Identity Connectors

About Identity Connectors

MidPoint is using ConnId framework. 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. Evolveum is a major contributor to the development of ConnId framework.

The ConnId **framework** is developed jointly by several companies and independent contributors. But not the connectors. The **connectors** are developed and maintained in separate 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.

Connector List

This table summarizes the connectors for use with midPoint. Some of those connectors are used routinely, other are used occasionally and some of those are experimental connectors in development. This table provides information from our (Evolveum) point of view.

Connector	Status	Maintained by (Origin)	Connects to	Note
DatabaseTable Connector	Supported Bundled	Evolveum (Sun ICF)	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 Evolveum.
CSV Connector	Supported Bundled	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	Supported Bundled	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. Distributed in LDAP connector bundle. There are support limitations.
Active Directory Connector (LDAP)	Supported Bundled	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. Distributed in LDAP connector bundle. There are support limitations. See Active Directory with LDAP connector.
UNIX	Supportable	ConnId	Linux (RedHat, Ubuntu)	
Microsoft Graph API Connector	Supportable Experimental	Evolveum	Microsoft Azure AD and Office365	Connector for Microsoft Graph API. It is meant to manage users in Microsoft cloud applications, such as Azure AD and Office365.
Office365	Deprecated	Evolveum (contributed)	Microsoft Office 365	This connector was contributed. However, due to lack of interest from midPoint subscribers the connector was not maintained. This connector is DEPRECATED. Please consider using of Graph API connector instead.
GitLab (REST)	Supportable Experimental	Evolveum	GitLab server	
Liferay	Supportable Limited	Evolveum	Liferay Portal	Tested on 6.2-ce-ga4, support ACCOUNT and assignments to Roles and Org. structure over ID

CMD	Supportable	ConnId	Executes arbitrary commands	Does not seem to support object renaming.
Google Apps	Supportable Community	Evolveum (contributed)	Google API and OAuth	
ScriptedSQL Connector	Supportable	Evolveum (SunICF /OpenICF)	Very generic database connector based on Groovy/JavaScript scripting.	For databases with data in more then one table with joins, or when procedures are to be called.
SAS	Supportable	Evolveum	SAS Metadata Server	This connector is quite outdated. But it can be supported if needed.
Atlassian Jira (REST)	Deprecated	Evolveum	Atlassian Jira	Development of this connector was not finished. The maintenance is now stopped. It can be resumed if there is enough financial incentive.
Box Connector	Experimental	Evolveum	Box (cloud service)	Development of this connector was not finished.
SAP Connector	Supportable	Evolveum	SAP R/3	Tested on SAP System (R07) Netweaver 7 EHP 2 (aka 7.31)
Drupal 7 Connector	Supportable	Evolveum	Drupal 7	Tested on Drupal 7.33, 7.53
SmartSmartRecruiters Connector	Supportable	Evolveum	SmartRecruiters	
SCIMv1 Generic Connector	Experimental	Evolveum	SCIM v1.1 compliant services	Generic connector with limited usability. Please consider using a service-specific connectors instead.
SCIM v1 Slack connector	Supportable	Evolveum	Slack services (SCIMv1.1 API)	Child extension of the SCIM v1 generic connector
SCIM v1 Salesforce connector	Supportable	Evolveum	Salesforce platform (SCIMv1.1 API)	Child extension of the SCIM v1 generic connector
OpenStack	Experimental	Evolveum	OpenStack REST API (keystone, nova)	Created as a PoC and demo purposes at FOSDEM2016.
Siebel Connector	Community Experimental	Evolveum (contributed)	Siebel customized SOAP WS exposed at 3rd party middleware integration layer.	Tested: Siebel 8.1.1 Connector code is experimental, it may be incomplete and it may require code changes.
PeopleSoft HCM connector	Supportable	Evolveum	XML exported files from the PeopleSoft Human Capital Management (HCM) software.	Connector used to pull data from XML file exports.
Coupa	Community	Evolveum (contributed)	Coupa Cloud Platform (REST Coupa API)	
SWITCH edu-ID Affiliation Connector	Supportable	Evolveum	SWITCH edu-ID SCIM API	
Scripted REST Connector	Deprecated	OpenICF (Evolveum modifications)	Generic REST service.	Needs customization with Groovy scripts for every operation. This connector it 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.
Waveset Connector	Experimental	Evolveum	Oracle Waveset (Sun Identity Manager)	
Grouper Connector	Supportable	Evolveum	Grouper (Internet2 et al.)	
Banner Connector	Community	community	Ellucian Banner	An initiative to develop source connector for Ellucian Banner.

Please see Legacy Identity Connectors for the list of legacy connectors and old connectors with an unknown status.

Connector Status Legend

Status	Description

Supported	Connector is fully supported by Evolveum. Connector support is routinely provided to customers. Note: This does **not** mean that connector support is included in all midPoint support contract. Connector support still may need to be purchased separately.
Supportable	Connector can be supported, but the support is not provided on a routine basis. Support of this connector may need additional effort or there may be limitations. Please consult Evolveum sales team for the details.
Bundled	Connector is bundled with midPoint. Some extent of connector support is also bundled with standard midPoint support agreements. Please see connector documentation, midPoint release notes and your support contract about the details and limitations of bundled support.
Limited	Connector capabilities are limited. Only some of the connector capabilities are supported.
Experimental	Connector development is not finished or the connector was created purely for exploration. Support status of this connector is not yet completely clear. Experimental connectors can be supported in some cases. Please consult Evolveum sales team for the details.
Deprecated	Connector is deprecated. Although the connector may work, it is not recommended to use this connector in new midPoint deployments. Some support for older connector installations is possible. However, this connector will not be developed any more and the support of this connector will end eventually. Therefore using this connector in new installations is likely to result in lost investment. It is recommended to use alternative connectors instead.
Legacy	Legacy connector. Such connector is obsolete. No support will be provided for this connector unless a significant investment is made. Any investment in this connector will most likely lead to a complete rewrite of connector code.
Community	This connector was contributed by the community or it is independently developed by the community. Please contact original connector author for inquiries about connector support. Evolveum will not provide support for this connector unless there is a substantial incentive to do so.
Not recommended	This connector may be in active development by other (non-Evolveum) teams. However, Evolveum provides better alternatives for this connector. Use of this connector is not recommended by Evolveum. Evolveum will not provide any support for this connector.
Unknown	Connector status is unkown. There is no reliable information about the quality or functionality of the connector. This connector is probably outdated. But if there is enough interest and incentive then the connector code may be analyzed and there is a chance to support this connector. Please consult Evolveum sales team for the details.

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 istance. 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

MidPoint can use ConnId-compatible connectors from a variety of sources. The connectors that are often used with midPoint are developed and maintained by Evolveum teams. However, other ConnId-compatible connectors can be used with midPoint.

Connector Support

Support for some (bundled) connectors is included in in basic midPoint support services. This "bundled" support is limited only to connectors that are distributed together with midPoint. And the support is limited, e.g. eDirectory is not supported at all, only standardized LDAP operations are supported for LDAP connector, there are limitations for Active Directory operations and so on. Please see midPoint release notes, connector documentation and your support contract for the details.

Support for other connectors can be purchased separately. All the connectors have their peculiarities that are often determined by the system that they connect to. Use and capabilities of the connector can also depend on the configuration of the target system. Therefore it is almost impossible to provide a comprehensive price list and support terms that would work for all the connectors and target systems. For that reason a pricing for many connectors is determined on case-by-case basis. Please contact Evolveum sales representative for the details.

Project Polygon

The code that is currently known as "ConnId" has a turbulent history. There were strange and uncertain times. Project Polygon was born during such times as an attempt to stabilize situation regarding connector framework and the basic connectors. The goal of Project Polygon was to create a set of reliable ConnId-compatible connectors that came from various sources. Now, Project Polygon is mostly just a historical name that does not have much meaning any more. All the Evolveum changes to ConnId framework are immediately contributed to upstream ConnId project. And support for each connector is provided separately.

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: Syncope. 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 Syncope 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. That was the time when our Project Polygon was born as an attempt to survive in this confusing situation. It took several years to make OpenICF and ConnId teams to talk to each other. But it finally happened in late 2013. The code of OpenICF was merged into ConnId. But later on, ForgeRock stopped contributing to ConnId. Without any official statement or notice to ConnId team, ForgeRock went on and developed OpenICF framework independently. Other ConnId contributors were puzzled, but the development of ConnId went on. Then in 2016 ForgeRock stopped to publish their day-to-day development, effectively making OpenICF a closed-source project. But Evolveum and Tirasa continued cooperation to maintain and extend ConnId framework.

Currently (2019) there is one common framework code maintained in ConnId Project. The idea is to use this framework in all open source IDM projects (midPoint, Syncope and possibly others). 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.

See also

- Connector Server
- Resource Schema
- Resource and Connector Schema Explanation
- Connector Upgrade
- Legacy Identity Connectors