

Identity Connectors

- [About Identity Connectors](#)
- [Project Polygon](#)
- [Connector List](#)
 - [Local vs Remote, Java vs .NET](#)
- [Connector Development and Maintenance](#)
- [The Story of Identity Connectors as Seen by Evolveum](#)
- [See also](#)
- [External links](#)

About Identity Connectors

MidPoint Open Source Identity and Access Management is using a connector framework that is based on Identity Connector Framework (ICF) originally created by Sun Microsystems. This framework provides a layer that separates the identity management system from the target and source systems. The framework supports Java and .NET connectors for that purpose. The original Sun ICF project is no longer publicly maintained. But the open source community picked up the development thread and the code of the framework is currently maintained by several independent companies. The two primary entities that maintain the framework code are:

- Evolveum (MidPoint project)
- Tirasa (ConnId project)

The framework code is maintained in [ConnId Project](#) at github. MidPoint is currently using this ConnId framework.

The **framework** is maintained jointly by all these three companies and also other contributors. But not the connectors. The **connectors** are developed and maintained in several projects. There are mostly non-technical reasons for this separation such as licensing issues, philosophical differences and business strategies. However the **common framework** makes the connectors **compatible**. Therefore connectors from several projects can be used together in one solution.

The midPoint team tends to abstract from non-technical reasons that hinder cooperation. Therefore we have started Project Polygon to collect **all usable connectors** that we are aware of across all the projects.

Project Polygon

The midPoint team tends to abstract from non-technical reasons that hinder cooperation. As all the connectors should now be compatible we have created a project (or rather a meta-project) that maintains references to **all the known connectors** that work with recent versions of ConnId framework. The goal of the project is to collect the connectors, test them, fix the issues (and contribute fixes back to original projects) and track the state of each connector. The project started in April 2014.

The list contains the best information about connectors that we have. Nevertheless some data may not be entirely correct. E.g. there is a couple of connectors "inherited" from Sun ICF and we are not sure about their status. Any reports about connector testing or usage are greatly appreciated.





Connector List

Connector	Maintained by (Origin)	Recommended Version	Tested by	Sample	Prov	Sync	Connects to	Note
DatabaseTable Connector	Evolveum (Sun ICF)	1.4.3.0	Evolveum	✓	✓	✓	Generic database table (JDBC). Tested with MySQL, PostgreSQL, Oracle, MS SQL	This connector originated from Sun ICF, taken over by OpenICF and then taken over by Polygon. We do not recommend use of the original or OpenICF version.
CSV Connector	Evolveum	2.1	Evolveum	✓	✓	✓	Manipulates content of CSV-formatted files.	Good for integration to HR-like source systems that export data to CSV. Rewrite of the CSV connector from scratch.
LDAP Connector	Evolveum	1.5.1	Evolveum	✓	✓	✓	LDAP-based directory servers. Complete rewrite based on Apache Directory API. Apache-licensed.	This is an LDAP connector completely rewritten from scratch. It is using Apache Directory API and it is designed and built to work with recent ConnId versions and to take all the advantages of that. Although this connector may not yet have all the exotic features (such as support for AD LDAP quirks) it is the way forward. Use this connector whenever possible.

Active Directory Connector (LDAP)	Evolveum	1.5.1	Evolveum	✓	✓	✓	LDAP-based connector for Active Directory servers. Microsoft Exchange support via WinRM interface and powershell scripts.	This is a specialization of the new LDAP Connector that supports Microsoft LDAP quirks. Tested on Windows 2008 R2 server. Distributed in LDAP connector bundle. See Active Directory with LDAP connector .
UNIX	Connld		Evolveum		✓		Linux (RedHat, Ubuntu)	
eDirectory Connector	Evolveum	1.5.1	Evolveum	✓	✓	?	Novell/NetIQ eDirectory	Distributed in LDAP connector bundle. Explicit subscription is needed to support this connector.
Office365	Evolveum (contributed)		Community	✓	✓		Microsoft Office 365 / Azure Active Directory	
GitLab	Evolveum	1.0	Evolveum	✓	✓		GitLab server	Deprecated
GitLab rest	Evolveum	1.0	Evolveum	✓	✓		GitLab server	
Liferay	Evolveum	1.0	Evolveum	✓	✓	✓	Liferay Portal	Tested on 6.2-ce-ga4, support ACCOUNT and assignments to Roles and Org. structure over ID
AD (JNDI) (obsolete)	Connld		Evolveum				Active Directory (using LDAP)	Not recommended for use. This connector has many limitations. Use Polygon Active Directory Connector instead.
CMD	Connld	0.2	Connld	✓	✓	✗	Executes arbitrary commands	Does not seem to support object renaming.
CSV Directory	Connld		Connld					
Google Apps	Evolveum (contributed)	1.4.2.17	Community	✓	✓		Google API and OAuth	
OpenAM	Connld							
DB2 Connector	OpenICF (Sun ICF)			?	?	?		No reports
MySQLUser Connector	OpenICF (Sun ICF)			✓	✓	✗	MySQL, manages MySQL database accounts (users).	This is not for table content, use DatabaseTable Connector instead. Activation (disable/enable) not supported by the connector.
Oracle Connector	OpenICF (Sun ICF)			✓	✓	✗	Oracle Database Server, manages Oracle database accounts (users).	This is not for table content, use DatabaseTable Connector instead.
ScriptedSQL Connector	OpenICF (Sun ICF)	1.1.2.0.em3	Evolveum	✓	✓	✓	Very generic database connector based on Groovy/JavaScript scripting.	For databases with data in more than one table with joins, or when procedures are to be called.
FlatFile Connector (historic)	OpenICF (Sun ICF)			?	?	?		Not tested. Seems to be obsoleted by CSV and CSVFile connector.
XML Connector	OpenICF (Sun ICF)			?	?	?		Not tested, probably obsolete.
VMS Connector	OpenICF (Sun ICF)			?	?	?		Not tested, probably obsolete.
OpenPortal Connector	OpenICF			?	?	?		Not tested, probably obsolete.
SPML Connector	OpenICF			?	?	?		Not tested
SAS Connector	Connld						SAS Metadata Server	
Lotus Notes Connector	OpenICF	1.0 (developed by Evolveum)	Evolveum	✗	✓	✗	Lotus Notes	Tested and deployed with older version of OpenICF framework
Atlassian JIRA	Evolveum	1.0	Evolveum	✓	✓	✗	JIRA	Only for push and get profile avatar picture with resizing - Deprecated
Atlassian JIRA	Evolveum	2.0	Evolveum	✓	✓	✗	JIRA	
Atlassian Confluence	Evolveum	1.0	Evolveum	✓	✓	✗	Atlassian Confluence (Wiki)	Only for push and get profile picture with resizing
Atlassian Bitbucket	Evolveum	1.0	Evolveum	✓	✓	✗	Atlassian Bitbucket	Only for push and get profile picture with resizing
Box Connector	Evolveum	1.0	Evolveum	✓	✓	✗		
SAP Connector	Evolveum	1.0.0.0	Evolveum	✓	✓	✓	SAP	Tested on SAP System (R07) Netweaver 7 EHP 2 (aka 7.31)

Drupal 7 Connector	Evolveum	1.0.0.2	Evolveum	✓	✓	✗	Drupal 7	Tested on Drupal 7.33, 7.53
SmartSmartRecruiters Connector	Evolveum	1.0.0.0	Evolveum	✓	✓	✗	SmartRecruiters	
SCIM v1 generic connector	Evolveum	1.4.2.16	Evolveum	✓	✓	✗	SCIM v1.1 compliant services	
SCIM v1 Slack connector	Evolveum	1.4.2.16	Evolveum	✓	✓	✗	SCIM v1.1 compliant services	Child extension of the SCIM v1 generic connector
SCIM v1 Salesforce connector	Evolveum	1.4.2.16	Evolveum	✓	✓	✗	SCIM v1.1 compliant services	Child extension of the SCIM v1 generic connector
OpenStack (experimental)	Evolveum	TODO	Evolveum	✓	✓	✗	OpenStack REST API (keystone, nova)	Created as a PoC and demo purposes at FOSDEM2016. Work continues in cooperation with Mirantis to evolve it.
Siebel Connector (may req. code changes)	Evolveum (contributed)	1.0.0	Community	✓	✓	✗	Siebel customized SOAP WS exposed at 3rd party middleware integration layer.	Tested: Siebel 8.1.1
PeopleSoft HCM connector	Evolveum	1.4.2.2	Evolveum	✗	✓	✗	XML exported files from the PeopleSoft Human Capital Management (HCM) software.	Connector used to pull data from XML file exports.
Coupa	Evolveum (contributed)	1.4.2.14	Community				REST Coupa API	
CSVFile Connector (deprecated) (DEPRECATED)	Evolveum (nLight)	1.4.2.0	Evolveum	✓	✓	✓	Manipulates content of CSV-formatted files, executes scripts.	Good for integration to HR-like source systems that export data to CSV. Originally contributed by the current Evolveum team to the OpenICF project. It was later taken over by the Polygon project. We do not recommend use of the original or OpenICF version. This connector is DEPRECATED. New CSV Connector is in development.
Scripted REST Connector (DEPRECATED)	OpenICF (Evolveum modifications)	1.1.1.e2	Evolveum				Generic REST service.	Needs customization with Groovy scripts for every operation. This connector is DEPRECATED. Using groovy scripts to write connectors is a maintenance nightmare. Evolveum created a generic superclass for REST connector (in Polygon project). The use of the superclass is recommended as a replacement for OpenICF Scripted REST connector.
Legacy LDAP Connector (DEPRECATED)	Evolveum (OpenICF, Sun ICF)	1.4.1.23	Evolveum	✓	✓	✓	LDAP-based directory servers. Also works for LDAP access to Active Directory. Evolution of original Sun LDAP connector. CDDL-licensed and JNDI-based.	This connector originated from Sun ICF, taken over by OpenICF and then taken over by Polygon. We do not recommend use of the original or OpenICF version. The Polygon version has significant improvements over all other versions. WARNING: This connector is a development dead-end. It is JNDI-based. JNDI is a very bad API for LDAP and it has severe limitations. This connector is maintained, but it is no longer actively developed. Use the new LDAP Connector instead whenever possible.
Active Directory Connector (.NET) (DEPRECATED)	OpenICF (Sun ICF)	1.4.1.20257 (contains Evolveum fixes)	Evolveum	✓	✓	✓	Active Directory (by ADSI)	Tested on Windows 2008 R2, 2012 server. This connector is DEPRECATED. Please use the Active Directory Connector (LDAP) instead.
Exchange Connector (.NET) (DEPRECATED)		1.4.1.20257 (contains Evolveum fixes)	Evolveum		✓		Microsoft Exchange	Tested on Windows 2008 R2 + Exchange 2010, 2013 Server This connector is DEPRECATED. Please use the Active Directory Connector (LDAP) instead.
Solaris Connector (DEPRECATED)	OpenICF (Sun ICF)	1.1.0.em77 (contains Evolveum fixes)	Evolveum	✓	✓	✗	Solaris, Linux, AIX	Tested Solaris, AIX, RedHat and Ubuntu with su and sudo DEPRECATED use Unix connector instead

Legend:

-  - Unknown status/ Not tested yet
-  - Tested and PASS
-  - Tested and PASS with small exceptions
-  - Not supported feature
- **Sample** - means, there is a resource sample
- **prov** - means that provisioning works and is tested
- **sync** - means synchronization works and is tested.

All connectors are Java connectors unless explicitly specified otherwise.

Local vs Remote, Java vs .NET

Connectors can be deployed in two ways:

- **Local connectors** are deployed to a midPoint instance. This is the usual way how connectors are used. The connector is executed inside a midPoint instance, has the same lifecycle (start/stop), etc. MidPoint can detect local connectors automatically and overall the connector management is easier.
- **Remote connectors** are executed in a different process or on a different node than midPoint instance. Remote connectors are deployed to a [Connector Server](#). There may be need to use a remote connector e.g. to access a file on a remote system (e.g. in case of CSV connector) or because of platform incompatibilities (e.g. .NET connectors)

Connector is **not** developed as local or remote. The placement of the connector is a deployment-time decision. There is just one connector package that can be deployed locally or remotely. However there may be deployment limitations when it comes to a platform. The ConnId framework is available for two platforms therefore also the connectors can be developed for one of the following two platforms:

- **Java** connectors can be both local or remote. Remote Java connectors are deployed in [Java Connector Server](#). Vast majority of connectors are Java connectors.
- **.NET (C#)** connectors can only be deployed as remote connectors into a [.NET Connector Server](#). Even if midPoint and the connector server is on the same node they are still considered remote connectors (communicating through *localhost* interface).

See [Connector Server](#) page for a more detailed description of remote connectors.

Connector Development and Maintenance

The Polygon project combines connectors from ConnId and other sources and it also adds its own connectors. Polygon is something like a *downstream* project to ConnId. It means that Polygon takes all the changes from ConnId and adds a couple of changes of its own. Therefore Polygon connectors should be the same or slightly more ahead when compared to the ConnId connectors. Polygon project also adds its own connectors. For example there is [LDAP connector](#) and [LDAP-based Active Directory connector](#) which is a complete next-generation rewrite when compared to the antiquated ICF connector or even the ConnId version of LDAP connector. You can think of Polygon as a connector collection and enhancement project.

Although we add our own changes to the connectors these changes are **not** private. All the changes that we made are public and are made available in our source code repositories. Upstream projects are free to "pull" these changes at any time. We also issue pull requests when the changes are ready to be pushed upstream if a suitable process for this is established. We try to be good open source citizens.

The Story of Identity Connectors as Seen by Evolveum

In the old days there were no identity connectors. Each IDM product used its own proprietary framework. Such frameworks originated together with the first generation of provisioning products therefore they usually were ugly, dirty and cumbersome. The product called "Lighthouse" developed by a company called Waveset was no exception. The company was acquired by Sun Microsystems and the product was renamed to Sun Identity Manager (Sun IDM). The engineers at Sun obviously realized how bad this "adapter" interface was and after few long years of hesitation finally created a new framework. It was still quite far from being perfect but there was one huge difference: it was **not** proprietary. Sun developed the framework as an open source project. This project was known simply as "Identity Connector Framework" (ICF). And so identity connectors were born. Before the ICF framework got any chance of major success Sun was acquired by Oracle. We can only speculate what happened inside Oracle but the result was that the ICF project effectively stopped all development activity. Last commit to the project was in May 2010.

But the acquisition of Sun was like a supernova. Engineers that worked with Sun technologies suddenly scattered around to other projects and companies. This also affected the team that now forms the core of midPoint project. In 2010 we were looking for a replacement of Sun IDM. We have realized very quickly that Oracle IDM or any similar commercial product just cannot satisfy our needs. We have decided to start a new open source project to fill this sudden technological gap. And it was early 2010 when we connected with ForgeRock and started work on OpenIDM version 1. The Sun ICF framework was an obvious choice for a connector layer. Although we were not aware of it another project was started approximately at the same time: Syncopé. This project has also chosen ICF as a connector framework. In early 2011 ForgeRock decided to drop OpenIDM version 1 code-base and this was an impulse that contributed to our decision to start independent development of midPoint. The ICF was kept as a connector layer. So now there were three open source projects that were using the framework. This finally seems like a success for the framework. But there was a glitch.

In mid-2011 it was quite clear that the original Sun ICF project is not going anywhere. ForgeRock decided to take over the development and formed the OpenICF project. We have been forming an independent stream of development at that very time. But we had seen the benefit of cooperation and therefore we have decided to cooperate on OpenICF. Approximately at the same time the ConnId project was created by the Syncopé team. This was also a fork of the original Sun ICF code. There were also rumours that Oracle continues development of ICF in a closed-source fashion. Therefore in late 2011 there were actually several versions of ICF:

- Original Sun Identity Connector Framework - in a clinical death state

- OpenICF maintained by ForgeRock with Evolveum as a major contributor
- ConnId maintained by the Syncope team
- Oracle closed-source version (rumoured)

The "forks" began independent development and they became incompatible. This was quite an awkward situation. We could do nothing about the original Sun ICF and it is unlikely that we could do anything about Oracle. But having two incompatible open source frameworks was just plainly insane. It took several years to make OpenICF and ConnId teams to talk to each other. But it finally happened in late 2013.

Currently (2018) there is one common framework code maintained in [ConnId Project](#) at github. The idea is to use this framework in all open source IDM projects (midPoint, Syncope and possibly others). This will make the connectors compatible once again. Teams from Evolveum and Tirasa contribute the code to ConnId framework. ConnId connectors are compatible and interchangeable. All the teams also take part of the design and future development of the framework. We are more than aware that the [ICF framework is not perfect](#). But we have plans to improve it. In a fully open and transparent fashion to make sure it does not become a proprietary technology.

In the meantime we hear reports about Oracle using something that resembles original Sun ICF in the Oracle Identity Manager (OIM) product. We are no longer working with Oracle technology therefore we cannot confirm it and we can only speculate. However we guess that Oracle continues development of the original Sun ICF framework. However ConnId has evolved in the meantime and it is likely that Oracle has evolved the framework as well. It is extremely unlikely that these frameworks are still compatible. Therefore we guess that OIM users will **not** be able to take advantage of the new ConnId-based open source connectors.

Therefore the situation of the framework was resolved. Starting from ConnId 1.4 these connectors are compatible but they are still maintained in separate projects - mostly because of non-technical reasons. What we call "polygon" is an attempt to put together a list of workign and maintained ConnId-compatible connectors in various repositories.

Later on, ForgeRock team silently stopped cooperation on ConnId framework. Or at least we think so as there was no contribution from ForgeRock in several years. Now it looks like that OpenICF connectors are, once again, not compatible with other ConnId connectors. But Evolveum and Tirasa still cooperate maintain ConnId framework.

See also

- [Connector Server](#)
- [Resource Schema](#)
- [Resource and Connector Schema Explanation](#)
- [Connector Upgrade](#)

External links

- [What is midPoint Open Source Identity & Access Management](#)
- [Evolveum](#) - Team of IAM professionals who developed midPoint