

ConnectWise Manage Invoicing Integration

Contents

[1. Purpose of the document 4](#_Toc132706457)

[2. Executive summary 4](#_Toc132706458)

[3. Invoices in AppXite platform 4](#_Toc132706459)

[4. Invoices in ConnectWise Manage 5](#_Toc132706460)

[5. Integration flow between AX platform and ConnectWise 5](#_Toc132706461)

[5.1. Triggering data exchange 5](#_Toc132706462)

[5.2. AX platform and ConnectWise supported (build-in) flow configurations 6](#_Toc132706463)

[5.3. AX platform and ConnectWise data exchange order 6](#_Toc132706464)

[5.3. Custom fields 7](#_Toc132706465)

[5.4. Logging 8](#_Toc132706466)

[6. Authentication in AX Platform and ConnectWise 8](#_Toc132706467)

[6.1. Authentication in AX Platform 8](#_Toc132706468)

[6.2. Authentication in CW 8](#_Toc132706469)

[7. Getting required data AppXite portal 8](#_Toc132706470)

[7.1 Getting a list of invoices from AppXite portal 8](#_Toc132706471)

[7.2. Getting Organization data 12](#_Toc132706472)

[8. Getting required data in ConnectWise 15](#_Toc132706473)

[8.1. Defining mapping identity fields 15](#_Toc132706474)

[8.2. Creating/Updating contacts in ConnectWise 16](#_Toc132706475)

[8.3. Creating/Updating companies in ConnectWise 16](#_Toc132706476)

[8.4. Creating/updating agreements in Connectwise 18](#_Toc132706477)

[8.5. Creating/updating product catalog items in ConnectWise 20](#_Toc132706478)

[8.6. Agreement products (attachments) 21](#_Toc132706479)

[8.7. Generating invoices in CW 22](#_Toc132706480)

[9. Synchronization run 22](#_Toc132706481)

[9.1. Running synchronization steps manually/Synchronization test 22](#_Toc132706482)

[9.1.1. Synchronization of a contact 22](#_Toc132706483)

[9.1.2. Synchronization of a company 22](#_Toc132706484)

[9.1.3. Synchronization of a product 23](#_Toc132706485)

[9.1.4. Synchronization of an agreement and agreement attachments 23](#_Toc132706486)

[9.2. Running synchronization automatically 23](#_Toc132706487)

[9.3. Manual mapping of AX and ConnectWise entities 23](#_Toc132706488)

[Appendix 1 – setting up ConnectWise API access for integration with AppXite platform 25](#_Toc132706489)

[1. Creating a security role 25](#_Toc132706490)

[2. Creating/assigning API member 26](#_Toc132706491)

[3. Create API key 27](#_Toc132706492)

[4. Creating Admin User for setting up API access 28](#_Toc132706493)

# Purpose of the document

This document on overall level describes AppXite platform integration with ConnectWise Manage. The scope is limited to invoicing related integration only.

# Executive summary

AppXite platform – further AX platform is used by vendors, distributors, resellers, managed services providers (MSPs) and end-customer organizations to build, sell, deliver, operate multi-cloud, SaaS, and managed services offerings.

One of the major purposes of AX platform is to calculate billings and generate invoices based on sales that are performed on the platform. For some users of AX platform, it is important to keep all invoicing data inside their own ERP/CRM or similar systems.

ConnectWise Manage has dedicated “Invoicing” section that contains invoicing and invoice lines data.

The objective of AX platform invoicing integration with ConnectWise Manage is to provide all necessary information from AX platform to ConnectWise Manage, to produce invoices.

Data that is needed to produce invoices will be imported using respective APIs of AX Platform and ConnectWise. Further in this document this process will be described in more detail.

# Invoices in AppXite platform

Invoices in AX platform are generated based on a billing cycle date set for offerings and contracts, usually once per month. Depending on vendors that provide services there can be several different billing cycles even for one vendor, for example Microsoft for old CSP legacy offers may have different billing cycle comparing to NCE (New commerce experience) offers, which always are billed based on a calendar month.

To check invoices for particular customer, users with appropriate permissions can select that customer and then navigate to Invoices section to see a list of invoices:

Graphical user interface

Description automatically generated

Picture 1. Customer organization incoming invoices.

Invoices can have different statuses, but only invoices in status “Verified” are valid for payment and import to ConnectWise.

To see individual invoice rows users should click on “Invoice Id” number:

Graphical user interface, application

Description automatically generated

Picture 2. Invoice rows

Information from “Incoming invoices” and “Incoming invoice details” will be exported to ConnectWise automatically when invoice status is changed to “Verified”. This is performed by a scheduled job that is executed nightly.

For further information on invoicing functionality in AX platform refer to knowledge base articles here: <https://support.appxite.com/hc/en-us/articles/360009963280-Where-can-I-find-invoices->

# Invoices in ConnectWise Manage

Invoices in ConnectWise Manage can be generated based on agreements or individual sales.

To view invoiceable items users should navigate to “Finance->Invoices” section.

Graphical user interface, text, application

Description automatically generated

Picture 3. Agreement invoicing sample

This integration always uses CW agreements when data that is needed for invoicing.

# Integration flow between AX platform and ConnectWise

## 5.1. Triggering data exchange

1. Invoices are generated based on a billing cycle. New, non-verified invoices have status “New” in AX platform. When invoices are verified by the finance dept then the status is changed to “Verified”. Verified invoices are processed by the scheduler that performs data export to ConnectWise.
2. If a reseller has enabled invoices integration with ConnectWise then AppXite Integration middleware (further - AI) executes ConnectWise integration flow. The flow can be fine-tuned per reseller, for example setting custom field mappings etc. AI allows re-configuration of integration flow without additional development, by simply uploading/updating integration scenario which is defined in JSON. AppXite onboarding and integration team performs these updates. In future versions of this integrations, it will be possible to do by other authorized users using AX platform UI from “Settings->Integrations->ERP/CRM->ConnectWise” section.

## 5.2. AX platform and ConnectWise supported (build-in) flow configurations

There are two major flows, how AX can be integrated with ConnectWise:

1. Fully automated flow: AX creates all data that is needed to generate invoices, such as companies, agreements, catalog items, contacts, agreement additions
2. Manual mapping flow: AX checks if entities, such as companies, agreements etc. exist in CW and are mapped and if yes – attaches or updates agreement additions (which serve as a base for invoices), if not – adds missing data entities for mapping, that further can be performed in a separate tool (described in “9.3. Manual mapping of AX and ConnectWise entities”). Normally it is only Companies that need to be mapped, if it is required to have a manual mapping, in order not to produce duplicates during invoice import.

## 5.3. AX platform and ConnectWise data exchange order

The description below applies both to fully automated flow and manual flow with the exception that in case of manual flow no new entities in CW will be created (apart from those that are agreed to be created automatically in CW). With the manual flow invoice line is skipped if some required entity is not mapped.

For each company listed in a billing data provided by AX the following steps are executed:

1. AI retrieves invoice data which also includes company data (AX Invoices API. Only invoices in status “Verified” are retrieved.)
2. AI checks if a company exists in ConnectWise (CompaniesAPI <https://developer.connectwise.com/Products/Manage/REST#/Companies/getCompanyCompanies> )
3. If a company doesn’t exist, then AI creates a contact and a company in ConnectWise (CompaniesAPI) using available data from AX platform. When a contact and company is created in ConnectWise its Ids are automatically mapped with company Id in AX platform. Note that step b is skipped if the mapping between company Ids already exists
4. AI checks if an agreement exists in ConnectWise (AgreementsAPI <https://developer.connectwise.com/Products/Manage/REST#/Agreements/getFinanceAgreements> )
5. If agreement doesn’t exist, then it is being created and mapped similar way as described in point c with company data. If it exists this step is skipped
6. Invoice rows are added/updated in CW as agreement additions

After this process users in CW can generate invoices in Finance->Invoicing->Agreement invoicing section

Below is the example with screenshots with invoice related information:

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

## 5.3. Custom fields

It is possible to attach custom fields to company, agreement, invoice entities and use them for data mapping if required. This is to be defined after investigation of reseller’s set-up.

To set up a custom field navigate to System->Setup Tables and search for Custom Fields as a table name.

Graphical user interface, application

Description automatically generated

Custom fields can be used for mapping entities between AX platform and CW, however there are also other options, how to map entities, refer to section 8.1. for more details. By default AX use Id fields for the integration mapping.

## 5.4. Logging

AX platform logs integration related events on several levels, depending which component performs integration related work: APIs, AppXite Integrations, Messaging system (for web hooks). Logs contain both successfully executed steps as well as errors.

ConnectWise stores logs in API Logs section:

Graphical user interface, text, application, website

Description automatically generated

# Authentication in AX Platform and ConnectWise

## 6.1. Authentication in AX Platform

Machine to machine authentication (M2M) is recommended to authenticate AX APIs. It is also possible to create a local user in AX platform, but it is not a recommended way, as in this case there is no centralized control over users using consumer’s identity provider, such as Azure Active Directory (AAD) or similar.

The procedure how to configure M2M authentication is described in a separate document. M2M configuration is performed by AppXite.

The procedure how to invite local users is described in AX knowledge base. [How can I invite user of Microsoft partner? – AppXite](https://support.appxite.com/hc/en-us/articles/360015987494-How-can-I-invite-user-of-Microsoft-partner-) and [How to invite users of non-Microsoft partners? – AppXite](https://support.appxite.com/hc/en-us/articles/360010063940-How-to-invite-users-of-non-Microsoft-partners-)

Authentication token should be used in all requests to AX Platform APIs, mentioned in this document.

## 6.2. Authentication in CW

Distributors or resellers should configure ConnectWise API access. This procedure is described in Appendix 1 of this document.

# Getting required data AppXite portal

Note: since this document describes build-in AppXite-ConnectWise integration all AppXite APIs mentioned below are internal AppXite APIs, you as external user do not need access to them directly and they are listed just for your information.

The required data from AX portal is:

* List of invoices and invoice rows
* Organizations, contacts, contracts, offers (catalog items), subscription details

## Getting a list of invoices from AppXite portal

To get a list of invoice collections the following API request should be used:

GET: <https://API_ROOT_URL/api/v1/Reconciliation/Invoices/Outgoing?skip=0&take=10&requireTotalCount=true>

Headers (note – same headers are required for all subsequent request to AX APIs):

Authorization: Bearer *token*

referrer: *FULL RESELLER’S PORTAL URL*

Response:

{

"data": [

{

"InvoiceId": "10c103ef-9d09-4849-8e1d-be3b9d7c11ed",

"InvoiceState": "Verified",

"InvoiceFormattedReferenceNumber": null,

"InvoiceNumber": "Standard Customer Contract - 2021.04.01",

"InvoiceType": "Standard Customer Contract",

"InvoiceDate": "2021-04-01T00:00:00",

"InvoiceDateStr": "2021.04.01",

"Currency": "EUR",

"CurrencyId": "03b8b1a9-2201-4938-984e-6f0c807a92b9",

"InvoiceAmount": 10682.06000000000,

"PreferredInvoiceAmount": 0.0,

"DueDate": "2021-05-01T00:00:00",

"ContractTypeId": "7f7f5483-7540-4c1c-8c58-241cfce92338",

"PreferredCurrencyPriceInfo": null

}]

}

Only invoices with state “Verified” should be used in integration. Invoices are grouped by invoice date, contract type (different contract types can have different billing periods) and currency.

The integration will iterate through the list of “Verified” invoice collections and retrieve customer invoices associated with a collection.

To get a list of customer invoices the following API request should be used:

GET: https://*API\_ROOT\_URL*/api/v1/Reconciliation/Invoices/Outgoing/Collection/ *InvoiceDateStr*/*ContractTypeId*/*CurrencyId*?skip=0&take=10&requireTotalCount=true

Response:

{

"data": [

{

"InvoiceId": "f80fb7a1-2976-49e8-b402-bf909c00e2f3",

"InvoiceFormattedReferenceNumber": null,

"InvoiceNumber": "51",

"InvoiceStatus": "Processed",

"InvoiceDate": "2021-04-01T00:00:00",

"CurrencyCode": "EUR",

"CurrencyId": "03b8b1a9-2201-4938-984e-6f0c807a92b9",

"InvoiceProviderName": "QA2 reseller",

"InvoiceReceiverName": "QA Kreator 3 Shorty",

"InvoiceReceiverStatus": "Active",

"InvoiceReceiverErpId": null,

"InvoiceReceiverCountry": "Latvia",

"InvoiceReceiverVATNumber": "",

"ContractNumber": null,

"ContractReferenceNr": null,

"ReferenceContactPerson": null,

"BillingAddress": "Latvia, Riga, Riga, Matrozu iela 15",

"AccountManager": "",

"AccountManagerEmail": "",

"Comments": null,

"InvoiceProviderId": "d227d98b-6681-4e13-8a73-d1ec6d4f1128",

"InvoiceReceiverId": "5bc73ef5-364d-4d2e-b9a0-c026e2f0bb1f",

"TotalCharges": 238.78000000000,

"PreferredTotalCharges": null,

"PreferredCurrencyPriceInfo": null

}

]

}

Finally, for the integration, invoice rows are retrieved using this request:

GET: https://API\_ROOT\_URL/api/v1/Reconciliation/Invoices/Details/f80fb7a1-2976-49e8-b402-bf909c00e2f3?group=[{%22selector%22:%22SubscriptionName%22,%22desc%22:false,%22isExpanded%22:false}]

Response:

[

{

"RecordId": "20d7dac2-1efd-4626-94a4-1173628073ba",

"ResellerName": "QA2 reseller",

"CustomerName": "QA Kreator 3 Shorty",

"OfferName": "Vendol Pt 1",

"SubscriptionName": "Bundle 123: Vendol Pt 1",

"SubscriptionStartDate": "2021-03-16T00:00:00",

"SubscriptionEndDate": "2022-03-16T00:00:00",

"ChargeStartDate": "2021-04-01T00:00:00",

"ChargeEndDate": "2021-04-30T23:59:59",

"ChargeType": "CycleFee",

"Currency": "EUR",

"CurrencyId": "03b8b1a9-2201-4938-984e-6f0c807a92b9",

"CustomOfferProperties": "",

"InvoiceFormattedReferenceNumber": "74023",

"CustomerUnitPrice": 6.05000000000,

"PreferredCustomerUnitPrice": null,

"CustomerQuantity": 2.00000000000,

"CustomerTotalPrice": 12.10000000000,

"PreferredCustomerTotalPrice": null,

"RetailPriceSource": "ResellerInPrice",

"RetailPriceMarkup": 10.000000,

"CustomerMarkup": 0.000000,

"PriceMarkupStartDate": "2021.03.16",

"InvoiceProviderId": "d227d98b-6681-4e13-8a73-d1ec6d4f1128",

"InvoiceReceiverId": "5bc73ef5-364d-4d2e-b9a0-c026e2f0bb1f",

"ResellerId": "d227d98b-6681-4e13-8a73-d1ec6d4f1128",

"CustomerId": "5bc73ef5-364d-4d2e-b9a0-c026e2f0bb1f",

"OfferId": "db99e8d9-be70-4d43-a77a-3e992e90d5aa",

"VendorOfferId": "db99e8d9-be70-4d43-a77a-3e992e90d5aa",

"SubscriptionId": "863c35ac-c78d-415b-b4bd-19f324827243",

"VendorSubscriptionId": "de281cbd-1cf2-454b-8cb0-6fe58b57b51f",

"OrderItemId": null,

"OrderItemProvisionedDate": null,

"PreferredCurrencyPriceInfo": null,

"BuyerPurchaseOrderNumber": "",

"SellerPurchaseOrderNumber": ""

}

]

## Getting Organization data

Organization data is required for the integration since it is needed to create invoices in ConnectWise and map them to respective companies.

Organizations are identified by OrganizationId in AX portal. Each invoice and invoice rows from API response have those values. Invoices have “InvoiceProviderId” and “InvoiceReceiverId” in respective APIs’ responses, which represent IDs of organizations in AX portal. Using those IDs additional organization details can be retrieved.

To get organization details the following API request is used:

GET: https://API\_ROOT\_URL/api/v1/customers/*OrganizationId*

Response:

{

"Id": "42ed70f3-3698-49d6-ba60-8cd8a08247bf",

"Addresses": [

{

"Id": "6567ff9b-62ac-433d-b81a-e2981c3414b9",

"CountryCode": "UA",

"AuditId": "00000000-0000-0000-0000-000000000000",

"CustomerId": null,

"AddressLine1": "aaaa.",

"AddressLine2": "",

"City": "city",

"State": "city",

"PostalCode": "06894",

"CountryId": "ce1b8b8b-4684-4633-a8b3-c3fdf07bcd5e",

"Type": "Primary"

}

],

"Contacts": [

{

"Id": "28003a0c-8366-43a8-b0b3-12dbab6bcbff",

"AuditId": "00000000-0000-0000-0000-000000000000",

"FirstName": "name",

"LastName": "name",

"Role": "role",

"Email": "bbb@vvv.sa",

"Phone": "88364873834",

"Type": "Primary"

}

],

"Status": "Active",

"PartnerName": "QA2 reseller",

"AuditId": "00000000-0000-0000-0000-000000000000",

"CustomerCategory": {

"Id": "47ff3264-131e-4130-9ed4-5d4bdb8768ac",

"Name": "All (default)"

},

"NumberOfOffices": 0,

"NumberOfUsers": 0,

"NumberOfItUsers": 0,

"ConcurrentItUsers": 0,

"OrganizationSize": "Small",

"FullName": "ApproveCustomer\_Test\_8",

"ShortName": "ApproveCustomer\_Test\_8",

"WebSite": null,

"InvoicingEmail": null,

"CurrencyId": "03b8b1a9-2201-4938-984e-6f0c807a92b9",

"RegistrationNumber": "2321321",

"VatNumber": "",

"ErpId": null,

"ParentErpId": null,

"Comments": null,

"BillingAddressAsPrimary": true,

"BillingContactAsPrimary": true,

"Industries": [

"PrivateBusiness"

]

}

# Getting required data in ConnectWise

The following entities are retrieved from ConnectWise when integration runs:

* Companies

Companies are retrieved using Id or a custom field that belongs to “Company” model in CW API .

* Agreements

Agreements

* Agreements are retrieved using Id or a custom field that belongs to “Agreement” model in CW API

Agreement Additions (products)

* Agreement should contain all products that can be invoiced. Before adding products, it should be checked if they exist in CW

Products/CatalogItems

* Products are retrieved using Id or a custom field that belongs to “CatalogItem” model in CW API

Depending on the configuration each of these entities can be created or updated when integration runs, for example it can be configured that all companies from AppXite portal should be mapped to companies in CW, and all other entities can be created automatically, or, alternatively the requirement can be that all entities except agreement additions should be mapped and so on.

## Defining mapping identity fields

Companies and catalog items in CW are mapped to respective entities of organizations and offers in AX platform. The field that is used for mappings in CW can be defined for each integration, but by default it is Company.Id field. In AX platform side the Id field of a respective entity is always used for the mapping: Organization.Id. So default mapping is always ConnectWise.Company.Id mapped to AppXite.Organization.Id

In general, there are 3 options in CW which fields can be used for mapping purposes:

1. Identifiers, such as Company.identifier. In this case same identifiers can be used both in AX and CW, however identifiers in AX are GUIDs, which are not meant to be human readable.
2. Custom fields: it is possible to define custom fields for all major entities in CW with various types. Custom fields should be defined by CW administrators per deployment. Custom fields are also visible in CW UI, so it is easy to relate a record in CW to its equivalent in AX, without accessing CW using APIs. The disadvantage is that it is a requirement to configure custom fields before enabling the integration.
3. User defined fields: for most major entities in CW it is possible to use user defined fields. Comparing to custom fields it is not required for administrators to define them in advance. User defined fields are named using a pattern userDefinedField*X* – where X can be a value from 1 to 10 and a value of such field can be any string type value.

In this document fields that are used for mapping are shown as “entityname.X” – where entityname is a name of an entity such as Company, and X is a field used for mapping (any of options described above). Before enabling the integration there should be an agreement between AppXite and administrators of CW instance which option and which fields to use.

## Creating/Updating contacts in ConnectWise

A contact is required to exist and to be attached to company entity in CW. It should be created/updated during the sync (depending on the configuration)

**Data mapping:**

|  |  |
| --- | --- |
| **AX Platform** | **ConnectWise** |
| Contact.Id | Contact.X |
| Contact.FirstName | Contact.firstName |
| Contact.LastName | Contact.lastName |

## Creating/Updating companies in ConnectWise

When Iterating through Organizations in verified invoices AX AI engine checks if a corresponding company exists in CW using company’s Id (or other mapped field). AX AI maintains mapping between organizations in AX platform and CW and this mapping is unique per CW instance (usually configured per reseller). Static (hardcoded) fields can be adjusted in the configuration as well.

**Data mapping:**

|  |  |
| --- | --- |
| **AX Platform** | **ConnectWise** |
| Organization.Id | Company.X |
| Organization.ShortName+ unique value (such as DateTime or ERPId) | Company.identifier |
| Organization.FullName | Company.Name |
| Organization.AddressLine1 | Company.addressLine1 |
| Organization.City | Company.city |
| Organization.State | Company.state |
| Organization.PostalCode | Company.zip |
| Contact.Phone | Company.phoneNumber |
| n/a | Company.site {“name”: “Main”} |
| n/a | Company.status {“name”: “active”} |
| n/a | Company.faxNumber |
| n/a | "defaultContact": {  “id” : contact.id  } |
|  |  |

To further customize the integration the following should be checked:

Check if taxCode is required for integration:

"taxCode": {

"id": 11,

"name": "State",

"\_info": {

"taxCode\_href": "https://api-staging.connectwisedev.com/v4\_6\_release/apis/3.0/finance/taxCodes/11"

}

}

Billing terms:

"billingTerms": {

"id": 2,

"name": "Net 10"

},

"billToCompany": {

"id": 5,

"identifier": "BlackRoosterInc",

"name": "Black Rooster, Inc.",

"\_info": {

"company\_href": "https://api-staging.connectwisedev.com/v4\_6\_release/apis/3.0/company/companies/5"

}

},

Invoice delivery methods:

"invoiceDeliveryMethod": {

"id": 1,

"name": "Mail"

},

## Creating/updating agreements in ConnectWise

To invoice a company on a recurrent basis an agreement should be created in CW.

**Data mapping:**

|  |  |
| --- | --- |
| **AX Platform** | **ConnectWise** |
| Contract.Id | Agreement.X |
| Organization.ShortName + “ “ + “Agreement” | Agreement.name |
| n/a | Agreement.type {“name”: “Managed Service”} |
| n/a | Agreement.company {“id” : company.Id} |
| n/a | "billToCompany": {  "id": company.Id,  } |
| n/a | "contact": {  "id": contact.id  } |
| n/a | "site": {  "name": "Main"  } |
| n/a | "restrictLocationFlag": false,  "restrictDepartmentFlag": false,  "startDate": "2021-05-03T00:00:00Z",  "noEndingDateFlag": true,  "cancelledFlag": false,  "applicationUnits": "Hours",  "applicationLimit": 0.00,  "applicationCycle": "CalendarMonth",  "applicationUnlimitedFlag": true,  "oneTimeFlag": false,  "employeeCompRate": "Actual",  "employeeCompNotExceed": "Billing",  "compHourlyRate": 0.00,  "compLimitAmount": 0.00,  "billingCycle": {  "id": 2,  "name": "Monthly"  },  "billOneTimeFlag": false,  "invoicingCycle": "CalendarYear",  "billToCompany": {  "id": company.id  },  "billAmount": 0.00,  "taxable": false,  "billStartDate": "2021-05-03T00:00:00Z",  "restrictDownPayment": true,  "prorateFlag": false,  "invoiceDescription": "",  "topComment": false,  "bottomComment": false,  "billTime": "Billable",  "billExpenses": "Billable",  "billProducts": "Billable",  "billableTimeInvoice": false,  "billableExpenseInvoice": false,  "billableProductInvoice": false,  "currency": {  "id": 7  },  "autoInvoiceFlag": false,  "nextInvoiceDate": "2021-05-01T00:00:00Z",  "agreementStatus": "Active" |
|  |  |

## Creating/updating product catalog items in ConnectWise

All invoiced SKUs/products should exist in CW. Only products that exist in invoices will be added to CW

**Data mapping:**

|  |  |
| --- | --- |
| **AX Platform** | **ConnectWise** |
| Offer.Id | CatalogItem.X |
| Offer.Name (no spaces) + unique value (such as DateTime or ERPId) | CatalogItem.identifier |
| Offer.Name | CatalogItem.description |
| n/a | "subcategory": {  "name": "Managed Services"  },  "category": {  "name": "Managed Services"  } |
| n/a | "type": {  "name": "Software"  } |
| n/a | "productClass": "Service",  "serializedFlag": false,  "serializedCostFlag": false,  "phaseProductFlag": false |
| n/a | "unitOfMeasure": {  "name": "User-based"  } |
| n/a | "minStockLevel": 0,  "dropShipFlag": false,  "specialOrderFlag": false,  "recurringFlag": false,  "recurringOneTimeFlag": false,  "calculatedPriceFlag": false,  "calculatedCostFlag": false, |
|  | CatalogItem.price |
|  | CatalogItem.cost |
|  | "priceAttribute": "FixedFee", |
|  | "manufacturer": {  "name": "Offer.VendorName"  } |
| Offer.Id | CatalogItem.IntegrationXRef |
|  |  |

## Agreement catalog items (attachments)

For invoice generation products (a.k.a. SKUs, offers, catalog items) should be included in CW agreements.

**Data mapping:**

|  |  |
| --- | --- |
| **AX Platform** | **ConnectWise** |
| Offer.Id | Agreement.X |
|  | "product": {  "id": CatalogItem.Id  } |
| InvoiceRow.CustomerQuantity | "quantity": { InvoiceRow.SubscriptionQuantity}, |
| InvoiceRow.CustomerUnitPrice | "unitPrice": 12.000000, |
| n/a | "billCustomer": "Billable" |
| InvoiceRow.ChargeStartDate | "effectiveDate": "2021-04-22T00:00:00Z" |
| InvoiceRow.ChargeEndDate | “cancelledDate” : "2021-05-22T00:00:00Z" |

## Generating invoices in CW

To generate invoices in CW navigate to Financing->Invoicing->Agreement invoicing, select period for which generate invoices and press “Search”. Then select companies for which invoices should be generated.

# Synchronization run

Synchronization runs automatically (in production) or can be triggered manually while integrating and testing. Exact time when automatic synchronization runs is to be agreed between AppXite and a reseller, but normally it should run outside of normal business hours of the reseller.

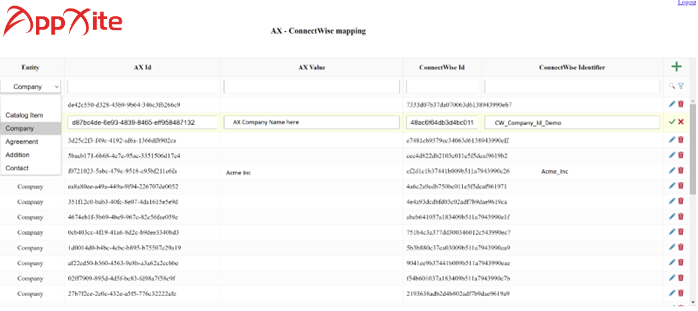
## 9.1. Manual mapping of AX and ConnectWise entities

To have full control over what is being mapped or to link entities in AX to relevant entities that has very specific configuration in CW it is possible to have manual mapping of Companies, SKUs, Contacts and Agreements. Manual mapping is also expected during the initial set up of a platform for testing of every step.

There is no need to map each and every entity (companies, agreements), but only those that have active subscriptions and billing information that can be used for invoicing.

It is possible to configure the synchronization not to create any records in CW if some required entities do not have mappings. Similar to when running synchronization automatically, AX processes all records that have billing information and tries to export them to CW. During the export it is being checked if those records are mapped to relevant records in CW and if not – they will be listed in AX-CW mapper tool with corresponding ConnectWise Id and ConnectWise Identifier values missing. Then users can map those entities manually and then they will be processed during the next export run. To map an entity user should select entity type (Company, CatalogItem etc.), find missing entity, ,press “pencil icon” on the right, enter ConnectWise identifier value and press on a checkbox icon. If enty is mapped successfully ConnectWise Id would appear in the corresponding column.

User interface of the mapping tool is shown below:



# Appendix 1 – setting up ConnectWise API access for integration with AppXite platform

AX platform needs to access to CW to create and modify certain data. Required access level depends on a level of automation that is required, e.g. full automation requires more permissions comparing to mostly manual integration flow when manual mapping of almost any entity is required and AX will not create or modify any entities except agreement additions. Note, that in this document it is considered that a default configuration of ConnectWise is in place, but some specific configuration may require more permissions than listed below.

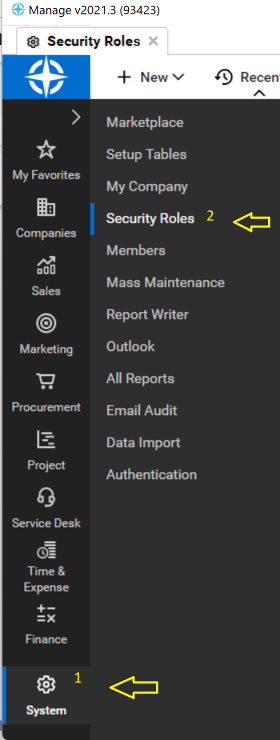
The access can be given on behalf of an existing user and security role or a new security role and a user can be created.

You need to have sufficient permissions to create new roles and users.

## Creating a security role

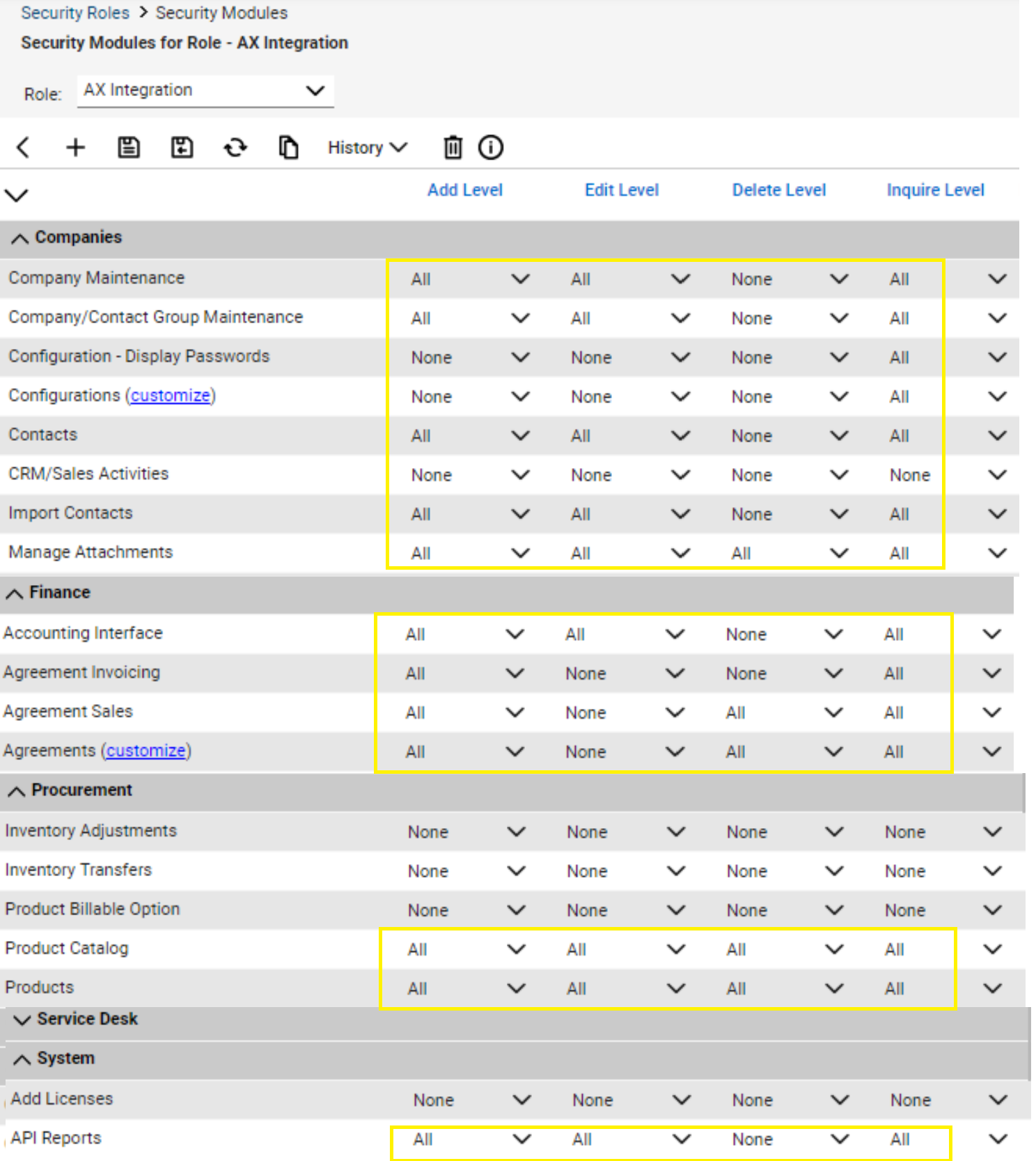
Note: this step is optional if you already have a security role that is suitable for integration.

Navigate to **System** and then to **Security Roles**. Click on **+** to create a new security role.



For **Role Id ­** - add any name you can recognize that will be used for the integration, such as ‘AX Integration and click **Save**.

Select the following permissions for the role:

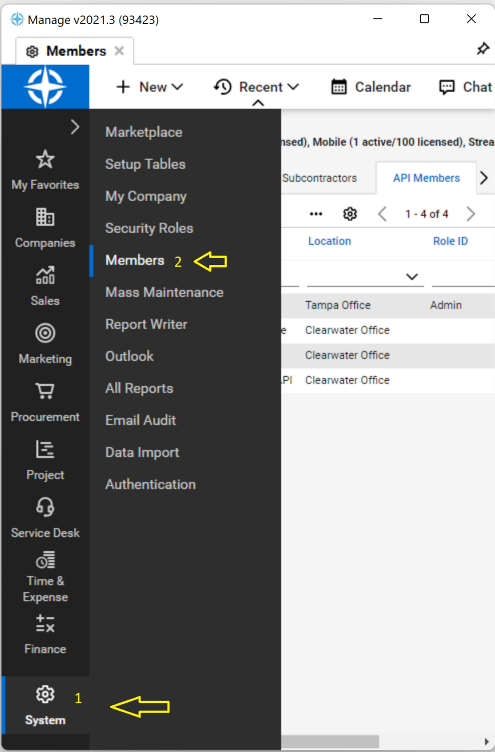


Note that permissions above are valid for default CW installation, for your configuration they may need to be adjusted.

## Creating/assigning API member

The integration will operate on behalf of API member so creating or assigning API member for the integration is required.

To set as API member navigate to **System->Members,**select the tab **API Members** and then press **+**



Set both **Member Id** and **Member Name** to AX Integration, as well select AX Integration (or whatever role you’ve specified during the previous step) in **Role Id** dropdown. As a **Business Unit** select **Admin** (or equivalent as per your configuration) and then save the record:

Graphical user interface, application

Description automatically generated

## Create API key

Finally, to create an API key select **API Keys** tab and click on **+** button. Type a description (such as “AX Integration”) and press on save icon:

Graphical user interface, text, application

Description automatically generated

Public and private keys will be generated. Please copy them both to a secure location and share with AppXite.

In addition to the keys, please provide **Site** and **Company** information (displayed on the login screen):

Graphical user interface, text, application

Description automatically generated

## Creating Admin User for setting up API access

Alternatively, you can create a new user with an Admin role for AppXite so that AppXite support can set up API access themselves. Once the API access is set up, you can disable or delete this user. To do so, navigate to System > Members and press the + button. Please refer to the example below, where important fields are highlighted, and other required fields should be filled in with default values:

Graphical user interface

Description automatically generated