



# Autotask

## Seamless endpoint guide

Version 1, 8 November 2018  
[info@recursyv.com](mailto:info@recursyv.com)

## Table of Contents

1. Document overview .....	2
2. Autotask integrations using the Seamless service .....	2
3. Data entities typically synced using the Autotask connector plugin .....	2
3.1 Account.....	3
Record matching.....	4
3.2 Contact.....	4
Record matching.....	5
3.3 Ticket .....	5
Record matching.....	7
3.4 Opportunity .....	7
Record matching.....	9
4. Service pre-requisites.....	9
4.1 Autotask .....	9
4.2 Other system .....	9
5. Appendix 1: The architecture of the Seamless service.....	11

## 1. Document overview

Author	David Reinhardt
Version	1
Publication date	04/07/2018
Version history	<ul style="list-style-type: none"> <li>1.0 – First published version</li> </ul>

This document details the typical data entities and associated fields synced using the Seamless Autotask connector plugin.

## 2. Autotask integrations using the Seamless service

The Seamless service is a cloud-based integration framework allowing you to connect different applications to one-another by passing data or events through a cloud hosted service built and operated by Recursyv.

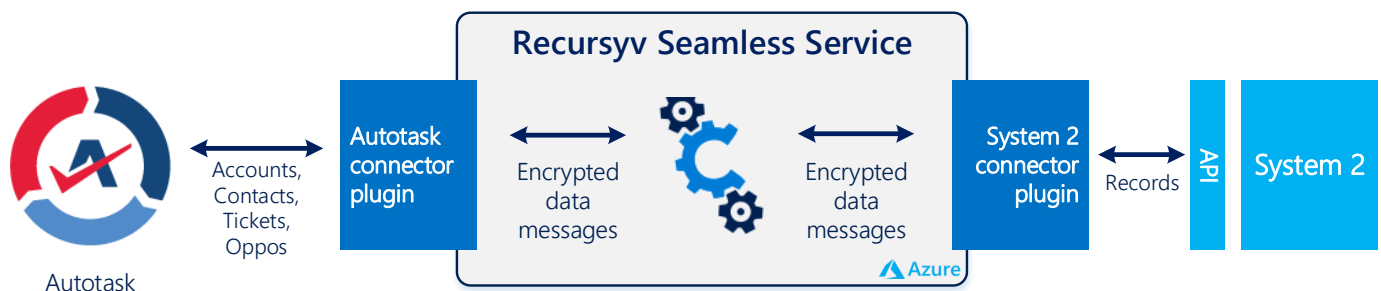


Figure 1: Seamless Overview Diagram

The Seamless Autotask connector allows you to use Seamless to sync Autotask with other applications.

## 3. Data entities typically synced using the Autotask connector plugin

Data Item	Notes
Account	Typically synced bidirectionally between Autotask and other systems.
Contact	Typically synced bidirectionally between Autotask and other systems.
Ticket	Typically synced one way between Autotask and other systems.
Opportunity	Typically synced one way between Autotask and other systems.

Data mappings are based on a “vanilla” implementation of Autotask. If custom list values have been added, these will need to be mapped as part of the initial configuration.

Where fields are indicated as mandatory, this is relevant only if the integration is set to create/update those records in Autotask. If the integration is from Autotask to the other system, this is not relevant.

It is important note that additional fields not specified here, including User Defined Fields, can be added to the scope of the sync. Subject to the volume of fields to be added to an integration, this may attract additional professional service fees.

### 3.1 Account

The table below details the fields that are typically synchronised with Autotask account.

Autotask Field	Description
Account ID	In order to enable bi-directional sync, this field needs to be added to the other system as a cross-reference field.
Record ID for the other system	In order to enable record updates from the other system, this field needs to be added to Autotask as a cross-reference field.
AccountName	Account name
Address 1	First address line
Address 2	Second address line
City	City
State	State
PostalCode	Post Code
Country	Country
Phone	Main telephone number
WebAddress	Website
AccountType	The option set could be mapped to the relevant options in the other system.
	<hr/> <b>Autotask values</b> <hr/> Customer (1) <hr/> Lead (2) <hr/> Prospect (3) <hr/> Dead (4) <hr/> Cancelation (6) <hr/> Vendor (7) <hr/> Partner (8)
ParentAccountID	Could be resolved to the parent account in the other system.
AccountNumber	Account number

CreateDate	Record create date. This can only be synced in cases where it can be written to the other system without affecting underlying operations of the other system.
LastActivityDate	Last time the record was updated in Autotask
Active	Record status indicator

## Record matching

The integration will look for existing records before creating new ones. If an existing record is found, it will be updated with any changes.

By default, Account record matching will be performed using the following fields:

- System record ID (Autotask ID / other system ID)

If a record cannot be identified, a new record will be created (which will contain the relevant system record ID to ensure duplicates are not created). Seamless can be configured to match records on additional fields if there is a requirement to do so.

## 3.2 Contact

The table below details the fields that are typically synchronised with Autotask Contact.

Autotask Field	Description
Autotask ID	In order to enable bi-directional sync, this field needs to be added to the other system as a cross-reference field.
Record ID for the other system	In order to enable record updates from the other system, this field needs to be added to Autotask as a cross-reference field.
Account ID	Identifier of the Account that this Contact belongs to.
FirstName	Contact's first name
LastName	Contact's surname
AddressLine	First address line
AddressLine1	Second address line
City	City
State	State
ZipCode	Post Code
Country	Country
Email address	Only the Autotask first email address can be synced.
Phone	Business phone number
MobilePhone	Mobile phone number

PrimaryContact	This is an indicator as to whether this Contact is the Primary Contact for the Account. This can be synced into the other system.
CreateDate	Record create date. This can only be synced in cases where it can be written to the other system without affecting underlying operations of the other system.
LastModifiedDate	Last time the record was updated.

## Record matching

The integration will look for existing records before creating new ones. If an existing record is found, it will be updated with any changes.

By default, Contact matching will be performed using the following fields:

- System record ID (Autotask ID / other system ID)
- Email Address

If a record cannot be identified, a new record will be created (which will contain relevant system record ID to ensure duplicates are not created). Seamless can be configured to match records on additional fields if there is a requirement to do so.

## 3.3 Ticket

The table below details the fields that are typically synchronised with Autotask Ticket.

Typically the Ticket is setup to sync one-way, either from Autotask into a different system or the other way around.

Autotask Field	Description
ID	Autotask ticket system identifier
Record ID for the other system	In order to enable record updates from the other system, this field needs to be added to Autotask as a cross-reference field.
Ticket Number	Autotask ticket number
Title	Ticket title
Description	Ticket description
Account ID	Identifier of the account that this ticket belongs to.
Contact ID	Identifier of the contact that this ticket belongs to.
Due Date (DueDateTime)	Date resolution is required
Ticket Category	The option set could be mapped to the relevant options in the other system.
Ticket Type	The option set could be mapped to the relevant options in the other system.

	<hr/> <p><b>Autotask values<sup>1</sup></b></p> <hr/> <p>Service Request (1)</p> <hr/> <p>Incident (2)</p> <hr/> <p>Problem (3)</p> <hr/> <p>Change Request (4)</p> <hr/> <p>Alert (5)</p>
Priority	The option set could be mapped to the relevant options in the other system.
	<hr/> <p><b>Autotask values<sup>2</sup></b></p> <hr/> <p>High (1)</p> <hr/> <p>Medium (2)</p> <hr/> <p>Low (3)</p> <hr/> <p>Critical (4)</p>
Status	The option set could be mapped to the relevant options in the other system.
	<hr/> <p><b>Autotask values<sup>3</sup></b></p> <hr/> <p>New (1)</p> <hr/> <p>Complete (5)</p> <hr/> <p>Waiting Customer (7)</p> <hr/> <p>In Progress (8)</p> <hr/> <p>Waiting Materials (9)</p> <hr/> <p>Scheduled (10)</p> <hr/> <p>Escalate (11)</p> <hr/> <p>Waiting Vendor (12)</p> <hr/> <p>Waiting Approval (13)</p> <hr/> <p>Waiting Dispatch (14)</p> <hr/> <p>Approved (15)</p> <hr/> <p>Denied (16)</p>

<sup>1</sup> Autotask ticket type reference can be found at <https://ww2.autotask.net/Help/Content/ServiceDesk/CreateTickets/TicketTypes.htm>

<sup>2</sup> Autotask ticket priority reference can be found at <https://ww2.autotask.net/Help/Content/AdminSetup/1FeaturesSettings/ServiceDeskTickets/Tickets/TicketPriorities.htm>

<sup>3</sup> Autotask ticket status reference can be found at <https://ww2.autotask.net/Help/Content/AdminSetup/1FeaturesSettings/ServiceDeskTickets/Tickets/TicketStatuses.htm>

In the event that Seamless encounters a value that is not mapped, it can be synced as In Progress.

Source	The option set could be mapped to the relevant options in the other system.
	<b>Autotask values<sup>4</sup></b>
	Insourced (-2)
	Client Portal (-1)
	Internal (1)
	Phone (2)
	Email (4)
	Web Portal (5)
	In-person / Onsite (6)
	Monitoring Alert (8)
AssignedResourceID	Autotask resource that the incident is currently assigned to.
CreatorResourceID	Autotask resource that created the ticket.
Last Activity Date	Last time the record was updated.

### Record matching

The integration will look for existing records before creating new ones. If an existing record is found, it will be updated with any changes.

By default, Ticket matching will be performed using the following fields:

- System record ID (Autotask ID / other system ID)
- Email Address

If a record cannot be identified, a new record will be created (which will contain relevant system record ID to ensure duplicates are not created). Seamless can be configured to match records on additional fields if there is a requirement to do so.

### 3.4 Opportunity

The table below details the fields that are typically synchronised with Autotask Opportunity.

<sup>4</sup> Autotask ticket source reference can be found at <https://ww2.autotask.net/Help/Content/AdminSetup/1FeaturesSettings/ServiceDeskTickets/Tickets/TicketSources.htm>



Typically the Opportunity is setup to sync one-way, either from Autotask into a different system or the other way around.

<b>Autotask Field</b>	<b>Description</b>
Autotask ID	Autotask opportunity system identifier
Record ID for the other system	In order to enable record updates from the other system, this field needs to be added to Autotask as a cross-reference field.
AccountID	Resolve to parent account of the Opportunity within Autotask
ContactID	Resolve to the contact of the Opportunity within Autotask
OwnerResourceID	Opportunity owner
Salesstage	The option set could be mapped to the relevant options in the other system.
	<hr/> <p><b>Autotask values</b></p> <hr/> <p>New Lead (29682769)</p> <hr/> <p>Stage 1 First Contact (29682770)</p> <hr/> <p>Stage 2 Qualification (29682771)</p> <hr/> <p>Stage 3 Proposal Sent (29682772)</p> <hr/> <p>Stage 4 Contract Sent (29682773)</p> <hr/> <p>Stage 5 Closed (29682774)</p>
Title	Opportunity title
Description	This needs to be added to Autotask as a UDF
Status	The option set could be mapped to the relevant options in the other system.
	<hr/> <p><b>Autotask values</b></p> <hr/> <p>Not Ready to Buy (0)</p> <hr/> <p>Active (1)</p> <hr/> <p>Lost (2)</p> <hr/> <p>Closed (3)</p> <hr/> <p>Implemented (4)</p>
	If the integration encounters a record that is not in this mapping, the Opportunity can be set to Active (1) in Autotask.
ProjectedCloseDate	Estimated close date
ClosedDate	Actual close date

Probability	Probability of winning the sale
CreateDate	Date opportunity was created
Amount	Estimated revenue of the sale
ClosedDate	Actual close date
Cost	A value for cost is mandatory in Autotask. In the event that the other system does not provide a value for cost, this can be set to 0.
UseQuoteTotals	A value for UseQuoteTotals is mandatory in Autotask. This will be set to False when writing Opportunities to Autotask.

It should be noted that following fields are mandatory in Autotask. Records without these fields captured cannot be synced to Autotask:

- Probability
- Estimated close date
- Estimated revenue

The integration can be configured to populate data into these fields if required.

### Record matching

The integration will look for existing records before creating new ones. If an existing record is found, it will be updated with any changes.

By default, Opportunity matching will be performed using the following fields:

- System record ID (Autotask ID / other system ID)
- Email Address

If a record cannot be identified, a new record will be created (which will contain relevant system record ID to ensure duplicates are not created). Seamless can be configured to match records on additional fields if there is a requirement to do so.

## 4. Service pre-requisites

In order to successfully deploy this integration, the following pre-requisites will be required.

### 4.1 Autotask

- An Autotask user account needs to be created or nominated and the credentials provided to Recursyv.
- It is suggested that this is a dedicated user with an identifiable name (e.g. Integration) as this will allow creation of views filtered by this user and identification of amendments made by the integration service.
- The other system ID user defined fields need to be created on the relevant record types in Autotask.

### 4.2 Other system

- A license / user account is required for most integrations. This needs to be a license of a suitable level that allows API access to amend the type of records within scope.
- It is suggested that this is a dedicated user with an identifiable name (e.g. Integration) as this will allow creation of views filtered by this user and identification of amendments made by the integration service.

## 5. Appendix 1: The architecture of the Seamless service

The Seamless service is a cloud-based integration framework allowing you to connect different pieces of software to one-another by passing data or events through a cloud hosted service built and operated by Recursyv.

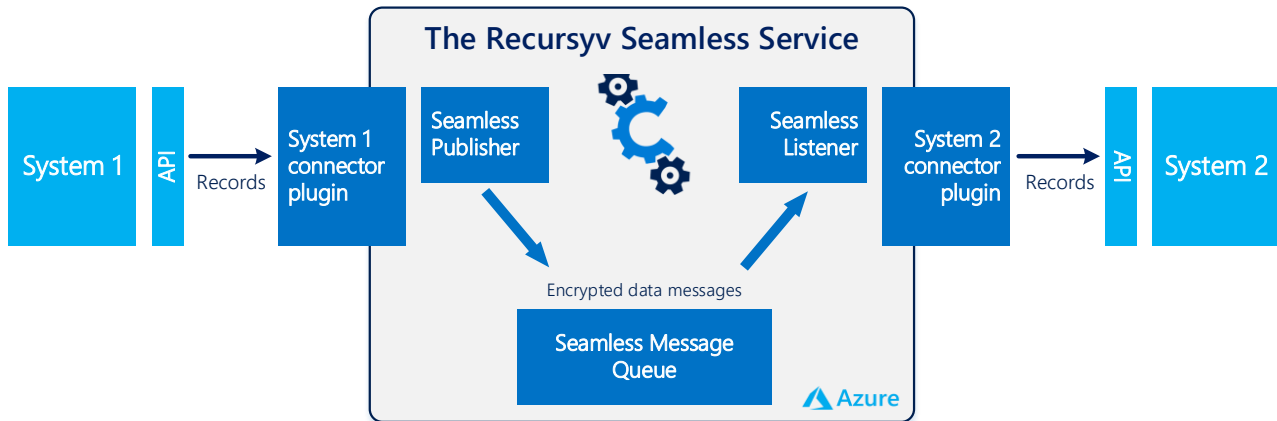


Figure 2: Generic Seamless overview diagram

The Seamless service is built using a micro-service architecture platform and is deployed within Azure to utilise a number of its integrated services. This also ensures that it takes advantage of the security and resilience of Azure, as well as providing the ability to specifically geolocate instances of Seamless as required.

The service is characterised by five key components:

- **Recursyv Seamless Service:** this is the control framework which governs the flow of data, serialised into data messages, including the process of looking for data or events in the source system, using the plugins to perform their roles, encrypting, compressing, decrypting and decompressing data messages and pushing data into the target systems. It also manages security, load distribution and system monitoring, etc.
- **Messages:** these are individual pieces of data that are processed by the service, typically from source to target. A message could contain raw data extracted from the source system that needs to be replicated to the target system(s), or it could represent an event that has occurred in the source system that the target system(s) might need to know about.
- **Source System Plugin:** this component retrieves data or events from the source system and converts them into messages so that the Seamless service can publish them onto the Azure service bus. This is specific to any scenario as there will be variations for each source system and business use.
- **Target System Plugin:** this component assesses, identifies and converts messages provided by the Seamless service into data for the target system to use. Like the source system plugin, this will typically be tailored to a particular scenario.
- **Azure Service Bus:** This acts as a staging medium for the messages, and eliminates any real-time dependency between the source and target systems and plugins. It can also be used as an aggregator to allow information from multiple sources to be brought together or as a disseminator to allow a single data source to be used multiple times.