For example,

```
<entry key="odataEventOptionIdMap">
  <value>
    <Map>
      <entry key="Hire" value="3669"/>
      <entry key="Rehire" value="3676"/>
      <entry key="SCWK" value="30768"/>
    </Map>
  </value>
</entry>
```

Note: Option id value associated with Picklist id events can be obtained by navigating to 'Admin Centre' then searching for 'Picklist Management' and then selecting the option 'Export all picklist(s)', which would generate the csv file containing the picklist value.

- **For Employee Status**

Identity status connected to the account would be based on employee status as specified below:

Employee Status	IdentityIQ Status
Active	
Dormant	
unpaid leave	Enabled
paid leave	
Suspended	

Schema attributes

Employee Status	IdentityIQ Status
Furlough	Disabled
Discarded	
Retired	
Terminated	

Default behavior can be modified by specifying **odataEventOptionIdMap** entry key in the application debug page as follows:

```
<entry key="odataEventOptionIdMap">
  <value>
    <Map>
      <entry key="EmplStatus-ActiveOptionIds" value=",
<OptionIdvalue2>"/>
      <entry key="EmplStatus-InActiveOptionIds" value=",
<OptionIdvalue2>"/>
    </Map>
  </value>
</entry>
```

Where **OptionIdvalue** is the option id associated with the **PicklistId** employee-status which can be obtained by navigating to **Admin Centre** and then searching for **Picklist Management** and then selecting the option **Export all picklist(s)**, which would generate the csv file containing the picklist value.

For example,

```
<entry key="odataEventOptionIdMap">
  <value>
    <Map>
      <entry key="EmplStatus-ActiveOptionIds" value="4595,4603,4596,4597,4599"/>
      <entry key="EmplStatus-InActiveOptionIds" value="4601,4602,4598,4600"/>
    </Map>
  </value>
</entry>
```

Note: Additional employee status other than default values if present on the managed system must be configured in the odataEventOptionIdMap, else the records corresponding to the missing option ID would not be aggregated.

Schema attributes

This section provides the different attributes of the Account attributes for SuccessFactors connector.

Account attributes

The application schema is used to configure the objects returned from a connector. When the connector operations are performed the schema is supplied to the methods on the connector interface. This connector currently supports account objects, Account objects are used when building identities Link objects.

Attributes	Description
PersonID	ID of the person
Username	Username
Userid	User ID
Salutation	Salutation
FormalName	Formal name
FirstName	First name
MiddleName	Middle name
LastName	Last name
PreferredName	Preferred name
Date of Birth	Date of birth for Employee
Gender	Gender
Department	Department name
Division	Represent Division name in the organization data
Company	The company under which Employee belongs
Location	Work location name
Country	Name of the country
Nationality	Nationality
PositionNumber	Represent position number associated with Employee
JobTitle	Represent job title associated with Employee
EmployeeType	Represent Employee type
EmployeeStatus	Represent Employee Status
PrimaryEmailAddress	Primary email address
Job Classification	Job classification
CostCenterID	Cost center ID associated with Employee
IsContingentWorker	Represent whether Employee is Contingent Worker or not
FLSA	FLSA status code
AssignmentType	Assignment type
ManagerID	Manager ID
CostCenter	Represent Cost center associated with Employee
EmployeeClass	Represent Employee Class
IsFullTime	Represent whether Employee is Full Time or Part Time
ServiceDate	Service start date
JobInfoLastModified	Date when Job Information was Last Modified

Additional Information

Attributes	Description
Position Entry Date	Position Start date for Employee
LastDateWorked	Last date worked
Address	Address of Employee
BusinessPhone	Business phone
BusinessPhoneCountryCode	Business phone country code
BusinessExtension	Business extension
Cell	Primary cell
CellCountryCode	Primary Cell Code
Fax	Fax number
FutureActions	Stores Information about Future Actions in the following JSON format: {"Actions": [{"ActionType" : "<ActionCode>","ActionReason" : "<Action Reason value>","ActionStartDate" : "<Start date>","ActionEndDate" : "<End date>"}]} For example, {"Actions": [{"ActionType" : "SCWK","ActionReason" : "Start CWK","ActionStartDate" : "2018-08-26","ActionEndDate" : "2018-09-08"}]} Note: This attribute will aggregate only the future employment actions.
Person ID External	Person ID External
BusinessUnit	BusinessUnit Name

Additional Information

This section describes the additional information of SuccessFactors Connector.

Support for Additional Parameters

SuccessFactors Connector provides support for the following additional parameters:

1. **Aggregation Page Size:** Value can be set as follows:

```
<entry key="aggPageSize" value="200"/>
```

Note: aggPageSize value is Number of records per page. Default: 200

2. **apiServerUrl:** Value can be set as follows:

```
<entry key="apiServerUrl" value="https://HOST:PORT"/>
```

where **apiServerUrl** value is the base URL for the REST API server.

3. **Default Time Out:** Value can be set as follows:

```
<entry key="apiTimeout" value="5"/>
```

Note: apiTimeout value is in terms of minutes.

Troubleshooting

1 - Test connection fails

Test connection fails with the following with error message:

Unable to verify the signature of the SAML assertion

Resolution: Values of Client ID and Private key must be provided correctly. These values must be a part of the same OAuth2 Client Applications.

Troubleshooting

Chapter 39: SailPoint SQL Loader Connector

The following topics are discussed in this chapter:

Overview.....	379
Supported features	379
Supported Managed Systems	380
Administrator permissions	380
Configuration parameters.....	380
Schema Attributes	382
Troubleshooting.....	382

Overview

The SailPoint SQL Loader Connector supports Read/Write operations on flat file data like CSV, TEXT flat files. The data in these files are separated with delimiters. This connector can handle aggregation for multiple file by defining complex SQL query.

This connector can be configured to enable the automatic discovery of schema attributes. See “Schema Attributes” on page 382.

Supported features

SailPoint SQL Loader Connector supports the following features:

- Account Management
 - Manages SQL Users as Accounts
 - Aggregation, Partitioning Aggregation, Refresh Accounts, Discover Schema
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - (Application Object Type)
 - Manages SQL groups as Account-(Application Object Type)
 - Aggregation, Refresh (Application Object Type)
 - Create, Update, Delete
- Permission Management
 - Application reads permissions directly assigned to application object types as direct permissions during account and application object type aggregation.
 - The connector does not support automated revocation of the aggregated permissions and creates work item for such requests.

Configuration parameters

SailPoint supports the following additional SQL Loader Connector features:

- Ability to provide the SQL statement or stored procedure during application configuration for automatic discovery of account-group (application object type) schema attributes from same or different files used for the account schema.
- Ability to define provisioning rule(s) called for each row in the data file to provision account and group (application object type) attributes.
- Ability to define separate provisioning rule for specific operation called for each row in the data file to provision account and group (application object type) attributes. Operation that include are Enable, Disable, Unlock, Delete, Create, and Modify.

Note: An example of a provisioning rule is located in `examplerules.xml` file.

References

- Appendix B: Partitioning Aggregation

Supported Managed Systems

SailPoint SQL Loader Connector supports flat file data which contains delimiter. The extension for the flat file can be `.txt` or `.csv`.

Administrator permissions

Administrator must have the read and write permission on the files in the given directory path.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The SQL Loader Connector uses the following connection attributes under different tabs (Settings, Merging and Iteration Partitioning):

Attribute	Description
Settings	
<i>SQL Loader Connection Settings</i>	
Directory Path*	The directory path in which the Target files are stored.
Delimiter*	Delimiter separator type with which the entire row gets separated with different columns.
Treat First Record As Header	Indicates whether the first record of CSV files container is provided with column headers as first column then URL would be as follows: <code>jdbc:csv:/c:/data?_CSV_Separator= ;_CSV_Header=true</code>

Attribute	Description
Database URL*	<p>URL to connect to the database. This will be automatically created when the delimiter and directory attributes are filled up with appropriate data.</p> <p>For example,</p> <ul style="list-style-type: none"> (For Windows) <code>jdbc:csv:/c:/data?_CSV_Separator= </code> (For UNIX) <code>jdbc:csv:///home/sqlloader/acc.csv?_CSV_Separator= </code>
JDBC Driver*	Enter the JDBC driver class path.
Query Settings	
SQL Statement*	<p>The SQL attribute can be used to customize the select statement that is generated when iterating over objects. You can specify the exact SQL that is executed if you want to filter out objects or only want to select a few objects from a table. Additionally, if you want to perform joins between more than one table, it is impossible to describe with the schema alone.</p> <p>By default if the SQL option is null when the query string is built using the schema attributes and nativeObjectType.</p>
getObjectSQL	The object SQL statement.
useExecuteQuery	Use Statement.executeQuery() instead of the default Statement.execute()
Direct Permission Execute Query	Enable this option to execute the query for direct permission.
Get Direct Perm Object SQL	<p>Direct Permission Execute Query is used to retrieve the direct permission data from permission file. Permission file should contain at least Identity attribute column. The permission data is retrieved by referring the identity attribute in the column at the time of aggregation through main SQL query in which the identity attribute is mentioned.</p> <p>Note: Query must be written in such a way that ResultSet data must contain first column as Target, second column as Permission and third column as annotation (optional).</p> <p>For example, <code>SELECT column4 AS TARGET, column5 AS PERMISSION FROM Permission p WHERE CONCAT(TRIM(CONCAT(p.column1, '\')), TRIM(p.column2)) = '\${identity}';</code></p> <p>Here file name is <code>Permission.csv</code> and <code>\$(identity)</code> is Identity attribute.</p>
Merging	
Data needs to be merged	<p>Select this option if the data for a single object spans multiple lines.</p> <p>This option enables the connector to verify the order of the data returned from the database when merging to prevent data loss. When merging, it is very important to have the ORDER BY clause in your SQL statement to prevent out of order errors.</p>
Index Column	Name of the index column that will be used when finding like objects in the dataset.

Schema Attributes

Attribute	Description
Which columns should be merged?	Names of the columns from the file from which values must be merged.
Note: User must discover the schema to get the suggested column values in index and merge columns for selection. Discover schema populates the values in multi-suggest attribute dropdown of index and merge columns which have the auto complete facility.	
Iteration Partitioning	
Partitioning Enabled	Select this checkbox to configure and enable partitioning.
Partitioning Statements	Enter the list of sql/stored procedure statements that must be executed when partitioning. The statements must include all of the rows and each line/statement so it can be proceeded in separate threads and/or multiple hosts. For more information, see Appendix B: Partitioning Aggregation.

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. Version 7.3 supports multiple types of objects, account and any number of group application object types. Account objects are used when building identities Link objects. Additional schema definitions can be used when building AccountGroup objects which are used to hold entitlements shared across identities.

The SQL Loader Connector's most important attribute is the SQL statement. In many cases this is a stored procedure (`call mystoredProcedure`). In other cases it is select from a table with any number of joins included. If this connector is configured to use the automatic discovery function, it connects to the database and executes the statement provided and then uses the meta-data returned from the result to build the column names.

Troubleshooting

1 - Database URL format requires change if application is on UNIX computer

The following error message is displayed:

```
Unable to discover the [account] schema for this
application.[sailpoint.connector.ConnectorException: Error while trying to discover
the columns. SQL[null]sailpoint.connector.ConnectorException: Failure
trying to get a pooled connection to
[jdbc:csv://home/manoj?_CSV_Separator=,]java.sql.SQLException: home/manoj doesn't
exist or can't be accessed. If you're using mapped drives to access database files,
you may need to check the security permissions.]
```

Resolution: Update the directory path field in the following format:

```
///DirectoryName/SubDirectoryName
```

The database URL should be of the following format for UNIX computers:

```
jdbc:csv:///DirectoryName/SubDirectoryName
```

2 - Issue of SQL Loader dropping records

Resolution: If the CSV data is provided with column headers as first column then URL would be as follows:

```
jdbc:csv:/c:/data?_CSV_Separator=|;_CSV_Header=true
```

In this case, perform the following:

1. Use CSV_Header=true in the URL. By adding this, you can directly use header_name of the csv file as column name in the SQL query.

For example, url would be as follows:

```
jdbc:csv:/c:/data?_CSV_Separator=\u003B;_CSV_Header=true
```

2. Instead of using column1, column2... in the SQL query use the header names.

For Example, earlier SQL query was as follows:

```
Select emp.column1 as pimId, emp.column2 as employeeId .... from table abc;
```

3 - Case sensitive query

Only those records which are of the same case (case insensitive) as provided in the string would be fetched.

For example, if the query is as follows then only the data with **Abc** would be fetched and data with **ABC** would be ignored.

```
select * from [file] where column3 = 'Abc'
```

Resolution: Add the following attribute with the value as true in the database URL:

```
caseInsensitive=true
```

For example, `jdbc:csv:/c:/data?_CSV_Separator=|;_CSV_Header=true;caseInsensitive=true`

Troubleshooting

Chapter 40: SailPoint System for Cross-Domain Identity Management Connector 2.0

The following topics are discussed in this chapter:

Overview.....	385
Supported features	385
Administrator permissions	386
Supported Managed System.....	386
Configuration parameters.....	386
Schema attributes	387
Provisioning Policy attributes.....	388
Update account attributes.....	388
Provisioning of extended attributes.....	388
Troubleshooting.....	389

Overview

The SCIM (System for Cross-Domain Identity Management) standard defines a schema and API to create, read, update, and delete identity and identity-related information on other systems. This standard creates a common language, by which a client system can communicate with many different servers in the same way. SaaS providers (such as Salesforce) and other software vendors are beginning to adopt this standard, and are exposing their identity management interfaces through SCIM.

Supported features

SailPoint SCIM 2 Connector supports the following features:

- Account Management
 - Aggregation, Discover Schema
 - Create, Update, Delete
 - Enable, Disable
 - Change Password
 - Add/Remove Entitlements

Note: Add/Remove Entitlement would work if entitlements, groups and roles are a part of SCIM 2 core schema. For more information on SCIM 2 core schema, see <https://tools.ietf.org/html/rfc7643>.
- Account - Group Management
 - Aggregation, Discover Schema

Administrator permissions

The required administrator permissions depends on which SCIM 2 server is being connected to. For example, IdentityIQ as SCIM 2 server has SCIM 2 executor capabilities which has minimal permission required for operations to be executed through SCIM 2 Connector.

Supported Managed System

SailPoint SCIM2 Connector supports SCIM servers which are compliant to SCIM 2 protocol.

Configuration parameters

The following table lists the configuration parameters of SCIM 2.0 Connector:

Parameters	Description
Base URL*	The base URL to connect to the SCIM 2.0 server.
Authentication Type*	Authentication method that is supported by the end managed system <ul style="list-style-type: none"> • OAuth 2.0 • API Token • Basic Authentication
Grant Type	Authorization grant to be used to obtain an access token. <ul style="list-style-type: none"> • Refresh Token • Client Credentials
Account Filter	Filter strings for Users.
Group Filter	Filter strings for Groups.
Role Filter	Filter strings for Roles.
Entitlement Filter	Filter strings for Entitlements.
Explicit Attribute Request	Select to create request for fetching required attributes in request. Note: If selected, Connector would request for attributes present in schema else will request for all attributes.
Accept Header	The http ‘Accept’ header value to use for SCIM requests. If blank, the default value is application/scim+json .
Content-type Header	The http ‘Content-type’ header value to use for SCIM requests. If blank, the default value is application/scim+json .
<i>Applicable if Authentication Type is selected as OAuth 2.0</i>	
Grant Type*	Select the type of Grant: <ul style="list-style-type: none"> • Refresh Token • Client Credentials

Parameters	Description
Client Id*	Client Id for OAuth 2.0 authentication. The client identifier issued to the client during the registration process for OAuth 2.0 on the SCIM 2.0 server.
Client Secret*	The client secret pertaining to the provided Client ID.
Token URL*	The endpoint to be used by the client to exchange an authorization grant for an access token.
Refresh Token*	(Applicable if Grant Type is selected as Refresh Token) A valid refresh token to be used to generate access token.
<i>Applicable if Authentication Type is selected as API Token</i>	
API Token*	The API Token to be used for authorization. Value must be added with the token type. For example, Bearer <AUTH TOKEN>
<i>Applicable if Authentication Type is selected as Basic Authentication</i>	
Username*	Username for the SCIM 2.0 Server.
Password*	Password for the SCIM 2.0 Server.

Note: Attributes marked with * sign are the mandatory attributes.

Additional configuration parameter

The SCIM 2.0 now supports authentication using **OAuth2** type and **Refresh Token** as the Grant Type for managed systems which issue a non-expiring/long validity refresh tokens. The administrator would require to configure parameters relevant to OAuth2 (Client Id, Client Secret, Token URL and Refresh Token) for enabling OAuth2 authentication.

For few managed systems, which require customized headers to be a part of the access token generation request, add the following additional configuration attribute in the application debug page:

```
<entry key="oauth_headers">
<value>
<Map>
    <entry key="Content-Type" value="application/x-www-form-urlencoded" />
</Map>
</value>
</entry>
```

Schema attributes

The application schema is used to configure the objects returned from connector. The following types of objects (account, group, entitlements, and roles) are supported:

- Account object type is mapped to SCIM 2 server User resource.
- Group object type is mapped to SCIM 2 server Group resource.
- Entitlements object type is mapped to SCIM 2 server Entitlement resource.
- Roles object type is mapped to SCIM 2 server Role resource.

Provisioning Policy attributes

Note: Discover schema populates schema attribute values for supported object type. The newly added extended schema attributes on SCIM 2 Server can be obtained into IdentityIQ schema by clicking on the Discover Schema option of the respective objectType. For example, to obtain the ‘test’ extended attribute on SCIM 2 Server into IdentityIQ schema, click on the account object type Discover Schema option.

Provisioning Policy attributes

This section lists the different policy attributes of SCIM 2 Connector.

Update account attributes

Update account provisioning policy varies according to SCIM 2 managed system. To update any attribute, add account schema attributes name as it is in update provisioning policy.

Pre-requisite:

Execute **Discover Schema** to obtain attributes present on managed system after creating application for accounts in IdentityIQ.

Provisioning of extended attributes

For provisioning of extended attributes write a Before Provisioning rule to modify the Provisioning Plan and prepare a AttributeRequest that includes only the right value (not the full JSON).

Following is an example of before provisioning rule for updating extended complex schema attributes when managed system is supporting HTTP PATCH method:

```
import com.google.gson.Gson;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;
import com.google.gson.reflect.TypeToken;
import org.codehaus.jackson.map.ObjectMapper;
import sailpoint.object.Application;
import sailpoint.object.ProvisioningPlan;
import sailpoint.object.ProvisioningPlan.AttributeRequest;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

try {
    AttributeRequest attrReq =
plan.getAccountRequest(application.getName()).getAttributeRequest("manager");
    String valueList = attrReq.getValue().toString();
    // Here you get a value in JSON format, fetch your interesting value from the JSON
    // Parser Logic goes here.
    ObjectMapper mapper = new ObjectMapper();
    Map map = mapper.readValue(valueList , Map.class);
    String value = (String) map.get("value");
    // Prepare the right AttributeRequest
    attrReq.setName("manager.value");
    attrReq.setValue(value);
```

```

} catch (Exception e) {
    e.printStackTrace();
}

```

Additional information

This section describes the additional information related to the SCIM 2.0 connector.

Upgrade considerations

When upgrading IdentityIQ to version 7.3, the **OAuth 2.0** Authentication Type does not work for the existing SCIM 2.0 application. Hence for **OAuth 2.0** Authentication Type to work, add the following entry key to the application debug page:

```
<entry key="encrypted"
value="client_secret,oauthBearerToken,oauthTokenInfo,refresh_token"/>
```

Troubleshooting

1 - Create account status is pending

During create account operation, the create account task action status remains in pending state.

Resolution: Perform the following:

1. Add **active** attribute in Create Account Provisioning Policy.
2. Perform create account operation.
3. Run **Perform Identity Request Maintenance** task.

2 - While aggregation an error message is displayed

During aggregation the following error message is displayed:

```
ResourceObject is returned with null identity error
```

Resolution: Verify if the identity attribute is present in the schema attribute else add the appropriate schema attribute name as identity attribute.

Troubleshooting

Chapter 41: SailPoint System for Cross-Domain Identity Management Connector

The following topics are discussed in this chapter:

Overview.....	391
Supported features	391
Administrator permissions	392
Configuration parameters.....	392
Schema attributes	396
Account attributes	396
Group attributes.....	400
Provisioning Policy attributes	401
Create account attributes	401
Update group attributes	401
Additional information	401
Troubleshooting.....	402

Overview

The SCIM (System for Cross-Domain Identity Management) standard defines a schema and API to create, read, update, and delete identity and identity-related information on other systems. This standard creates a common language, by which a client system can communicate with many different servers in the same way. SaaS providers (such as Salesforce) and other software vendors are beginning to adopt this standard, and are exposing their identity management interfaces through SCIM.

Supported features

SailPoint SCIM Connector supports the following features:

- Account Management
 - Manage SCIM Users as Account
 - Aggregation, Refresh Accounts, Discover Schema
 - Create, Update, Delete
 - Enable, Disable
 - Change Password
 - Add/Remove Entitlements

Configuration parameters

- Account - Group Management
 - Aggregation, Discover Schema
 - Create, Update, Delete
 - Enable, Disable
 - Change Password

Note the following:

- SCIM1.1 Connector supports proxy host and proxy port for SCIM compliant servers.
- SCIM1.1 Connector supports filtering option for SCIM compliant servers. Users may request a subset of Resources by specifying the filter criteria for Account and Group.
- SCIM1.1 connector supports PATCH operation for PATCH supporting SCIM compliant servers. PATCH enables users to send only those attributes requiring modification. Attributes may be deleted, replaced, merged, or added in a single request.

User can enable PUT operation for SCIM compliant servers by explicitly setting **HttpPatchSupport** value to false. For more information, see “Additional configuration parameters” on page 393.

Administrator permissions

The required administrator permissions depends on which SCIM server is being connected to.

Configuration parameters

The following table lists the configuration parameters of SCIM Connector:

Parameters	Description
Base URL*	The base URL to connect to the SCIM server.
Authentication Type*	Authentication method that is supported by the end managed system <ul style="list-style-type: none">• Basic (username and password)• OAuth 2.0 (bearer token)• OAuth 1.0 Token (a bearer token using the “Bearer” header)
Page Size	The maximum size of each data set when querying large number of objects. Specify the page size if SCIM Service Provider supports pagination. Pagination would not be applied for no value. Minimum allowed value: 0
Content Type	The data format for SCIM resources. Either XML or JSON (default).
Filter Condition for Accounts	Optional condition to filter Accounts during aggregation. Example: <code>userName sw "S"</code>
Filter Condition for Groups	Optional condition to filter Groups during aggregation. Example: <code>displayName sw "S"</code>

Parameters	Description
<i>Applicable if Authentication Type is selected as OAuth 2.0</i>	
Grant Type*	Select the type of Grant: <ul style="list-style-type: none"> • Client Credentials • Refresh Token • JWT
Client Id*	Client Id for OAuth 2.0 authentication. The client identifier issued to the client during the registration process for OAuth 2.0 on the SCIM server.
Client Secret*	The client secret pertaining to the provided Client ID.
Token URL*	The endpoint to be used by the client to exchange an authorization grant for an access token.
Refresh Token*	(Applicable if Grant Type is selected as Refresh Token) A valid refresh token to be used to generate access token.
Private Key*	(Applicable if Grant Type is selected as JWT) The private key to be used to sign the JWT.
Private Key Password*	(Applicable if Grant Type is selected as JWT) Password for the provided private key.
<i>Applicable if Authentication Type is selected as API Token</i>	
API Token*	The API Token to be used for authorization. Only bearer API tokens are supported.
<i>Applicable if Authentication Type is selected as Basic</i>	
Username*	Username for the SCIM Server.
Password*	Password for the SCIM Server.

Note: Attributes marked with * sign are the mandatory attributes.

Additional configuration parameters

Add the following parameters in the application debug page:

Parameters	Description
skipSchemaAttributes	During update operation, SCIM connector skips the specified attributes. For example: <pre><entry key="skipSchemaAttributes"> <value> <List> <String>alias</String> <String>groups</String> </List> </value> </entry></pre>

Configuration parameters

Parameters	Description
queryAccountSchemaAttributes and queryGroupSchemaAttributes	<p>Retrieves specified query attributes during account and group aggregation.</p> <p>For example:</p> <pre data-bbox="736 439 1416 686"><entry key="queryAccountSchemaAttributes"> <value> <List> <String>userName</String> <String>urn:scim:schemas:extension:enterprise:1.0:manager</String> </List> </value> </entry></pre> <p>Note: Specify the fully qualified extended attributes with the associated Resource URN.</p> <p>For example, the 'manager' attribute defined in urn:scim:schemas:extension:enterprise is fully encoded as urn:scim:schemas:extension:enterprise:manager.</p>
extensionSchemaAttributes	<p>Provides the list of fully qualified extended attributes with the associated Resource URN.</p> <p>For example:</p> <pre data-bbox="736 1072 1416 1305"><entry key="extensionSchemaAttributes"> <value> <List> <String>urn:scim:schemas:extension:enterprise:manager</String> </List> </value> </entry></pre>
HttpPatchSupport	<p>Enables PUT operation for SCIM compliant servers.</p> <pre data-bbox="736 1389 1176 1535"><entry key="HttpPatchSupport"> <value> <Boolean>false</Boolean> </value> </entry></pre>

Parameters	Description
scimAttrMapping	<p>Defines mapping of connector schema attributes with SCIM schema attributes.</p> <p>For example:</p> <pre data-bbox="736 424 1302 952"><entry key="scimAttrMapping"> <value> <Map> <entry key="name"> <value> <Map> <entry key="familyName" value="familyName"/> <entry key="formattedName" value="formatted"/> <entry key="givenName" value="givenName"/> </Map> </value> </entry> </Map> </value> </entry></pre> <p>In the above example, <code><entry key="name"></code> is the SCIM complex attribute created with subAttributes <code>familyName</code>, <code>formatted</code> and <code>givenName</code> which are mapped to <code>familyName</code>, <code>formattedName</code> and <code>givenName</code> Connector schema attributes respectively.</p>
For Grant Type as JWT	
oAuthJwtHeader	<p>Contain the alg (algorithm that is used for signing the JWT assertion). Must be the same algorithm used for generating the private key.</p> <pre data-bbox="736 1326 1286 1522"><entry key="oAuthJwtHeader"> <value> <Map> <entry key="alg" value="RS256"/> </Map> </value> </entry></pre> <p>If required additional header attributes can be provided in this map.</p>

Schema attributes

Parameters	Description
oAuthJwtPayload	<p>Contains the aud (Audience), Expiry of the JWT assertion (exp), iss (Issuer), sub (Subject).</p> <pre><entry key="oAuthJwtPayload"> <value> <Map> <entry key="aud" value="" /> <entry key="exp" value="15f" /> <entry key="iss" value="" /> <entry key="sub" value="" /> </Map> </value> </entry></pre> <p>If required additional payload attributes can be provided in this map.</p>

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
id	A unique identifier for a SCIM resource.
userName	Name of the user.
externalId	A String that is an identifier for the resource.
displayName	The name of the user.
nickName	The casual way to address the user in real life.
profileUrl	A fully qualified URL to a page representing the Users Online profile.
title	The user's title, such as Vice President .
userType	Used to identify the organization to user relationship. Values can be: Contractor , Employee , Intern , Temp , External , and Unknown
preferredLanguage	Indicates the User's preferred written or spoken language.
locale	Used to indicate the User's default location for purposes of localizing items such as currency, date time format, numerical representations, and so on.

Attributes	Description
timezone	The User's time zone in the Olson timezone database format.
formattedName	The full name, including all middle names, titles, and suffixes as appropriate, formatted for display.
familyName	The family name of the User, or Last Name in most Western languages.
givenName	The given name of the User, or First Name in most Western languages.
middleName	The middle name(s) of the User.
honorificPrefix	The honorific prefix(es) of the User, or Title in most Western languages. For example, Ms. given the full name Ms. Barbara Jane Jensen, III.
honorificSuffix	The honorific suffix(es) of the User, or Suffix in most Western languages. For example, III. given the full name Ms. Barbara Jane Jensen, III.
employeeNumber	Numeric or alphanumeric identifier assigned to a person, typically based on order of hire or association with an organization.
costCenter	Identifies the name of a cost center.
organization	Identifies the name of an organization.
division	Identifies the name of a division.
department	Identifies the name of a department.
managerId	The id of the SCIM resource representing the Users manager.
managerName	Name of the manager.
emails	E-mail addresses for the User. The value must be canonicalized by the Service Provider. For example, <code>bjensen@example.com</code> instead of <code>bjensen@EXAMPLE.COM</code> .
emails_objects	A list of all of the users email addresses, including their type (for example, home, work) and whether it is their primary address.
emails_primary	The users primary email address.
phoneNumbers	Phone numbers for the User. The value must be Canonicalized by the Service Provider.
phoneNumbers_objects	A list of all of the users phone numbers, including their type (for example, home, work) and whether it is their primary phone number.
phoneNumbers_primary	The users primary phone number.
ims	Instant messaging address for the User.

Schema attributes

Attributes	Description
ims_objects	A list of all of the instant messaging usernames, including their type (for example, aim, gtalk) and whether it is their primary messaging username.
ims_primary	The users primary instant messaging username.
photos	URL of a photo of the User.
photos_objects	A list of URLs of all of the users photos, including the photo type (for example, photo, thumbnail) and whether it is their primary photo.
photos_primary	The URL of the users primary photo.
addresses	A physical mailing address for this User.
addresses_objects	A list of all of the users physical mailing addresses, including their type (for example, home, work) and whether it is their primary address.
addresses_primary	The users primary physical mailing address.
groups	A list of groups that the user belongs to, either thorough direct membership, nested groups, or dynamically calculated.
groups_objects	A list of all of the users group memberships, including how they are assigned to the group (for example, direct : if assigned directly or indirect : if assigned indirectly through another group) and whether it is their primary group.
groups_primary	The users primary group membership.
entitlements	A list of entitlements for the User that represent a thing the User has.
entitlements_objects	A list of all of the users entitlements, including whether it is their primary entitlement.
entitlements_primary	The users primary entitlement.
roles	A list of roles for the User that collectively represent who the User is. For example, Student, Faculty.
roles_objects	A list of all of the user's roles, including whether it is their primary role.
roles_primary	The users primary role.
created	The DateTime the Resource was added to the Service Provider.
lastModified	The most recent DateTime the details of this Resource were updated at the Service Provider.
location	The URL of the Resource being returned. This value MUST be the same as the Location HTTP response header.

Attributes	Description
version	The version of the Resource being returned.
Complex Schema attributes	
emails.home.primary.value	Primary home email address for the User.
emails.home.secondary.value	Secondary home email addresses for the User.
emails.work.primary.value	Primary work email address for the User.
emails.work.secondary.value	Secondary work email addresses for the User.
emails.other.primary.value	Primary other email address for the User.
emails.other.secondary.value	Secondary other email addresses for the User.
phoneNumbers.home.primary.value	Primary home phone number for the User.
phoneNumbers.home.secondary.value	Secondary home phone numbers for the User.
phoneNumbers.work.primary.value	Primary work phone number for the User.
phoneNumbers.work.secondary.value	Secondary work phone numbers for the User.
phoneNumbers.mobile.primary.value	Primary mobile phone number for the User.
phoneNumbers.mobile.secondary.value	Secondary mobile phone numbers for the User.
phoneNumbers.other.primary.value	Primary other phone number for the User.
phoneNumbers.other.secondary.value	Secondary other phone numbers for the User.
phoneNumbers.fax.primary.value	Primary fax number for the User.
phoneNumbers.fax.secondary.value	Secondary fax numbers for the User.
phoneNumbers.pager.primary.value	Primary pager number for the User.
phoneNumbers.pager.secondary.value	Secondary pager numbers for the User.
ims.aim.value	AIM instant messaging address for the User.
ims.gtalk.value	Gtalk instant messaging address for the User.
ims.icq.value	ICQ instant messaging address for the User.
ims.xmpp.value	XMPP instant messaging address for the User.
ims.msn.value	MSN instant messaging address for the User.
ims.skype.value	Skype instant messaging address for the User
ims.qq.value	QQ instant messaging address for the User.
ims.yahoo.value	Yahoo instant messaging address for the User.
photos.photo.value	URL of a photo of the User.
photos.thumbnail.value	URL of a thumbnail photo of the User.
addresses.home.formatted	Formatted home address for this User.
addresses.home.streetAddress	Home Street address for this User.
addresses.home.locality	Home address locality address for this User.
addresses.home.region	Home address region for this User.

Schema attributes

Attributes	Description
addresses.home.postalCode	Home address postal code for this User.
addresses.home.country	Home address country for this User.
addresses.work.formatted	Formatted work address for this User.
addresses.work.streetAddress	Work Street address for this User.
addresses.work.locality	Work address locality address for this User.
addresses.work.region	Work address region for this User.
addresses.work.postalCode	Work address postal code for this User.
addresses.work.country	Work address country for this User.
addresses.other.formatted	Formatted other address for this User.
addresses.other.streetAddress	Other Street address for this User.
addresses.other.locality	Other address locality address for this User.
addresses.other.region	Other address region for this User.
addresses.other.postalCode	Other address postal code for this User.
addresses.other.country	Other address country for this User.
roles.primary.value	User's primary role.
roles.secondary.value	User's secondary roles.

Group attributes

The following table lists the group attributes:

Attributes	Description
id	A unique identifier for a SCIM resource.
externalId	A String that is an identifier for the resource.
displayName	The name for the Group.
members	A list of members of the Group.
memberGroups	A list of the sub-groups of this group.
created	The DateTime the Resource was added to the Service Provider.
lastModified	The most recent DateTime the details of this Resource were updated at the Service Provider.
location	The URI of the Resource being returned. This value MUST be the same as the Location HTTP response header.
version	The version of the Resource being returned.

Provisioning Policy attributes

This section lists the different policy attributes of SCIM Connector.

Note: The attributes marked with * sign are the required attributes.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
User Name	Name of the user to create.
FirstName	First name of the user.
Full Name	Full name of the user.
Last Name	Last name of the user.
Password	Password of the user.
Email	Email ID of the user.
Email Type	Email type of the user.
Email Primary	Determine if the email is primary.

Update group attributes

The following table lists the provisioning policy attributes for Update Group:

Attributes	Description
Display Name	The display name of the group.
External ID	The external identifier of the group, which can natively identify the group on the resource.

Additional information

This section describes the additional information related to the SCIM Connector.

Upgrade considerations

- Before upgrading IdentityIQ to version 7.3 from 7.2 Patch 1 or earlier version, if the **Authentication Type** is selected as **Basic**, then after upgrading IdentityIQ to version 7.3 the **Authentication Type** would be displayed as **Basic Authentication**.
- After upgrading IdentityIQ to version 7.3 from 7.2 Patch 1 or earlier version, ensure that you select the required **Authentication Type** again and enter the relevant configuration parameters.
- After upgrading IdentityIQ to version 7.3 from 7.2 Patch 1 or earlier version, add the following entry key in the application debug page of the existing application:

```
<entry key="encrypted"
value="client_secret,refresh_token,oauthBearerToken,oauthTokenInfo,apiToken,private_key,private_key_password"/>
```

Troubleshooting

1 - Mapping of connector schema attributes with SCIM schema attributes fails with an error message

If **scimAttrMapping** attribute is missing in the application debug page, the following error message is displayed:

Unable to add entry 'uid=john,ou=people,dc=example,dc=com' because it violates the provided schema: The entry is missing required attribute cn

Resolution: Add the **scimAttrMapping** attribute to the application debug page.

For more information, see “Additional configuration parameters” section.

2 - Provisioning failed with an error message

Provisioning fails with the following error message:

Resource 'User' is malformed: Attribute urn:scim:schemas:core:1.0:alias is not defined for resource User

Resolution: Add alias attribute in **skipSchemaAttributes** list, which is not supported by SCIM server.

3 - Create account fails for Salesforce SCIM server with an error message

Create account fails for Salesforce SCIM server with the following error message:

REQUIRED_FIELD_MISSING:user_must_have_one_entitlement_which_must_be_a_profileid;

Resolution: While performing create any new account with **Manage User Access**, select at least one valid entitlement from **Manage User Access ==> Manage Access ==> Entitlements**.

4 - While using refresh token with WSO2 server, token generation fails.

While using refresh token with WSO2 server, token generation fails with the following error message:

"sailpoint.connector.ConnectorException: Token generation failed. Unable to generate access token. Response returned: {"error_description":"Client Authentication failed.", "error":"invalid_client"}"

Resolution: Add the following application configuration entry in the application debug page:

Key=oauth_body_attrs_to_exclude
Value=Comma separated oauth attributes
For example,
<entry key="oauth_body_attrs_to_exclude" value="client_id, client_secret" />

5 - Test Connection fails for Salesforce SCIM server with an error message

Test Connection fails for Salesforce SCIM server with the following error message:

com.unboundid.scim.sdk.UnauthorizedException: Unauthorized

Test Connection fails due to the **OAuth2** bearer token generated from other machine.

Resolution: For **OAuth2.0** type of authentication, use the OAuth2 bearer token generated on the machine where IdentityIQ is hosted.

Troubleshooting

Chapter 42: SailPoint Sybase Connector

The following topics are discussed in this chapter:

Overview.....	405
Supported features	406
Supported Managed Systems	406
Pre-requisites	406
Administrator permissions	407
Configuration parameters.....	408
Schema attributes	409
Account attributes	409
Group attributes.....	410
Provisioning policy attributes	410
Additional information	411
Identity and Entitlement representation	411
Performance Optimization	412
Troubleshooting.....	412

Overview

Sybase Adaptive Server Enterprise (ASE) is widely used database Server, mainly used to store data for different business modules like Sales, Production, Human Resource, Finance and Accounting. It requires the user to authenticate in order to connect to database to manipulate business data. It controls the users/roles logging in to Sybase ASE Managed System and performs other activities like processing transactions, writing logs, updating database files and so on.

A group is a means of organizing users, where as a role is usually a means of organizing rights. User roles are aggregated as Account Groups as it is widely used by the customers.

SailPoint Sybase Adaptive Server Enterprise Connector manages the following entities on Sybase Adaptive Server Enterprise:

- Login User
- Database User
- Roles
- Database Groups
- Aliases

Supported features

SailPoint Sybase Connector provides support for the following features:

- Account Management
 - Manages Sybase Users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
(Aliases, database_groups, roles)
- Account - Group Management
 - Supports multiple group functionality.
 - Manages Sybase server roles as group
 - Aggregation, Refresh Groups
 - Create, Update, Delete
 - Manages database groups as database_group
 - Aggregation, Refresh Groups

Note: If user adds the database group to login user, the database user is created with the name of login user on the respective database.

For example: Login user: JamesSmith. If user adds database group (master.public) to JamesSmith then the database user (master.JamesSmith) is created on the 'master' database.

Supported Managed Systems

Following versions of Sybase ASE are supported by the SailPoint Sybase ASE Connector:

- SAP ASE 16.0
- Sybase ASE 15.7

Pre-requisites

Sybase JDBC Driver is required for proper functioning of SailPoint Sybase ASE Connector. For example, jconn4.jar. This JDBC driver must be copied in the ..\identityiq\WEB-INF\lib directory.

Note: It is recommended that, on Managed System the server name and database name must not be the same.

Upgrade

Before upgrading IdentityIQ version from 6.4 to 7.3, user must execute the following:

- All the certifications must be executed.

After upgrade, user can view the certification history by right clicking on the specific entity and selecting the **View History** in the certification panel.

- The certifications related to 6.4 must be executed.
- Before aggregating the data, user must select the **Detect deleted accounts** option in task menu.

Administrator permissions

Respective administrator permissions must be provided for the following operations:

- Test Connection
- Aggregation
- Provisioning

Following script must be used to create minimum permission user and role. These permissions must be run by sa/administrator level login user.

Test Connection

1. Set the enable granular permissions configuration parameter to 1 as follows in the application debug page:
`sp_configure enable_granular_permissions, 1`

2. Login using administrator credentials and create a new user on managed system using the following command:

```
use [databaseName]
go
sp_addlogin [loginname], [password]
go
sp_adduser [username]
go
```

where **loginname** is login user on Sybase, **password** is the user provided password and **username** is the corresponding DatabaseUser.'

Note: Access must be provided to all databases.

Aggregation

1. Create role and assign it to login user:

```
create role [rolename]
go
grant role [rolename] to [loginname]
go
```

2. Activate role

```
exec sp_modifylogin '[loginname]', 'add default role', '[rolename]'
go
```

3. Assign the following privileges to role:

- Service account must have associated user on respective databases to fetch aliases:

```
GRANT select on master..sysloginroles to [rolename]
```

Configuration parameters

```
go
GRANT select on master..syssrvroles to [rolename]
go
GRANT select on master..syslogins to [rolename]
go
GRANT select on sysusers to [rolename]
go
GRANT select on master..sysdatabases to [rolename]
go
GRANT select on sysroles to [rolename]
go
GRANT select on sysalternates to [rolename]
go
```

Provisioning

User would require the following permissions to perform provisioning operations:

```
grant manage any login to [rolename]
go
grant manage server to [rolename]
go
grant manage roles to [rolename]
go
```

For database related operations [add/remove user, add/remove database group, add/drop alias]:

Prerequisite: Service account must have associated user on respective databases and dbo must have **manage database permissions** privilege.

```
use [databaseName]
go
grant manage any user to [username]
go
```

Configuration parameters

The following table lists the configuration parameters of SailPoint Sybase ASE Connector:

Parameters	Description
url*	A valid URL of Sybase ASE Connector which directly interacts with the managed system. In case of jconn2.jar, use the following url: jdbc:sybase:Tds:<host>[:<port>] For example, jdbc:sybase:Tds:ACHAUDHARI:5000
user*	Administrative Account to connect to Sybase ASE.

Parameters	Description
password*	Administrative Account Password.
driverClass*	The name of the Driver class supported by JDBC Type 4. For example, In case of <code>jconn2.jar</code> , use the following driverClass: <code>com.sybase.jdbc2.jdbc.SybDataSource</code>
Included Databases	List of comma separated database names to be included in the aggregation operation.
Excluded Databases	List of comma separated database names to be excluded in the aggregation operation. Note: If the Include Database parameter is populated, the Exclude Database parameter would be ignored.
Force Delete Login User	Deletes the login user. Options are: <ul style="list-style-type: none"> • Yes: Deletes the login user • (Default) No: Does not delete the login users which have the database users attached.

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attribute name	Description
name	Login user name.
server_user_id	Server User ID.
default_database	Default database. For example: master
default_language	Default language. For example: us_english
full_name	Full name of login user.
create_date	Date on which login user is created.
password_chg_date	Date on which password got changed.
last_login_date	Last login date of the user.
native_identity	An attribute which acts like a primary key during aggregation.
status	Status of login user: enable/disable
roles	Roles associated with login user.

Provisioning policy attributes

Attribute name	Description
database_groups	Database groups.
aliases	Aliases associated with login user.

Custom attributes

Perform the following to support the custom attributes:

- Click Add attribute button to add the custom attribute name in the schema.
- Add the following lines in the application debug page:

```
<entry key = "customAttribute" >
    <value>
        <List>
            <String>custom1</String>
            <String>custom2</String>
        </List>
    </value>
</entry>
```

Group attributes

The following table lists the group attributes:

Attribute name	Description
Group Object Type = Groups	
server_role_id	ID of the server Role.
native_identity	An attribute which acts like a primary key during aggregation.
name	Name of the Role.
password_chg_date	Date on which password got changed.
member_roles	Roles which are present under the hierarchy of the main role.
Group Object Type = Database Groups	
Group_name	Database Group Name.
native_identity	An attribute which acts like a primary key during aggregation.
Group_id	Database Group ID.

Provisioning policy attributes

This section lists the single provisioning policy attributes of SailPoint Sybase ASE Connector that allows to select the type of user, login, or group.

Attribute name	Description
Creating Group (User Role)	

Attribute name	Description
Role name*	Name of the role created.
Role Password	
Member Roles	
Creating User (Login User)	
Name*	Name of the Login User.
password*	Password for LoginUser.
Default database	Default database for Login User.
Default language	Default language.
Full name	Full name of the Login User.

Note: All the parameters marked with the * sign in the above table are the mandatory parameters.

Additional information

This section describes the additional information related to the Sybase Connector.

Identity and Entitlement representation

This section describes the Identity and Entitlement representation for SailPoint Sybase Adaptive Server Connector.

Identity representation

Account: The Account in Sybase ASE Connector is represented as follows:

- For Server Login it is represented as <Account name>
For example, login_name

Entitlement representation

- **Groups:** The Groups in Sybase ASE Connector are represented as follows:
For Application Role it is represented as <Group name>
- **Database Groups:** The Database Groups in Sybase ASE Connector are represented as follows:
For database groups it is represented as <database_name>. <Group name>
For example: **master.public**
- **Alias:** The Alias in Sybase ASE Connector are represented as follows:
For Alias it is represented as <database_name>. <Alias name>
For example: **master dbo**

Performance Optimization

During aggregation in Sybase ASE, if a performance degradation is observed, perform the following steps on the Sybase Server Management Console:

1. To view the number of open databases in the Sybase Server, use the following command:

```
sp_monitorconfig 'open'
```

This command displays the number of open databases under the **Number of Open Databases** column.

2. To increase the number of open databases, run the following command:

```
sp_configure "number of open databases", x ;
```

where x is the number of open databases that must be set.

3. To view the number of auxiliary scan descriptors set, use the following command:

```
sp_monitorconfig 'aux scan descriptors'
```

This command displays the value of **Num_free** and **Num_active** columns. The sum of the values of **Num_free** and **Num_active** columns represents the number of auxiliary scan descriptors.

4. To set the value of the auxiliary scan descriptors, use the following command:

```
sp_configure "aux scan descriptors", x;
```

where x is the number of the auxiliary scan descriptors that must be set.

The number of auxiliary scan descriptors must be more than three times the number of open databases. If this is not set properly than lower of the two values (number of open databases or auxiliary scan descriptors/3) is used during aggregation and this may affect the aggregation performance.

Note: The maximum number of open databases during aggregation is limited to 100. If the number of open databases is greater than 100 with appropriate auxiliary scan descriptors than only 100 open databases would be considered.

5. By default performance optimization is enabled. To use minimum permissions user must add the following entry key in the application debug page:

```
<entry key="isMinimumPermissionUser" value="true"/>
```

For performance optimization, it is required that the user has elevated roles.

Note: User having earlier set of permissions must add the following line in application debug page to use minimum permissions to complete aggregation successfully:

```
<entry key="isMinimumPermissionUser" value="true"/>
```

In case the following entry key is set to false, the Service account must have `mon_role` to complete the aggregation successfully:

```
<entry key="isMinimumPermissionUser" value="false"/>
```

Troubleshooting

1 - Aggregation fails

When a login user is created in Sybase and is granted permission only on some of the Databases present on the server and if aggregation task is run for that application, Aggregation fails as the user is not able to access other databases.

Resolution: In application Configuration page under the “Include Databases” section, provide the complete list of databases (comma separated list) for which the login user have accesses.

This completes the aggregation successfully, and only details of the users present in the list of included database will be fetched.

Troubleshooting

Chapter 43: SailPoint Tivoli Access Manager Connector

The following topics are discussed in this chapter:

Overview.....	415
Supported features	415
Supported Managed System.....	416
Pre-requisites	416
Configuration parameters.....	418
Schema attributes	418
Account attributes	418
Group attributes.....	419
Provisioning Policy attributes	419
Create account attributes	419
Create group attributes	420
Additional information	420
Unstructured Target Collector	420
Troubleshooting.....	421

Overview

SailPoint Tivoli Access Manager Connector manages Users and their Entitlements through groups present in Tivoli Access Manager system.

Supported features

SailPoint Tivoli Access Manager Connector supports the following features:

- Account Management
 - Manages Tivoli Access Manager Users as Accounts
 - Aggregation, Partitioning Aggregation, Refresh Accounts, Pass Through Authentication
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages Tivoli Access Manager Group as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete

Overview

- Permission Management
 - Application can be configured to read file permissions directly assigned to accounts and groups using Unstructured Target Collector.
 - The connector also supports automated revocation of the aggregated permissions for accounts and groups.

References

- “Unstructured Target Collector” on page 420.
- “Appendix B: Partitioning Aggregation”

Supported Managed System

SailPoint Tivoli Access Manager Connector supports IBM Security Access Manager for Web version 9.0 and 7.0.

Pre-requisites

Ensure that LDAP directory server associated with IBM Access Manager is already configured and functional. Following steps must be followed after successful integration of LDAP Directory Server and IBM Security Access Manager.

1. **Install PDJRTE:** Install PDRTE to configure IBM Security Access Manager Runtime for Java component to enable the Java application to use Security Access Manager security.
Perform the following steps to install PDJRTE on the LDAP Directory Server Machine:
 - a. Copy the license file (that is, **PDLIC.txt**) from **PDJRTE** directory to the root directory (for example, **C:** or **** in Unix).
 - b. Navigate to **PDJRTE** directory through command prompt available on LDAP Directory Server machine (for example, **C:\pdjrte-x.x.x-0\pdjrte\sbin**)
where **x.x.x** is 7.0.0 or 9.0.0
 - c. Open Command Prompt and execute the following command:
 - **For Windows:** **pdjrtecfg.bat -action config -interactive**
 - **For UNIX:** The file will be shell script
 - d. On UI dialogue box navigate to configure the java run-time environment for Access Manager. Select the valid JRE path and click **Next**.
 - e. Enter the existing policy server information (where your security access manager policy server is running [machine details]) as follows:
Host name: IP of the configured IBM Access Manager
Port: 7135(Default port)
Domain: Default (Recommended)
 - f. Enable Tivoli common directory logging (recommended to keep it on for troubleshooting).
 - g. Click **Finish**.
- This adds additional **.jar** file in the **\$JDK_HOME\jre\lib\ext** directory which is used by Tivoli Access Manager Connector.

For example,

- PD.jar
 - ibmjcefips.jar
 - ibmjcefw.jar
 - ibmjceprovider.jar
 - ibmjsseprovider2.jar
 - ibmpkcs.jar
 - local_policy.jar
 - US_export_policy.jar
2. On the LDAP Directory Server computer, use `com.tivoli.pd.jcfg.SvrSslCfg` command to generate the **config** and **keyfile** required to communicate with the IBM Security Access Manager. The file path of **config** file must be configured in application configuration.

For example:

```
>java com.tivoli.pd.jcfg.SvrSslCfg -action config -admin_id sec_master -admin_pwd
<password> -appsvr_id server1 -host <host> -port <port_number> -mode remote
-policysvr <host:7135:1> -authzsvr <host:7136:2> -domain default -cfg_file <path
of config file to be generated> -key_file <Path of key file to generate> -cfg_action
create
```

Integration of IdentityIQ and IBM Security Access Manager

Perform the following steps to integrate IdentityIQ with IBM Security Access Manager

1. Copy the following jar files which are generated from `jre/lib/ext` file in IBM LDAP Directory Server to `$JDK_HOME\jre\lib\ext` file on the IdentityIQ computer.
 - Ibmjcefips.jar
 - Ibmjcefw.jar
 - Ibmjceprovider.jar
 - ibmjsseprovider2.jar
 - ibmpkcs.jar
 - local_policy.jar
 - US_export_policy.jar
 - PD.jar
2. Copy the **Policy Director** directory from LDAP Directory Server computer to `$JAVA_HOME\jre` directory on IdentityIQ computer.
3. Copy the **config** and **keystore** files from LDAP server machine generated by `SvrSslCfg` command on IdentityIQ computer.

Note: It is recommended is to place the ‘config’ and ‘keystore’ files in the root directory as absolute path of the ‘config’ file would be referred and used in the application configuration.
4. Restart the Web Application Server (for example, Tomcat/WebSphere and so on).

Java Authentication and Authorization Service (JAAS)

Tivoli Access Manager Authorization APIs uses the Java Authentication and Authorization Service (JAAS). For supporting JAAS, perform the following changes in `java.security` file:

1. **Specify the login file location:** Point to the login configuration file from the `JAVA_HOME/jre/lib/security/java.security` file.

Configuration parameters

For example, a sample entry from the java.security file would be displayed as
login.config.url.1=file:\${java.home}/lib/security/login.pd

2. **Creating a login configuration file:** Create login.pd on the specified location. If it does not exist add an entry as follows:

```
pd {  
    com.tivoli.pd.jazn.PDLoginModule required;  
};
```

Configuration parameters

The following table lists the configuration parameters of Tivoli Access Manager Connector:

Parameters	Description
Admin Name*	Tivoli Access Manager administrator name.
Admin Password*	Password of the administrator.
Domain	The domain that is to be managed by the connector.
Configuration file URL*	A URL reference to configuration data file generated by com.tivoli.pd.jcfg.SvrSslCfg utility.

Schema attributes

This section describes the different schema attributes.

Note: All the attributes marked with * sign are the mandatory attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
userid	User ID of the user.
first_name*	The user's first name.
last_name*	The user's last name.
registryUID*	The account name stored in the user registry.
description	Text describing the user.
groups	The Access Manager groups that the user is a member of.
noPwdPolicy	Indicates whether a password policy is enforced.
ssoUser	Indicates whether the user has single sign-on abilities.
passwordvalid	The valid password.
accountValid	Indicates whether the account is disabled.

Attributes	Description
gsoWedCreds	gsoWedCreds.
gsoGroupCreds	Shows the list of gso group credentials assigned to a user. Will be displayed as <userid>:<gso group name>.
importFromRegistry	Indicates that the new user must be imported from registry server and not created in registry server. Note that the user must be present in the registry server.

Group attributes

The following table lists the group attributes:

Attributes	Description
GroupName*	Name of the group.
registryUID*	The group name stored in the user registry.
description	Text describing the Group.

Provisioning Policy attributes

This section lists the different policy attributes of Tivoli Access Manager Connector.

Create account attributes

The following table lists the provisioning policy attributes for Create Account:

Attributes	Description
UserID	User ID for the user.
First Name	The user's first name.
Last Name	The user's last name.
registryUID	The account name stored in the user registry.
Description	Text describing user.
Password	Password for the user.
Password Valid	Indicates whether the password will be expired.
Account Valid	Indicates whether the account is disabled.
GSO User	Indicates whether the user has single sign-on abilities.
GSO Web Credentials	List of gso credentials to be given to the new user. It should be as follows: <gso name>:<userid>:<gso-password>

Additional information

Attributes	Description
GSO Group Credentials	List of gso group credentials to be given to the new user. It should be as follows: <code><gso group name>:<userid >:<gso-password></code>
ImportFromRegistry	Indicates whether the user is to be imported from registry server.
No Password Policy	Indicates whether a password policy is enforced.

Configuration settings

Field separator for **GSO Web Credentials** and **GSO Group Credentials** can be defined in application template using the debug page. Default delimiter is ':'.

For example, entry: `<entry key="gso_field_seperator" value="#" />`

The above example will set the field separator for mentioned attributes to '#'.

Note: The delimiter selected should not be a part of any of the subfields in the mentioned attribute. For above example character '#' should not be part of **gso name** or **userID** or **gso password** on **GSO Web Credentials** attribute.

Create group attributes

The following table lists the provisioning policy attributes for Create Group:

Attributes	Description
Name	Name of the group.
registryUID	The group name stored in the user registry.
Last Name	The user's last name.
Description	Text describing the user.

Additional information

This section describes the additional information related to the Tivoli Access Manager Connector.

Unstructured Target Collector

Tivoli Access Manager uses a data structure which requires the configuration in the **Unstructured Targets** tab to collect targeted data and correlate it with **account identityAttribute** for Accounts and **group identityAttribute** for AccountGroups. For more information on the **Unstructured Targets** tab, see “Unstructured Targets Tab” section of the *SailPoint User’s Guide*.

The Unstructured Targets functionality will be enabled for Tivoli Access Manager connector if **UNSTRUCTURED_TARGETS** feature string is present in the application.

Tivoli Access Manager Target Collector supports aggregation of Access Control List (ACL). Access permissions on ACL will be correlated to Users and Groups.

Following is the configuration parameter in **Unstructured Targets** tab for Tivoli Access Manager Connector:

Attribute	Description
IBM Tivoli Access Manager Application Name	Name of the IBM Tivoli Access Manager application.

Provisioning related parameters

Select the following settings for provisioning to Tivoli Access Manager:

- **Override Default Provisioning:** Overrides the default provisioning action for the collector.
- **Provisioning Action:** The overriding provisioning action for the collector.

Troubleshooting

1 - Connector not aggregating all accounts

When you have the LDAP user registry setup for Tivoli Access Manager, the Connector might not aggregate all accounts from Tivoli Access Manager.

Resolution: The Maximum search results is controlled by the following parameters:

- The **max-search-size** stanza entry in the [ldap] stanza of the **ldap.conf** configuration file:
To indicate that there is no limit, set the stanza entry `max_search_size = 0`.
For example: `max-search-size = 0`
Note: **Restart the Tivoli Access Manager servers for the required changes.**
- The **ibm-slapdSizeLimit** parameter in the Tivoli Directory Server server **slapd32.conf** or **ibmslapd.conf** configuration file:
To indicate there is no limit, set the size limit to 0.
For example: `ibm-slapdSizeLimit = 0`
Note: **This parameter affects all LDAP searches.**

Note: **Ensure that both parameters are set to value greater than or equal to the total number of records in Tivoli Access Manager.**

Troubleshooting

Chapter 44: SailPoint Top Secret Connector

The following topics are discussed in this chapter:

Overview.....	423
Supported features	423
Configuration parameters.....	424
Schema Attributes	425

Overview

The SailPoint Top Secret Connector is a *read only* connector developed to read the TSSCFILE export.

Note: The Top Secret Full Connector supports the provisioning operations. For more information, see *SailPoint IdentityIQ Connector for CA-Top Secret Administration Guide*.

Supported features

SailPoint Top Secret Connector supports the following features:

- Account Management
 - Manages TOP SECRET Users as Accounts
 - Aggregation, Refresh Accounts, Discover Schema
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages TOP SECRET Groups as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete
- Permission Management
 - Application reads permissions directly assigned to accounts and groups as direct permissions during account and group aggregation.
 - The connector does not support automated revocation of the aggregated permissions and creates work item for such requests.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The Top Secret connector uses the following connection attributes:

Table 1—Top Secret Connector - Configuration parameters

Parameters	Description
filetransport	local, ftp, scp
host	The host of the server to which you are connecting.
transportUser	The user to use with ftp and scp. Not valid with local.
transportUserPassword	The password to use with of ftp and scp. Not valid with local.
file	The fully qualified path to the file.
fileEncoding	Specify the file encoding to be used by the connector. Valid values for this attribute can be found at: http://www.iana.org/assignments/character-sets If this field is empty, the default encoding (the value of file.encoding specified by the jvm) is used.
mapToResourceObjectRule	Rule that is called to override the transformation of the data from the Map<String, String> form into a ResourceObject.
filterString	Filter lines that match this string.
filterEmptyRecords	If activated, records that have no data are filtered.
preIterativeRule	The pre-iterate rule will check for a specially named Configuration object that will hold the last run statistics that can be compared against the current values. This rule is called after the file has been transferred, but before iteration over the objects in the file is started. For validation this rule can use the existing statistics stored by the postIterationRule during the last aggregation. The rule can compare the stored values with the new values to check for problems
postIterativeRule	The post-iterate rule can store away the configuration object and rename/delete the file if desired. This rule is called after aggregation has completed and ALL objects have been iterated.
accountTypes	The type of account use to connect to the server. The default value is USER, but additional values can be specified.

Table 1—Top Secret Connector - Configuration parameters

Parameters	Description
groupTypes	The group type of the connector. The default values are GROUP ACID and PROFILE ACID.
Top Secret Attribute Customization Rule	The rule used to extend the parsing capabilities to customer records or redefine existing record configurations. TopSecret records hold a record identifier and all of the fields that are part of that record. This rule use the TopSecretRecord and TopSecretField classes to work with that information.

Schema Attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group. Account objects are used when building identities Link objects. The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 2—Top Secret Connector - Account Attributes

Attribute	Description
XAUTH	The authorized level at which the user can access the resource.
VMMDISK	The VM minidisks owned by the user.
ACTION	Specifies which action(s) CA-Top Secret will take when access to a resource is attempted.
LOCK TIME(MINUTES)	The time interval before unattended or inactive terminals are locked.
LOCK TIME FACILITY	The lock time for all terminals connected to the specified facility.
LANGUAGE PREFERENCE	The language preference code the user.
VOLSER(OWNED)	The volumes to which the user has access.
NAME	Identifies the ACID name. Names can be up to 32 characters in length, must be surrounded by single quotes if embedded with blanks, and can use letters, numbers, and special characters.
SITRAN	Specifies which CICS transaction CA-Top Secret automatically executes after an ACID successfully signs on to a facility. Note: If a SITRAN is added to an ACID that already has a CICS transaction defined, the transaction is replaced.
HOME	Defines the initial directory pathname. This is the initial directory used when a user enters the OMVS command or enters the ISPF shell. The HOME keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If HOME isn't defined, OpenEdition MVS sets the initial directory for the user to the root directory. HOME is optional.
XA ACID	XAUTH Resource Class Name

Schema Attributes

Table 2—Top Secret Connector - Account Attributes (Continued)

Attribute	Description
MULTIPW	Used to assign or remove multiple password attributes, which means ACIDs need a different password to access each facility.
NOADSP	Used to prevent data sets, created by an ACID, from being automatically secured by MVS by setting the RACF bit. NOADSP is used to define an ACID that will be used to create data sets that cannot be automatically protected by CA-Top Secret.
AUDIT	Used to allow an audit of ACID activity.
NOPWCHG	To prevent ACIDs from changing passwords at either signon or initiation.
OIDCARD	Used to support the physical identification of users through operator identification cards.
TRACE	Used to activate a diagnostic trace on all ACID activity (initiations, resource access, violations, user's security mode, etc.)
SUSPEND	Used to prevent ACIDs from accessing the system when a violation occurs.
MRO	Used to support the use of the multi-region option.
CONSOLE	Used to grant or remove an ACID's ability to modify control options. For VM, options are modified via the TSS MODIFY command only. With VSE and OS/390, options are modified at the O/S console or via the TSS MODIFY command function.
GAP	Used to specify that a profile will become, or will cease to be, globally administrable.
DUFXTR	Used to add or remove the DUFXTR attribute to an ACID. DUFXTR enables an ACID to use a RACROUTE REQUEST=AUTH (RACHECK) macro or the CA-Top Secret Application Interface to extract installation data (INSTDATA) or field data from a Security Record. DUFXTR is a component of the CA-Top Secret Dynamic Update Facility (DUF).
DUFUPD	Used to add or remove the DUFUPD attribute to an ACID. DUFUPD enables an ACID to use the CA-Top Secret Application Interface to update the installation data (INSTDATA) or field data from a Security Record. DUFUPD is a component of the CA-Top Secret Dynamic Update Facility (DUF).
TSOMPW	Used to support multiple TSO UADS passwords, on a user-by-user basis.
NOATS	Used to prevent an ACID in CICS and CA-IDMS from signing on via ATS (Automatic Terminal Signon).
ACEDEFAU	
ASUSPEND	Used to remove the suspension of an ACID that was suspended for administrative reasons.
WHO HAS RESOURCE	
PROFILE ACID	Used to assign profiles to an ACID.
PASSWORD	Used to assign a password, along with values that control its use, to a previously defined ACID.

Table 2—Top Secret Connector - Account Attributes (Continued)

Attribute	Description
PASSWORD EXPIRES DATE	The date, in string format, that the password expires.
PASSWORD INTERVAL	The interval in which the password must be changed.
PASSWORD FACILITY	The facility name applied to ACIDs with the multipw attribute.
OPIDENT	Used to assign or remove a CICS operator identification value that is equal to the ACID's OPIDENT entry in the CICS SNT (Signon Table). The OPIDENT value is placed into the ACID's TCT at signon.
OPPRTY	Used to assign or remove a CICS operator priority from the associated ACID. The OPPRTY value is placed into the ACID's TCT (Terminal Control Table) at signon.
PROGRAM	Used to secure system programs and utilities.
WHOHAS ADMIN	Used to determine who has administrative authority on the application.
ACIDS2	
SOURCES	
TSOLPROC	<p>Used to provide a default procedure to be used for TSO logon.</p> <p>The one- to eight-character logon procedure name. Procedure names are also TSO-related resources and the user must be permitted to any procedure name with which he attempts to log on.</p>
DIV ACID	Specifies the Division ACID to which the ACID is attached.
DIV NAME	The name assigned to the ACID within the zone.
SUSPENDED	The date, in string format, that the suspension ends.
WHOHAS XAUTH	A list of resources that may be accessed by the ACID shown in the command, the level at which the ACID may access the resource, and the owner of the resource.
TSOUNIT	<p>The default unit name to be used for dynamic allocations under TSO.</p> <p>The one- to eight-character unit (device) name for dynamically allocated data sets. The name must be a defined generic unit class name at the installation. This field is not alterable by the user at logon and is not required for successful logon.</p>
PHYSKEY	PHYSKEY (physical security key) supports external authentication devices.
ACID WITHIN DEPT/DIV/ZONE	Used to specify department, division, and zone to include.
DATE CREATED	The date on which the ACID was created.
DATE LAST MODIFIED	The date on which the ACID was last modified.
TIME LAST MODIFIED	The time at which the ACID was last modified.
ROOM NUMBER	The room number assigned to the ACID.
MISC2	Used to give, or to remove, a CA-Top Secret administrator's authority to perform one or more administrative functions.
ACCESSLEVELS	

Schema Attributes

Table 2—Top Secret Connector - Account Attributes (Continued)

Attribute	Description
MISC8	Used to give, or to remove, a CA-Top Secret administrator's authority to list the contents of the RDT, FDT or STC or to use the ASUSPEND administrative function.
XA MINIDISK	The minidisk authorization information for the ACID.
SCOPE	Used to give CA-Top Secret administrators the authority, or to remove their authority, to assign the SCOPE of an LSCA.
DIGITAL CERT NAME	The name of the digital certification.
DEPARTMENT	The Department ACID to which the ACID is attached.
DLFTGRP	The default group for the ACID.
WHO OWNS RESOURCE	The resources owned by the ACID.
TSOOPT	The default options that a TSO user may specify at logon.
WANAME	The person to whom SYSOUT information should be delivered for this ACID.
XA	
SYSID	The SYSID (which is actually the SMFID) that the authorizations for the ACID apply to.
BUILDING	The building in which the ACID is located.
TSOCOMMAND	Default commands issued upon login of the ACID.
DIGITAL CERT STARTS	The date, in string format, that the digital certification starts.
XAUTH LIBRARY	The libraries for which the ACID has authority.
WHOHAS FACILITY	Returns facility information for the ACID.
RESOURCE CLASS NAME	The resource class for which the ACID has authority.
FCT/PREFIX(OWNED)	
FACILITIES	The facilities to which the user has access.
TSOHCLASS	The default hold class for TSO generated JCL for TSO the user.
DIGITAL CERT EXPIRES	The date, in string format, when the digital certification expires.
ZONE ACID	The Zone ACID to which the ACID is attached.
ZONE NAME	The name assigned to the ACID within the zone.
ADDRESS1	Physical address for the ACID.
XAUTHDAYS	Days of the week the ACID is authorized on this application.
ACID TYPE	The ACID type, for example zone, division, or department.
ACID SIZE	The size of the ACID.
RESTRICT	
ADDRESS4	Alternative physical address for the ACID.
NODSNCHK	To specify that no data set name check will be performed. That is, CA-Top Secret will bypass all data set access security checks. All data set access will be audited.

Table 2—Top Secret Connector - Account Attributes (Continued)

Attribute	Description
NOVOLCHECK	
NOLCFCHK	Used to allow an ACID to execute any command or transaction for all facilities, regardless of LCF (Limited Command Facility) restrictions. No auditing is done.
NOSUBCHK	Used to allow an ACID to bypass alternate ACID usage as well as all job submission security checking. Thus, associated ACIDs may submit all jobs regardless of the (derived) ACID on the job card being submitted.
NORESCHK	Used to allow an ACID to bypass security checking, including auditing, for all owned resources except data sets and volumes.
NOVMDCHK	Used to allow an ACID to bypass all checking for minidisk links. All links will be audited. NOVMDCHK is intended only to be applied to special products such as DASD space managers, which may link to many minidisks.
NOSUSPEND	Used to allow an ACID to bypass suspension due to violations.
TSODEST	The default destination identifier for TSO generated JCL for TSO users.
XA VOLUMN	
TSODEFPRFG	The default TSO performance group.
RESOURCE CLASS NAME2	An additional resource class for which the ACID has authority.
MISC1	A CA-Top Secret administrator's authority to perform one or more administrative functions.
GID	IDs of the groups to which the ACID belongs.
TSOUDATA	The site-defined data fields for a TSO user.
ACCESSLEVELS2	
DSN/PREFIX(OWNED)	
TSOMSIZE	The maximum region size (in kilobytes) that the TSO user can specify at logon.
EXPIRES	The date on which the ACID expires.
TSOSCLASS	The default SYSOUT class for TSO generated JCL for the TSO users.
XAUTH FAC	
DEPT ACID	The Department ACID to which the ACID is attached.
DEPT NAME	The name assigned to the ACID within the department.
DATE LAST USED	Date the ACID was last used.
TIME LAST USED	Time the ACID was last used.
CPU	Name of the CPU on which the ACID was used.
FAC	System facilities defined to CA-Top Secret: BATCH, STC, TSO, IMS, CICS, NCCF, CA-Roscoe, WYLBUR, or any installation-defined facility.
COUNT	

Schema Attributes

Table 2—Top Secret Connector - Account Attributes (Continued)

Attribute	Description
SEGMENT	Used to allow TSS administrators to list data about fields in a specific segment.
RESOURCES	
TSOJCLASS	The default job class for TSO generated job cards from TSO users.
ADMIN BY	
XAUTH MODE	
TSOLACCT	The default account number used for TSO logon.
TSOLSIZE	The default region size (in kilobytes) for TSO.
LISTDATA	
OMVSPGM	The user's OpenEdition MVS shell program. This is the first program started when the OMVS command is entered, or when an OpenEdition MVS batch job is started using the BPXBATCH program.
SMSSTOR	The default storage keyword for the ACID.
UID	The unique user ID for the ACID.
ADDRESS3	Alternative physical address for the ACID.
XAUTH PRIVPGM	The program pathing, if privileged program is in use.
TIME ZONE	The time zone attached to the ACID.
MASTER FACILITY	
LCF FACILITY	
FACILITY NAME	
FACILITY UNTIL DATE	
INSTDATA	Used to record or remove information about an ACID. Up to 255 characters of information about an associated ACID may be used for convenient record keeping, or for interrogation by a user-written Installation Exit.
ADDRESS2	Alternative physical address for the ACID.
GROUP ACID	The Group ACID to which the ACID is attached.
TSOMCLASS	The default message class for TSO generated JCL for TSO users.
MISC9	To give, or to remove, a TSS administrator's authority to perform one or more high-level administrative functions.

Table 3—Top Secret Connector - Group Attributes

Attribute	Description
XAUTH	The authorized level at which the user can access the resource.
VMMDISK	The VM minidisks owned by the user.
ACTION	Specifies which action(s) CA-Top Secret will take when access to a resource is attempted.

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
LOCK TIME(MINUTES)	The time interval before unattended or inactive terminals are locked.
LOCK TIME FACILITY	The lock time for all terminals connected to the specified facility.
LANGUAGE PREFERENCE	The language preference code the user.
VOLSER(OWNED)	The volumes to which the user has access.
NAME	Identifies the ACID name. Names can be up to 32 characters in length, must be surrounded by single quotes if embedded with blanks, and can use letters, numbers, and special characters.
SITRAN	Specifies which CICS transaction CA-Top Secret automatically executes after an ACID successfully signs on to a facility. Note: If a SITRAN is added to an ACID that already has a CICS transaction defined, the transaction is replaced.
HOME	Defines the initial directory pathname. This is the initial directory used when a user enters the OMVS command or enters the ISPF shell. The HOME keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If HOME isn't defined, OpenEdition MVS sets the initial directory for the user to the root directory. HOME is optional.
XA ACID	XAUTH Resource Class Name
MULTIPW	Used to assign or remove multiple password attributes, which means ACIDs need a different password to access each facility.
NOADSP	Used to prevent data sets, created by an ACID, from being automatically secured by MVS by setting the RACF bit. NOADSP is used to define an ACID that will be used to create data sets that cannot be automatically protected by CA-Top Secret.
AUDIT	Used to allow an audit of ACID activity.
NOPWCHG	To prevent ACIDs from changing passwords at either signon or initiation.
OIDCARD	Used to support the physical identification of users through operator identification cards.
TRACE	Used to activate a diagnostic trace on all ACID activity (initiations, resource access, violations, user's security mode, etc.)
SUSPEND	Used to prevent ACIDs from accessing the system when a violation occurs.
MRO	Used to support the use of the multi-region option.
CONSOLE	Used to grant or remove an ACID's ability to modify control options. For VM, options are modified via the TSS MODIFY command only. With VSE and OS/390, options are modified at the O/S console or via the TSS MODIFY command function.
GAP	Used to specify that a profile will become, or will cease to be, globally administrable.

Schema Attributes

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
DUFXTR	Used to add or remove the DUFXTR attribute to an ACID. DUFXTR enables an ACID to use a RACROUTE REQUEST=AUTH (RACHECK) macro or the CA-Top Secret Application Interface to extract installation data (INSTDATA) or field data from a Security Record. DUFXTR is a component of the CA-Top Secret Dynamic Update Facility (DUF).
DUFUPD	Used to add or remove the DUFUPD attribute to an ACID. DUFUPD enables an ACID to use the CA-Top Secret Application Interface to update the installation data (INSTDATA) or field data from a Security Record. DUFUPD is a component of the CA-Top Secret Dynamic Update Facility (DUF).
TSOMPW	Used to support multiple TSO UADS passwords, on a user-by-user basis.
NOATS	Used to prevent an ACID in CICS and CA-IDMS from signing on via ATS (Automatic Terminal Signon).
ACEDEFAU	
ASUSPEND	Used to remove the suspension of an ACID that was suspended for administrative reasons.
XA DATASET	
WHO HAS RESOURCE	
PROFILE ACID	Used to assign profiles to an ACID.
PASSWORD	Used to assign a password, along with values that control its use, to a previously defined ACID.
PASSWORD EXPIRES DATE	The date, in string format, that the password expires.
PASSWORD INTERVAL	The interval in which the password must be changed.
PASSWORD FACILITY	The facility name applied to ACIDs with the multipw attribute.
OPIDENT	Used to assign or remove a CICS operator identification value that is equal to the ACID's OPIDENT entry in the CICS SNT (Signon Table). The OPIDENT value is placed into the ACID's TCT at signon.
OPPRTY	Used to assign or remove a CICS operator priority from the associated ACID. The OPPRTY value is placed into the ACID's TCT (Terminal Control Table) at signon.
PROGRAM	Used to secure system programs and utilities.
WHOHAS ADMIN	Used to determine who has administrative authority on the application.
ACIDS2	
SOURCES	
TSOLPROC	Used to provide a default procedure to be used for TSO logon. The one- to eight-character logon procedure name. Procedure names are also TSO-related resources and the user must be permitted to any procedure name with which he attempts to log on.
DIV ACID	Specifies the Division ACID to which the ACID is attached.

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
DIV NAME	The name assigned to the ACID within the zone.
SUSPENDED	The date, in string format, that the suspension ends.
WHOHAS XAUTH	A list of resources that may be accessed by the ACID shown in the command, the level at which the ACID may access the resource, and the owner of the resource.
TSOUNIT	The default unit name to be used for dynamic allocations under TSO. The one- to eight-character unit (device) name for dynamically allocated data sets. The name must be a defined generic unit class name at the installation. This field is not alterable by the user at logon and is not required for successful logon.
PHYSKEY	PHYSKEY (physical security key) supports external authentication devices.
ACID WITHIN DEPT/DIV/ZONE	Used to specify department, division, and zone to include.
DATE CREATED	The date on which the ACID was created.
DATE LAST MODIFIED	The date on which the ACID was last modified.
TIME LAST MODIFIED	The time at which the ACID was last modified.
ROOM NUMBER	The room number assigned to the ACID.
MISC2	Used to give, or to remove, a CA-Top Secret administrator's authority to perform one or more administrative functions.
ACCESSLEVELS	
MISC8	Used to give, or to remove, a CA-Top Secret administrator's authority to list the contents of the RDT, FDT or STC or to use the ASUSPEND administrative function.
XA MINIDISK	The minidisk authorization information for the ACID.
SCOPE	Used to give CA-Top Secret administrators the authority, or to remove their authority, to assign the SCOPE of an LSCA.
DIGITAL CERT NAME	The name of the digital certification.
DEPARTMENT	The Department ACID to which the ACID is attached.
DLFTGRP	The default group for the ACID.
WHO OWNS RESOURCE	The resources owned by the ACID.
TSOOPT	The default options that a TSO user may specify at logon.
WANAME	The person to whom SYSOUT information should be delivered for this ACID.
XA	
SYSID	The SYSID (which is actually the SMFID) that the authorizations for the ACID apply to.
BUILDING	The building in which the ACID is located.
TSOCOMMAND	Default commands issued upon login of the ACID.
DIGITAL CERT STARTS	The date, in string format, that the digital certification starts.

Schema Attributes

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
XAUTH LIBRARY	The libraries for which the ACID has authority.
WHOHAS FACILITY	Returns facility information for the ACID.
RESOURCE CLASS NAME	The resource class for which the ACID has authority.
FCT/PREFIX(OWNED)	
FACILITIES	The facilities to which the user has access.
TSOCLASS	The default hold class for TSO generated JCL for TSO the user.
DIGITAL CERT EXPIRES	The date, in string format, when the digital certification expires.
ZONE ACID	The Zone ACID to which the ACID is attached.
ZONE NAME	The name assigned to the ACID within the zone.
ADDRESS1	Physical address for the ACID.
XAUTHDAYS	Days of the week the ACID is authorized on this application.
ACID TYPE	The ACID type, for example zone, division, or department.
ACID SIZE	The size of the ACID.
RESTRICT	
ADDRESS4	Alternative physical address for the ACID.
NODSNCHK	To specify that no data set name check will be performed. That is, CA-Top Secret will bypass all data set access security checks. All data set access will be audited.
NOVOLCHECK	
NOLCFCHK	Used to allow an ACID to execute any command or transaction for all facilities, regardless of LCF (Limited Command Facility) restrictions. No auditing is done.
NOSUBCHK	Used to allow an ACID to bypass alternate ACID usage as well as all job submission security checking. Thus, associated ACIDs may submit all jobs regardless of the (derived) ACID on the job card being submitted.
NORESCHK	Used to allow an ACID to bypass security checking, including auditing, for all owned resources except data sets and volumes.
NOVMDCHK	Used to allow an ACID to bypass all checking for minidisk links. All links will be audited. NOVMDCHK is intended only to be applied to special products such as DASD space managers, which may link to many minidisks.
NOSUSPEND	Used to allow an ACID to bypass suspension due to violations.
TSODEST	The default destination identifier for TSO generated JCL for TSO users.
XA VOLUMN	
TSODEFPRFG	The default TSO performance group.
RESOURCE CLASS NAME2	An additional resource class for which the ACID has authority.

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
MISC1	A CA-Top Secret administrator's authority to perform one or more administrative functions.
GID	IDs of the groups to which the ACID belongs.
TSOUDATA	The site-defined data fields for a TSO user.
ACCESSLEVELS2	
DSN/PREFIX(OWNED)	
TSOMSIZE	The maximum region size (in kilobytes) that the TSO user can specify at logon.
EXPIRES	The date on which the ACID expires.
TSOSCLASS	The default SYSOUT class for TSO generated JCL for the TSO users.
XAUTH FAC	
DEPT ACID	The Department ACID to which the ACID is attached.
DEPT NAME	The name assigned to the ACID within the department.
DATE LAST USED	Date the ACID was last used.
TIME LAST USED	Time the ACID was last used.
CPU	Name of the CPU on which the ACID was used.
FAC	System facilities defined to CA-Top Secret: BATCH, STC, TSO, IMS, CICS, NCCF, CA-Roscoe, WYLBUR, or any installation-defined facility.
COUNT	
SEGMENT	Used to allow TSS administrators to list data about fields in a specific segment.
RESOURCES	
TSOJCLASS	The default job class for TSO generated job cards from TSO users.
ADMIN BY	
XAUTH MODE	
TSOLACCT	The default account number used for TSO logon.
TSOLSIZE	The default region size (in kilobytes) for TSO.
LISTDATA	
OMVSPGM	The user's OpenEdition MVS shell program. This is the first program started when the OMVS command is entered, or when an OpenEdition MVS batch job is started using the BPXBATCH program.
SMSSTOR	The default storage keyword for the ACID.
UID	The unique user ID for the ACID.
ADDRESS3	Alternative physical address for the ACID.
XAUTH PRIVPGM	The program pathing, if privileged program is in use.
TIME ZONE	The time zone attached to the ACID.

Schema Attributes

Table 3—Top Secret Connector - Group Attributes (Continued)

Attribute	Description
MASTER FACILITY	
LCF FACILITY	
FACILITY NAME	
FACILITY UNTIL DATE	
INSTDATA	Used to record or remove information about an ACID. Up to 255 characters of information about an associated ACID may be used for convenient record keeping, or for interrogation by a user-written Installation Exit.
ADDRESS2	Alternative physical address for the ACID.
GROUP ACID	The Group ACID to which the ACID is attached.
TSOMCLASS	The default message class for TSO generated JCL for TSO users.
MISC9	To give, or to remove, a TSS administrator's authority to perform one or more high-level administrative functions.

Chapter 45: SailPoint UNIX Connector

The following topics are discussed in this chapter:

Overview.....	437
Supported features	437
Configuration parameters.....	437
Schema attributes	438

Overview

The SailPoint UNIX Connector is a *read only* connector developed to read and parse the **passwd** and **group** file from UNIX servers to build identities and groups. Since this connector is file based, there is some synergy between the UNIX and Delimited File connector.

Depending on your application configuration, the SailPoint UNIX Connector determines login success by authenticating using the ftp or scp service with the provided login credentials. Therefore, the **passwdfile** attribute of the UNIX application must be the same password file used by the system for authentication. This password file is typically `/etc/passwd`, but might be different in an environment where NIS is used.

Supported features

SailPoint UNIX Connector supports the following features:

- Account Management
 - Manages UNIX Users as Accounts
 - Aggregation
- Account - Group Management
 - Manages UNIX Groups as Account-Groups
 - Aggregation

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The UNIX Database connector uses the following connection attributes:

Table 1—UNIX Connector - Configuration parameters

Parameters	Description
host	The host of the server to which you are connecting.

Schema attributes

Table 1—UNIX Connector - Configuration parameters

Parameters	Description
filetransport	local, ftp, scp
transportUser	The user to use with ftp and scp. Not valid with local.
transportUserPassword	The password to use with of ftp and scp. Not valid with local.
passwdfile	The fully qualified path to the <code>passwd</code> file.
groupfile	The fully qualified path to the <code>group</code> file.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, account and group.

Account attributes

Account objects are used when building identities Link objects.

Table 2—UNIX Connector - Account Attributes

Attribute	Description
homedir	The path to the user's home directory on the host system. The home directory is the directory in which the user keeps personal files such as initialization files and mail.
shell	The shell, or program, preferred by the user for accessing the command line interface.
info	The information pertaining to the user.
groups	The groups to which the user belongs.

Group attributes

The group schema is used when building AccountGroup objects which are used to hold entitlements shared across identities.

Table 3—UNIX Connector - Group Attributes

Attribute	Description
groupname	A name associated with the group. The group names are listed in the first comma-delimited field of the groups text file.
groupid	A group id used to identify the group. The group ids are listed in the third comma-delimited field of the groups text file.

Table 3—UNIX Connector - Group Attributes

Attribute	Description
members	A comma-delimited list of users who are members of the group. Members are listed in the forth comma-delimited field of the groups text file.

Schema attributes

Chapter 46: SailPoint Web Services Connector

The following topics are discussed in this chapter:

Overview.....	441
Supported features	441
Supported Managed Systems	442
Pre-requisites	442
Administrator permissions	442
Configuration parameters.....	442
Schema attributes	451
Additional information	451
Upgrade considerations.....	451
Web Services Class used in Before/After operation Rule.....	452
Use of Quotes	458
Pagination	459
Configuration for Response.....	464
Configuration for Multiple endpoints.....	466
Configuring Multiple Entitlement Requests.....	467
Other Operations	467

Overview

The Web Services Connector is developed with an idea where any web service supported target system can be configured. This connector would be able to perform read and write operation on the target system using the respective target system web services.

Note: Web Services Connector supports JSON and XML for read and write.

Supported features

SailPoint Web Services Connector supports the following features:

- Account Management
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Aggregation, Refresh Groups

Additional supported feature

SailPoint Web Services Connector provides additional support for pagination.

Configuration parameters

For more information on embedding pagination support in Web Service Connector, see “[Pagination](#)” on page 459.

Supported Managed Systems

SailPoint Web Services Connector supports web services with JSON/XML response.

Pre-requisites

Web Services must be accessible.

Administrator permissions

The user/administrator must have the required permissions to call the web services API of the target system.

Configuration parameters

This section provides the following type of configuration parameters of SailPoint Web Services Connector:

- Basic configuration parameters
- Operation specific configuration parameters

(General Settings) Basic configuration parameters

The following table lists the basic configuration parameters of SailPoint Web Services Connector:

Parameters	Description
Base URL*	The base URL to connect to the web service target system.
Authentication Method*	Authentication method that is supported by the target system <ul style="list-style-type: none">• OAuth2• API Token• Basic Authentication• No Authentication <p>Note: SOAP Web Services supports only Basic Authentication method.</p>
Schema Attribute for Account Enable status	Attribute name and value required to be provided to check the Enable status. For example, status=Active
Request Timeout (In Seconds)	Request Timeout Value in seconds.

Parameters	Description
<i>Applicable if Authentication Method is selected as OAuth2</i>	
Grant Type*	Select the type of Grant: <ul style="list-style-type: none"> • Refresh Token • JWT • Client Credentials
Client Id*	(Optional for JWT) Client Id for OAuth2 authentication.
Client Secret*	(Optional for JWT) Client secret for OAuth2 authentication.
Token URL*	URL for generating access token.
Refresh Token*	(Applicable if Grant Type is selected as Refresh Token) A valid refresh token for grant type authentication.
Private Key*	(Applicable if Grant Type is selected as JWT) The private key to be used to sign the JWT.
Private Key Password*	(Applicable if Grant Type is selected as JWT) Password for the provided private key.
<i>Applicable if Authentication Method is selected as API Token</i>	
API Token*	Enter the API token specific to the Managed System.
<i>Applicable if Authentication Method is selected as Basic</i>	
Username*	Username that holds permission to execute the Web Service.
Password*	Password of the user name.

Note: Attributes marked with * sign are the mandatory attributes.

Additional configuration parameters

Add the following attributes in the application debug page:

Attributes	Description
throwProvBeforeRuleException	<p>During Provisioning, GetObject and Test Connection if an exception is thrown from WebServiceBeforeOperationRule, then Provisioning would fail. Hence to dispose of the exception in the log file and proceed with provisioning, set the value of throwProvBeforeRuleException to false in the application debug page as follows:</p> <pre><entry key="throwProvBeforeRuleException"> <value> <Boolean>true</Boolean> </value> </entry></pre> <p>Note: The default value of ‘throwProvBeforeRuleException’ flag is set to true for new Web Services application and false for existing application (before upgrading to IdentityIQ version 7.3). The ‘throwProvBeforeRuleException’ flag can be set for all operations except Account and Group aggregation.</p>

Configuration parameters

Attributes	Description
throwBeforeAfterRuleException	<p>During aggregation if an exception is displayed from WebServiceBeforeOperationRule or WebServiceAfterOperationRule, then aggregation continues and completes successfully.</p> <p>Set the value of the following flag to true in the application debug page to terminate the aggregation by displaying an error message:</p> <pre>throwBeforeAfterRuleException</pre> <p>Note: This flag can be set only for Account and Group aggregation.</p> <p>The default value of the <code>throwBeforeAfterRuleException</code> flag is set to false.</p>
oauth_headers	<p>(Applicable if Authentication Method is selected as OAuth2)</p> <ul style="list-style-type: none"> To have customized headers as a part of the access token generation request, add the oauth_headers parameter to the application debug page as follows: <pre><entry key="oauth_headers"> <value> <Map> <entry key="Content-Type" value="application/x-www-form-urlencoded" /> </Map> </value> </entry></pre> <p>Note: Web Services connector now uses access token configured in the application as authorization header for each endpoint, users would no longer require to specify the authorization header for each endpoint. If authorization is provided at endpoint level then it would precede over the access token.</p> <p>SailPoint recommends to provide authorization header suffix in the access token provided. For example, Bearer <Access Token>. If no prefix is provided, then connector would by default provide Bearer as Authorization header prefix.</p> <ul style="list-style-type: none"> To send additional headers for token generation, add the oauth_headers parameter to the application debug page as follows: <pre><entry key="oauth_headers"> <value> <Map> <entry key="customHeaderKey" value="customHeaderValue" /> </Map> </value> </entry></pre>

Attributes	Description
isQuotesEnabled	<p>(Applicable only for JSON Web Services) To send the data in a type as mentioned in provisioning plan or schema type, set the value of isQuotesEnabled to true in the application debug page as follows:</p> <pre data-bbox="659 403 1078 544"><entry key="isQuotesEnabled"> <value> <Boolean>true</Boolean> </value> </entry></pre> <p>For more information, see “Use of Quotes” on page 458.</p>
createAccountWithEntReq	<p>To enable the functionality of sending entitlements with create account in a single request to the managed system, set the value of createAccountWithEntReq parameter to true as follows:</p> <pre data-bbox="659 745 1192 887"><entry key="createAccountWithEntReq"> <value> <Boolean>true</Boolean> </value> </entry></pre> <p>Default value: false</p>
enableHasMore	<p>If enableHasMore is set to true as follows then the termination of aggregation would depend on the value of hasMore attribute:</p> <pre data-bbox="659 1036 1253 1068"><entry key="enableHasMore" value="true"/></pre> <p>The hasMore attribute is the boolean attribute which is to be set in the transientValues map in the before/after operation rule. Unless the value of hasMore attribute is false aggregation would not be terminated.</p> <p>If enableHasMore is set to false as follows, then the aggregation would be terminated if the number of accounts returned is zero:</p> <pre data-bbox="659 1326 1269 1358"><entry key="enableHasMore" value="false"/></pre>
oauth_request_parameters	<p>To send additional parameters for token generation, add the following entry in the application debug page:</p> <pre data-bbox="659 1474 1209 1698"><entry key="oauth_request_parameters"> <value> <Map> <entry key="customParamKey" value="customParamValue"/> </Map> </value> </entry></pre>

Configuration parameters

Attributes	Description
oauth_body_attrs_to_exclude	<p>To delete parameters for token generation, add the following entry in the application debug page:</p> <pre><entry key="oauth_body_attrs_to_exclude"> <value> <Map> <entry key="customParamKey" value="customParamValue"/> </Map> </value> </entry></pre>
oAuthJwtHeader	<p>Contains the alg (algorithm that is used for signing the JWT assertion) as follows:</p> <pre><entry key="oAuthJwtHeader"> <value> <Map> <entry key="alg" value="RS256"/> </Map> </value> </entry></pre> <p>If required additional header attributes can be provided in this map.</p>
oAuthJwtPayload	<p>Contains the aud (Audience), Expiry of the JWT assertion (exp), iss (Issuer), sub (Subject) as follows:</p> <pre><entry key="oAuthJwtPayload"> <value> <Map> <entry key="aud" value="" /> <entry key="exp" value="15f" /> <entry key="iss" value="" /> <entry key="sub" value="" /> </Map> </value> </entry></pre> <p>Note: If required additional payload attributes can be provided in this map. For additional attributes like jti, iat, nbf if only key (not value) is available in the map then it would consider the default values for the same.</p>

(Connector Operations) Operation specific configuration parameters

Note: No default provisioning template is provided. The template may vary from one target system to another.

Perform the following procedure to add and configure the specific operations:

1. Click **Add Operation**.
2. Select the operation from the drop down list of **Choose Operation**.

3. Provide a unique name to the operation. For example: Account Aggregation-1
4. Select the configure option (Pencil image) on the same row.

Allows user to provide additional options.

The following table lists the operation specific configuration parameters of SailPoint Web Services Connector:

Parameters	Description
ContextURL	<p>Context URL specific to the operation.</p> <p>For example, <code>/api/core/v3/securityGroups?startIndex=0&count=100&fields=%40all&sort=lastNameAsc</code></p>
Method	<p>Select one of the following type of HTTP method supported by the respective operation:</p> <ul style="list-style-type: none"> • GET • PUT • POST • DELETE • PATCH
Header	<p>(Optional) To view the header value in plain text, user must provide it in encrypted form. The encrypted value can be obtained from IdentityIQ Console.</p> <p>For example: The following example displays the sample header key and header value, where Authorization is header key and 1:vQaPY5LvJVbpsaig0nE56Q== is the header value:</p> <pre>Authorization 1:vQaPY5LvJVbpsaig0nE56Q==</pre> <p>Note:</p> <ul style="list-style-type: none"> • Content-Type header value must contain type matching any XML formats that is, application/XML or text/XML or */XML. • JSON request, JSON response: Content-Type= application/JSON (<i>optional</i>), Accept (<i>optional</i>) • XML request, XML response: Content-Type= application/XML or text/XML or */XML (<i>required</i>), Accept (<i>optional</i>) • JSON request, XML response: Content-Type=application/JSON (<i>optional</i>), Accept= application/XML or text/XML or */XML (<i>required</i>)

Configuration parameters

Parameters	Description
Body	<p>Standard http body used to post data with request. User can send data in either of the following format:</p> <ul style="list-style-type: none">• form-data: (<i>Applicable only for JSON</i>) Key value. User must set the data that has to pass in the key value• raw: Data to be sent in request body. For endpoint configuration, user must provide the XML payload by selecting the raw format. <p>For example,</p> <ul style="list-style-type: none">• (For JSON)<pre>{ "limit": 10, "cursor": "abcd1234" }</pre>• (For XML)<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:bsvc="urn:com.workday/bsvc"> <soapenv:Header> <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis- 200401-wss-wssecurity-secext-1.0.xsd"> ... </Security> </soapenv:Header> <soapenv:Body> ... </soapenv:Body> </soapenv:Envelope></pre>

Parameters	Description
Response	<p>(For XML Web Services) XPath Namespace Mapping: XML Namespace Prefix and corresponding Namespace URL identify uniquely named elements and attributes in XML request/response.</p> <p>If there exists any non-standard XML Namespace in the response, configure the same in the XML Namespace mapping where the key is Namespace Prefix and value is the Namespace URL.</p> <p>Note: Absence of non-standard Xml Namespaces would result in errors while response parsing.</p> <p>If a default Namespace is present, add a temporary Namespace Prefix with the default Namespace URL in the XML Namespace mappings. Further, use this temporary prefix in the XPATH elements within the scope of the default Namespace.</p> <p>For example, see “XML response for mapping:” example of payload.</p> <p>Root Path: Common path present in the JSON/XML response.</p> <p>It must be common for all the above attribute mentioned in the Response Attribute Mapping</p> <p>For example,</p> <p>(For JSON) <code>\$.members.profile</code></p> <p>(For XML) <code>//wd:Response_Data/wd:Worker/wd:Worker_Data</code></p> <p>Successful Response Code: Successful response code expected by the respective Web Service operation.</p> <p>This field accepts HTTP status code in csv format (For example, 200, 201, 203). If the list does not contain any value, the status code from 200 to 299 would be checked.</p> <p>There could be situation where successful status code may start with 2, in this situation user can provide 2**.</p>
Before Rule	Before Operation Rule: Rule that will be invoked before performing any operation (account aggregation, enable, disable account and so on).
After Rule	After Operation Rule: Rule that will be invoked after performing any operation (account aggregation, enable, disable account and so on)

Note: For more information on operation specific configurations, see “Additional information” on page 451.

Keywords

Web Service application supports the following keywords for various configuration attributes such as context URL, Headers, Body (JSON and Form-data) for a single or multiple endpoints:

Configuration parameters

Keywords	Description
plan	<p>Used for configuring the provisioning operations such as, create account, update account among others.</p> <p>For example,</p> <ul style="list-style-type: none"> • (Context URL): <code>/api/core/v3/people/\$plan.nativeIdentity\$</code> • (JSON Body example for plan): <pre> { "new_members": [{ "email": "\$plan.email\$", "first_name": "\$plan.first_name\$", "surname": "\$plan.surname\$", "send_welcome_email": \$plan.send_welcome_email\$, "role": { ".tag": "member_only" } }] } </pre>
response	<p>Used for multiple endpoints, where the response from the first endpoint is provided as an input for the second endpoint.</p> <p>For example, there are two endpoints for account aggregation.</p> <ul style="list-style-type: none"> • The first endpoint returns a response as a list of member_ids that is an input for the second endpoint as mentioned in the next point. • Second endpoint's JSON body is: <pre>{"members_info": [{"member_id": "\$response.member_id\$"}]}</pre>
application	<p>Used to get other configuration attributes from the current application.</p> <p>For example, <code>"\$application.accesstoken\$"</code>, where the <code>accesstoken</code> is an application configuration attribute.</p>

Keywords	Description
getObject	<p>Used while performing Aggregate Account (get a single account details).</p> <p>For example,</p> <ul style="list-style-type: none"> • (JSON body): <pre>{ "members": [{ ".tag": "member_id", "member_id": "\$getobject.nativeIdentity\$" }] }</pre> • (Context URL for get object): <pre>/api/v4/admin/\$getobject.nativeIdentity\$</pre>
nativelidentity	<p>Signifies the AccountID (identity attribute) in the plan or during getObject operation.</p> <p>For example, nativelidentity would be used along with the keyword as follows:</p> <ul style="list-style-type: none"> • getObject: \$getobject.nativeIdentity\$ • plan: \$plan.nativeIdentity\$

Schema attributes

Discover schema functionality is not available. Hence user must add the schema attributes manually for the respective Web Service based target system.

Additional information

This section describes the additional information related to the Web Services Connector.

Upgrade considerations

After upgrading IdentityIQ to version 7.3, add the following additional configuration attributes, if the **Authentication Method** is selected as **OAuth2** and the **Grant Type** is **JWT**:

- AuthJwtHeader
- AuthJwtPayload

For more information on the above additional configuration attributes, see “Additional configuration parameters” on page 443.

Web Services Class used in Before/After operation Rule

This section describes the following types of Web Services class:

- WebServicesClient
- EndPoint Class

WebServicesClient

The following table list the different rules and their description:

Rule	Description
Constructor Detail	
WebServicesClient	<p>Default constructor.</p> <pre>public WebServicesClient()</pre> <p>Constructor that configures the client using the given args.</p> <pre>public WebServicesClient(java.util.Map args) throws java.lang.Exception</pre> <p>Throws: java.lang.Exception</p>
Method Detail	
configure	<p>Configure connection parameters. See the ARG_* constants.</p> <pre>public void configure(java.util.Map args) throws java.lang.Exception</pre> <p>Throws: java.lang.Exception</p>
executeGet	<ul style="list-style-type: none"> • Execute method GET with headers <pre>public java.lang.String executeGet(java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: headers (Request headers) and allowedStatuses (Allowed status codes) • Execute method GET with URL and headers <pre>public java.lang.String executeGet(java.lang.String url, java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), headers (Request headers) and allowedStatuses (Allowed status codes) <p>Returns: Response object</p> <p>Throws: java.lang.Exception</p>

Rule	Description
executePost	<ul style="list-style-type: none"> Execute method POST with URL, payload and headers <pre>public java.lang.String executePost(java.lang.String url, java.lang.Object payload, java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body), headers (Request headers) and allowedStatuses (Allowed status codes) Execute method POST with URL and payload <pre>public java.lang.String executePost(java.lang.String url, java.lang.Object payload, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body) and allowedStatuses (Allowed status codes) <p>Returns: Response object</p> <p>Throws: java.lang.Exception</p>
executePut	<ul style="list-style-type: none"> Execute method PUT with URL and payload <pre>public java.lang.String executePut(java.lang.String url, java.lang.Object payload, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body) and allowedStatuses (Allowed status codes) Execute method PUT with URL, payload and headers <pre>public java.lang.String executePut(java.lang.String url, java.lang.Object payload, java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body), headers (Request headers) and allowedStatuses (Allowed status codes) <p>Returns: Response object</p> <p>Throws: java.lang.Exception</p>

Additional information

Rule	Description
executePatch	<ul style="list-style-type: none"> Execute method PATCH with URL and payload <pre>public java.lang.String executePatch(java.lang.String url, java.lang.Object payload, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body) and allowedStatuses (Allowed status codes) Execute method PATCH with URL, payload and headers <pre>public java.lang.String executePatch(java.lang.String url, java.lang.Object payload, java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), payload (Request body), headers (Request headers) and allowedStatuses (Allowed status codes) <p>Returns: Response object</p> <p>Throws: java.lang.Exception</p>
executeDelete	<ul style="list-style-type: none"> Execute method DELETE with URL <pre>public java.lang.String executeDelete(java.lang.String url, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL) and allowedStatuses (Allowed status codes) Execute method DELETE with URL and headers <pre>public java.lang.String executeDelete(java.lang.String url, java.util.Map headers, java.util.List<java.lang.String> allowedStatuses) throws java.lang.Exception</pre> Parameters: url (Request URL), headers (Request headers) and allowedStatuses (Allowed status codes) <p>Returns: Response object</p> <p>Throws: java.lang.Exception</p>
getResponseHeaders	<p>Get last executed Request's Response headers.</p> <pre>public java.util.Map<java.lang.String,java.lang.String> getResponseHeaders()</pre>

EndPoint Class

The following table list the different rules and their description:

Rule	Description
Constructor Detail	
EndPoint	public EndPoint()

Rule	Description
Method Detail	
setAfterRule	<p>Setting the after rule name</p> <pre>public void setAfterRule(java.lang.String value)</pre>
setBeforeRule	<p>Setting the before rule name</p> <pre>public void setBeforeRule(java.lang.String value)</pre>
setParseRule	<pre>public void setParseRule(java.lang.String value)</pre>
setContextUrl	<p>Set the context url for the particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setContextUrl(java.lang.String value)</pre>
setHttpMethodType	<p>Set the http method (put, post, get, patch and delete) for the particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setHttpMethodType(java.lang.String value)</pre>
setOperationType	<p>Set the operation (Account Aggregation, Group Aggregation, Create Account etc) for the particular operation record (create user, update user, account aggregation, and so on)</p> <pre>public void setOperationType(java.lang.String value)</pre>
setRootPath	<p>Set the root of the JSON response returned from the target system (Managed system) for the particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setRootPath(java.lang.String value)</pre>
setFullUrl	<p>set the complete url (endpoint) of the operation that need to be performed for the particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setFullUrl(java.lang.String value)</pre>
setBaseUrl	<p>Set the base url (the machine id or IP and the port where the service is executing) for the operation that need to be performed. Ideally this would be common for all the operation.</p> <pre>public void setBaseUrl(java.lang.String value)</pre>
setSequenceNumberForEndpoint	<p>Set the Sequence number particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setSequenceNumberForEndpoint(int value)</pre>
setUniqueNameForEndPoint	<p>Set Unique operation name for particular operation (create user, update user, account aggregation, and so on)</p> <pre>public void setUniqueNameForEndPoint(java.lang.String value)</pre>

Additional information

Rule	Description
setResMappingObj	<p>Set the Response mapping for the response attribute returned in the JSON response from the target system (Managed system) for the particular operation like create user, update user, account aggregation, and so on. Here the key would be attribute name (attribute in the schema) and value would be the JSON response path after the root path mentioned above.</p> <pre>public void setResMappingObj(java.util.Map value)</pre>
setHeader	<p>Set HTTP header information in the form of key value (For example, key="ContentType" Value="Application/JSON")</p> <pre>public void setHeader(java.util.Map value)</pre>
addHeader	<p>Adding key value if header exists or will create header and add</p> <pre>public void addHeader(java.lang.String entry, java.lang.String value)</pre>
setBody	<p>Set http body information as a Map. Here the map would contain three keys bodyFormat, bodyFormData and jsonBody. The bodyFormat value can be raw that means user has provided values as raw JSON string else user has provided value in the key value format that must be converted into JSON.</p> <pre>public void setBody(java.util.Map value)</pre>
setResponseCode	<p>Set the value of successful response code as list (200, 299, 201). This would be respected by the connector if any other response code would be consider as request failure.</p> <pre>public void setResponseCode(java.util.List value)</pre>
getAfterRule	<p>Fetch the name of after rule assigned to the particular operation like create, update user, account aggregation, and so on:</p> <pre>public java.lang.String getAfterRule()</pre>
getBeforeRule	<p>Fetch the name of before rule assigned to the particular operation like create, update user, account aggregation, and so on.</p> <pre>public java.lang.String getBeforeRule()</pre>
getParseRule	<pre>public java.lang.String getParseRule()</pre>
getContextUrl	<p>Fetch the contextUrl provided to the particular operation like create, update user, account aggregation, and so on</p> <pre>public java.lang.String getContextUrl()</pre>
getHttpMethodType	<p>Fetch the httpMethodType (get,put,post,delete and patch) provided to the particular operation like create, update user, account aggregation, and so on.</p> <pre>public java.lang.String getHttpMethodType()</pre>
getOperationType	<p>Fetch the operationType (Account Aggregation,Create Account,Group Aggregation etc) provided to the particular operation like Create, update user, account aggregation, and so on.</p> <pre>public java.lang.String getOperationType()</pre>

Rule	Description
getRootPath	Fetch the rootPath provided to the particular operation like Create, update user, account aggregation, and so on. public java.lang.String getRootPath()
getFullUrl	Fetch the fullUrl that is a combination of basicUrl + contextUrl for the particular operation like Create, update user, account aggregation, and so on. public java.lang.String getFullUrl()
getBaseUrl	Fetch the baseUrl which is common for all operation like Create, update user, account aggregation, and so on. public java.lang.String getBaseUrl()
getSequenceNumberForEndpoint	Fetch the sequenceNumber for the particular operation (Create, update user, account aggregation, etc) that decide the priority of execution for operation, if there are multiple endpoint of same operation like account aggregation. public int getSequenceNumberForEndpoint()
getUniqueNameForEndpoint	Fetch the uniqueName provided to the particular operation like Create, update user, account aggregation, and so on. public java.lang.String getUniqueNameForEndPoint()
getResMappingObj	Fetch the responseMapping map that will have key as schema attribute and value as JSON path in the JSON response for particular operation like Create, update user, account aggregation, and so on. public java.util.Map getResMappingObj()
getHeader	Fetch the Http header map that holds the header information for particular operation like Create, update user, account aggregation, and so on. public java.util.Map getHeader()
getBody	Fetch the body map that holds the body information with keys like bodyFormat , jsonBody and bodyFormData . The bodyFormat can have raw or formData value. bodyFormData will have value as map jsonBody will have value as string with whole JSON. public java.util.Map getBody()
getResponseCode	Fetch the success response code (list) value which will decide whether the operation was successful or not for particular operation like Create, update user, account aggregation, and so on. public java.util.List getResponseCode()
getAttributes	public sailpoint.object.Attributes getAttributes()
getAttribute	public java.lang.Object getAttribute(java.lang.String name)
getBooleanAttributeValue	public boolean getBooleanAttributeValue(java.lang.String name)
getStringAttributeValue	public java.lang.String getStringAttributeValue(java.lang.String name)
setAttribute	public void setAttribute(java.lang.String name, java.lang.Object value)

Additional information

Rule	Description
getPaginationSteps	Fetch the paging steps as a string which will decide how account/group paging will work. public java.lang.String getPaginationSteps()
setPaginationSteps	Set the paging steps as a string which will decide how account/group paging would work. public void setPaginationSteps(java.lang.String paginationSteps)
toString	public java.lang.String toString() Overrides: toString in class java.lang.Object
getResponseBody	Retrieve last executed Request's Response Body. public java.lang.String getResponseBody()
setXpathNamespaces	Sets XPath namespaces using the supplied Map object. public void setXpathNamespaces(Map<String, String> xpathNamespaces)
getXpathNamespaces	Retrieves XPath namespaces. public Map<String, String> getXpathNamespaces()
getPagingInitialOffset()	Retrieves the initial page offset. public int getPagingInitialOffset()
setPagingInitialOffset()	Sets the initial paging offset. public void setPagingInitialOffset(int pagingInitialOffset)
getPagingSize()	Retrieves the page limit. public int getPagingSize()
setPagingSize	Sets the page limit. public void setPagingSize(int pagingLimit)

Use of Quotes

(Applicable only for JSON) For application prior to version 7.3, extra quotes with plan placeholder must be send the attribute value as a String. For example, {firstName: "\$plan.firstName\$"}.

After upgrading to IdentityIQ version 7.3,

- to send data type in a type mentioned in provisioning plan or schema type add the **isQuotesEnabled** in the application debug page and set it to true.

For example, { "user": { "firstName": "\$plan.firstName\$", "isActiveAsAString": "\"\$plan.isActive\$\"", "isActive": "\$plan.isActive\$", "userId" : "\$plan.Id\$"} }

where

- *firstName* is a String
- *isActive* is a Boolean
- *Id* is an Integer

The above mentioned attributes type can be part of Provisioning plan/Schema attributes/Provisioning policy form and the results would be as follows:

```
{ "user": { "firstName": "Testfirstname", "isActiveAsAString": "true", "isActive": true, "userId" : 987654321 } }
```

- For formdata to work as mentioned in the following example, set the value of **isQuotesEnabled** to true:

For example,

- *firstName \$plan.firstName\$*
- *isActive \$plan.isActive\$*
- *isActiveAsAString "\$plan.isActive\$"*
- *userId \$plan.Id\$*
- *passwordresetCount 0*

The above mentioned attributes type can be a part of Provisioning plan/Schema attributes/Provisioning policy form and the results would be as follows:

```
{ "firstName": "Testfirstname", "isActiveAsAString": "true", "isActive": true, "userId" : 987654321, "passwordresetCount" : "0" }
```

Pagination

Web Services Connector supports generic paging for Account and Group Aggregations. Following are the methods for embedding paging in Web Service Connector:

- Using BEFORE and AFTER operation rules
- Or
- Using Paging tab

BEFORE and AFTER operation rules

To embed pagination in Web Service Connector, manual processing is required in BEFORE and AFTER operation rules of Web Service Connector.

1. The Web Service Connector relies on a temporary information stored in the application object in form of a map which has the name as **transientValues**.
2. The administrator must write the Before Rule and After Rule for account/group aggregation as follows:
 - **Web Service Before Rule:** The Before Rule alters the URL/request parameters if the value of the **hasMore** parameter is set to **TRUE** and the request to fetch further accounts is triggered. If **hasMore** parameter is not set or is set to **FALSE** the pagination request would be terminated.

For example, see sample Before Rule for account aggregation request in Web Services Connector for Dropbox using V2 in **examplerules.xml** file by name **Example WSBeforeRI DropboxPaging** as follows:

```
import sailpoint.tools.Util;
Map obj = (Map) application.getAttributeValue("transientValues");
```

Additional information

```
System.out.println("BEFORE RULE: Transient Values ==> " + obj);
if(null != obj) {
    String offset = obj.get("offset");
    System.out.println("BEFORE RULE: offset value ==> " + offset);
    String urlString = (String) requestEndPoint.getFullUrl();
    if(Util.isNotNullOrEmpty(offset)) {
        System.out.println("BEFORE RULE: requestEndpoint ==> " + requestEndPoint);
        System.out.println("BEFORE RULE: URL ==> " + urlString);
    URL tempUrl = new URL(urlString);
    String queryString = tempUrl.getQuery();
    System.out.println("BEFORE RULE: Query String ==> " + queryString);

    if(Util.isNotNullOrEmpty(queryString)) {
        StringBuffer queryParams = new StringBuffer();
        String[] params = tempUrl.getQuery().split("&");
        for (String param : params) {
            if(queryParams.length() > 0)
                queryParams.append("&");
            queryParams.append(param);
        }
        urlString = urlString.replace(tempUrl.getQuery(), queryParams.toString());
    }
}

System.out.println("BEFORE RULE: Updated Query String ==> " + urlString);
requestEndPoint.setFullUrl(urlString);
}
System.out.println("BEFORE RULE: requestEndpoint Updated ==> " + requestEndPoint);
return requestEndPoint;
```

In case of Dropbox V2, the **cursor** returned from the previous team membership listing API would be stored in the **transientValues** map in the application by the Web Service AFTER Rule. The url is modified to direct to the paging API and the cursor would be sent as a part of the form data. Ensure that the **hasMore** flag is set by the earlier requests AFTER RULE

- **Web Service After Rule:** The AFTER Rule deduces whether the managed system has more records which can be fetched and added as an entry in **transientValues** with **hasMore** key and value as TRUE/FALSE depending upon the condition deduced.

For example, see sample AFTER Rule for account aggregation request in Web Services Connector for Dropbox using V2 in **examplerules.xml** by name **Example WSAfterRI DropboxPaging** as follows:

```
Integer fetchedRecordsCount = 0;
if(null != processedresponseObject) {
    fetchedRecordsCount = ((List) processedresponseObject).size();
}

Integer expectedCount = null;
Integer offset = null;
URL url = new URL(requestEndPoint.getFullUrl());
System.out.println("AFTER RULE: Original Url ==> " + url);
String[] params = url.getQuery().split("&");
for (String param : params) {
```

```

String name = param.split("=')[0];
String value = param.split("=')[1];

switch(name) {
    case "sysparm_limit":
        expectedCount = Integer.parseInt(value);
        break;

    case "sysparm_offset":
        offset = Integer.parseInt(value);
        break;

    default:
}
}

System.out.println("AFTER RULE: Fetch Count ==> " + fetchedRecordsCount);
System.out.println("AFTER RULE: Limit Count ==> " + expectedCount);
System.out.println("AFTER RULE: Fetch Offset ==> " + offset);

boolean hasMore = (fetchedRecordsCount != 0 && null != expectedCount &&
fetchedRecordsCount.equals(expectedCount) && null != offset);
System.out.println("AFTER RULE: Has More? ==> " + hasMore);

Map transientValues = application.getAttributeValue("transientValues");
if(transientValues == null) {
    transientValues = new HashMap();
    application.setAttribute("transientValues", transientValues);
}
transientValues.put("hasMore", hasMore);
if (hasMore) {
    if(null != offset) {
        System.out.println("AFTER RULE: New Offset ==> " + (offset + expectedCount));
        transientValues.put("offset", String.valueOf(offset + expectedCount));
    }
}
}

```

In case of Dropbox, Dropbox V2 for team membership response contain the following elements:

- **cursor**: is an encrypted token which represents the next page to be fetched, if any, and would form part of the subsequent API calls.
- **has_more**: is a boolean value which explicitly indicates whether more records are available for fetching.

AFTER Rule stores the **cursor** and **has_more** values from the response in the **transientValues** map in the Application object. This map stores the necessary information which would be used by the BEFORE RULE to manipulate the next API call. Ensure that the flag indicating whether the managed system contains more records is stored by the key named **hasMore**. This field is mandatory as it is the deciding factor for aborting the pagination requests.

Paging tab

Paging can be configured in Account/Group Aggregation endpoints through Paging tab.

Note: If there are multiple Account or Group Aggregation endpoints configured, paging would be supported only for the first endpoint of Account and Group Aggregation each.

Paging mechanism has the following predefined set of keywords:

Additional information

Keywords	Description	
application	Represents the application.	
	baseUrl	Base URL configured in the application.
endpoint	Represents endpoint configuration.	
	relativeURL	Relative URL of the endpoint.
	fullUrl	Full URL contained within the endpoint.
limit	Page limit.	
offset	Initial page offset.	
request	Represents request body.	
requestHeaders	Represents request header.	
response	Represents response body object.	
responseHeaders	Represents response header.	
TERMINATE_IF	Indicates termination condition, multiple conditions can exist.	
	NO_RECORDS	Indicates no records retrieved; to be used in conjunction with TERMINATE_IF, evaluates to TRUE / FALSE
	RECORDS_COUNT	Indicates number of records retrieved.
NULL	Indicates null or empty object.	
REMOVE	Indicates to remove an attribute.	

The following table lists the supported and conditional supported operations:

Type	Operations
Regular operations	+, -, *, /, =, && and
Conditional operations	<, >, <=, >=, == and !=

The following table lists the examples for the respective Target System:

Managed System	Examples
Dropbox	<pre>TERMINATE_IF \$response.has_more\$ == FALSE \$endpoint.fullUrl\$ = \$application.baseUrl\$ + \$endpoint.relativeUrl\$ + "/continue" \$request.cursor\$ = \$response.cursor\$ REMOVE \$request.limit\$</pre>

Managed System	Examples
ServiceNow	<p>Using limit-offset:</p> <pre>\$sysparm_limit\$ = 100 TERMINATE_IF \$RECORDS_COUNT\$ < \$sysparm_limit\$ \$sysparm_offset\$ = \$sysparm_offset\$ + \$sysparm_limit\$ \$endpoint.fullUrl\$ = \$application.baseUrl\$ + "/api/now/v1/table/sys_user?sysparm_fields= sys_id&sysparm_limit=100&sysparm_offset=" + \$sysparm_offset\$</pre>
	<p>Using response header link:</p> <pre>TERMINATE_IF \$responseHeaders.Link.next\$ == NULL \$endpoint.fullUrl\$ = \$responseHeaders.Link.next\$</pre>
Workday	<pre>TERMINATE_IF \$response.wd:Response_Results.wd:Page\$ > \$response.wd:Response_Results.wd:Total_Pages\$ \$offset\$ = \$response.wd:Response_Results.wd:Page\$ + 1 \$request.bsvc:Response_Filter.bsvc:Page.text() [1]\$ = \$offset\$</pre>
Salesforce	<pre>TERMINATE_IF \$response.ns:result.ns:done\$!= FALSE \$request.soapenv:Body\$ = "<urn:queryMore><urn:queryLocator>" + \$response.ns:result.ns:queryLocator\$ + "</urn:queryLocator></urn:queryMore>"</pre>
Successfactor	<pre>TERMINATE_IF \$response.ns:result.ns:hasMore\$!= TRUE \$request.soapenv:Body\$ = "<urn:queryMore><urn:querySessionId>" + \$response.ns:result.ns:querySessionId\$ + "</urn:querySessionId></urn:queryMore>"</pre>

Paging configuration caveats:

1. Every paging configuration step must start on a new line.
2. SailPoint recommends to provide a <space> after every operator, condition or placeholder for correct evaluation of paging expression.
3. Paging mechanism follows the placeholder notation for resolution of attribute values, that is., `$response.attribute_key$`. Any attribute which follows the placeholder notation would be resolved or assigned a value depending upon the operator being used.
4. Intermediate values can also be stored between page request by using the placeholder notation. In order to achieve this, any attribute key which does not match any of the predefined keywords can be used. For more information, see the example mentioned in the above table for **ServiceNow (Using limit-offset)** where `$sysparm_offset$` is being updated and used between page requests.
5. For complex expressions or conditions, multiple conditions can be clubbed together using '(' and ')'. For example, `TERMINATE_IF ($someattribute$ == TRUE) && ($otherattribute$ == NULL)`

Configuration for Response

When configuring the Web Services application, map the schema attribute as follows:

- **For JSON:** Refer the following example:

Figure 1—Example for mapping the schema attributes with JSON

```
"list": [
    {
        "id": "2124",
        "resources": {
            "securityGroups": {
                "ref": "https://mydomain.jive.com/api/core/v3/people/2124/securityGroups"
            },
            "displayName": "Bill Jackson",
            "emails": [
                {
                    "value": "bill.jackson@mydomain.com",
                }
                {
                    "value": "admin@mydomain.com",
                }
            ],
            "jive": {
                "enabled": true,
                "level": {
                    "name": "Level 0",
                },
                "username": "bill.jackson",
            },
        },
    }
]
```

In the above JSON response, all the attributes can be mapped as follows considering Root Path as \$.list:

Id = id
displayName=displayName
username=jive.username
enabled =jive.enabled
emails=emails[*].value

Figure 2—Mapped schema attributes

The screenshot shows the 'Response Attribute Mapping' interface. It consists of two main sections: 'Schema Attribute' and 'Attribute Path'. Below these are sections for 'XPath Namespace Mapping (For XML web services)' and 'Successful Response Code'.

Schema Attribute	Attribute Path
id	id
enabled	jive.enabled
username	jive.username
emails	emails[*].value
displayName	displayName

XPath Namespace Mapping (For XML web services)

Root Path: \$list

Successful Response Code: 2**

- **For XML:**

- XML response for mapping:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <....>
    <....>
  </soapenv:Header>
  <soapenv:Body>
    <queryResponse>
      <result xsi:type="QueryResult">
        <done>true</done>
        <queryLocator xsi:nil="true"/>
        <records xsi:type="sf:sObject">
          <sf:type>User</sf:type>
          <sf:Id>123456</sf:Id>
          <sf:Id>123456</sf:Id>
          <sf:Alias>Alias</sf:Alias>
          <sf:City xsi:nil="true"/>
          <sf:CommunityNickname>CName1</sf:CommunityNickname>
          <sf:Email>test@test.com</sf:Email>
          <sf:IsActive>false</sf:IsActive>
          <sf:Username>test@test.com</sf:Username>
          <sf:FirstName>Test</sf:FirstName>
          <sf:LastName>Test</sf:lastName>
        </records>
      <size>1</size>
    </result>
  </queryResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Additional information

```
</result>
</queryResponse>
</soapenv:Body>
</soapenv:Envelope>
```

- See the following figure which mentions the XPath Namespace mapping for XML Web Services:

The screenshot shows the configuration interface for an XML Web Services endpoint. At the top, there are fields for 'Context URL' (set to '/services/Soap/u/33.0/00D|0000000bTeC') and 'Method' (set to 'POST').

Response Attribute Mapping: This section maps schema attributes to attribute paths. It includes columns for 'Schema Attribute' (e.g., ProfileId, firstName, UserRoleId, Username, Alias, IsActive, ProfileName, LastName, id, email) and 'Attribute Path' (e.g., sf:ProfileId, sf:FirstName, sf:UserRoleId, sf:Username, sf:Alias, sf:IsActive, sf:Profile/sf:Name, sf:LastName, sf:id[1], sf:Email). There are 'Add Row' and 'Delete' buttons.

XPath Namespace Mapping (For XML web services): This section maps namespace prefixes to URIs. It includes columns for 'Namespace Prefix' (sf, soapenv, ns, xsi) and 'Namespace URI' (urn:object.partner.soap.sforce.com, http://schemas.xmlsoap.org/soap/envelope/, urn:partner.soap.sforce.com, http://www.w3.org/2001/XMLSchema-instance). There are 'Add Row' and 'Delete' buttons.

At the bottom right, there are 'Cancel' and 'Save' buttons.

Configuration for Multiple endpoints

Perform the following to obtain the properties of account/group from multiple endpoints:

1. The basic attribute is obtained from the first endpoint and is then used for fetching the data from rest of the endpoints.

For example, during aggregation of Jive some attributes are obtained from first endpoint ("Figure 2— Mapped schema attributes") using the following URL:

<https://myDomain.jive.com /api/core/v3/people>

2. To fetch additional attribute from another endpoint use the `id` attribute from the previous response. Add these attributes in Schema Attribute of Response Attribute Mapping and response as follows:

- **Schema Attribute**

Response Attribute Mapping	
Schema Attribute	Attribute Path
Email	profile.email
groups	profile.groups

- **Response:** The following context URL contains `id` which fetches all the groups connected to that account:

[https://myDomain.jive.com/api/core/v3/people/\\$response.id\\$/securityGroups](https://myDomain.jive.com/api/core/v3/people/$response.id$/securityGroups)

Configuring Multiple Entitlement Requests

To enable the functionality of sending multiple entitlement request of different type of entitlements (role, permission, groups and so on) in a single request to the managed system, set the value of `addRemoveEntInSingleReq` parameter to true as follows:

```
<entry key="addRemoveEntInSingleReq">
    <value>
        <Boolean>true</Boolean>
    </value>
</entry>
```

1. If `addRemoveEntInSingleReq` is set to true, then the payload for entitlements must be as given in the following example:


```
{
        "group_id" : $plan.groups$,
        "permission":$plan.permission$,
        "roles": $plan.roles$
      }
```
2. If `addRemoveEntInSingleReq` is set to false, then the payload for entitlements must be as given in the following example:


```
{
        "group_id" : "$plan.groups$",
        "permission":'$plan.permission$',
        "roles": "$plan.roles$"
      }
```

Other Operations

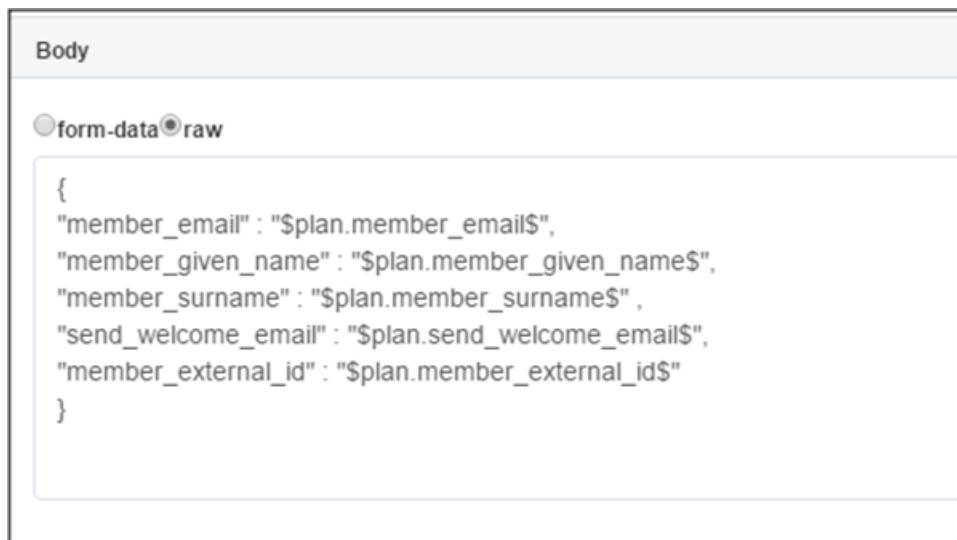
For certain operations, the Body must be updated accordingly.

Additional information

Create Account

This section provides an example for updating the Body for create account in Dropbox. For fetching attribute through Provisioning Plan, the body must be updated in the following manner. This fetches the attribute detail through Provisioning Form and updates the endpoint.

- (For JSON) In the following Body,
 - **\$plan** represents the Provisioning Plan that is passed to provision method
 - **\$plan.member_surname**: the connector checks for **member_surname** in the attribute request and updates in the body after it is found



The screenshot shows a JSON editor interface with a single tab labeled "Body". Below the tab, there are two radio buttons: "form-data" (selected) and "raw". The main area contains a JSON object:

```
{  
  "member_email": "$plan.member_email$"  
  "member_given_name": "$plan.member_given_name$"  
  "member_surname": "$plan.member_surname$"  
  "send_welcome_email": "$plan.send_welcome_email$"  
  "member_external_id": "$plan.member_external_id$"  
}
```

- (For XML) To create account for XML payload:

Body

[form-data](#) [raw](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:partner.soap.sforce.com"
xmlns:urn1="urn:sobject.partner.soap.sforce.com">
<soapenv:Header>
<urn:SessionHeader>
<urn:sessionId>$application.accesstoken$</urn:sessionId>
</urn:SessionHeader>
</soapenv:Header>
<soapenv:Body>
<urn:create>
<!--Zero or more repetitions:-->
<urn:sObjects>
<urn1:type>User</urn1:type>
<!--Zero or more repetitions:-->
<urn1:Username>$plan.Username$</urn1:Username>
<urn1:LastName>$plan.LastName$</urn1:LastName>
<urn1:FirstName>$plan.FirstName$</urn1:FirstName>
<urn1:Email>$plan.Email$</urn1:Email>
<urn1:Alias>$plan.Alias$</urn1:Alias>
<urn1:CommunityNickname>$plan.CommunityNickname$</urn1:CommunityNickname>
<urn1:isActive>true</urn1:isActive>
<urn1:TimeZoneSidKey>America/Los_Angeles</urn1:TimeZoneSidKey>
<urn1:LocaleSidKey>en_US</urn1:LocaleSidKey>
<urn1:LanguageLocaleKey>en_US</urn1:LanguageLocaleKey>
<urn1:ProfileId>00ei0000000ye0AAQ</urn1:ProfileId>
<urn1:EmailEncodingKey>UTF-8</urn1:EmailEncodingKey>
<!--You may enter ANY elements at this point-->
</urn:sObjects>
</urn:create>
</soapenv:Body>
</soapenv:Envelope>
```

To get object for XML payload:

Body

[form-data](#) [raw](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:partner.soap.sforce.com">
<soapenv:Header>
<urn:SessionHeader>
<urn:sessionId>$application.accesstoken$</urn:sessionId>
</urn:SessionHeader>
</soapenv:Header>
<soapenv:Body>
<urn:query>
<urn:queryString>Select Id , Alias , City , CommunityNickname , CompanyName , CallCenterId , Country , Department , Email ,
Division , EmployeeNumber , Extension , Street , Fax , IsActive , Username , FirstName , LastName , EmailEncodingKey , Name ,
UserPermissionsMarketingUser , UserPermissionsMobileUser , UserPermissionsOfflineUser , UserPermissionsSFContentUser , Phone ,
ProfileId , Profile.Name , ReceivesAdminInfoEmails , UserRoleld , UserRole.Name , UserType , State , Title , ReceivesInfoEmails , Profile.Id ,
UserRole.Id from user Where Id = '$getobject.nativ entityId$'
</urn:queryString>
</urn:query>
</soapenv:Body>
</soapenv:Envelope>
```

Additional information

Enable/Disable

Set the get object endpoint for enable/disable operation as in the POST method the complete object would be required to update and not single attribute. Hence first endpoint getObject would fetch the whole account and later the endpoint would update the payload with all the required attributes using the response of the first endpoint.

Perform the following steps to get object for Enable operation with PUT method

1. Configure the first endpoint to get object for Enable.

		Enable Account	▼	Enable ONE - getObject	Configured		
		Enable Account	▼	Enable TWO - enable	Configured		

2. Configuration for the first endpoint.

Context URL	Method																														
/api/core/v3/people/\$getObject.nativeIdentity\$	GET																														
<table border="1"><tr><td>Header</td></tr><tr><td>Body</td></tr><tr><td>Response</td></tr><tr><td>Before Rule</td></tr><tr><td>After Rule</td></tr></table>	Header	Body	Response	Before Rule	After Rule	<table border="1"><tr><td>Response Attribute Mapping</td><td>Add Row </td></tr><tr><td><table border="1"><tr><td>Schema Attribute </td><td>Attribute Path </td></tr><tr><td>country</td><td>addresses[0].country</td></tr><tr><td>Email</td><td>emails[0].value</td></tr><tr><td>displayName</td><td>displayName</td></tr><tr><td>street</td><td>addresses[0].value.streetAddress</td></tr><tr><td>givenName</td><td>name.givenName</td></tr><tr><td>name</td><td>displayName</td></tr><tr><td>id</td><td>id</td></tr><tr><td>region</td><td>addresses[0].value.region</td></tr><tr><td>type</td><td>type</td></tr><tr><td>status</td><td>jive.enabled</td></tr></table></td></tr></table>	Response Attribute Mapping	Add Row	<table border="1"><tr><td>Schema Attribute </td><td>Attribute Path </td></tr><tr><td>country</td><td>addresses[0].country</td></tr><tr><td>Email</td><td>emails[0].value</td></tr><tr><td>displayName</td><td>displayName</td></tr><tr><td>street</td><td>addresses[0].value.streetAddress</td></tr><tr><td>givenName</td><td>name.givenName</td></tr><tr><td>name</td><td>displayName</td></tr><tr><td>id</td><td>id</td></tr><tr><td>region</td><td>addresses[0].value.region</td></tr><tr><td>type</td><td>type</td></tr><tr><td>status</td><td>jive.enabled</td></tr></table>	Schema Attribute	Attribute Path	country	addresses[0].country	Email	emails[0].value	displayName	displayName	street	addresses[0].value.streetAddress	givenName	name.givenName	name	displayName	id	id	region	addresses[0].value.region	type	type	status	jive.enabled
Header																															
Body																															
Response																															
Before Rule																															
After Rule																															
Response Attribute Mapping	Add Row																														
<table border="1"><tr><td>Schema Attribute </td><td>Attribute Path </td></tr><tr><td>country</td><td>addresses[0].country</td></tr><tr><td>Email</td><td>emails[0].value</td></tr><tr><td>displayName</td><td>displayName</td></tr><tr><td>street</td><td>addresses[0].value.streetAddress</td></tr><tr><td>givenName</td><td>name.givenName</td></tr><tr><td>name</td><td>displayName</td></tr><tr><td>id</td><td>id</td></tr><tr><td>region</td><td>addresses[0].value.region</td></tr><tr><td>type</td><td>type</td></tr><tr><td>status</td><td>jive.enabled</td></tr></table>	Schema Attribute	Attribute Path	country	addresses[0].country	Email	emails[0].value	displayName	displayName	street	addresses[0].value.streetAddress	givenName	name.givenName	name	displayName	id	id	region	addresses[0].value.region	type	type	status	jive.enabled									
Schema Attribute	Attribute Path																														
country	addresses[0].country																														
Email	emails[0].value																														
displayName	displayName																														
street	addresses[0].value.streetAddress																														
givenName	name.givenName																														
name	displayName																														
id	id																														
region	addresses[0].value.region																														
type	type																														
status	jive.enabled																														

This endpoint retrieves getObject for account for which Provisioning Operation is performed.

3. Configuration for second endpoint for Enable endpoint as shown in the following figure:

Context URL [?](#) /api/core/v3/people/\$plan.nativelidentity\$

Method [?](#) PUT

Header	
Body	
Response	
Before Rule	
After Rule	

form-data raw

```
{
  "emails": [
    {
      "value": "$response.Email$",
      "jive_label": "Email"
    }
  ],
  "jive": {
    "enabled": "true",
    "federated": "false",
    "visible": "true",
    "username": "$response.Email$"
  },
  "name": {
    "formatted": "$response.givenName$",
    "familyName": "$response.familyName$",
    "givenName": "$response.givenName$"
  }
}
```

Note: It may be required to update few attribute for performing enable/disable operation

Similar steps are to be performed for Disable operation.

Add/Remove Entitlement

Following is an example of the Body entry for Add Entitlement:

Context URL [?](#) /1/team/groups/members/add

Method [?](#) POST

Header	
Body	
Response	
Before Rule	
After Rule	

form-data raw

```
{
  "group_id": "$plan.groups$",
  "members": [
    { "team_member_id": "$plan.nativelidentity$",
      "access_type": "member" }
  ]
}
```

On similar basis as above example the Body entry must be updated for Remove Entitlement.

Additional information

Update Account

Following is an example of the Body entry for Update Account:

Context URL [?](#) /1/team/members/set_profile Method [?](#) POST

Header **Body** **Response** **Before Rule** **After Rule**

Body

form-data raw

```
{  
    "member_id": "$plan.nativeIdentity$"  
    "new_given_name": "$plan.given_name$"  
    "new_email": "$plan.email$"  
}
```

Delete Account

Following is an example of the Body entry for Delete Account:

Context URL [?](#) /1/team/members/remove Method [?](#) POST

Header **Body** **Response** **Before Rule** **After Rule**

Body

form-data raw

```
{  
    "member_id": "$getobject.nativeIdentity$"  
}
```

Change Password

Following is an example of the Body entry for Change Password:

Details **Configuration** Correlation Accounts Risk Activity Data Sources Rules Password Policy

Settings Schema Provisioning Policies

Back Connection Settings

Context URL [?](#) /api/now/v1/import/x_sapo_iiq_connect_sysuser Method [?](#) POST

Header **Body** **Response** **Before Rule** **After Rule**

Body

form-data raw

```
{"user_sys_id": "$plan.nativeIdentity$","password": "$plan.password$","password_needs_reset": "false"}
```

Cancel Save

Additional information

Chapter 47: SailPoint WebEx Connector

The following topics are discussed in this chapter:

Overview.....	475
Supported features	475
Pre-requisites	475
Administrator permissions	476
Configuration parameters.....	476
Schema attributes	476
Account attributes	476
Group attributes.....	478
Provisioning Policy attributes.....	479

Overview

The SailPoint WebEx Connector manages WebEx accounts and groups (Meeting Types). It supports read and write for WebEx accounts. The WebEx connector supports creation, deletion, retrieval, authentication and unlock for users and retrieval for groups.

Note: In the WebEx connector, Meeting Types are treated as Groups.

Supported features

SailPoint WebEx Connector supports the following features:

- Account Management
 - Manages Webex Users as Accounts
 - Aggregation, Refresh Accounts, Pass Through Authentication
 - Create, Update, Delete

Note: When performing Delete operation, the account does not get deleted but gets disabled.
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages Webex meeting types as Account-Groups
 - Aggregation, Refresh Groups

Pre-requisites

Note: If WebEx Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

Configuration parameters

Administrator permissions

The user must be a **Site Administrator**.

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The WebEx Connector uses the connection attributes listed in the following table:

Parameters	Description
webExID	WebEx user ID for the meeting host.
password	The password for the user with a webExID.
siteID	The WebEx-assigned identification number that uniquely identifies your website.
siteName	The first string in your WebEx site URL, provided by WebEx. For example, if acme is the siteName for the https://acme.webex.com site.
partnerID	(Optional) A reference to the WebEx partner, provided by WebEx.
xmlURL	XML URL of the site. For example, WBXService/XMLService
Manage Disabled Accounts	If set to yes, the disabled accounts will be a part of the Aggregation.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following types of objects:

Account: Account objects are used when building identities Link objects.

Group: The group schema is used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes:

Attributes	Description
WebexID	WebEx ID of the user.
FirstName	First Name of the user.
LastName	Last Name of the user.

Attributes	Description
Email	Email id of the user.
RegistrationDate	The creation date of the user
Active	Determines whether the user account has been staged for use. Default: ACTIVATED Note: If you set the user's active parameter to ACTIVATED such that the WebEx sites host limit is exceeded, the CreateUser or SetUser command displays the following error: exceededSiteHostLimit
TimezoneID	Determines the time zone for the geographic location of the meeting.
Company	The user's company name.
Description	A description of the user's virtual office.
CategoryID	A reference to the office category for the user's office.
AddressType	Determines whether the meeting participant is a personal contact of the meeting host or is a site-wide (global) contact.
Country	The country for the user.
Phone	The user's Office Profile phone number.
MobilePhone	The attendee's mobile phone number.
Fax	Indicates the fax number for the user.
Pager	The user's Office Profile pager number.
PersonalURL	The user's website.
ExpirationDate	A WebEx-maintained date and time at which the user's account expires.
Prod/ServiceAnnouncement	Indicates product or service announcements.
TrainingInfo	Indicates training information.
ElectronicInfo	Indicates electronic information.
Promos	Indicates promotions and special offers.
PressRelease	Indicates press releases.
UserEmail	The email address as stored in the user profile.
UserPhone	Indicates the phone number for the user.
MailInfo	Indicates the mail information for the user.
TimeZone	Determines the time zone for the geographic location of the user or user's office.
TimeZoneWithDST	A timezone description which is adjusted by DST. For example, GMT-7:00, Pacific (San Francisco)
Service	The type of service that the user has.

Schema attributes

Attributes	Description
Host	Indicates whether the user is the host for the meeting.
TelephoneConferenceCallOut	Indicates whether conference calling out of meetings is supported for the meeting.
TelephoneConferenceCallOutInternational	Indicates whether international calling out of meetings is supported for the meeting.
TelephoneConferenceCallIn	Indicates whether conference calling into meetings is supported for the meeting.
TelephoneConferenceTollFreeCallIn	Indicates whether toll-free calling into meetings is supported for the user.
SiteAdmin	Indicates whether the user has administrative privilege for the meeting.
VOIP	Specifies whether Voice Over IP telephony is enabled.
SiteAdminwithViewOnly	Indicates whether the current user is a site administrator with view only privilege.
LabAdmin	If TRUE, then user has access to the Hands-on Lab administration pages.
OtherTeleConferencing	Specifies whether a user account has the privilege to schedule a session with the other teleconferencing feature enabled. Default value depends on the configurations on the user's website.
TeleConferenceCallInInternational	Allows a user to access WebEx teleconferencing through international local call-in telephone numbers.
AttendeeOnly	If the value is TRUE, indicates that the user's role is attendee only. If the value is set to TRUE, then the host , siteAdmin , labAdmin and roSiteAdmin elements should be FALSE.
RecordingEditor	Indicates whether a user has the privilege to download WebEx Recording Editor from My WebEx > Support.
MeetingAssist	Enables Meeting Assist.
MeetingType	The meeting types of which the account is a part of.

Group attributes

The following table lists the group attributes:

Attributes	Description
ProductCodePrefix	Indicates the product label for the type of meeting.
Active	Indicates whether the type of meeting represented by an object of this type is enabled or disabled.
DisplayName	The display name for the meeting type.
PrimaryTollCallInNumber	The telephone number for a toll call-in teleconference.
PrimaryTollFreeCallInNumber	The telephone number for a toll free call-in teleconference.

Attributes	Description
GroupName	The name of the group or meeting type
MeetingTypeID	Specifies IDs for the meeting types whose detailed information you want to get.
ServiceType	The type of meeting being returned.

Provisioning Policy attributes

The following table lists the provisioning policy attributes for create:

Attributes	Description
AccountType	Host: Indicates whether the user is the host for the meeting. SiteAdmin: Indicates whether the user has administrative privilege for the meeting. SiteAdminWithViewOnly: Indicates whether the current user is a site administrator with view only privilege.
WebexID	WebEx ID of the user.
FirstName	First Name of the user.
LastName	Last Name of the user.
Email	Email ID of the user.
TelephoneConferenceCallOut	Indicates whether conference calling out of meetings is supported for the meeting.
TelephoneConferenceCallOutInternational	Indicates whether international calling out of meetings is supported for the meeting.
TelephoneConferenceCallIn	Indicates whether conference calling into meetings is supported for the meeting.
TelephoneConferenceTollFreeCallIn	Indicates whether toll-free calling into meetings is supported for the user.
VOIP	Specifies whether Voice Over IP telephony is enabled.
LabAdmin	If TRUE, then user has access to the Hands-on Lab administration pages.
OtherTeleConferencing	Specifies whether a user account has the privilege to schedule a session with other teleconferencing feature enabled. Default value depends on the configurations on the user's website.
TeleConferenceCallInInternational	Allows a user to access WebEx teleconferencing via international local call-in telephone numbers.
RecordingEditor	Indicates whether a user has the privilege to download WebEx Recording Editor from My WebEx > Support.
WelcomeMessage	Holds a welcome message for when people enter the meeting room.

Provisioning Policy attributes

Chapter 48: SailPoint Windows Local Connector

The following topics are discussed in this chapter:

Overview.....	481
Supported features	481
Supported Managed Systems	482
Pre-requisites	482
Administrator permissions	482
Configuration parameters.....	482
Schema attributes	483
Account attributes	483
Group attributes.....	484
Provisioning Policy attributes	485
Install and register IQService	485
Additional information	486
Unstructured Target Collector	486
Troubleshooting.....	487

Overview

The SailPoint Windows Local Connector manages User Accounts and Groups on Windows Operating System based computers through IQService. IQService uses WinNT ADSI service provider to connect to local users/groups for all versions of windows.

Supported features

SailPoint Windows Local Connector supports the following features:

- Account Management
 - Manages Windows Local Users as Accounts
 - Aggregation, Refresh Accounts
 - Create, Update, Delete
 - Enable, Disable, Unlock, Change Password
 - Add/Remove Entitlements
- Account - Group Management
 - Manages Windows Local Groups as Account-Groups
 - Aggregation, Refresh Groups
 - Create, Update, Delete

Configuration parameters

- Permission Management
 - Application can be configured for the following unstructured target collector to read permissions from respective end system:
 - Windows File Share - Read Windows File Share permissions directly assigned to accounts and groups.
 - The connector supports automated revocation of the aggregated permissions.
- Supports executing native before/after scripts for provisioning requests

References

- “Unstructured Target Collector” on page 486
- “IQService Before/After Scripts” on page 557”

Supported Managed Systems

Following versions of Microsoft Windows are supported by the SailPoint Windows Local Connector:

- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2008

Pre-requisites

- IQService must be installed on a Windows system. For more information, see “Appendix D: IQService”.
- Remote registry service must be started on the target system.
- Allow Exception for **File and Printer Sharing** in windows firewall.

Administrator permissions

User should be a member of **Administrators** group of Windows host computer which is to be managed.

Configuration parameters

The following table lists the configuration parameters of SailPoint Windows Local Connector:

Parameters	Description
IQService Host*	Host name or IP address where IQservice is installed.
IQservice port*	The TCP/IP port where the IQService is listening for requests (Default: 5050).
UserName*	User name of the account with administrator rights on the managed system (Syntax: <i>computerName\userName</i> or <i>userName</i>). For domain users it will be: <i>domainName\userName</i>
Password*	Password of user account mentioned in UserName field.

Parameters	Description
Server*	Host name or IP address of windows computer which is to be managed.
disableQualifyingLocalObjects	Flag to indicate whether aggregated objects must not be prefixed with server name. (Defaults to false. If set to true then aggregated object will not be prefixed with server name).
pageSize	Number of objects to fetch in a single request. Defaults to 1000

Note: Attributes marked with * sign are the mandatory attributes

Additional configuration parameters

The following table lists the additional configuration parameters of SailPoint Windows Local Connector:

Parameters	Description
disableNonLocalLookup	Set this parameter to false to read non-local (domain users/groups) group membership of local groups. Run IQService from any domain member server to read non-local group membership. The connector does not support provisioning of non-local group membership. Default: true

Schema attributes

This section describes the different schema attributes.

Account attributes

The following table lists the account attributes:

Attributes	Description
AutoUnlockInterval	Time interval for auto unlocking of locked user account.
Disabled	Flag to indicate if the user is disabled.
Description	User's description.
DirectoryPath	Fully qualified directory path WinNT://...
FullName	User's fullname.
groups	List of groups assigned to a user.
HomeDirectory	Location of the user's home directory.
Lockedout	Flag to indicate a user is locked out.
MaxStorage	The maximum amount of disk space the user can use.
MinPasswordLength	Minimum length of the user's password.
Name	Name of the account unqualified SAMAccountName.
objectSid	Windows SID.

Schema attributes

Attributes	Description
PasswordAge	Time duration of the password in use. This property indicates the number of seconds that have elapsed since the password was last changed.
PasswordExpired	Indicates if the password is expired.
PasswordNotRequired	Flag to indicate if the user requires a password.
PasswordUnchangeable	Flag to indicate if the user password can be changed.
Profile	User's profile.
PrimaryGroupID	ID of the user's primary group.
sAMAccountName	Fully qualified version of the sAMAccountName.
UserFlags	User Flag defined in ADS_USER_FLAG_ENUM.
BadPasswordAttempts	Number of consecutive Bad Password Attempts made last time.
LoginScript	File path of Login script file.
HomeDirDrive	Home Directory Drive of the user.
PasswordNeverExpires	Flag to indicate if the password never expires.
MaxPasswordAge	Indicates the maximum time interval, in seconds, after which the password must be changed.
MinPasswordAge	Indicates the minimum time interval, in seconds, before the password can be changed.
LastLogin	Date and time when user logged in last time.

Group attributes

The following table lists the group attributes:

Attributes	Description
Description	User's description.
DirectoryPath	Fully qualified directory path <code>WinNt://...</code>
GroupMembers	List of groups assigned to a group.
GroupType	Windows SID.
Members	List of users assigned to a group.
objectSid	Windows SID.
sAMAccountName	Fully qualified version of the sAMAccountName.

Provisioning Policy attributes

This section lists the provisioning policy attributes of SailPoint Windows Local Connector for create Account, create Group, and update Group.

Attributes	Description
For create Account	
sAMAccountName*	Name for user account to create. (Syntax: if disableQualifyingLocalObjects attribute in application configuration is unchecked then the format is sAMAccountName = <i>hostName\userName</i> . Otherwise sAMAccountName = <i>userName</i> .)
Password*	Password for new user account.
Description	Description of new user account.
Full Name	Full name of the user account.
Disable user account	Flag to create disabled user account.
User must change password on next logon	Flag to indicate whether user must change his password on next logon.
User cannot change Password	Flag to indicate whether user is allowed to change his password. If the value is false , user can change his password. Otherwise only system administrator can change his password.
Password never expires	Flag to indicate that user account password never expires until next password set.
For create Group	
sAMAccountName*	Name for group to create. (Syntax: if disableQualifyingLocalObjects attribute in application configuration is unchecked then sAMAccountName = <i>hostName\groupName</i> . Otherwise sAMAccountName= <i>groupName</i> .)
For update Group	
Description	Description of the group.
GroupType	Type of the group.
objectSid	Windows SID of group.
DirectoryPath	Fully qualified directory path <code>WinNt://...</code>

Note: Attributes marked with * sign are the mandatory attributes.

Install and register IQService

To install and register IQService, perform the following:

1. Create a directory in which you want to download the service. For example, `c:\iqservice`.
2. Extract the `IQService.zip` archive from the `IIQHOME\WEB-INF\bin\win` directory of the IdentityIQ installation into the created directory.

Additional information

3. Run the following command to install a Windows service named IQService.
`IQService.exe -i`
4. Start the service either from the Services Applet or from the command line by running the following command:
`IQService.exe -s`

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information
- **-u**: Uninstall the service. Removes the service components and clears the registry entries.

Trace Parameters (require a restart of the IQService):

- **-l [level]**: Trace Level 0-3
 - 0: Off
 - 1: Information
 - 2: Error
 - 3: Debug
- **-f [fileName]**: Trace File Name (For example, "C:\IQService\IQServiceLog.log")

Additional information

This section describes the additional information related to the Windows Local Connector.

Unstructured Target Collector

Windows Local unstructured target collector supports aggregating direct access permissions on resources such as shared files and folders from target system and correlate it with aggregated user accounts and groups using objectSid as the correlation key.

Pre-requisites for target aggregation

IQservice needs to be installed on Target Windows computer.

Target aggregation configuration parameters

The following table lists the different target aggregation configuration parameters:

Attributes	Description
IQService configuration parameters	
IQservice Host*	The host on which the IQService resides.
IQservice Port*	The TCP/IP port where the IQService is listening for requests.

Attributes	Description
Number of targets per block	Number or targets (files) to include in each block of data returned.
File share configuration parameters	
Path*	Path of file or directory. You can target a specific file or a directory and its sub-directories containing multiple files from which to extract the required data. If you target a directory, use the Wildcard and Directory Depth fields to narrow the query if possible.
Directories Only	Use to instruct to the collector to ignore files and just report back directory permission information. Valid only if Path value is directory path.
Directory Depth	The sub-directory depth from which to extract data. The Directory Depth field enables you to extend your query up to ten (10) sub-directories below the one specified in the Path field.
Wildcard	Use wild cards to target a particular file type of naming scheme. For example, to search only exe, use *.exe or to search only files with names beginning with New_ and New_.*
Administrator*	The administrator that has access to this share so you can collect permissions. This value can be domain\userName, computerName\userName, or userName.
Password*	The password associated with the specified administrator.
Rule configuration parameters (used to transform and correlate the targets)	
Creation Rule	The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ.
Correlation Rule*	The rule used to determine how to correlate account information from the application with identity cubes in IdentityIQ.
Provisioning related parameters	
Override Default Provisioning	Overrides the default provisioning action for the collector.
Provisioning Action	The overriding provisioning action for the collector.

Note: Attributes marked with * sign are the mandatory attributes.

Troubleshooting

1 - Error returned from IQService: Unspecified Error

The following error message is displayed for any Windows Local application request:

```
Error returned from IQService:Unspecified Error
```

Resolution: Perform the following:

1. Ensure that the Target system is up and accessible from IQservice host.
2. Ensure that the Username and Password provided in application configuration are correct.

Troubleshooting

3. If the target system is in a workgroup **Guest Only** option for **Sharing and security model for local accounts** in local policy will force all incoming network file sharing connections to authenticate as **Guest**.
To resolve this problem perform the following steps:
 - a. On the Windows Start menu, click **Start => Control Panel => Administrative Tools => Local Security Settings**.
 - b. In the left pane, expand **Local Policies => Security** options.
 - c. In the right pane, double-click **Network access: Sharing and security model for local accounts**.
 - d. Select **Classic - local users authenticate as themselves** and click **OK**.
4. If the target system is Windows Server 2003 Service Pack 2 then some Windows updates are missing from the system. Turn on the Windows updates and install the latest updates.
5. Ensure that exception for **File and Printer Sharing** in windows firewall is enabled.
6. If the problem still persists try restarting IQservice.

2 - Error returned from IQService: The network path was not found

When Remote registry service is not started on Windows computer the following error message is displayed:

Error returned from IQService: The network path was not found

Resolution: Ensure that Windows Service named, **Remote Registry Service** is started on the Windows managed system.

3 - Unspecified Error

The following error message is displayed for any Windows Local Connector operation after upgrading to latest version from version 6.0 Patch 5 or below.

Unspecified Error

Resolution: Perform following:

1. Navigate to IdentityIQ debug page.
2. Select **Application** from the object browser.
3. Select and open your application from the list.
4. If a line exists with the following text as the starting text, then delete the line and save the application
 "`<entry key="domain"`"

4 - Target aggregation failed as one of the path was not accessible

The target Aggregation failed as one of the path was not accessible.

Resolution: The `continueOnError` attribute must be set to true in the `targetSource` xml file to continue the target aggregation for other paths configured in unstructured target configuration.

5 - Access Denied error

Provisioning operation fails with the following error, when User Account Control (UAC) is enabled:

Access Denied

Resolution: Use one of the following options:

- To **Turn off User Account Control** for Microsoft Windows Vista or later, perform the following steps:
 - a. For Microsoft Windows Vista and Microsoft Windows Server 2008, open the **Control panel => User Accounts => Turn User Account Control** on or off

- b. For Microsoft Windows 7 onwards, open the **Control panel => User Accounts => User Accounts => Change User Account Control settings**

Or

- To have more granular control, without disabling UAC you can add the following entry in **Registry Editor:**
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy
Key Value: 1

Troubleshooting

Chapter 49: SailPoint Workday Connector

The following topics are discussed in this chapter:

Overview.....	491
Supported features	491
Prerequisites	492
Administrator permissions	492
Configuration parameters.....	495
Additional configuration parameter.....	496
Configuration to fetch the Custom attributes in Workday.....	501
Configuration to update Custom attributes in Workday	503
Schema attributes	504
Account attributes	504
Additional information	510
Future dated Workers	511
Fetch Workers by Organization type	512
Upgrade considerations.....	512
Troubleshooting.....	516

Overview

SailPoint Workday Connector is used to aggregate Worker records from Workday, and update email and phone.

Note: As Workday is in the process of decommissioning the support for TLS version 1.0, SailPoint Workday Connector would use TLS version 1.2 for connection purpose.

Supported features

SailPoint Workday Connector supports the following features:

- Account Management
 - Aggregation
 - Delta Aggregation

Note: Any changes done to custom/calculated attributes are not aggregated during Delta Aggregation.
 - Future data: supported for Hire, Terminate and On Boarding events
 - Custom attributes: ability to update custom attributes through REST API's
 - Update of following attributes:
 - EMAIL_ADDRESS_HOME
 - EMAIL_ADDRESS_WORK
 - ADDITIONAL_EMAIL_ADDRESS_HOME

Overview

- ADDITIONAL_EMAIL_ADDRESS_WORK
- WORK_MOBILE
- WORK_TELEPHONE
- HOME_MOBILE
- HOME_TELEPHONE
- USER_ID

Note: **USER_ID** is internally mapped to **USER_NAME** of Workday managed system.

Note: When upgrading IdentityIQ to version 7.3, ***PROVISIONING*** must be added as a feature string in application debug page.

Prerequisites

Note: If Workday Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

- A Workday administrator account and password for connecting to the given URL of the Workday tenant

Note: For more information on Workday administrator permissions, see “Administrator permissions”.

- User must create a Workday Integration System

For more information on creating Workday Integration System, “Configuring an Integration System in Workday” on page 501.

- Custom attributes configuration

For more information on configuring the procedures for Workday connector to be able to update the custom attributes through REST APIs, see “Configuration to update Custom attributes in Workday” on page 503.

Administrator permissions

Administrator login must have the following privileges:

- Create a user for accessing the Workday Integration System
- Create Integration System Security Group (Unconstrained)
- Provisioning Administrator Permissions: Workday connector now accepts second users in configuration. The first user would have read permission and second user would have write permission. Hence the updating account would be performed by the second user credentials. If the second users Username or Password is not provided then it would use the first users credentials to perform the update operation.
- For updating **User ID (User_Name on Workday managed system)** provide GET and PUT permission for the following security group for the Domain Security Policy:

Workday Accounts (Functional area: System)

- Perform the following for Integration System Security Group:
 - Add the user in the Integration System Security Group (Unconstrained)
 - Modify the Integration System Security Group to associate required permissions for the following Workday Web Services:
 - Maintain Contact Information Domain (Get and Put permissions required)
 - Get References (Only Get permission is required)
 - Modify the Integration System Security Group to associate the following domain required by Workday Integration System:

Get Permission	Get and Put Permission
<ul style="list-style-type: none"> - Public Worker Reports - General Staffing Information - Current Staffing Information - Time in Position - Organizations - Worker Data: Active and Terminated Workers - security policy to user group - Manage: Organization Integration 	<ul style="list-style-type: none"> - Worker Data: Home Contact Information - Worker Data: Work Contact Information - Worker Data: Work Email - Worker Data: Home Email

Note: The Workday fields on Workday Managed System marked as private are not accessible through Workday API and the fields marked as Public are accessible.

- (For Delta Aggregation only) Integration System Security Group which has the respective administrator user associated with, must have **View Completed Only** permission for the related Business Process Security Policy.

Note: If schema attributes are not related to any of the following events, then permission for that Business Process Type is not required.

Following are the list of events and related business process supported in delta aggregation:

Events	Business Process Type
Hire Employee	Hire
Onboarding	Onboarding
Terminate Employee	Termination
Change Personal Information	Personal Information Change
Contact Information Event	Contact Change
Change Job	Change Job
Change Legal Name	Legal Change Name
Change Business Title	Title Change

Overview

Events	Business Process Type
Add Retiree Status	Add Retiree Status
Assign Organization Roles	Assign Roles
Change Owner	Assign Self-Assign Roles
Assign Superior	Assign Superior
EMERGENCY_CONTACT_EVENT	Change Emergency Contacts
Change Organization Assignments for Worker	Change Organization Assignments for Worker
Change Primary Address	Change Primary Address
Contract Contingent Worker	Contract Contingent Worker
Create Change Order from Contingent Worker Contract	Create Change Order from Contingent Worker Contract
Create Primary Address	Create Primary Address
Edit Worker Additional Data	Edit Worker Additional Data
Maintain Employee Contracts	Employee Contract
End Additional Job	End Additional Job
End Contingent Worker Contract	End Contingent Worker Contract
Change Marital Status	Marital Status Change
Move to New Manager	Move to New Manager
Assign Worker	Move Worker (By Organization)
Move Workers Staffing	Move Worker (Supervisory)
Assign Workers	Move Workers (By Organization)
New Hire Provisioning	New Hire Provisioning
Change Preferred Name	Preferred Name Change
Request Worker	Request Worker
Submit Resignation	Submit Resignation
Transfer Contingent Worker	Transfer Contingent Worker
Transfer Employee	Transfer Employee

(Optional) Additional administrator permissions

- To access custom or calculated fields the user must have access to all the fields referenced in the calculation.
For more information on accessing the custom or calculated fields, see "Fetching calculated field on Workday to get WORKER_STATUS_c" on page 502.
- To access Additional Standard attribute user must have access to corresponding standard attributes.
For more information on accessing the additional standard attribute, see "XPATH to support additional standard attribute" on page 510.

Configuration parameters

The following table lists the configuration parameters of Workday Connector:

If the configuration attribute Don't allow terminated Accounts is checked then the account aggregation in workday will aggregate the user using the cost center organization where it will initially fetch all the cost center organization and then fetch workday account from each organization in chunks.

Note: Attributes marked with * sign are the mandatory attributes.

Parameters	Description
Workday URL*	This is valid URL to connect to the Human Resource module of workday.
Username*	The name of administrative user. Note: Username must always be in the following format: username@tenantname
Password*	Password of administrative user.
Effective Date Offset	Enter the number of days in advance to aggregate future hires and terminates. For example, enter 14 to aggregate workday worker, 14 days in advance of his joining date. Note: SailPoint recommends that the end user must perform full aggregation if the Effective Date Offset is changed. For more information, see "Future dated Workers" on page 511.
Chunk Size	The number of account to be fetched per page (Limit 1 to 999).
Integration System ID	Provide System ID of Integration System to fetch custom or calculated attributes.
Server Time Zone	Set this parameter, if termination related data must be fetched according to a particular time zone. By default the value is UTC . SailPoint recommends the use of Server Time Zone parameter during the Delta Aggregation operation. For example, if Workday server is in PST time zone then enter PST in this field.
Connection TimeOut	Provide the timeout value in minutes. Default value is 1 minute.
Don't Allow Terminated Accounts	Terminated or disabled accounts will not be aggregated if checked.
Credentials to update Workday fields	
Provisioning Administrator	User connected to Integration security group who would have permission to update Workday account.
Provisioning Administrator Password	Password of Provisioning Administrator.
REST API Custom attributes	
Enable REST API	Enables rest API configuration (Rest API credentials would be used to update custom attributes).
REST API Endpoint	REST API Endpoint of Workday server. For example, https://WORKDAY_HOST/ccx/api/v1/TENANT_NAME

Configuration parameters

Parameters	Description
Token Endpoint	Workday token API endpoint. For example, https://WORKDAY_HOST/ccx/oauth2/TENANT_NAME/token
Authorization Endpoint	Workday authorization API endpoint. For example, https://impl.workday.com/TENANT_NAME/authorize
Note: The values of REST API Endpoint, Token Endpoint and Authorization Endpoint can be obtained from View API Client in Workday application.	
Client ID	Workday API client ID.
Client Secret	Workday API client secret.
Refresh Token	Workday API refresh token. Note: Ensure that the token never expires.
Note: The values of Client ID, Client Secret and Refresh Token can be obtained from API Clients for Integrations==> API Client in Workday application.	

Note: (*Applicable only for full aggregation*) If the Don't Allow Terminated Accounts parameter is checked, the account aggregation in Workday application would aggregate the user using the cost center organization where it would initially fetch all the cost center organization and then fetch Workday account from each organization in chunks.

Note: SailPoint recommends to perform full aggregation if the following parameters are changed:
 - Effective Date Offset
 - Integration System ID
 - Server Time Zone
 - XPATH of existing attribute
 - New attribute added with XPATH to get correct result

Additional configuration parameter

Add the following entry in the application debug page:

Parameter	Description
pastEffectiveDateOffset	To aggregate an account using delta aggregation that has been back dated (for example, account details such as termination date which has been set prior to the last aggregation): <entry key="pastEffectiveDateOffset" value="30"/> In the above example, 30 is the number of days.

Parameter	Description
Effective_Date_For_Contact_Update	<p>Used in provisioning policy while updating contact details to provide effective date.</p> <p>Data type: Date</p> <p>Note: If the Worker is FUTURE HIRE and Effective_Date_For_Contact_Update is not provided in provisioning policy then Worker's Hire Date would be used as Effective Date for updating the contact details.</p>
Future_Hire_Events	<p>This list contains events from which worker can be hired in workday and these events would be used while fetching future hiring workers. For example,</p> <pre data-bbox="659 684 1426 984"><entry key="Future_Hire_Events"> <value> <List> <String>Contract Contingent Worker</String> <String>Onboarding</String> <String>Hire Employee</String> <String>Change Job</String> </List> </value> </entry></pre>
Future_Termination_Events	<p>This list contains from which workers can be terminated in Workday and these events would be used while fetching future terminating workers. For example,</p> <pre data-bbox="659 1148 1426 1427"><entry key="Future_Termination_Events"> <value> <List> <String>End Contingent Worker Contract</String> <String>Terminate Employee</String> <String>Change Job</String> </List> </value> </entry></pre>
Past_Termination_Events	<p>This list will be used for fetching past inactive workers till day of full aggregation. This list will contain termination events through which workers were terminated in WorkDay. Default certain events will be provided in this list.</p> <pre data-bbox="659 1550 1426 1818"><entry key="Past_Termination_Events "> <value> <List> <String>End Contingent Worker Contract</String> <String>Terminate Employee</String> <String>Change Job</String> </List> </value> </entry></pre>

Configuration parameters

Parameter	Description
Termination_Events	<p>Used for fetching past inactive workers till day of full aggregation. This list contains termination events through which workers were terminated in WorkDay. Default certain events would be provided in this list.</p> <pre><entry key="Termination_Events "> <value> <List> <String>End Contingent Worker Contract</String> <String>Terminate Employee</String> </List> </value> </entry></pre>
Past_Termination_Offset	<p>This offset would be used while fetching past inactive workers through mentioned past termination event list.</p> <pre><entry key="Past_Termination_Offset" value="100"/></pre> <p>If no offset is provided, then Connector would use 1 Jan 1970 to start of fetching terminated workers till today.</p>
SKIP_FUTURE_TERMINATION	<p>Boolean attribute to decide whether to skip future termination events while requesting the future data.</p> <pre><entry key="SKIP_FUTURE_TERMINATION_RECORDS" value="true"/></pre>
disableMultipleWorkerRecords	<p>Boolean attribute to decide whether to perform aggregation depending on LATEST_WORKER_RECORD calculated field result or not.</p> <pre><entry key="disableMultipleWorkerRecords" value="true"/></pre>
Termination_Attributes	<p>List would contain additional attributes to be populated when certain worker is terminating in future. Default four attributes would be there in list.</p> <p>For example,</p> <pre><entry key="Termination_Attributes"> <value> <List> <String>TERMINATION_DATE</String> <String>LAST_DAY_OF_WORK</String> <String>CONTRACT_END_DATE</String> <String>PRIMARY_TERMINATION_REASON_REFERENCE</String> </List> </value> </entry></pre> <p>Note: By default the above mentioned attributes would be a part of the future 'Termination Attributes'. Additional attributes can be appended in the above list.</p>

Parameter	Description
Delta _Aggregation_Events	<p>This list contains events list which would be used while performing delta aggregation. This list can be extended or any event can be removed from existing events list.</p> <pre data-bbox="669 380 1387 1562"><entry key="Delta_Aggregation_Events"> <value> <List> <String>Hire Employee</String> <String>Onboarding</String> <String>Terminate Employee</String> <String>End Contingent Worker Contract</String> <String>Change Personal Information</String> <String>Contact Information Event</String> <String>Change Job</String> <String>Change Legal Name</String> <String>Change Business Title</String> <String>Add Retiree Status</String> <String>Assign Organization Roles</String> <String>Change Owner</String> <String>Assign Superior</String> <String>EMERGENCY_CONTACT_EVENT</String> <String>Change Organization Assignments for Worker</String> <String>Change Primary Address</String> <String>Contract Contingent Worker</String> <String>Create Change Order from Contingent Worker Contract</String> <String>Create Primary Address</String> <String>Edit Worker Additional Data</String> <String>Maintain Employee Contracts</String> <String>End Additional Job</String> <String>Change Marital Status</String> <String>Move to New Manager</String> <String>Assign Worker</String> <String>Move Workers Staffing</String> <String>Assign Workers</String> <String>New Hire Provisioning</String> <String>Change Preferred Name</String> <String>Request Worker</String> <String>Submit Resignation</String> <String>Transfer Contingent Worker</String> <String>Transfer Employee</String> </List> </value> </entry></pre> <p>Note: To include or exclude any events in delta aggregation for the existing application, add Delta_Aggregation_Events entry and modify accordingly.</p>

Additional attributes for filter supported in Workday aggregation

Configuration parameters

Parameter	Description
Response Filter Map	The Configure_Response_Group map contains the list of attributes which can be customized based on the requirement and it would filter out the non required data from aggregation. Used to improve the performance of Workday Connector. For more information, see “Performance improvement” on page 513.
Past_Termination_Offset	Limits the aggregation of termination records based on offset days provided in the application. For example, <entry key="Past_Termination_Offset" value="100"/>. In this example, the aggregation would fetch the terminated records for the last 100 days only.
SKIP_FUTURE_TERMINATION_RECORDS	Boolean attribute to decide whether to skip future termination events while requesting future data. For example, <entry key="SKIP_FUTURE_TERMINATION_RECORDS" value="true"/> When set to true in the application, the aggregation would exclude all the future termination records.

Note

Application containing **Future_Data_Business_Process** would required to be removed and split the entry in child entry format for accurate results, that is **Future_Hire_Events** and **Future_Termination_Events**.

For example,

```
<entry key="Future_Data_Business_Processes">
<value>
<List>
    <String>Contract Contingent Worker</String>
    <String>Edit Position</String>
    <String>End Contingent Worker Contract</String>
    <String>Terminate Employee</String>
</List>
</value>
</entry>
```

The above entry key (Future_Data_Business_Processes) contains termination and hiring events as combined entry. This entry must be split as follows:

- **Future_Hire_Events**

```
<entry key="Future_Hire_Events">
<value>
<List>
    <String>Contract Contingent Worker</String>
    <String>Edit Position</String>
</List>
</value>
```

```

</entry>
• Future_Termination_Events
<entry key="Future_Termination_Events">
<value>
<List>
    <String>End Contingent Worker Contract</String>
    <String>Terminate Employee</String>
</List>
</value>
</entry>

```

Configuration to fetch the Custom attributes in Workday

This section describes the procedure for fetching the custom attributes in Workday.

The Workday connector supports aggregating custom fields defined as part of Worker attributes in a Workday tenant. To aggregate the values of custom and calculated fields from Workday into IdentityIQ, perform the following:

- Configure an Integration System in Workday
For more information on configuring an Integration System in Workday, see “Configuring an Integration System in Workday” on page 501.
- Fetch calculated field on Workday to get WORKER_STATUS__c

The WORKER_STATUS__c default schema attributes require additional configuration before their values can be aggregated from a Workday application.

Configuring an Integration System in Workday

Perform the following steps to configure **Workday Integration System**:

1. Search and click on **Create Integration Field Override Service** in Workday search available in the upper left of the interface to create new Integration Field Override Service.
2. Assign a name to the Integration Field Override Service and select the business object as Worker.
3. Add fields as per your requirement and provide a name for each field:
 - a. Click the **Plus** icon to add a new row.
 - b. Click **OK**.
 - c. Choose a name that will match the schema attribute in IdentityIQ.
 - d. Define the required settings for the new fields.

Note: For details related to LATEST_WORKER_RECORD and WORKER_STATUS, see “Fetching calculated field on Workday to get WORKER_STATUS__c” on page 502.

 - e. Click **Done**.
 - f. Repeat Step a through Step e for each new field.
4. Create a new Workday Integration System as follows:
 - a. On your Workday Home page, navigate to **Integration System ==> Create Integration System**.
 - b. Provide a name for the Integration System
 - c. Select **New Using Template** and provide the value as **Document Transformation**.
 - d. Click **OK**.
 - e. Navigate to **Actions ==> Integration System==> Configure Integration Attachment Service**.
 - f. Click on Attachment column and select **Create Integration Attachment Service**.

Configuration parameters

- g. Attach an empty text file and click **OK**.
5. In your new Workday Integration System, add the field override service created in steps 1-2, as follows:
 - a. Navigate to **Custom Integration Services**.
 - b. Click on the **Plus** icon. Under **Custom Integration Services** add the field override service created in step 1-2.
 - c. Click **OK**.
6. Note the System ID for this Workday Integration System. It is required in for retrieving the custom or calculated attributes.
7. Search for **View Integration System**.
8. Enter the name of Workday Integration System you created in Step 4.
9. Click **OK**.
10. Map the new fields to the correct value:
 - a. Select **Integration System ==> Configure Integration Field Overrides**.
 - b. Select the calculated or custom field and map the correct value to it in **Override External Field**.
 - c. Click **OK**.
 - d. Click **Done**.
11. When applicable, grant the required permissions to the Service Account associated with the IdentityIQ application to aggregate the new custom or calculated fields.
12. In IdentityIQ, add the following to the Workday application configuration:
 - Add the Workday Integration System ID from Step 6 to the Integration System ID field.
 - For each field in Step 3, add related attributes to the account schema.

Note: Custom and calculated attributes added to the Workday schema must have "**_c**" appended to the field names created in Workday. For example, if you added **PREVIOUS_EMPLOYER** to Workday, you would add "**PREVIOUS_EMPLOYER_c**" in the application schema.

Fetching calculated field on Workday to get WORKER_STATUS_c

The **WORKER_STATUS** attribute can be added as a custom calculated field on the Workday Managed System. This can return a status of Active, Terminated, Retired, and so on.

There is already a Worker Status field in Workday. For this to be aggregated in IdentityIQ, you must create a new field with a name IdentityIQ can recognize.

Pre-requisites

Complete Step 1 - Step 2 and Step 4 through Step 6 of "Configuring an Integration System in Workday" on page 501.

Perform the following steps:

1. After creating the Integration Field Override Service, add a calculated field and provide a name for it as follows:
 - a. Click the **Plus** icon to add a new row.
 - b. Type a name
 - c. Click **OK**.
 - d. Click **Done**.
2. In the Search field in the top left corner, search for **View Integration System**.
3. Click **View Integration System**.

4. In the Integration System field, enter the name of integration system you created in Step 4 of “Configuring an Integration System in Workday” on page 501.
5. Select the item in the drop down and click **OK**.
6. Map the new fields to the correct value:
 - a. To the right of the ID, click the Actions icon and select **Integration System ==> Configure Integration Field Overrides**.
 - b. In the Configuration page, find the row for WORKER_STATUS
 - c. Click in **Override External Field** column and select the Workday field **Worker Status**.
 - d. Click **OK**.
 - e. Click **Done**.
7. Add the **Active and Terminated Workers** security policy permission to the user group Worker Data.
8. After you finish configuring the Workday system, sign in to IdentityIQ as an administrator and select one of the following options:
 - **New Workday Application:** Add the Workday Integration System ID from Step 6 to the Integration System ID field.
 - **Existing Workday Application:** Add the following to the application configuration:
 - Add the Workday Integration System ID from Step 6 above to the Integration System ID field.
 - Add the WORKER_STATUS__c attribute to the account schema.

Configuration to update Custom attributes in Workday

Workday connector provides the ability to update the custom attributes through REST APIs.

Perform the following procedure to update the custom attributes:

1. Using the **Create Custom Object** task, setup the Custom object on the Workday Managed System.
2. Using the **Edit Custom Object** task, add the following security domains to the custom object on Workday Managed System:
 - Integration Process
 - Integration Security
3. Using the **Register API Client for Integrations** task, register a new API Client for Integrations with the following configurations:
 - a. Client Name
 - b. Grant type: Authorization Code Grant
 - c. Access token: Bearer
 - d. Select the **Non-Expiring Refresh Tokens** checkbox.
 - e. Unselect the **Disabled** checkbox.
 - f. Select Scope (Functional Areas): Personal Data, Integration
4. Use the **View API Client for Integrations** to generate a refresh token.
 - a. Select the API client created in Step 3, navigate to **API Client Actions ==> API Client ==> Manage Refresh Tokens for Integration**
 - b. Select the Workday Account for which the refresh token must be generated and click **Next**.
 - c. Select **Generate New Refresh Token** checkbox and click **OK**.
5. Add custom attribute created in Step 1 in workday schema with **_c** appended at the end. For example, **vehicleType__c**.
Manually add the **restCustomObjectAliasMap** in the application map. If required the map can have multiple values.

Schema attributes

Syntax: <entry key="CustomAttribute_c" value="Customobject.customattribute"/>

For example,

```
<entry key="restCustomObjectAliasMap">
  <value>
    <Map>
      <entry key="vehicleType_c" value="vehicledetails.vehicletype"/>
    </Map>
  </value>
</entry>
```

Note: Aggregation takes more time for larger number of configure custom attributes and calculated fields.

Aggregation is delayed due to the following reasons:

- The value of calculated fields and custom attributes are calculated on run time, due to which aggregation is delayed as it requires extra amount of time for fetching the calculated values.
- If there are any customization rule in the configuration, the aggregation would be delayed due to multiple database calls (if set in the rule).

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following type of objects:

- Account: objects used when building identities Link objects.

Account attributes

The following table lists the account attributes:

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
USERID (Identity Attribute)	Worker ID of the worker.	Worker_Data/User_ID
WORKER_DESCRIPTOR (Display attribute)	Descriptor of the worker. Full name.	Worker_Reference/@Descriptor
FILENUMBER	Employee ID of the worker.	Worker_Data/Worker_ID
MANAGER	Current manager of the worker.	Worker_Data/Management_Chain_Data/Worker_Supervisory_Management_Chain_Data/Management_Chain_Data[last()]/Manager_Reference/@Descriptor
JOBTITLE	Business title of the worker.	Worker_Data/Employment_Data/Position_Data/Business_Title

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
JOBCODE	Job profile of the worker.	Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Job_Profile_Summary_Data/Job_Profile_Reference/ID[@type='Job_Profile_ID']
EMPLOYEE_TYPE	Type of Employee	Worker_Data/Employment_Data/Position_Data/Worker_Type_Reference/ID[@type='Employee_Type_ID']
FIRST_NAME	Legal first name of the worker.	Worker_Data/Personal_Data/Name_Data/Legal_Name_Data/Name_Detail_Data/First_Name
LAST_NAME	Legal last name of the worker.	Worker_Data/Personal_Data/Name_Data/Legal_Name_Data/Name_Detail_Data/Last_Name
CLASS	Combination of Position, Time Type and Employment Type.	concat(Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Position_Title,Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Position_Time_Type_Reference/ID[@type='Position_Time_Type_ID'],Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Worker_Type_Reference/ID[@type='Employee_Type_ID'])
DEPARTMENT	Cost center.	Concatenation of Worker/Cost Center/Name and Worker/Cost Center/Code
LOCATION	Location of the worker.	Worker_Data/Employment_Data/Position_Data/Business_Site_Summary_Data/Name
TEAM	Team in the organization data of worker.	Worker_Data/Organization_Data/Worker_Organization_Data/Organization_Data[Organization_Type_Reference/ID]
DIVISION	Sales channel in the organization data of worker.	Worker/Sales Channel/Name
COST_CENTER_HIERARCHY	Cost center hierarchy of the worker.	Worker/Cost Center Hierarchy/Name
MANAGER_ID	Employee ID of the current manager of the worker.	Worker/Manager/Employee ID

Schema attributes

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
HIREDATE	Hire date of the worker.	Worker_Data/Employment_Data/Worker_Status_Data/Hire_Date
TERMINATION_DATE	Termination date of the terminated employee.	Worker_Data/Employment_Data/Worker_Status_Data/Termination_Date
FULLPARTTIME	Type of employment full time or part time.	concat(Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Position_Title,Worker_Data/Employment_Data/Worker_Job_Data/Position_Data/Position_Time_Type_Reference/ID[@type='Position_Time_Type_ID'])
WORKER_NAME	Name of the worker	Worker_Data/Personal_Data/Name_Data/Preferred_Name_Data/Name_Detail_Data/@Formatted_Name
MIDDLE_NAME	Preferred Middle Name	Worker_Data/Personal_Data/Name_Data/Legal_Name_Data/Name_Detail_Data/Middle_Name
ON_LEAVE	Status of worker whether is he on leave	Worker_Data/Employment_Data/Worker_Status_Data/Leave_Status_Date[1]/@On_Leave
POSTAL_CODE	Postal Code of the city of a worker	Worker_Data/Personal_Data/Contact_Data/Address_Data[Usage_Data/Type_Data/Type_Reference/ID[@type='Communication_Usage_Type_ID']='WORK']/Postal_Code
COUNTRY	Country of a worker	Worker_Data/Personal_Data/Contact_Data/Address_Data[Usage_Data/Type_Data/Type_Reference/ID[@type='Communication_Usage_Type_ID']='WORK']/Country_Reference/ID[@type='ISO_3166-1_Alpha-3_Code']
CITY	City of a worker	Worker_Data/Personal_Data/Contact_Data/Address_Data[Usage_Data[@Public='true']/Type_Data[@Primary='true']/Type_Reference/ID[@type='Communication_Usage_Type_ID']='WORK']/Municipality
LEGAL_MIDDLE_NAME	Legal Middle Name of a worker	Worker/Legal Middle Name

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
FUTURE_DATE	Fetches the future date when the respective future action will be effective.	
FUTURE_ACTION	Fetches the future action that will be performed on the respective worker. For example, Onboarding, Hire, Terminate	
PHONE_WORK	Primary business phone number of worker	Worker_Data/Personal_Data/Contact_Data/Phone_Data[Usage_Data[@Public='true']/Type_Data[@Primary='true']/Type_Reference/ID[@type='Communication_Usage_Type_ID']='WORK']/@Formatted_Phone
PHONE_HOME	Primary home phone number of a worker	Worker_Data/Personal_Data/Contact_Data/Phone_Data[Usage_Data[@Public='true']/Type_Data[@Primary='true']/Type_Reference/ID[@type='Communication_Usage_Type_ID']='HOME']/@Formatted_Phone
COST_CENTER	Represents the organization name whose type is COST_CENTER	Worker_Data/Organization_Data/Worker_Organization_Data/Organization_Data[Organization_Type_Reference[ID[@type='Organization_Type_ID']='COST_CENTER']]//Organization_Name
WORKER_STATUS_c	Represent the current status of the worker, that is, active/terminated/retired	
Optional attributes		
<i>If user requires the image as type string add the following attributes</i>		
Note: After adding the image attribute in the schema, the time taken to fetch the response from workday would be more.		
IMAGE_NAME (type as String)	Image file name	Worker/Employee_Image/file name
IMAGE (type as String)	Image value as string.	Worker/Employee_Image/Image
<i>If required user must add the following attributes manually to Workday schema after upgrading to IdentityIQ version 7.3</i>		

Schema attributes

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
EMAIL_ADDRESS_WORK	Email address of the worker.	Worker_Data/Personal_Data/Contact_Data/Email_Address_Data[Usage_Data[@Public='true']/Type_Data[@Primary='true']/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'WORK']/Email_Address
EMAIL_ADDRESS_HOME	Home email address of the worker.	Worker_Data/Personal_Data/Contact_Data/Email_Address_Data[Usage_Data[@Public='true']/Type_Data[@Primary='true']/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'HOME']/Email_Address
ADDRESS_HOME	Home address of the worker.	Worker_Data/Personal_Data/Contact_Data/Address_Data[Usage_Data/Type_Data/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'HOME']/@Formatted_Address
ADDRESS_WORK	Work address of the worker.	Worker_Data/Personal_Data/Contact_Data/Address_Data[Usage_Data/Type_Data/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'WORK']/@Formatted_Address
ADDITIONAL_PHONE_WORK	Additional business phone number of the worker.	Worker_Data/Personal_Data/Contact_Data/Phone_Data[Usage_Data[@Public='true']/Type_Data[@Primary='false']/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'WORK']/@Formatted_Phone
ADDITIONAL_PHONE_HOME	Additional home phone number of a worker.	Worker_Data/Personal_Data/Contact_Data/Phone_Data[Usage_Data[@Public='true']/Type_Data[@Primary='false']/Type_Reference/ID[@type='Communication_Usage_Type_ID']= 'HOME']/@Formatted_Phone

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
ADDITIONAL_EMAIL_ADDRESS_WORK	Additional Email address of the worker.	Worker_Data/Personal_Data/Contact_Data/Email_Address_Data[Usage_Data[@Public='true']/Type_Data[@Primary='false']/Type_Reference/ID[@type='Communication_Usage_Type_ID']='WORK']/Email_Address
ADDITIONAL_EMAIL_ADDRESS_HOME	Additional Home email address of the worker.	Worker_Data/Personal_Data/Contact_Data/Email_Address_Data[Usage_Data[@Public='true']/Type_Data[@Primary='false']/Type_Reference/ID[@type='Communication_Usage_Type_ID']='HOME']/Email_Address
ORGANIZATION_NAME	Represent organization name whose type is BUSINESS_UNIT	Worker_Data/Organization_Data/Worker_Organization_Data/Organization_Data[Organization_Type_Reference[@Descriptor ='Business Unit']]/Organization_Name
CONTRACT_END_DATE	Represent the contract end date of worker.	Worker_Data/Employment_Data/Worker_Contract_Data/Contract_End_Date
ADDRESS_LINE_1	Represent business site's Address_line_1 of worker.	Worker_Data/Employment_Data/Position_Data/Business_Site_Summary_Data/Address_Data/Address_Line_Data[@Type ='ADDRESS_LINE_1']
STATE	Represent business site's country region.	Worker_Data/Employment_Data/Position_Data/Business_Site_Summary_Data/Address_Data/Country_Region_Reference/@Descriptor
LAST_DAY_OF_WORK	Represents the last day of work of the worker.	Worker_Data/Employment_Data/Worker_Status_Data/Termination_Last_Day_of_Work
COST_CENTER_REFERENCE_ID	Represent reference ID of organization type COST CENTER	Worker_Data/Organization_Data/Worker_Organization_Data/Organization_Data[Organization_Type_Reference[@Descriptor ='Cost Center']]/Organization_Reference_ID

Additional information

Attributes	Description	Source Data Element WD Object /Field Web Service Element File field
COMPANY_NAME	Represent organization name whose type is COMPANY	Worker_Data/Organization_Data/Worker_Organization_Data/Organization_Data[Organization_Type_Reference/ID[@type='Organization_Type_ID']='COMPANY']/Organization_Name

XPATH to support additional standard attribute

With this release of IdentityIQ, Workday connector provides support to aggregate fields in addition to the fields present in the default account schema.

To fetch the additional standard Workday fields during account aggregation, perform the following:

1. Add the attributes to the account schema.
2. Provide the XPATH for those attributes.

For example, to fetch work email address, add an entry in the **XpathAttributesMap** as follows, where the value of the entry would be the XPATH required to fetch the required field:

```
<entry key="XpathAttributesMap">
    <value>
        <Map>
            <entry key="EMAIL_ADDRESS_WORK"
value="ns1:Worker_Data/ns1:Personal_Data/ns1>Contact_Data/ns1>Email_Address_Data
[ns1:Usage_Data[@ns1:Public='true']/ns1>Type_Data[@ns1:Primary='true']/ns1>Type_
Reference/ns1>ID[@ns1:type='Communication_Usage_Type_ID']
='WORK']/ns1>Email_Address"/>
        </Map>
    </value>
</entry>
```

3. Add above entry key in the application debug.

Additional information

This section describes the additional information related to the Workday Connector.

Future dated Workers

1. Workday Connector fetches FUTURE DATED workers for the following events:

- Hire Employee
- Terminate Employee
- Onboarding
- End Contingent Worker Contract

This creates a mismatch in Delta Aggregation task result as Connector receives two records in the following scenarios:

- Contract Ended employee is Hire in future within Effective Offset Date
- Terminated employee is Hire as Contingent worker in future within Effective Offset Date
- Contracted employee is Hire as full time employee in future within Effective Offset Date

For example, Account scanned 446 and actual account would be 445.

2. To fetch future data using any additional business process apart from above listed events. Add **Future_Data_Business_Process** entry in application which will contain list of additional business processes. Add any additional business process event type ID in the above list.

For example,

```
<entry key="Future_Data_Business_Process">
    <value>
        <List>
            <String>Contract Contingent Worker</String>
            <String>Edit Position</String>
            <String>Change Business Title</String>
            <String>Transfer Employee</String>
        </List>
    </value>
</entry>
```

To find Business process transaction event types ID as follows:

- a. Search for **Transaction Types available for Subscription** report.
 - b. Find required transaction event type and click on related action of transaction.
 - c. Navigate to **Integrations IDs** and click on **View IDs**.
3. For example, if a contractor is future hired using **Contract Contingent Worker** which is not a part of the above mentioned default business processes, and FUTURE_ACTION and FUTURE_DATE is required in this case then update XPATH of FUTURE_ACTION and FUTURE_DATE as given follows:

• FUTURE_ACTION

```
ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Transaction_Log_Data/ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_Process_Type']='Hire Employee']/ns1:ID[@ns1:type='Business_Process_Type']|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Transaction_Log_Data/ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_Process_Type']='Terminate Employee']/ns1:ID[@ns1:type='Business_Process_Type']|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Transaction_Log_Data/ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_Process_Type']='Contract Contingent Worker']/ns1:ID[@ns1:type='Business_Process_Type']|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Transaction_Log_Data/ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_Process_Type']]
```

Additional information

```
_Process_Type']= 'End Contingent Worker  
Contract']/ns1:ID[@ns1:type='Business_Process_Type']  
  
• FUTURE_DATE  
ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Tra  
nsaction_Log_Data[ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_  
Process_Type']= 'Hire Employee']]/ns1:Transaction_Effective_Moment  
|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Tr  
ansaction_Log_Data[ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_  
Process_Type']= 'Terminate Employee']]/ns1:Transaction_Effective_Moment  
|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Tr  
ansaction_Log_Data[ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_  
Process_Type']= 'Onboarding']]/ns1:Transaction_Effective_Moment  
|ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Tr  
ansaction_Log_Data[ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_  
Process_Type']= 'End Contingent Worker  
Contract']]/ns1:Transaction_Effective_Moment |  
ns1:Worker_Data/ns1:Transaction_Log_Entry_Data/ns1:Transaction_Log_Entry/ns1:Tr  
ansaction_Log_Data[ns1:Transaction_Log_Type_Reference[ns1:ID[@ns1:type='Business_  
Process_Type']= 'Contract Contingent Worker']]/ns1:Transaction_Effective_Moment
```

Fetch Workers by Organization type

When **Don't allow terminated account flag** is selected, Workday connector uses default **COST_CENTER** organization type in full aggregation.

To change the default organization type, add the following entry in application debug page:

```
<entry key="Organization_Type_ID" value="organization_type_id_value"/>
```

For example, to change the default organization type to **SUPERVISORY** instead of **COST_CENTER**, add the following entry key in application debug page:

```
<entry key="Organization_Type_ID" value="SUPERVISORY"/>
```

Upgrade considerations

- Workday application has the following phone attributes:

- PHONE_WORK
- PHONE_HOME
- ADDITIONAL_PHONE_HOME
- ADDITIONAL_PHONE_WORK

When upgrading IdentityIQ to version 7.3 to workday managed system, default device type is set as Telephone.

For example, if user provides phone number +1 123 1234567, then in workday managed system phone number would be updated +1 (123) 1234567 (Telephone) means with default device type.

However, if customer wants to update other supported phone device type like Mobile, Fax, or Pager, provide the entry as follows from IdentityIQ:

- +1 640 8900837 (Mobile)
- +1 640 8900837 (Fax)
- +1 640 8900837 (Pager)

After providing the number in above format for phone device type the same will be updated in Workday managed system.

Provide the entry in the following format for defaultPhoneDevice in application debug page:

```
<entry key="defaultPhoneDevice" value="Mobile"/>
```

This adds Mobile as default phone device type if any other device type is not provided with phone number in Workday managed system.

By default the international code is +1. This code can be changed by providing the entry as follows in the application debug page:

```
<entry key="defaultCountryCode" value="+44"/>
```

This adds +44 as country code if any other country code is not provided in Workday managed system.

- When upgrading IdentityIQ to version 7.3 to workday managed system, the following attributes must be added to the application debug page with default value as **telephone**:

```
<entry key="primary_Work_Phone_Type" value="telephone"/>
<entry key="primary_Home_Phone_Type" value="telephone"/>
```

For example, the default value (**telephone**) can be changed as follows:

```
<entry key="primary_Work_Phone_Type" value="Mobile"/>
<entry key="primary_Home_Phone_Type" value="Fax"/>
```

Note: Workday Connector now supports updating of the following attributes:

- WORK_MOBILE
- WORK_TELEPHONE
- WORK_PAGER
- WORK_FAX
- HOME_MOBILE
- HOME_TELEPHONE
- HOME_PAGER
- HOME_FAX

After adding the above attributes to the application debug page, the respective XPATH must also be added in the application debug page.

- When upgrading IdentityIQ to version 7.3, a change in XPATH would change the value displayed by **State** schema attribute.

For example,

Old value: California

New value: USA-CA

Performance improvement

With the option of filter out response data, only the required data can be fetched for better performance in account aggregation.

When upgrading to IdentityIQ version 7.3, adding the following mentioned filter (filters help in adding/removing the attributes as required) in the application debug page would enhance the performance of the aggregation.

```
<entry key="Configure_Response_Group">
  <value>
    <Map>
      <entry key="Exclude_Companies" value="false"/>
      <entry key="Exclude_Company_Hierarchies" value="true"/>
      <entry key="Exclude_Contingent_Workers" value="false"/>
      <entry key="Exclude_Cost_Center_Hierarchies" value="false"/>
      <entry key="Exclude_Cost_Centers" value="false"/>
```

Additional information

```

<entry key="Exclude_Custom_Organizations" value="false"/>
<entry key="Exclude_Employees" value="false"/>
<entry key="Exclude_Location_Hierarchies" value="true"/>
<entry key="Exclude_Matrix_Organizations" value="true"/>
<entry key="Exclude_Organization_Support_Role_Data" value="true"/>
<entry key="Exclude_Pay_Groups" value="true"/>
<entry key="Exclude_Region_Hierarchies" value="true"/>
<entry key="Exclude_Regions" value="true"/>
<entry key="Exclude_Supervisory_Organizations" value="true"/>
<entry key="Exclude_Teams" value="true"/>
<entry key="Include_Account_Provisioning" value="false"/>
<entry key="Include_Background_Check_Data" value="false"/>
<entry key="Include_Benefit_Eligibility" value="false"/>
<entry key="Include_Benefit_Enrollments" value="false"/>
<entry key="Include_Career" value="false"/>
<entry key="Include_Compensation" value="false"/>
<entry key="Include_Development_Items" value="false"/>
<entry key="Include_Employee_Contract_Data" value="false"/>
<entry key="Include_Employee_Review" value="false"/>
<entry key="Include_Employment_Information" value="true"/>
<entry key="Include_Feedback_Received" value="false"/>
<entry key="Include_Goals" value="false"/>
<entry key="Include_Management_Chain_Data" value="true"/>
<entry key="Include_Organizations" value="true"/>
<entry key="Include_Personal_Information" value="true"/>
<entry key="Include_Qualifications" value="false"/>
<entry key="Include_Reference" value="false"/>
<entry key="Include_Related_Persons" value="false"/>
<entry key="Include_Roles" value="false"/>
<entry key="Include_Skills" value="false"/>
<entry key="Include_Succession_Profile" value="false"/>
<entry key="Include_Talent_Assessment" value="false"/>
<entry key="Include_User_Account" value="false"/>
<entry key="Include_Worker_Documents" value="false"/>
<entry key="Include_Benefit_Eligibility" value="true"/>
</Map>
</value>
</entry>

```

The following table mentions:

- the Schema attributes and flag mapping in reference to the **Configure_Response_Group** map
- it also states the required values of the flags to fetch the corresponding schema attribute during aggregation

Schema Attributes	Flag/s
COST_CENTER_HIERARCHY	"Include_Organizations" = "true" && "Exclude_Cost_Center_Hierarchies" ="false"
DEPARTMENT	"Include_Organizations" = "true" && "Exclude_Cost_Centers" ="false"
COST_CENTER	"Include_Organizations" = "true" && "Exclude_Cost_Centers" ="false"

Schema Attributes	Flag/s
COST_CENTER_REFERENCE_ID	"Include_Organizations" = "true" && "Exclude_Cost_Centers" ="false"
COMPANY_NAME	"Include_Organizations" = "true" && "Exclude_Companies"="false"
ORGANIZATION_NAME	"Include_Organizations" = "true" && "Exclude_Custom_Organizations"="false"
DIVISION	"Include_Organizations" = "true" && "Exclude_Custom_Organizations"="false"
TEAM	"Include_Organizations" = "true" && "Exclude_Custom_Organizations"="false"
ADDRESS_LINE_1	Include_Employment_Information="true"
STATE	Include_Employment_Information="true"
LOCATION	Include_Employment_Information="true"
JOBTITLE	Include_Employment_Information="true"
JOBCODE	Include_Employment_Information="true"
POSITION	Include_Employment_Information="true"
FULLPARTTIME	Include_Employment_Information="true"
CLASS	Include_Employment_Information="true"
EMPLOYEE_TYPE	Include_Employment_Information="true"
CONTRACT_END_DATE	Include_Employment_Information="true"
HIREDATE	Include_Employment_Information="true"
ON_LEAVE	Include_Employment_Information="true"
PRIMARY_TERMINATION_REASON	Include_Employment_Information="true"
TERMINATION_DATE	Include_Employment_Information="true"
LAST_DAY_OF_WORK	Include_Employment_Information="true"
MANAGER_ID	Include_Management_Chain_Data="true"
COUNTRY	Include_Personal_Information="true"
POSTAL_CODE	Include_Personal_Information="true"
ADDRESS_HOME	Include_Personal_Information="true"
ADDRESS_WORK	Include_Personal_Information="true"
CITY	Include_Personal_Information="true"
ADDITIONAL_EMAIL_ADDRESS_HOME	Include_Personal_Information="true"
ADDITIONAL_EMAIL_ADDRESS_WORK	Include_Personal_Information="true"
EMAIL_ADDRESS_HOME	Include_Personal_Information="true"
EMAIL_ADDRESS_WORK	Include_Personal_Information="true"
ADDITIONAL_PHONE_HOME	Include_Personal_Information="true"

Troubleshooting

Schema Attributes	Flag/s
ADDITIONAL_PHONE_HOME_DESCRIPTOR	Include_Personal_Information="true"
ADDITIONAL_PHONE_WORK	Include_Personal_Information="true"
ADDITIONAL_PHONE_WORK_DESCRIPTOR	Include_Personal_Information="true"
PHONE_HOME	Include_Personal_Information="true"
PHONE_HOME_DESCRIPTOR	Include_Personal_Information="true"
FIRST_NAME	Include_Personal_Information="true"
LAST_NAME	Include_Personal_Information="true"
LEGAL_MIDDLE_NAME	Include_Personal_Information="true"
MIDDLE_NAME	Include_Personal_Information="true"
WORKER_NAME	Include_Personal_Information="true"
CHECK_LATEST_WORKER_RECORD	No Flag
FUTURE_ACTION	No Flag
FUTURE_DATE	No Flag
USERID	No Flag
FILENUMBER	No Flag
WORKER_DESCRIPTOR	No Flag

Troubleshooting

1 - Test Connection is failing because of invalid workday host URL

Resolutions: Following are the possible exceptions and their solution:

- Test Connection Failed
- UnknownHostException
- FileNotFoundException
- HTTP response code: 500

Ensure that the Workday URL is correct and is case sensitive.

2 - Test connection fails with an error message

Test connection fails with the following error message due to Weblogic Server not using standard SUN HTTPS implementation provided by jdk:

```
openconnector.ConnectorException: javax.net.ssl.SSLKeyException: FATAL  
Alert:BAD_CERTIFICATE - A corrupt or unuseable certificate was received
```

the Weblogic Server uses weblogic.net.http.SOAPHttpsURLConnection class instead of
javax.net.ssl.HttpsURLConnection class.

Resolution: Start the Weblogic Server with the following argument so that it will use SUN HTTPS default handler:

```
-DUseSunHttpHandler=true jvm
```

Perform the following steps:

1. Open WebLogic Sever installed directory and navigate to \user_projects\domains\base_domain\bin directory.
2. Edit the startManagedWebLogic.cmd file and set JAVA_OPTIONS=-DUseSunHttpHandler=true%JAVA_OPTIONS%
Save the startManagedWebLogic.cmd file.
3. Edit the startManagedWebLogic.sh file and set JAVA_OPTIONS="-DUseSunHttpHandler=true" \${JAVA_OPTIONS}
Save the startManagedWebLogic.sh file.
4. Start the Managed Server.

3 - While provisioning of PHONE_HOME, PHONE_WORK, ADDITIONAL_PHONE_HOME, ADDITIONAL_PHONE_WORK attributes an error message appears

While provisioning of PHONE_HOME, PHONE_WORK, ADDITIONAL_PHONE_HOME and ADDITIONAL_PHONE_WORK attributes the following error message appears:

not Valid ID value for type="Phone_Device_Type_ID"

The above error message appears as the Workday Connector uses the default phone device reference id. Phone device reference id varies according to tenant. Workday application contains **ReferenceIDMap** that contains phone device type id map.

Resolution: To change the reference id of specific phone device type, modify **Phone_Device_Type_ID** map as follows:

```
<entry key="ReferenceIDMap">
    <value>
        <Map>
            <entry key="Phone_Device_Type_ID">
                <value>
                    <Map>
                        <entry key="Mobile" value="1063.1"/>
                        <entry key="Telephone" value="1063.5"/>
                        <entry key="Fax" value="1063.4"/>
                        <entry key="Pager" value="1063.6"/>
                    </Map>
                </value>
            </entry>
        </Map>
    </value>
</entry>
```

Perform the following to find **Phone_Device_Type_Reference_Id**:

1. Search **View Workday ID** report.

Troubleshooting

2. Enter **Phone Device Type** as class name in the report and click **OK**.
3. Click on **Related action** of phone device type ==> Integration IDs ==> View Reference IDs
The Reference ID values of Phone Device type ID are listed.

4 - If EMPLOYEE_TYPE value for any contingent worker is not correct

If EMPLOYEE_TYPE value for any contingent worker is not correct, for example, 382.2 which is the ID of the EMPLOYEE_TYPE.

Resolution: The application contains default map as follows for mapping the incorrect IDs with employee type:

```
<entry key="ReferenceIDMap">
  <value>
    <Map>
      <entry key="Employee_Type_ID">
        <value>
          <Map>
            <entry key="382.1" value="Consultant"/>
            <entry key="382.2" value="Contractor"/>
            <entry key="382.3" value="Vendor"/>
          </Map>
        </value>
      </entry>
    </Map>
  </value>
</entry>
```

The above default map, would map 382.2 key with **Contractor** string literal. For numeric value issues in EMPLOYEE_TYPE account attribute in IdentityIQ, if the actual Contractor string literal is not obtained then copy the value of the appropriate reference id of that employee type from Workday managed system and add it in Workday application **Employee_Type_ID** map.

For example, in IdentityIQ for a value of 382.2 in EMPLOYEE_TYPE perform the following steps:

1. In Workday managed system search for **Contingent Worker Type** and click on it.
2. For each row a related action option exists. Open **Related action ==> Integration ID ==> View IDs** and find to which type of contingent worker is 382.2 assigned.
3. After finding the correct value of the contingent worker type, add the value to the **Employee_Type_ID** map in IdentityIQ ==> Application debug page.

For the correct changes to be reflected perform full aggregation task.

5 - While updating contingent worker data an error message is displayed

While updating contingent worker data for existing applications, the following error message is displayed:

```
Invalid ID value. '21430' is not valid ID value for type='Employee_ID'
```

Resolution: If the provided Employee_ID belongs to the contingent worker, add the following map in the application debug page:

```
<entry key="EmployeeTypeMap">
  <value>
    <Map>
      <entry key="EmployeeTypes">
        <value>
```

```

<List>
  <String>Regular</String>
  <String>Casual</String>
  <String>Expatriate</String>
  <String>Fixed Term Contract</String>
  <String>Seasonal</String>
  <String>Temporary</String>
  <String>Trainee</String>
</List>
</value>
</entry>
<entry key="ContingentWorkerTypes">
  <value>
    <List>
      <String>Contractor</String>
      <String>Consultant</String>
      <String>Vendor</String>
    </List>
  </value>
</entry>
</Map>
</value>
</entry>

```

Note: This map would be present for new applications.

Perform the following to find the supported employee type:

1. In Workday managed system, search for reports of type **Employee Types**.
2. In the displayed report, verify if any additional employee type is listed on Workday managed system. If a new employee type is listed, add it in **EmployeeTypeMap ==> EmployeeTypes**.
For example, if additional employee type is found in **Trainee** then add the following entry under **EmployeeTypeMap ==> EmployeeTypes**:

```
<String>Trainee</String>
```

Perform the following to find supported contingent worker types:

1. In Workday managed system, search for reports of type **Contingent worker Types**.
2. In the displayed report, verify if any additional contingent worker type is listed on Workday managed system. If a new contingent worker type is listed, add it in **EmployeeTypeMap ==> ContingentWorkerTypes**.
For example, if additional employee type is found in **Trainee Consultant** then add the following entry under **EmployeeTypeMap ==> ContingentWorkerTypes**:

```
<String>Consultant</String>
```

6 - While updating Workday account email or phone attribute an error message appears

While updating Workday account email or phone attribute, the following error message appears:

The task submitted is not authorized

Resolution: The error message indicates that user is not authorized with some permission. The Workday administrator must verify if specific Integration security Group is added to **Maintain Contact Information (Web Service)**.

Troubleshooting

7 - While retrieving the Workers issue is been observed for BenefitEligibility or Supervisory Organization data

- By default connector retrieves **BenefitEligibility** data while retrieving the workers.

Resolution: To exclude retrieving **BenefitEligibility** data, add the following in application Configuration:

```
<entry key="Configure_Response_Group">
  <value>
    <Map>
      <entry key="excludeBenefitEligibility" value="true"/>
    </Map>
  </value>
</entry>
```

- By default Connector does not retrieve Supervisory Organization data while retrieving the workers.

Resolution: To retrieve **Supervisory Organization** data, add the following in application Configuration:

```
<entry key="Configure_Response_Group">
  <value>
    <Map>
      <entry key="Exclude_Supervisory_Organizations" value="false"/>
    </Map>
  </value>
</entry>
```

8 - When phone reference, employee type or Contingent worker type IDs are updated in Workday managed system, reference ID map is not updated in workday application xml

Resolution: When Reference IDs are updated in Workday, user must perform the following:

1. Delete the **ReferenceIDMap** entry from the debug page
2. Delete the following relevant entries from the XpathAttributesMap:
 - (*Phone Reference IDs*) WORK_MOBILE, WORK_TELEPHONE, HOME_MOBILE, and HOME_TELEPHONE
 - (*Employee Type Reference ID*) EMPLOYEE_TYPE
 - (*Contingent Worker Type Reference ID*) Contingent_Worker_Type
3. Perform aggregation.

Workday Connector updates the latest IDs in the application along with latest Xpath for the attributes listed in Step 2 above.

9 - When machine time is not in synchronize with Server Time Zone, an error message appears.

Delta Aggregation uses the last aggregation time set by IdentityIQ which results in the time difference on the machine with the following error message:

```
java.lang.RuntimeException - sailpoint.connector.ConnectorException:
openconnector.ConnectorException: Updated From must be less than or equal to Updated Through!
```

Resolution: Synchronize the machine system time with current time of the **Server Time Zone** added in the Workday application configuration.

For more information on Server Time Zone, see “Configuration parameters” on page 495.

10 - XPATH is not added to the new phone attributes added in the default Workday schema

After aggregation blank values are obtained in IdentityIQ as XPATH is not added to the following new phone attributes in default Workday schema:

- WORK_PAGER
- WORK_FAX
- HOME_PAGER
- HOME_FAX

Resolution: Add the XPATH attributes to the phone attributes as mentioned in “XPATH to support additional standard attribute” on page 510.

11 - Fixed intermittent issue of Timeout waiting for connection

Fixed intermittent issue of Timeout waiting for connection with the following error message:

Exception: org.apache.axis2.AxisFault: Timeout waiting for connection

Resolution: Specify http connection pool size by adding the following application attribute:

```
<entry key="http_conn_pool_size" value="50"/>
```

By default the value for http_conn_pool_size is 50.

12 - Workday certificate error

With 23 March 2018, Workday managed system has new Server certificates. Hence for customers on Workday Connector prior to 23 March 2018, the following error message is displayed:

```
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
```

Resolution: Perform the following steps:

1. Log on to Workday application instance or the login page.
2. Find the certificate details by clicking the padlock icon in the browser and copy the new certificate name.
3. Search the above copied certificate name in <https://www.digicert.com/digicert-root-certificates.htm> and download the root certificate.
4. Save the downloaded certificate in a folder. For example, D:\\workdaycert\\DigiCertGlobal-RootG2.crt
5. Navigate to the **JRE path => lib\\security** folder and ensure that the cacerts file is present.
6. As an administrator open a command prompt and navigate to the **JRE path => bin** folder.
7. Execute the following command:
keytool -import -alias <your alias> -file "<path to new downloaded certificate including filename>" -keystore "<path to cacerts including filename>" -storepass changeit

Troubleshooting

In the above command:

- <paths...> replace with the paths as identified in the above steps

- <alias> select an appropriate alias

For example, keytool -import -alias workday-cert -file
D:\\workdaycert\\DigiCertGlobalRootG2.crt -keystore cacerts -storepass changeit

8. Respond as Y when prompted to import the certificate.

9. Restart the Web Server.

Chapter 50: SailPoint XML Connector

The following topics are discussed in this chapter:

Overview.....	523
Supported features	523
Configuration parameters.....	523
Additional information	524
1 - Using XML Schema Definition (XSD)	524
2 - Using Document Type Definition (DTD)	525

Overview

The SailPoint XML Connector is a *read only* connector used to extract data from XML files. The Document Type Definition (DTD) or XML Schema Definition (XSD) can be used for data validation and to discover the schema attributes.

Supported features

SailPoint XML Connector supports the following features:

- Account Management
 - Aggregation, Discover Schema
- Account - Group Management
 - Aggregation

Configuration parameters

This section contains the information that this connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

The XML Connector uses the following connection attributes:

Attribute	Description
XML Data File Path	A semicolon separated list of XML files that contains account/group data.
XML Schema/DTD File Path	Enter the path and name of the XSD/DTD file that must be used for Discover Schema and Data Validation. You can specify only single schema file.
XML Element for Account	Specify XML ELEMENT to map with the Account.
XML Element for Account Group	Specify XML ELEMENT to map with the Account Group.

Additional information

Attribute	Description
File Transport	Specify how the file will be transferred. If the file resides locally on the application server select Local. If the file is on remote host, select FTP or SCP transport.
Host	In case of FTP/SCP transport, specify the hostname where the file is located.
User	In case of FTP/SCP transport, specify the username that will be used for authentication during the file transfer.
Password	In case of FTP/SCP transport, specify the password for the user that will be used for authentication during the file transfer.

Additional information

This section describes the additional information related to the XML Connector.

Following are the examples for XML Element for Account and XML Element for Account Group attributes of Application Configuration:

- 1 - Using XML Schema Definition (XSD)
- 2 - Using Document Type Definition (DTD)

1 - Using XML Schema Definition (XSD)

Using XML Schema Definition (XSD)

- XML Element for Account - "orderperson"
- XML Element for Account Group - "item"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:shiporder"
  xmlns:bks="urn:shiporder">

  <xsd:element name="shiporder">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="shipto" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="address" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:element name="orderperson" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="id" type="xsd:string" />
            <xsd:element name="email" type="xsd:string" maxOccurs="5"/>
          </xsd:sequence>
          <xsd:attribute name="id" type="xsd:string"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="title" type="xsd:string"/>
          <xsd:element name="note" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Figure 1—File-shiporder.xsd

```
<?xml version="1.0"?>
<x:shiporder xmlns:x="urn:shiporder">
  <orderperson id="ord1">
    <name>Smith</name>
    <id>001</id>
    <email>smith@test.com</email>
  </orderperson>
  <orderperson id="ord1">
    <name>Tom</name>
    <id>002</id>
    <email>tom@test.com</email>
  </orderperson>
  <item>
    <title>Box</title>
    <note>Type: Large</note>
  </item>
</x:shiporder>
```

Figure 2—File-shipping.xml

2 - Using Document Type Definition (DTD)

Using Document Type Definition (DTD)

- XML Element for Account - “user”
- XML Element for Account Group - “group”

```
<!ELEMENT MyUnix (user*,group*)>
<!ELEMENT user (name,id,home,phone*)>
<!ELEMENT group (grpname,comment?)>
<!ELEMENT name      (#PCDATA)>
<!ELEMENT id       (#PCDATA)>
<!ELEMENT home     (#PCDATA)>
<!ELEMENT phone    (#PCDATA)>
<!ELEMENT grpname  (#PCDATA)>
<!ELEMENT comment  (#PCDATA)>
```

Figure 3—File-UNIX.dtd

Additional information

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MyUnix SYSTEM "file:///C:/XMLs/unix.dtd">
<MyUnix>
    <user>
        <name>Test User</name>
        <id>1004</id>
        <home></home>
        <phone>84245</phone>
        <phone>66666</phone>
    </user>
    <user>
        <name>System user</name>
        <id>1005</id>
        <home>/local/home/scorp</home>
    </user>
    <group>
        <grpname>Artist</grpname>
        <comment>Group of artists</comment>
    </group>
</MyUnix>
```

Figure 4—File-UNIX.xml

Important notes

- Identifying XML Element for Account/Account Group mapping for XML Schema:
 - It must be an XML Element
 - The XML Element type must be **complexType**
 - Elements of this element should not have been defined as **complexType**
- Multi-valued attributes for DTD:
 - An asterisk or plus sign (* or +) can be used to define a multi-valued attribute. For example,

```
<attribute name>*
```

Or

```
<attribute name>+
```

This is same as DTD syntax to allow 1 or more and 0 or more occurrences of an element.
- Multi-valued attributes for XML Schema:
 - Add Element's attribute **maxOccurs** with value greater than 1 or **unbounded** to make an attribute a multi-valued attribute. This is also an XML Schema Syntax to allow more than 1 occurrence.
- Turning off XML validation:
 - Set the application configuration attribute **xmlValidation** to **false**.

Chapter 51: SailPoint Yammer Connector

The following topics are discussed in this chapter:

Overview.....	527
Supported features	527
Pre-requisites	527
Administrator permissions	527
Configuration parameter.....	528
Schema attributes	528
Account attributes	528
Group attributes.....	529

Overview

The SailPoint Yammer Connector is a *read only* connector which aggregates accounts and groups from one or more networks on Yammer (Enterprise Social Network).

Supported features

SailPoint Yammer Connector supports the following features:

- Account Management
 - Manages Yammer Users as Accounts
 - Aggregation, Partitioning Aggregation, Refresh Accounts
- Account - Group Management
 - Manages Yammer Groups as Account-Groups
 - Aggregation, Refresh Groups

Pre-requisites

Note: If Yammer Connector is behind proxy server, see the “Special Java Considerations” section of the *SailPoint IdentityIQ Installation Guide*.

The user will be walked through the **OAuth2** flow to generate the access token using the Cloud Commander and then pass it down to the Yammer connector. The connector will use this Access Token to make calls to any Yammer REST API.

Administrator permissions

The Administrator should be configured to have proper access rights for reading people information in the social network within the organization.

Configuration parameter

Access Token: A valid Access Token for the user is required which enables your application to access the user's information and take actions on their behalf. The application and user are verified with each API call by passing an access token along with each request.

Schema attributes

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports the following types of objects:

- **Account:** objects used when building identities Link objects.
- **Group:** schema used when building AccountGroup objects that are used to hold entitlements shared across identities.

Account attributes

The following table lists the account attributes ([Table 1—Account attributes](#)):

Table 1—Account attributes

Attributes	Description
Admin	The user is an administrator in a specified network or not.
Department	The department of the user in the company.
Email	Email of the user .
EmailType	Type of Email (primary or secondary).
FullName	Full name of the user.
Groups	Groups to which user is a member of.
JobTitle	The job title of the user.
NetworkDomain	The Domain of the network of which the user is a member of.
NetworkID	The ID of the network of which user is a member of.
NetworkName	The name of the network of which user is a member of.
UserID	The ID of the user.
UserName	The user name internally stored by Yammer for each user.
UserType	The retrieved identity is the user.
UserURL	The url which stores the property of user.
UserWebURL	The url for the web page of the user on Yammer.
Location	The location of the user.
Summary	The summary of the user.

Group attributes

The following table lists the group attributes ([Table 2—Group attributes](#)):

Table 2—Group attributes

Attributes	Description
GroupState	The group is active or not.
GroupType	The retrieved identity is group.
GroupWebURL	The URL of the web page for that group on Yammer.
GroupPrivacy	The group is private or public.
GroupURL	The URL stores the property of the group.
GroupDescription	The description given for the creation of the group.
GroupFullName	The full name of the group.
GroupName	The name of the group.
GroupMembers	Contains all the members of the group.
GroupID	The ID of the group.

Schema attributes

Appendix

This section contains information on the following:

- "A: Delta Aggregation" on page 533
- "B: Partitioning Aggregation" on page 539
- "C: Before and After Provisioning Action" on page 547
- "D: IQService" on page 553

Appendix A: Delta Aggregation

This appendix describes the following information.

Overview.....	533
Delta aggregation for Active Directory.....	533
Delta aggregation for Azure Active Directory	535
Delta aggregation for ADAM, SunOne and Tivoli	535
Configuring server for Delta Aggregation.....	536
Testing Delta Aggregation	536
Delta aggregation for JDBC.....	537
Delta aggregation for Lotus Domino	538
Delta aggregation for Google Apps	538

Note: For Delta Aggregation of SAP HR/HCM Connector, see ‘Delta Aggregation’ section of “Chapter 34: SailPoint SAP HR/HCM Connector”.

Overview

Delta aggregation can be requested by checking a box on the task definition that is then passed through to the connector. If the connector does not support delta aggregation, then it ignores this flag and performs normal aggregation. The connectors supporting delta aggregation uses various mechanisms depending on the managed system to read the changes that have taken place after certain benchmark. It can be **lastModData**, **usnChanged**, or so on, else that indicates the last aggregation benchmark. This marker is stored on the application. Hence to take advantage of delta aggregation at least one full aggregation is required which will allow the connector to store the starting point for next delta aggregation.

If the volume of changes are more than 40% of the total data on the server, a normal aggregation run is recommended before over delta aggregation. The delta aggregation run can be scheduled at suitable interval considering the amount of changes happening on the managed system.

Delta aggregation for Active Directory

Note: This includes changes such as user/group has been added/updated/deleted on the managed system. This version now supports aggregation of delta changes for Move and Rename operations.

Pre-requisite

- After creating a fresh version of IdentityIQ application of type Active Directory - Direct or after an upgrade to latest version of IdentityIQ, open the application configuration file in debug mode and ensure that the GROUPS_HAVE_MEMBERS (the following attribute) feature string has been added under the Group schema:

```
<entry key='groupMemberAttribute' value='member' />
```

Delta aggregation for Active Directory

For **DirSync** delta aggregator of Active Directory, user must be a member of Domain administrators group or must have **Read and Replicating directory changes** permissions along with read permissions on **Deleted objects container**.

- To provide **Replicating directory changes** permissions to the user, perform the following actions:
 - a. In the Active Directory Users and Computers browser menu, select the **View** option, right-click and ensure that **Advanced features** check box is enabled.
 - b. Right-click the domain node and select **property** option and open the Security tab.
 - c. Add user to the list of Security Principals.
 - d. Select the user and select **Allow** checkbox for Replicating Directory Changes permission.
- To provide **Read** permissions on **Deleted Objects Container** to user, perform the following actions:
 - a. Log on to any domain controller in the target domain with a user account that is a member of the Domain Administrators group.
 - b. Open a command prompt: navigate to Start, enter `cmd` and click **Enter**. Enter the following command and press **Enter**:

```
dsacls <Deleted objects container DN> /<takeownership>
```

In the above command line, Deleted objects container DN is the distinguished name of the deleted objects container.
For example, `dsacls "CN=Deleted Objects,DC=SailPoint,DC=Com" /takeownership`
 - c. To grant **Read** permission to the objects in the **Deleted Objects container** to a user type, enter the following command and press **Enter**:

```
dsacls < Deleted objects container DN > /G <domainName\userName >: LCRP
```

In the above command line, LCRP stands for the list object and read properties permission.
For example, `dsacls "CN=Deleted Objects, DC=SailPoint,DC=Com" /G Sailpoint\John:LCRP`

Configuring server for Delta Aggregation

The following delta aggregation modes are supported for Active Directory:

- **uSNChanged**: Based on uSNChanged attribute of Active Directory.
- **DirSync**: Based on DirSync feature of Active Directory.

In order to change the Delta iteration mode, modify the following entry (uSNChanged or DirSync):

```
<entry key="deltaIterationMode" value="dirSync"/>
```

Note: A full aggregation is required after selecting the above delta aggregation modes.

Testing Delta Aggregation

For delta aggregation to work properly, a start point would be required from where it would detect changes. To retrieve changes from the last iteration, a full aggregation must be performed first during which the reference point is maintained. Once the full aggregation completes, create a separate delta aggregation task to retrieve delta changes that occurred post the full aggregation.

Perform the following steps to test delta Aggregation:

1. Execute Account and Account - Group Aggregation task.
2. Create a task with delta aggregation flag set for Account and Account - Group Aggregation.

3. Perform Create/Update/Delete/Revoke operations for Accounts/Groups on the directory server.
4. Execute the respective delta aggregation task.
5. Confirm the changes have been retrieved into IdentityIQ.

Delta aggregation for Azure Active Directory

Delta Aggregation is supported for SailPoint Azure Active Directory. On Full Aggregation, the respective delta link of account and group aggregation are stored in the Application object which are used by Delta Aggregation to retrieve the changed data into IdentityIQ. Same values are updated after each respective delta aggregation.

- Note:**
- Account Delta Aggregation does not capture role assignment changes.
 - Delta Aggregation triggered without first full aggregation would trigger respective full aggregation by default.

Delta aggregation for ADAM, SunOne and Tivoli

Delta aggregation is supported for the following directory server types:

- ADAM - Direct
- SunOne - Direct
- IBM Tivoli DS - Direct

- Note:** This includes changes such as user/group has been added/updated/deleted on the managed system. This version now supports aggregation of delta changes for Move and Rename operations.

Pre-requisite

After creating a fresh version of IdentityIQ application of type ADAM - Direct, SunOne - Direct and IBM Tivoli DS - Direct or after an upgrade to latest version of IdentityIQ, open the application configuration file in debug mode and ensure that the GROUPS_HAVE_MEMBERS feature string have been added in respective Group schema. Ensure that the following entries exist for respective application types:

- (For ADAM - Direct): <entry key='groupMemberAttribute' value='member' />

For existing Active Directory - Direct or ADAM - Direct applications, add the following key into the applications configuration file. For new applications, modify the following key:

```
<entry key="deletedObjectsContainer" value="CN=Deleted Objects,DOMAIN"/>
```

Where DOMAIN is a place holder for the naming context where the account and accountgroup objects reside. Replace DOMAIN with the corresponding naming context.

For example,

```
<entry key="deletedObjectsContainer" value="CN=Deleted Objects,dc=sailpoint,dc=com"/>
```

- (For SunOne - Direct and IBM Tivoli Directory Server - Direct):

```
<entry key='groupMemberAttribute' value='uniqueMember' />
```

Configuring server for Delta Aggregation

The mechanism used under the hood for Delta Aggregation are:

- (For ADAM) The following delta aggregation mode is supported for ADAM:

- **uSNChanged**: Based on uSNChanged attribute of ADAM.

Note: A full aggregation is required after selecting the ‘Delta Aggregation Mode’ application configuration attribute.

- (For SunOne/Tivoli) changeLog

Following sections describe how to configure SunOne and Tivoli directory servers for delta Aggregation.

Note: After enabling the changelog on directory server, run Account and Account-Group full aggregation task before running delta aggregation.

Configuring SunOne Directory Server for Delta Aggregation

1. Locate the `dsconf` command of the SunOne directory server installation.
2. Using the command prompt execute the following command:
`> dsconf set-server-prop --unsecured -h <host> -p <non ssl port>
retro-cl-enabled:on retro-cl-deleted-entry-attr:nsUniqueId`
3. Enter the password for the directory server administrator.
4. Restart the server.

Configuring IBM Tivoli Directory Server for Delta Aggregation

1. Stop the Tivoli Directory server instance.
2. Locate the `idscfgchglg` for your Tivoli Directory Server installation.
3. To configure a change log for directory server instance, run the following command:
`idscfgchglg -I <Tivoli instance> -m 0`
4. Start the directory server instance.

Note: Confirm the server has been enabled for changelog, open a ldap browser and bind it to the ldap server instance and view the cn=changelog naming context. You should be able to see this naming context and the relevant change objects. Ensure this before you proceed with delta aggregation for SunOne and Tivoli directory servers.

Testing Delta Aggregation

For delta aggregation to work properly, it needs a start point from where it would detect changes. To retrieve changes from the last iteration, it needs to first perform a full aggregation during which it maintains its reference point. Once the full aggregation completes, you may create a separate delta aggregation task to retrieve delta changes that occurred post the full aggregation.

Perform the following steps to test delta Aggregation:

1. Execute Account and Account - Group Aggregation task.
2. Create a task with delta aggregation flag set for Account and Account - Group Aggregation.
3. Perform Create/Update/Delete/Revoke operations for Accounts/Groups on the directory server.

4. Execute the respective delta aggregation task.
5. Confirm the changes have been retrieved into IdentityIQ.

Note: For SunOne-Direct and Tivoli-Direct applications, the delta aggregation task would fail even though the full aggregation is successful in case if the server has not been configured for changelog. Hence, before performing full aggregation ensure the changelog has been configured for the directory server.

Delta aggregation for JDBC

1. Create two tables to capture the identities whose data is modified in the master tables:
 - One table for capturing the account whose attributes, entitlements, or direct permissions are modified in the master table for account, master table for its entitlements, or master table for its direct permissions respectively
 - Another table for capturing the account-group whose attributes are modified in the master table for account-group
 2. Each of the two tables must contain two columns such that
 - the first column will store the identity attribute defined in IdentityIQ and
 - the second column will store the action. Values of action can be Insert, Update, or Delete.
- (For Oracle database) For example, the SQL to create such a table
- ```
CREATE TABLE USER_DELTA(USER_ID VARCHAR2(20), ACTION VARCHAR2(10));
```
3. Assign the privileges to read and delete the records from the tables created in step 1. above to the connection user defined in the application configuration.
  4. Create the triggers on the following master tables:
    - Account table
    - Entitlements table for account
    - Permissions table for account
    - Account-Group table

The triggers on the Account table, the Entitlements table for the Account, and the Permissions table for the Account would write the Account Identity attribute, in the first table created in step 1., whose attributes, entitlements, or permissions have changed in the master tables. Similarly, the trigger on the Account-Group table would write the Account-Group Identity attribute, in the second table created in step 1., whose attributes have changed.

(For Oracle database) For example, the following trigger writes the user IDs in the first table, that have undergone some modifications, along with the respective action:

```
CREATE OR REPLACE TRIGGER T1
AFTER DELETE OR INSERT OR UPDATE ON USER_MASTER
FOR EACH ROW
BEGIN
 IF INSERTING THEN
 INSERT INTO USER_DELTA (USER_ID, ACTION)
 VALUES (:NEW.USER_ID, 'Insert');
 END IF;
 IF UPDATING THEN
```

## **Delta aggregation for Lotus Domino**

```
INSERT INTO USER_DELTA (USER_ID, ACTION)
VALUES (:NEW.USER_ID, 'Update');
END IF;
IF DELETING THEN
 INSERT INTO USER_DELTA (USER_ID, ACTION)
 VALUES (:OLD.USER_ID, 'Delete');
END IF;
END;
/
```

5. Once the above-mentioned steps are performed, the tables would start capturing the user or group IDs that have undergone changes in their master tables.
6. After the delta aggregation is finished, the tables would be reset/re-initialized and would start capturing the delta afresh.

## **Delta aggregation for Lotus Domino**

---

Delta Aggregation is supported for SailPoint Lotus Domino Connector. On Full Aggregation, the respective time and date values of account and group aggregation are stored in the Application object which are used by Delta Aggregation to retrieve the changed data into IdentityIQ.

### **Pre-requisites**

---

IQService is required for Delta Aggregation.

## **Delta aggregation for Google Apps**

---

Delta Aggregation is supported for SailPoint Google Apps Connector. On Full Aggregation, the respective time and date values of account and group aggregation are stored in the Application object which are used by Delta Aggregation to retrieve the changed data into IdentityIQ. Same values are updated after each respective delta aggregation.

- Note:**
- Group Delta aggregation does not capture the deleted groups. After group delta aggregation, groups deleted from managed system must be present in IdentityIQ.
  - Account Delta aggregation does not capture role assignment changes.
  - Group Delta aggregation does not capture create, update, delete role changes.

### **Pre-requisites**

---

Refresh token used in application must be generated with reports API scope in addition to the existing scopes.

# Appendix B: Partitioning Aggregation

---

This appendix describes the following information.

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| Overview.....                                                       | 539 |
| Partitioning Aggregation for JDBC Connector .....                   | 540 |
| Partitioning Aggregation for Active Directory Connector.....        | 540 |
| Partitioning Aggregation LDAP Connector.....                        | 541 |
| Partitioning Aggregation for Delimited Connector.....               | 542 |
| Partitioning Aggregation for IBM i Connector .....                  | 543 |
| Partitioning Aggregation for GoogleApps .....                       | 543 |
| Partitioning Aggregation for Tivoli Access Manager.....             | 544 |
| Partitioning Aggregation for Azure Active Directory Connector ..... | 545 |

**Note:** For Partitioning Aggregation of SAP HR/HCM Connector, see ‘Partitioning’ section of “Chapter 34: SailPoint SAP HR/HCM Connector”.

## Overview

---

Partitioning aggregation processes the connector data in parallel, across multiple threads and multiple hosts to help increase the performance of aggregation tasks.

- Partitioning Aggregation can be requested by clicking the **Enable Partitioning** check box on the aggregation task definition. When the partitioning is enabled during aggregation, the aggregation task builds separate request for each partition. For more information, see “Connector specific Partitioning”.
- Some connectors do not support “Partitioning Aggregation” feature. For such connectors Default Partitioning is applied. For more information, see “Default Partitioning”.

### Default Partitioning

Default Partitioning processes the connector data in parallel, across multiple threads and multiple hosts to help increase the performance of Account Aggregation tasks.

Default Partitioning can be requested by clicking the **Enable Partitioning** check box on the Account Aggregation task definition and by specifying objects-per-partition field (default:1000).

To use the Default Partitioning feature for any connectors, perform the following:

- Select the **Enable Partitioning** check box in Account Aggregation Task.
- In **Objects per partition** textbox specify how many object should be used in one partition.
- Save and execute the task.

**Note:** The 'Objects per partition' field is invalid if the connector already supports partitioning. Connector specific partitioning is applied and gets preference if configured. In case, when connector supports partitioning but it is not configured then Default Partitioning is applied. If object per partition field is blank then default value 1000 is considered.

### Connector specific Partitioning

When Connector Partitioning is used, then the partitioning criteria must be provided by the connector. Each partition is handled independently and configured at the application level. While forming the partitioning criteria ensure that all the objects on the server have been processed and nothing is skipped.

## Partitioning Aggregation for JDBC Connector

---

JDBC Connector supports the manual partitioning through configured SQL statements.

The **Partitioning Enabled** configuration parameter must be selected and the list of SQL statements/parameterized stored procedures must be specified in the **Partitioning Statements** textbox.

For example, if there is an employee data-set that has 100,000 rows with a sequential **employeeId** field, the partitioning statements that can be used are as follows:

```
select x,y,z from a where employeeId <= 10000;
select x,y,z from a where employeeId > 10000 AND employeeId <= 20000;
select x,y,z from a where employeeId > 20000 AND employeeId < =30000;
...
select x,y,z from a where employeeId > 90000;
```

The above example would have 10 partitions, handling approximately 10,000 accounts and the last sentence (with `employeeId > 90000`) handling larger number of accounts depending on the total number of employees in the system.

**Note:** An additional aggregation option “noAttributePromotion” has been added. If this attribute is set to true, the attribute promotion would be skipped during aggregation.

## Partitioning Aggregation for Active Directory Connector

---

With IdentityIQ version 7.3, auto partitioning can be performed using the **Allow Auto Partitioning** checkbox (See “Configuring searchDNs” on page 18).

The following section describes the manual configuration for partitions.

### Configuring partitions manually

---

Active Directory Connector supports the Partitioning Aggregation feature to enable faster retrieval of Active Directory data.

In Active Directory Connector, data can be partitioned by specifying a **searchDN** and/or a **searchFilter** as a partition entry. Active Directory Connector partition entries are the application configuration searchDNs list with each entry of the list treated as a single partition.

Typically, for a container based partitioning of data, define the searchDNs or partition list as follows:

```
<entry key="searchDNs">
 <value>
 <List>
 <Map>
```

```

<entry key="searchDN" value="ou=test1,DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(&(objectclass=user))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
<Map>
<entry key="searchDN" value="ou=test2,DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(&(objectclass=user))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
</List>
</entry>

```

And for filter based partition, define the searchDNs list or partition list as follows:

```

<entry key="searchDNs">
<value>
<List>
<Map>
<entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(&(objectclass=user) (sn=a*))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
<Map>
<entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(&(objectclass=user) (sn=b*))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
</List>
</entry>

```

As seen above, in the first example, the OUs on which the search is performed are different although the **searchFilter** is the same. Whereas, in the second partitions entry, the OUs are same, but the **iterateSearchFilter** values are different. Since the required key values are similar, we could have both the above examples coupled together into the application configuration of a single Active Directory Connector application. Active Directory Connector combines the **searchDN** value and the **iterateSearchFilter** value and considers it as the partition context, avoiding any additional required configurations.

- Note:** Each of the partitions specified has to be unique by way of the searchDN value or the iterateSearchFilter value. If not, the first partition would get aggregated skipping the subsequent duplicate ones.
- When there is no partitions list defined, the aggregation would execute over the baseDN and the iteraterSearchFilter only, even though the task definition has partitioning, enabled.
- Similarly, with partition list defined and partitioning is not enabled on the task definition, IdentityIQ would retrieve data from each searchDN entry in a sequential manner.

## Partitioning Aggregation LDAP Connector

---

LDAP Connector supports the Partitioning Aggregation feature to enable faster retrieval of LDAP Directory data.

In LDAP, objects can be retrieved by means of a **searchDN**, **searchFilter** and **searchScope**.

Typically, for a container based partitioning of data, define the searchDNs or partition list as follows:

```

<entry key="searchDNs">
<value>
<List>
<Map>

```

## Partitioning Aggregation for Delimited Connector

```
<entry key="searchDN" value="ou=test1,DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(objectclass=user))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
<Map>
<entry key="searchDN" value="ou=test2,DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(objectclass=user))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
</List>
</entry>
```

And for filter based partition, define the searchDNs list or partition list as follows:

```
<entry key="searchDNs">
<value>
<List>
<Map>
<entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(objectclass=user) (sn=a*))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
<Map>
<entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
<entry key="iterateSearchFilter" value="(objectclass=user) (sn=b*))"/>
<entry key="searchScope" value="SUBTREE"/>
</Map>
</List>
</entry>
```

As seen above, in the first example, the OUs on which the search is performed are different although the **searchFilter** is the same. Whereas, in the second partitions entry, the OUs are same, but the **iterateSearchFilter** values are different. Since the required key values are similar, we could have both the above examples coupled together into the application configuration of a single LDAP Connector application. LDAP Connector combines the **searchDN** value and the **iterateSearchFilter** value and considers it as the partition context, avoiding any additional required configurations.

- Note:** Each of the partitions specified has to be unique by way of the searchDN value or the iterateSearchFilter value. If not, the first partition would get aggregated skipping the subsequent duplicate ones.  
When there is no partitions list defined, the aggregation would execute over the baseDN and the iterateSearchFilter only, even though the task definition has partitioning, enabled.  
Similarly, with partition list defined and partitioning is not enabled on the task definition, IdentityIQ would retrieve data from each searchDN entry in a sequential manner.

## Partitioning Aggregation for Delimited Connector

---

Delimited Connector support the following types of partitioning modes:

- **Auto:** Auto mode will automatically calculate the number of partitions and objects per partition based on the hints provided by the aggregator.
- **Manual:** Manual mode allows to specify the number of objects per partition and would be split equally as possible in partitions. By default partitioning uses the **Auto** mode.

In manual mode, user has to click on Manually Defined radio button and provide the value for Number of objects per partition field.

For example, if the total number of entities in the file is 1050 and the number of objects specified is 100 per partition, then there would be 11 partition task created. The first 10 partitions would fetch 100 objects per partition. The last would fetch 50 objects.

## Partitioning Aggregation for IBM i Connector

---

To use the partitioning aggregation feature in IBM i Connectors, perform the following:

1. Enable partitioning on the aggregation task definition page by selecting the **Enable Partitioning** check box.
2. On the Define application page in the iteration partitioning section, enter the list of statements for partitioning. Separate each statement by new line. Each statement can be an expression as mentioned below:

```
A*
AB*
C*
GA*
```

The user profiles matching with that specific name will be retrieved. For example, if A\*, D\* or AB\* is mentioned then it will return all user profiles starting with A, D or AB respectively.

## Partitioning Aggregation for GoogleApps

---

GoogleApps Rewrite Connector supports Partitioning Aggregation by using email, givenName (First Name) and familyName (Last Name) as filters. An asterisk is required in the value.

The filter names for the attributes are as mentioned below:

Attribute Name	Filter Name
email	partitionEmail
givenName	partitionGivenName
familyName	partitionFamilyName

The filters are to be added as entries into the application xml file.

For example:

- One partition which brings all users with email ID's starting with an 'a' and another partition which brings all users whose first name starts with an 's' appear as follows:

```
<entry key="partitionEmail">
 <value>
 <List>
 <String>a*</String>
 </List>
 </value>
</entry>
<entry key="partitionGivenName">
 <value>
 <List>
```

## Partitioning Aggregation for Tivoli Access Manager

```
<String>s*</String>
</List>
</value>
</entry>
```

- Partition of users with last name starting with an F appear as follows:

```
<entry key="partitionFamilyName">
 <value>
 <List>
 <String>F*</String>
 </List>
 </value>
</entry>
```

# Partitioning Aggregation for Tivoli Access Manager

---

To use the partitioning aggregation feature in Tivoli Access Manager Connectors, perform the following:

1. Select the **Partition Enabled** check box.
2. Specify the criteria for partitioning in the **Partition Statements** text-box of the configuration parameter.  
**For example**, the statement A-M would be treated as one partition:
  - Tivoli Access Manager Connector would aggregate accounts, whose names start with the character between A and M, with A and M inclusive.

Following are the different types of partitioning supported by Tivoli Access Manager:

- **Range base partitioning**
    - **Aggregated Account with partitions having range 1-20:** Aggregates all users which start with 1,2,3,.....,20)
    - **Aggregated Account with partitions having range A-F:** Aggregate all users which start with A,B,C,D,E and F)
    - **Aggregated Account with partitions having range 1-20 and 30-40 (specify each range on new line):** Aggregates all users which start with 1,2,3,.....,20 and 30,31,32,.....,40)
  - **Wild card base partitioning**
    - **Start with wildcard (for example, John\*, 1\*):** Aggregates all users whose value start with John and 1
    - **Single characters (for example, John, Bill, Mahesh {each name on new line, new line characters are used as a separators}):** Aggregates only John, Bill and Mahesh
    - **Single numeric values (for example, 100, 12, 46 and so on with each on new line):** Aggregates only users which exactly matches with 100, 12, 46
    - **End with wildcard (for example, \*Joe):** Aggregates all users whose value end with Joe
- A new partition can be mentioned in the new line.

# Partitioning Aggregation for Azure Active Directory Connector

---

Azure Active Directory Connector supports partitioning aggregation based on search filters. To use partitioning feature perform the following:

1. Enable Partitioning on the aggregation task definition page by selecting the **Enable Partitioning** check box.
2. Add the following application configuration attribute:

```
<entry key="userPartitions">
```

The **userPartitions** configuration attribute is a multi-valued attribute. It's value consists of different search filters for the attributes which are filterable like accountEnabled, city, displayName, mail, usageLocation and so on.

For example,

```
<entry key="userPartitions">
 <value>
 <List>
 <String>startswith(displayName, 'J')</String>
 <String>startswith(givenName, 'Smith')</String>
 <String>accountEnabled eq true</String>
 <String>userPrincipalName eq 'Paul@contoso.onmicrosoft.com'</String>
 </List>
 </value>
</entry>
```

For large environments, for faster delta aggregation of the accounts, the connector supports partition delta aggregation.

*Supported operators are*

- Logical operators: **and**, **or**
- Comparison operators: '**eq**'(equal to), '**ge**' (greater than or equal to) and '**le**'(Less than or equal to)
- **startswith**
- **any** is supported while querying multi valued properties

For example,

- proxyAddresses/any(c:c eq 'smtp:Mary@contoso.com')
- proxyAddresses/any(c:startswith(c,'smtp:Mary@contoso.com'))



# Appendix C: Before and After Provisioning Action

---

This appendix describes the following information.

Overview.....	547
Before and After Provisioning Action for AIX/Linux/Solaris Connectors.....	547
Before and After Provisioning Action for IBM i Connector.....	549

## Overview

---

While managing account and group on any managed system, you may need to perform some custom action which is not available out of the box on the Managed System. This can be achieved through before and After action. Some of the connectors support Before and after action using Before and After rules configuration in Application. This appendix describes the same. The java code executed in rule would be specific to Connector and can perform any custom action.

## Before and After Provisioning Action for AIX/Linux/Solaris Connectors

---

For AIX/Linux/Solaris Connectors, you can configure before and after provisioning rule to support Before/After Actions. In the Before/After provision rule we can carry out any operation before/after the provisioning operation. This document describes the different steps required to perform the same.

### Pre-requisite

---

AIX/Linux/Solaris Connector application must be configured in IdentityIQ.

### Creating Before and After Provisioning Action

---

Perform the following procedure to use the Before and After Action functionality for UNIX Connectors:

1. Navigate to where UNIX application is configured.  
Open UNIX application Rules tab. Select the following option as required:
  - Before Provisioning Rule
  - After Provisioning Rule
2. Write java code in Rule Editor section. Specify the Rule Name and Save it.  
Select the rule name you saved in earlier step by using Select Rule option.  
Perform any provisioning task and check if before/after provisioning rule gets executed.  
For example, java code for After provisioning action which creates directory for user after Unix account is created:

## Before and After Provisioning Action for AIX/Linux/Solaris Connectors

```
import java.io.InputStreamReader;
import ch.ethz.ssh2.Connection;
import ch.ethz.ssh2.Session;
import ch.ethz.ssh2.StreamGobbler;
import java.util.*;
import sailpoint.object.ProvisioningPlan.ObjectOperation;
import sailpoint.object.ProvisioningPlan.ObjectRequest;
import sailpoint.object.ProvisioningPlan.AttributeRequest;
import sailpoint.object.ProvisioningPlan.AccountRequest;
import sailpoint.object.ProvisioningPlan.GenericRequest;
import sailpoint.api.*;
// Here I have hardcoded hostname, user, password,
// we can take this from Application config
String hostname = "127.0.0.1";
String username = "joe";
String password = "joespass";
try
{
String userId = null;
boolean operationCreate = false;
// Get the reuquest
List accountRequests = plan.getAccountRequests();
if(accountRequests != null){
for (AccountRequest acctReq : accountRequests) {
// Get the opertion
AccountRequest.Operation op = acctReq.getOperation();
if (op == AccountRequest.Operation.Create){
userId = acctReq.getNativeIdentity();
operationCreate = true;
}
}
}
if (operationCreate)
{
// Create a connection instance
Connection conn = new Connection(hostname);
// Now connect
conn.connect();
// Authenticate. Here we have used password authentication,
// you can use public key authentication as well.
boolean isAuthenticated = conn.authenticateWithPassword(username,
password);
if (isAuthenticated == false)
throw new IOException("Authentication failed.");
// Create a session
Session sess = conn.openSession();
// To customize implementation,
// you can execute any command/shell script here
if(userId != null){
String command="mkdir /tmp/"+ userId ;
sess.execCommand(command);
}
// Show exit status, if available (otherwise "null")
System.out.println("ExitCode: " + sess.getExitStatus());
// Close this session
sess.close();
// Close the connection
```

```

 conn.close();
}
}
catch (IOException e)
{
e.printStackTrace(System.err);
}

```

**Note:** This is an example of After Provisioning Rule for Create operation. User can configure rule for Create/Delete/Update operation as required. The java code which is executed in Rule should be modified accordingly.

## Before and After Provisioning Action for IBM i Connector

---

For IBM i Connector, Customer can configure before and after provisioning action to support Before/After Actions. JTOpen library provides API to execute Command or CL scripts on IBM i host which are used to perform pre/post provisioning actions.

### Pre-requisites

---

- IBM i application configured
- CL scripts configured on IBM i host.

### Creating CL scripts

---

Perform the following procedure to create the CL scripts on IBM i computer:

1. Create a library as follows:

CRTLlib library-name

For example, CRTLlib MTEST

2. Make the library as current library CHGCURLIB library-name

For example, CHGCURLIB MTEST

3. Create source physical file in that library CRTSRCDF QCLSRC

(QCLSRC is the standard naming convention in the IBM i for CLP source members).

4. Add member to ADDPFM file and enter the following details:

Physical File - QCLSRC

Library - MTEST

Member - Test123

Text - \*BLANK

Press <ENTER>

5. Enter the following command:

STRPDM

Select option “3 – work” with members and enter the following details:

File - QCLSRC

Library - MTEST

Press <ENTER>

6. To create members press **F6** option and enter the following details:

## Before and After Provisioning Action for IBM i Connector

```
Source member - TEST123
Source Type - CLP
Press <ENTER>
```

7. List of members will appear. To write CL script or edit member file:

```
Opt - 2 (edit)
```

```
Member - Test123
```

```
Type - CLP
```

The member file opens in seu editor.



```
Columns . . . : 1 71 Edit YASIRU/QCLSRC
SEU=> _____ SRCMBR
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
***** Beginning of data *****
***** End of data *****
```

Type I (for insert) on the first line as shown in the following figure and press **Enter**.



```
Columns . . . : 1 71 Edit YASIRU/QCLSRC
SEU=> _____ SRCMBR
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
I ***** Beginning of data *****
***** End of data *****
```

For more information of seu editor, see the following link:

<http://as400iseries.wordpress.com/2013/03/13/using-the-seu-editor/>

8. Write your CL script program in the seu editor and press **F3**.

Options for saving the file are displayed.

Enter **Y (Yes)** and press **Enter**. The file is saved.

9. To compile, use option 14 in front of member file name. Enter **Y** in the following field:

Delete existing object ..... Y Y=Yes, N=No

10. Navigate to where IBM i application is configured. Select IBM i application.

Open IBM i application Rules tab. Select the following option as required:

- Before Provisioning Rule
- After Provisioning Rule

11. Write java code in Rule Editor section. Specify the Rule Name and Save it.

Select the rule name you saved in earlier step by using Select Rule option.

Perform any provisioning task and check if before/after provisioning action gets executed.

For example, java code to run CL-script

```
import java.io.IOException;
import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AS400Exception;
import com.ibm.as400.access.AS400Message;
import com.ibm.as400.access.AS400SecurityException;
```

```

import com.ibm.as400.access.CommandCall;
import com.ibm.as400.access.ErrorCompletingRequestException;
import com.ibm.as400.access.ObjectDoesNotExistException;
AS400 system = null;
String host = "NDS400.isr.bmc.com";
String user = "UMITTAL";
String password = "UMITTAL4";
system = new AS400(host, user, password);
System.out.println ("Connected sucessfully!!!!");
CommandCall cmd = new CommandCall(system);

try{
if(cmd.run("CALL PGM(MTEST/Test123) != true){
// Test123 is member file in library MTEST which gets called here.
//similarly we can use - command.run("CRTLlib FOREST");
}
else {
AS400Message[] messagelist = cmd.getMessageList();
for (int i = 0; i < messagelist.length; ++i){
System.out.println(messagelist[i].getText());
System.out.println("Command Success!!!");
}
}
}catch(Exception e){
System.out.println("error" + e.getMessage());
}
System.out.println("ending program");

```

**Note:** In this example, hostname and related parameters have been hardcoded, you can access these parameter from IdentityIQ objects ( Application, AccountRequest, Items).

## **Before and After Provisioning Action for IBM i Connector**

# Appendix D: IQService

---

This appendix describes the following information.

Install and register the IQService for Windows.....	553
IQService Before/After Scripts .....	557

## Install and register the IQService for Windows

---

The IQService is a native Windows service that enables IdentityIQ to participate in a Windows environment and access information only available through Windows APIs. Following are connectors for which IQService must be installed and registered on windows host computer from where the respective connectors are accessible.

Connector	Location	Other Libraries
Active Directory	Remote or Local	<ul style="list-style-type: none"> <li>• .NET Framework version 4.5.x onwards</li> <li>• For Exchange Server <ul style="list-style-type: none"> <li>- Windows PowerShell version 3.0</li> </ul> </li> <li>• For Lync/Skype Server <ul style="list-style-type: none"> <li>- (<i>For Microsoft Lync server 2013</i>) Microsoft Lync Server Administrative Tools 2013</li> <li>- (<i>For Microsoft Skype for Business Server 2015</i>) Microsoft Skype Server Administrative Tools 2015</li> <li>- Windows PowerShell version 3.0</li> </ul> </li> </ul> <p><b>Note:</b> See “ Pre-requisite for Active Directory Connector”.</p> <p><b>Note:</b> IQService host must be in the same domain as that of Microsoft Lync\Skype for Business Server.</p>

## Install and register the IQService for Windows

Connector	Location	Other Libraries
Lotus Domino	Remote or Local	<ul style="list-style-type: none"> <li>.NET Framework version 4.5.x onwards on Windows 32-bit or 64-bit</li> </ul> <p><b>Note:</b> See “Pre-requisite for Lotus Domino Connector”.</p> <ul style="list-style-type: none"> <li>Lotus Notes client</li> <li>The PATH environment variable must contain the Notes data folder. For example, C:\Program Files\IBM\Notes</li> <li>Microsoft Visual Studio C++ 2015 Redistributable 14.0 (32-bit)</li> </ul>
Microsoft SharePoint Server	Must be installed on the computer having the same domain as that of SharePoint Server	<ul style="list-style-type: none"> <li>.NET framework version 4.5.x onwards</li> <li>Windows PowerShell version 3.0</li> </ul>
Microsoft Windows Local - Direct	Remote or Local	.NET Framework version 4.5.x onwards Must be installed locally to support revoking permissions.
Forefront Identity Manager Provisioning Integration Module	Remote	.NET Framework version 4.5.x onwards

**Note:** The \* sign represents that Framework version 4.0 is required. Rename app.config file present in the IQService directory to IQService.exe.config file and restart IQService.

### Pre-requisite for Lotus Domino Connector

IQService must be running as a 32-bit process in order to interact with 32-bit Lotus Domino Client. If IQService must be installed on a 64-bit Windows system perform the following before installing IQService:

1. Download Microsoft Windows SDK or .NET Framework SDK.
2. Run the following command from command prompt:  

```
<SDK Bin>\CorFlags.exe <IQServiceHome>\IQService.exe /32BIT+
For example, C:\Program Files\Microsoft SDKs\Windows\v7.1\Bin\x64>CorFlags.exe
C:\IQService\IQService.exe /32BIT+
This command converts the IQService.exe to a 32-bit application.
```

### Pre-requisite for Active Directory Connector

Active Directory Connector provides support for Serverless configuration. To get the closest DomainController for particular domain, connector calls IQService. As a pre-requisite to return the DomainController, Active Directory module for Windows PowerShell on IQService computer must be installed.

Active Directory module for Windows PowerShell must be enabled as follows:

1. Using Server Manager click **Add features**.

2. Under **Remote Server Administration Tools**, select **Role Administration Tools**.
3. Select Active Directory module for Windows PowerShell option under **AD DS and AD LDS Tools**.

## Installing and registering IQService

---

### Note the following:

- If IQService is installed, the IQService version must match the IdentityIQ server version, including the patch version. If you upgrade one you must upgrade the other, ensuring that the IQSERVICE patch version matches the IdentityIQ Application server.
- IQService can be installed on the following Windows Operating Systems:
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows Server 2012
  - Microsoft Windows Server 2008 R2
- While installing new instance of IQService, SailPoint recommends the following:
  - back up the current installation before uninstalling to aid with troubleshooting the new version.
  - uninstall IQService and then install the new version of IQService. This clears the entry in registry associated with existing IQService.

To install and register the IQService, perform the following:

1. Create a directory in which you want to download the service. For example, `c:\IQService`.
2. Extract the `IQService.zip` archive from the `IIQHOME\WEB-INF\bin\win` directory of the IdentityIQ installation into the created directory.
3. Run the following command to install a Windows service.  
`IQService.exe -i`

The above command installs IQService with name `IQService-Instance1` and at port 5050 (if available)

It registers the service with the new registry path,

`HKEY_LOCAL_MACHINE\SOFTWARE\SailPoint\IQService_Instances\IQService-Instance1`  
with the following keys:

- **port**: port to listen
- **tracefile**: path to the tracefile
- **tracelevel**: 0 (off)
  - 1 (Error)
  - 2 (Information) (This is default)
  - 3 (verbose)
- **maxTraceFiles**: maximum number of Trace log files that must get created before overwriting the older files.
- **traceFileSize**: maximum file size of a trace file in bytes
 

A new file is created when the current file exceeds this limit.
- **installDir**: absolute path of installation directory
- **serviceStatus**: indicates whether service is installed or un-installed

## Install and register the IQService for Windows

4. Start the service either from the Services Applet or from the command line by running the following command:

```
IQService.exe -s
```

Other command line options with this service are:

- **-d**: run in the foreground in debug mode instead of in the background using the service control manager
- **-k**: stop the service
- **-r**: remove the service
- **-v**: display version information
- **-u**: Uninstall the service. Removes the service components and clears the registry entries
- **-n**: (*Optional*) name of IQService for installing multiple instances. Default: IQService-Instance $x$ , where  $x$  is an incremental integer value.
- **-p**: (*Optional*) unique available port number specified at the time of IQService installation. Default: 5050. Incremental based on the next available port.
- **-? | h**: this is help output

### Example

To change the default name of IQService and port number, user can execute the following command:

```
IQService.exe -i -n "IQService-Demo" -p 6060
```

**Note:** Each IQService instance would have separate installation directory and no two instances would share the same directory.

Trace Parameters (require a restart of the IQService):

- **-l [level]**: Trace Level 0-3
  - 0: Off
  - 1: Error
  - 2: Information
  - 3: Debug
- **-f [fileName]**: Trace File Name. By default it is created in the installation directory created in Step 1.  
For example, "C:\IQService\IQServiceLog.log"

## Upgrading IQService

---

Perform the following:

1. Take backup of existing IQService installation.
2. Ensure that IQService is stopped through Applet Services or by using the following command:  
`IQService.exe -k`
3. Uninstall IQService using the following command:  
`IQService -u`
4. Extract latest IQService in the existing installation directory.
5. Install IQService using the following command:  
`IQService -i`
6. Start IQService from services or by using the following command:

```
IQService.exe -s
```

**Note:** If you have executed "IQService Public Key Exchange Task" for existing IQService, then it is recommended to follow the instructions mentioned in "Installing and registering IQService" on page 555 for installing and registering new IQService.

## IQService Public Key Exchange Task

---

By Default, the IQService uses a shared key encryption technique where IdentityIQ server and the IQService encrypts and decrypts data using a common key.

Optionally, communication between IdentityIQ and IQService can be configured to use dynamic keys. When configured, it uses the public/private pair key approach, where each side uses a public key to encrypt data sent to each side. The receiving end then decrypts the data using the local private key.

This approach requires the key exchange to be performed in IdentityIQ and IQService as part of securing a communication channel by public/private key.

To achieve this, **IQService Public Key Exchange Task** must be run which takes in a list of applications as input. For each application, the key is updated and used in the transmission.

**Note:**

- Once secured using the new dynamic key, only one IdentityIQ server (or cluster using the same database) can talk to a single IQService.
- Applications using the same IQService will be using the same public/private key.
- Ensure to pair the IQService version with the IdentityIQ server version deployed in the production environment.
- To use the dynamic keys for any newly added application using different IQService, the IQService Public Key Exchange Task must be executed.

## IQService Before/After Scripts

---

IdentityIQ provides most of the provisioning functionality for many systems through its connectors. Some systems provide better integration interface from Windows platform compared to other platforms. Hence connectors for such systems require IQService deployed on a Windows system. The IQService implementation performs the provisioning functions (such as Add User, Connect User to a Group) that are supported by the respective System. The IQService functions are called by the IdentityIQ connector implementation.

In addition to the basic action, some organizations may require supplementary actions performed by each function from Windows system. The IQService supports customization of the functions by allowing integrating before / after scripts implemented in any language. Following are some features of the IQService Before/After script:

- Centralized configurations (in IdentityIQ) for setting up Before/After scripts
- Supports Object Oriented scripting
- Script refers SailPoint library to get the request, result classes
- Can be executed with specific context
- Script can modify request/result

A script is a group of statements that perform one or more actions and manipulate request / result attributes. Scripts can be used to automate any required actions that are currently performed manually. Scripts called before processing the request are referred to as native before scripts and scripts called after processing the request are referred to as native after scripts.

## IQService Before/After Scripts

The scripts needs to be defined in a Rule and then configured for an Application in IdentityIQ. Based on the rule type, the connector would send the scripts to IQService that needs to be executed before / after processing the request. The IQService supports executing before and after Rules for Create, Modify, and Delete request operations.

## Writing a script

---

IQService divides scripts in the following categories:

- Scripts with Object Oriented support
- Scripts without Object Oriented support

### Scripts with Object Oriented support

Scripting languages with Object Oriented capabilities (for example, PowerShell) are popular because of their simplistic nature and easy to use. These scripts can form objects of any type by referring any library/assembly implemented in any language and call its methods.

Native scripts implemented in these languages have easier and more powerful access to request and result objects. IQService comes with a class library named `Utils.dll` which bundles all required classes to access the request and result objects. The inputs provided to the script would be in the form of process environment variables. The following table describes the environment variables created by IQService:

Name	Type	Before Script	After Script
Application	System.Collections.Hashtable	Read Only	Read Only
Request	SailPoint.Utils.objects.AccountRequest	Read/Write	Read Only
Result	SailPoint.Utils.objects.ServiceResult	Not Available	Read/Write

The data in the environment variables is in XML. The script creates respective objects using `Utils.dll`. Once the object is modified, the script should convert it to XML by calling `toxml()` method of the object and write the xml to a file at the path that is passed as the only argument to the script. The script returns non-zero value in case of error and writes the error message in the file at the path that is passed as the argument to the script. This failure is communicated to IdentityIQ as part of result.

### *Sample PowerShell before script*

Following is a sample PowerShell before script which modifies value of an attribute and add one new attribute to the request:

```
Refer to SailPoint class library Requires PowerShell v2 installed on the system.
Add-type -path utils.dll

Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);

Form the xml reader object
$xmlReader =
[System.xml.XmlTextReader] ([SailPoint.Utils.xml.XmlUtil]::getReader($sReader));

Create SailPoint Request object
$requestObject = New-Object SailPoint.Utils.objects.AccountRequest($xmlReader);

Loop through the attributes from the request
foreach ($attribute in $requestObject.AttributeRequests){
 if($attribute.Name -eq "description"){
 $attribute.value = "my description";#change value of the attribute
```

```

 }

Add a new attribute to request
$attributeObject = New-Object SailPoint.Utils.objects.AttributeRequest;
$attributeObject.Name = "otherMobile";
$otherMobileValues = New-Object System.Collections.ArrayList;
$otherMobileValues.Add("222-292-2929");
$otherMobileValues.Add("333-292-2929");
$attributeObject.Value= $otherMobileValues;
$attributeObject.Operation = "Set";
$requestObject.AttributeRequests.Add($attributeObject);

Write the request xml to file at the path passed as argument
$requestObject.toxml() |out-file $args[0];

```

### **Sample PowerShell after script**

Following is a sample PowerShell after script which ensures that the request was processed successfully and creates home directory at the path specified in the request:

```

Refer to SailPoint class library. Requires PowerShell v2 installed on the system.
Add-type -path E:\SVN\trunk\src\WinRPCGateway\IQService\bin\Debug\utils.dll

Read the environment variables
$srReader = New-Object System.IO.StringReader([System.String]$env:Request);
$srResult = New-Object System.IO.StringReader([System.String]$env:Result);

Form the xml reader objects
$xmlReader = [
System.xml.XmlTextReader] ([sailpoint.utils.xml.XmlUtil]::getReader($srReader));
$xmlReader_Result = [
System.xml.XmlTextReader] ([sailpoint.utils.xml.XmlUtil]::getReader($srResult));

Create SailPoint objects
$requestObject = New-Object Sailpoint.Utils.objects.AccountRequest($xmlReader);
$resultObject = New-Object Sailpoint.Utils.objects.ServiceResult($xmlReader_Result);

#Check if the request was processed successfully
if($resultObject.Errors.Count -eq 0){

 #Get Home directory path
 foreach ($attribute in $requestObject.AttributeRequests) {
 #Create Home directory
 if($attribute.Name -eq "TS_TerminalServicesHomeDirectory"){
 New-Item $attribute.Value -itemtype directory;
 }
 }
}

```

### **Scripts without Object Oriented support**

Non Object Oriented scripts do not support referring to the class library or a way of parsing XML. To have easy access to each attribute along with their operation and values, IQService creates process environment variables for each of the application and request attribute with name in the form **SP\_<OPERATION>\_<NAME>** for requests and **SP\_APP\_<NAME>** for application. For native identity, the environment variable would be **SP\_NativeIdentity**. These types of scripts have limited access to result and get only **SUCCESS** or **FAIL** in the **Result** environment variable. Hence the after scripts implemented using these scripting languages cannot modify any attribute/result. The before scripts can add, modify, or remove any attribute from the request. The script needs to write the newly added or modified attribute to the file at the path passed as an argument to the script in the form

## IQService Before/After Scripts

**SP\_<OPERATION>\_<NAME>=<VALUE>**. For removing the attribute from the request, write **/~<ATTRIBUTE\_NAME>** to the file. Value for the multivalued attribute is delimited by **/#**

Following is a sample batch after script which ensures that the request was processed successfully and creates home directory at the path specified in the request:

```
IF %Result% ==SUCCESS md %SP_Set_TS_TerminalServicesHomeDirectory%
```

## Creating a Rule

---

IdentityIQ (6.0) user interface does not have facility to create Native Rule applicable for IQService. Create a rule with any supported type from the user interface. Add the script to the Rule source and save the Rule. Navigate to the debug page, open the newly created Rule and perform following steps:

1. Change the rule type to one of the following types as appropriate:

Type name	Description
ConnectorBeforeCreate	Before script for create operation.
ConnectorAfterCreate	After script for create operation.
ConnectorBeforeModify	Before script for modify operation includes enable/disable, unlock.
ConnectorAfterModify	After script for modify operation includes enable/disable, unlock.
ConnectorBeforeDelete	Before script for delete operation.
ConnectorAfterDelete	After script for delete operation.

2. Add the following attributes to the Rule in the form:

```
<Attributes>
 <Map>
 <entry key=<NAME> value=<VALUE>/>
 </Map>
</Attributes>
```

Name	Description	Default Value
ObjectOrientedScript	Whether the rule source uses object oriented scripting.	False
disabled	Set to true if the rule should not be executed on the IQService side.	False
extension	Extension of the script.	.bat
program	Program/application that can execute this type of script.  <b>Note: Ensure that this program is installed on the system where IQService is running and is properly configured to execute the scripts.</b>	cmd.exe or cmd
timeout	Time interval (in seconds) for which IQService should wait for script to return. After this interval, IQService aborts the script.	10

## Configuring the Rules in Application

With this releases, IdentityIQ user interface does not have facility to configure Native Rule applicable for IQService in Application. Navigate to the debug page, open the application and add <nativeRules> under Attributes map with list of names of the Rules that must be configured for this application.

For example:

```
<entry key="nativeRules">
 <value>
 <List>
 <String>AfterCreate-Powershell</String>
 <String>BeforeCreate-Powershell</String>
 <String>BeforeModify-Batch</String>
 </List>
 </value>
</entry>
```

## **IQService Before/After Scripts**