datto | AUTOTASK PSA

Sage Peachtree

Last updated on Friday, May 21, 2021

Table of Contents

The Sage 50™ Integration

The Sage 50 Integration

Autotask PSA has integrated with Sage 50 to allow customers to transfer Autotask PSA billing items to Sage 50.

The integration takes advantage of a middleware called Boomi, a tool that enables you to build, deploy and manage simple to complex integration processes using only a Web browser. The Boomi AtomSphere manages the integration between Autotask PSA and an on-premise application like Sage 50. It features a visual interface that eliminates the need to learn complicated APIs.

Autotask PSA customers receive the standard integration with Sage 50 which will work well for most customers as-is, but given that Sage 50 is highly customizable, they can also develop a customized integration to match by engaging Boomi, Autotask Professional Services, or do the work themselves.

Supported Sage 50 versions

This integration supports the following Sage 50 versions:

- US Pro, Complete, Premium, and Quantum 2007-2011
- US Pro, Complete, Premium, and Quantum 2012, 2014, 2015, 2016, 2018

Canadian versions are not supported.

> IMPORTANT Our middleware provider Boomi doesn't support versions of Sage 50 later than 2018. Therefore, we only support legacy customers.

About this document

This document describes the features of the standard integration provided by Autotask PSA.

- The integration matches Autotask PSA companies to Sage 50 customers. If a match cannot be found, the integration can either generate an error message, or create a Customer in Sage 50 from the Autotask PSA company details. Refer to "Entering the Sage 50 customer ID into the Autotask company number" on page 12.
- This document also describes the initial configuration steps in both Autotask PSA and Sage 50 that allow us to match Autotask PSA billing items to Sage 50 inventory items using the External Number on Autotask Billing Code lists. Refer to "Entering the Sage 50 item number on Autotask billing codes" on page 14.
- The integration provides the default field mapping details for all supported item types. Refer to "Default field mapping to Sage 50" on page 15.
- The document provides a complete workflow for each billing item type from creation in Autotask PSA to the export of the XML file that Boomi uses to populate the Sage 50 items. Refer to "Workflow of billable items in Autotask" on page 47.

Initial setup tasks

Before the Boomi process can be configured, and the Boomi Atom deployed, initial setup tasks need to be completed by the customer. These tasks are brought together in the "Sage 50™ Setup Checklist" on page 7. In large part, the items listed below provide detailed information to accomplish the more involved tasks from the checklist.

Reference the Sage 50™ Customer ID in Autotask

New Customers who use the Sage 50 Integration should ensure that the Sage 50 Customer ID (Customer_ID) is used to populate the Autotask Company Number field when initially importing Companies.

This number will act as the key field for assigning each billing transaction to the proper customer in Sage 50. In some cases with existing Customers, this number may have to be added manually or through a back-end process. If you are an existing Autotask PSA customer and would like Autotask PSA to update your Company Numbers with the Sage 50 Customer_ID, contact Autotask Customer Service.

Refer to "Entering the Sage 50 customer ID into the Autotask company number" on page 12.

Reference the Sage 50 item number on Autotask billing codes

Invoice Items are mapped to their Sage 50 counterparts by referencing the Sage 50 Item ID (**Item_ID**) in the External Number field of the Autotask Billing Code associated with the transaction. Refer to "Entering the Sage 50 item number on Autotask billing codes" on page 14.

Setup for handling billing code matching failures

To transfer a transaction with an Billing Code that is missing the Sage 50 **Item_ID** or where the Item ID was entered incorrectly, you must create a Sage 50 Item with the description “Autotask – No Allocation Match Found”.

This can be accomplished by selecting **Maintain > Inventory Items** from the menu and selecting the “New” icon on the resulting Maintain Inventory Items page. Note that in addition to the **Item ID** (“AUTOTASK”) that **Item Class** is also a required field. **Item Class** should be set to “Description only”. The short **Description** should be set to “No Allocation Match Found” and the **Description** under the General tab set to “Autotask – No Allocation Match Found”.

For more information on creating a Sage 50 inventory item, refer to your Sage 50 documentation.

Sage 50™ Setup Checklist

The following guide outlines the setup process for integration between Sage 50 and Autotask PSA. Please complete this checklist and return it to your Autotask Implementation Manager.

Step 1: Sage 50™ setup

1. Determine which version of **Sage 50** you are running. Indicate your version here:
   __ Pro, Complete, Quantum or Premium 2007
   __ Pro, Complete, Quantum or Premium 2008-2010
   __ Pro, Complete, Quantum or Premium 2011
2. __ In **Sage 50**, create an Inventory Item with the Item ID = **AUTOTASK** and the Description = **No Allocation Match Found**. The Inventory List to add this can be accessed through the menu selection **Lists > Inventory and Services**. Refer to "Setup for handling billing code matching failures" on page 5 for more information on why this is done.
3. __ Ensure that any potential Autotask PSA account managers are listed as Sales Reps in the Employee List in Sage 50. List them using their last name in all caps as their Employee ID. The Employee List can be accessed through the menu selection **Lists > Employees & Payroll > Employees**.

Step 2: Autotask setup

1. __ Open the Company Detail page for each customer record, click **Edit Company** and update the **Company Number** field with the Sage 50™ **Customer ID**.
2. __ In order to ensure that Billing Codes are always available on tickets, enable the "Require Billing Code Name field for tickets" system setting. Refer to the topic "Entering the Sage 50 item number on Autotask billing codes" on page 14 for more information.

Step 3: Invoice processing setup

1. __ Provide the Sage 50 Accounts Receivable Company number you want to use for invoices processed from Autotask PSA ____________________________

____

2. __ Provide the number of days from the Invoice date you want to have as the invoice due date ____ ________

3. If the integration process cannot match an Autotask PSA Company to a Sage 50 Customer ID, you must select which one of the following you want to take place:
    - __ Create a customer in Sage 50 from the Autotask PSA company information.
    - __ Generate an error message indicating that the customer did not exist in Sage 50.

4. Do you want the Invoice number in Sage 50 to be the Autotask PSA Invoice number or do you want Sage 50 to always assign a new number?
    - __ Use the Autotask PSA Invoice number.
    - __ Use the Sage 50 assigned Invoice number.

Step 4: Integration setup – labor items

1. __ In Autotask, set the Work Types – **External Number** field equal to the Sage 50 Inventory Item ID to use for each Autotask code the integration will use. Refer to the topic "Mapping labor items to Sage 50™" on page 23 for more information.

2. The Sage 50 Comment Text for the transferred billing item will default to **"Ticket or Task Title <carriage return> Ticket or Task Number"**.Indicate if you need a fixed Comment Text Instead: __
Enter your Comment Text: ____ ________

Step 5: Integration setup – charges

1. __ In Autotask PSA, set the Material Codes – **External Number** field equal to the Sage 50 Inventory Item Id to use for each Autotask PSA code the integration will use. Refer to the topic "Mapping charges to Sage 50™" on page 27 for more information.

2. __ Set the Autotask PSA Product **External Number** equal to the Sage 50 Inventory Item Id to use for any product that you want mapped directly to Sage 50 (rather than using the mapping of the Material Code above). Refer to the topic "Mapping charges to Sage 50™" on page 27 for more information.
3. The Sage 50 Comment Text for a transferred **Ticket** charge billing item will default to the **"Charge Name:" + Item Name + Ticket or Task Number**. Indicate if you need a fixed Comment Text instead: __
   Enter your Comment Text: ____
   ____
4. The Sage 50 Comment Text for a transferred **Project** charge billing item will default to **"Charge Name:" + Item Name + Project Name**. Indicate if you need a fixed Comment Text instead: __
   Enter your Comment Text: ____
   ____
5. The Sage 50 Comment Text for a transferred **Contract** charge billing item will default to **"Charge Name:" + Item Name + "Contract" + Contract Name**. Indicate if you need a fixed Comment Text instead: __
   Enter your Comment Text: ____
   ____

Step 6: Integration setup - recurring service items

1. __ In Autotask PSA, set the Service Codes - **External Number** field equal to the Sage 50 Inventory Item ID. Refer to the topic "Mapping recurring service items to Sage 50™" on page 32 for more information.
2. The Sage 50 Comment Text for the transferred billing item will default to the **Recurring Service Name +[Recurring Service Date Range]**. Indicate if you need a fixed Comment Text instead: __
   Enter your Comment Text:____
   ____

Step 7: Integration setup - recurring contract services adjustments

1. Adjustments are mapped to the same Inventory Item ID in Sage 50 as the underlying recurring service item.
2. The Sage 50 Comment Text for the transferred billing item will default to **Recurring Service Name + [Recurring Service Date Range] + "(Prorated)"**.

Indicate if you need a fixed Comment Text instead: __
Enter your Comment Text:____
____

Step 8: Integration setup - milestone items

1. __ In Autotask PSA, set the Milestone Codes - **External Number** field equal to the Sage 50 Inventory Item ID. Refer to the topic "Mapping billing milestones to Sage 50™" on page 38 for more information.
2. The Sage 50 Comment Text for the transferred billing item will default to **“Milestone:” + Item Name + "Contract" + Contract Name**. Indicate if you need a fixed Comment Text instead: __
Enter your Comment Text:____
____

Step 9: Integration setup - subscription items

1. [ ] In Autotask, set the billing codes - **External Number** field equal to the Sage 50 Item Inventory ID. Refer to the topic "Mapping subscription items to Sage 50™" on page 41 for more information.
2. The Sage 50 Comment Text for the transferred billing item will default to **“Subscription:” + Item Name**. Indicate if you need a fixed Comment Text instead:
__
Enter your Comment Text:____
____

Step 10: Integration setup - expenses

1. [ ] All expenses will be mapped to the same Sage 50 Inventory Item ID. Specify the Inventory Item ID for mapping expenses:_______________________
2. The Sage 50 Comment Text for the transferred expense item will default to **“Expense:” + Item Name + [Expense Category]**. Indicate if you need a fixed Comment Text instead: __
Enter your Comment Text:____
____

Step 11: Autotask setup - system setting

1. ___ In order to ensure that Work Types are available on tickets, enable the "Require Work Type Name field for tickets" system setting. Refer to "Entering the Sage 50 item number on Autotask billing codes" on page 14 for more information.

Step 12: Boomi setup - directories and locations

1. ___ The Boomi Atom should be setup on the same server that is running Sage 50. Indicate the machine name for this server:______________________
2. ___ Specify the directory where you will place the XML invoice files exported from Autotask for the process to find:

   ____

   ____
3. Please indicate if the above directory is on a local or mapped drive.
   - ___ Local drive
   - ___ Mapped drive

Entering the Sage 50 customer ID into the Autotask company number

During the invoice transfer to Sage 50, each billing transaction must be matched to the correct Sage 50 customer. The integration uses the Autotask Company Number field as the key field for assigning each billing transaction to the proper customer in Sage 50. The integration uses Autotask Web Services to look up the company number based on the Autotask Account_ID found in the XML export file.

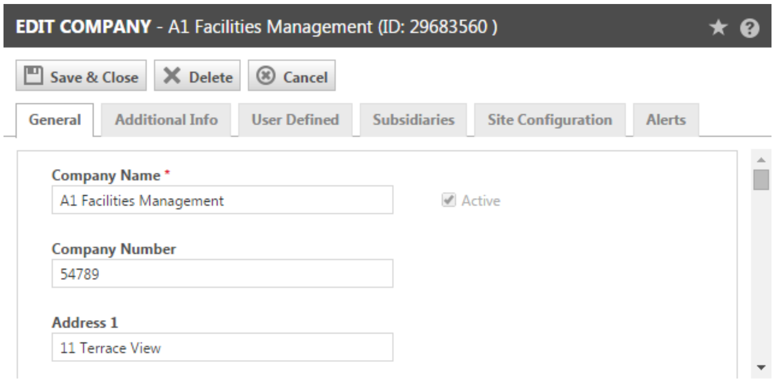
Before your first invoice transfer, you must enter the Sage 50 customer ID into the Autotask Company Number field.

How to...

Enter the company number

To add a company number to the Autotask PSA company record, do the following:

1. From the list of companies, right-click the company and select **Edit Company** The Edit Company form will open.
2. Enter the Sage 50 Customer ID into the **Company Number** field.



EDIT COMPANY - A1 Facilities Management (ID: 29683560)

Company Name *
A1 Facilities Management

☑ Active

Company Number
54789

Address 1
11 Terrace View

3. **Save & Close**.

The following steps will be followed to match the invoice items to the appropriate Sage 50 account if the company number is missing.

Missing company numbers

If the Company Number field in Autotask PSA has not been populated with the Sage 50 customer ID, the integration attempts to match the Autotask PSA company name to the Sage 50 customer name. An exact match is required. If the integration is unable to match the company name to the customer name, there are two options:

1. The integration can be configured to create a Sage 50 customer from Autotask PSA company details. For information on how Autotask PSA fields are mapped to Sage 50, refer to "Mapping company fields to Sage 50™" on page 16.
2. The integration can be configured so the transfer of associated items will fail, and an error message is emailed to a designated recipient.

Entering the Sage 50 item number on Autotask billing codes

Autotask billing items are mapped to Sage 50 billing items by referencing the Sage 50 Item ID (Item_ID) in the External Number field on the Autotask billing code.

Billing codes are required for all billing transactions, with one exception: Work Types on tickets may not be required in your Autotask PSA database. If you are using the Sage 50™ Integration, you must make Work Type required. refer to Configuring field properties.

To ensure the successful transfer of billing items to Sage 50, do the following:

1. Navigate to ☰ **> Admin >Finance, Accounting & Invoicing > Billing Codes**.
2. Hover over the context menu and select **Edit**.
3. On the form that opens, enter the Sage 50 Item ID into the **External Number**field.
4. Be sure to enter the Sage 50 Item ID into all billing codes on all tabs.

Default field mapping to Sage 50

The following topics provide the default mapping details for Company & Invoice items:

"Mapping company fields to Sage 50™" on the next page

"Mapping invoice level fields to Sage 50™ " on page 19

The following topics provide the default mapping details for supported billing item types:

"Mapping labor items to Sage 50™" on page 23

"Mapping charges to Sage 50™" on page 27

"Mapping recurring service items to Sage 50™" on page 32

"Mapping transaction adjustments to Sage 50™" on page 35

"Mapping expense items to Sage 50™" on page 44

Mapping company fields to Sage 50™

If during your integration, you selected to allow Autotask PSA to create a new customer in Sage 50, the Autotask PSA company fields will be mapped to Sage 50 PAW Customer fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
ID	ID=cust_ID if it exists in XML; Otherwise, use Sage 50 Customer Number obtained via query using AT cust-name. Normally Customer_ID=cust_ID. If a match is not found, a match based on cust_name to customer_name is attempted. If that fails, the integration logic will follow the setting you selected during implementation. Refer to "Entering the Sage 50 customer ID into the Autotask company number" on page 12.	ADDCUSTOMER (Setup) cust_ID cust_name
Name	1:1	cust_name
BillToAddress/Line1	1:1	billing_addr1
BillToAddress/Line2	1:1	billing_addr2
BillToAddress/City	1:1	billing_city
BillToAddress/State	1:1	billing_region
BillToAddress/Zip	1:1	billing_postal_code
BillToAddress/Country	1:1	billing_country
ShipToAddress/Name	1:1	cust_name

Sage 50	Integration Logic	Autotask PSA XML Field
ShipToAddress/Line1	1:1	addr1
ShipToAddress/Line2	1:1	addr2
ShipToAddress/City	1:1	city
ShipToAddress/State	1:1	account_region
ShipToAddress/Zip	1:1	postal_code
ShipToAddress/Country	1:1	country
PhoneNumber	Reformat for Sage 50 compatibility	phone
FaxNumber	Reformat for Sage 50 compatibility	fax
Web_Address	Web service call again Company entity using account_id to obtain Web Address. Web_Address =WebAddress	account_id Web_Address (WSAPI)
Terms	Use terms as set in Sage 50 Customer Defaults	Default terms (Sage 50)

Mapping invoice level fields to Sage 50™

The fields in the XML export file that contain invoice-level information such as Invoice Date or Invoice Total are mapped to Sage 50 PAW_Invoice fields as follows:

Sage 50	Integration Logic	Autotask PSAXML Field
Customer_ID	Normally Customer_ID= Cust_id. If a match is not found, a match based on Cust_name to Customer_Name is attempted. If that fails, the integration logic will follow the settings you selected during implementation. Refer to "Entering the Sage 50 customer ID into the Autotask company number" on page 12.	cust_id cust_name Invoice_id AccountNumber (WSAPI) account_id AccountName (WSAPI)
Customer_Name	1:1	cust_name
Invoice_Number	If INVOICEOVERRIDE is True, then Invoice_Number=blank. Otherwise, if invoice_number is populated, Invoice_Number=invoice_number. Otherwise Invoice_Number=invoice_id	INVOICEOVERRIDE (setup) invoice_id invoice_number
Date	1:1	invoice_date
ShipToAddress/Name	Currently not connected. Need to use cust_name instead	cust_name
ShipToAddress/Line1	1:1	addr1

Sage 50	Integration Logic	Autotask PSAXML Field
ShipToAddress/City	1:1	city
ShipToAddress/State	1:1	region
ShipToAddress/Zip	1:1	postal_code
ShipToAddress/Country	1:1	country
Customer_PO	1:1	purchase_order_number
Date_Due	Date_Due=invoice_date + OFFSETDAYS	invoice_date OFFSETDAYS (setup)
Displayed_Terms	Query from Sage 50 to get Customer terms.	cust_name CustomerID (Sage 50) Terms_Type (Sage 50)
Sales_Representative_ID	Through Web Service Calls: - Using the Company ID to obtain the OwnerResourceID from the Company entity. - Using the OwnerResourceID obtain FirstName and LastName from the Resource entity. Assign the LastName in all caps to Sales_Representative_ID	Account_Id OwnerResourceID (WSAPI) FirstName (WSAPI) LastName (WSAPI)

Sage 50	Integration Logic	Autotask PSAXML Field
Accounts_Receivable_ Account	1:1	ARACCOUNT (Setup)
Accounts_Receivable_ Amount	1:1	Invoice_total
Note	1:1	comments

Mapping labor items to Sage 50™

When the transaction type of a billing item is equal to Labor, the Autotask PSA XML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
Item_ID	Set the Item_ID =allo-ocation_code_ external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to "AUTOTASK"	allocation_code_ external_number
Unit_Price	if (hourly_billing_rate-e==0||hourly_billing_ rate=="") { Unit_Price =extended_price; } Else Unit_Price =hourly_billing_rate if (billable_hours-s=0||billable_hours=1) { Unit_Price =exte-ended_Price/billable_ hours	billable_hours, hourly_billing_rate, extended_price
Amount	Amount=extended_ price	extended_price
Quantity	if (Billable_Hours=0) {QUANTITY=1;} else Quantity=Billable_ Hours;	billable_hours
GL_Account	Sage 50 Query using Item_ID to obtain GL_	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSA XML Field
	Sales_Account. Set GL_Sales_Account=GL_Account	
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Type=Tax_Type.	Item_ID (Sage 50)
SalesOrderDistributionNumber	1:1	Invoice_item_id
GL_Inventory_Account	Query via Sage 50 Item_ID	Item_ID (Sage 50)
Description	- Sage 50 Query using Item_ID to obtain Description - If LABORCOMMENT is blank, then Description = ([Sage 50 item description] + [task_or_ticket_title]+ [task_or_ticket_number]) - Otherwise, Description = [Sage 50 item description] + [LABORCOMMENT]	Item_ID (Sage 50) LABORCOMMENT (Setup) task_or_ticket_number task_or_ticket_title

NOTE Although there is a default for the second half of the Description field, (the part after the Sage 50 item description), this portion can be set to be overridden with fixed

> ext that a customer has specified to use per their Sage 50 setup checklist for Labor
> ems.

Mapping charges to Sage 50™

In Autotask PSA, "charge" billing items are products or material items that can be associated with tickets, projects or contracts. Charges are mapped to Sage 50 either by the External Product ID for products, or by External Number for material codes. The only difference seen between the different associations (tickets, projects or contracts) is the Autotask PSA fields that are used to populate the line item Description field in Sage 50.

When the transaction type of a billing item is equal to Charge, the Autotask PSA XML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSAXML Field
Item_ID	Set the Item_ID =allocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to “AUTOTASK”	allocation_code_external_number
Unit_Price	If allocation_code_non_billable =Yes, Unit_Price=0; If allocation_code_non_billable=No, and hourly_billing_rate=0, then Unit_Price=extended_price/quantity; Otherwise, Unit_Price= hourly_billing_rate	allocation_code_non_billable hourly_billing_rate extended_price quantity
Amount	If allocation_code_non_billable=Yes, then Amount=0; Otherwise, Amount= extended_price	allocation_code_non_billable extended_price
Quantity	If allocation_code_non_billable =Yes, QUANTITY=1; If allocation_code_non_billable=No, and quantity=0, then QUANTITY=1, Otherwise QUANTITY=quantity	allocation_code_non_billable quantity
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSAXML Field
Tax_Type	Sage 50 Query using Item_ ID to obtain Tax_Type. Set Tax_Type = Tax_Type	Item_ID (Sage 50)
SalesOrder-DistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ ID to obtain GL_Inventory_ Account. Set GL_Inventory_ Account=GL_Inventory_ Account	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSAXML Field
Description		

NOTE Although there is a default for the second half of the Description field, (the part after the Sage 50 item description), this portion can be set to be overridden with fixed text that a customer has specified to use per their Sage 50 setup checklist for the individual charge associations.

Mapping recurring service items to Sage 50™

Recurring service items may be recurring services or recurring service bundles.

When the transaction type of a billing item is equal to Recurring Service, the Autotask XML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask XML Field
Item_ID	Set the Item_ID =allo-ocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to "AUTOTASK"	allocation_code_external_number
Unit_Price	1:1	service_unit_price
Amount	1:1	extended_price
Quantity	1:1	service_units
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Type=Tax_Type	Item_ID (Sage 50)
SalesOrder-DistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ID to obtain GL_Inventory_Account. Set GL_Inventory_Account=GL_Inventory_Account	Item_ID (Sage 50)
Description	1. Sage 50 Query using Item_ID to obtain item	Item_ID (Sage 50) RECURRINGSERVICECOMMENT (Setup)

Sage 50	Integration Logic	Autotask XML Field
	Description 2. If RECURRINGSERVICECOMMENT is blank, then Description = [Sage 50 item description] + [item_name] 3. Otherwise, Description = [Sage 50 item description] + [RECURRINGSERVICECOMMENT]	item_name (Note that in the XML item_name is the combination of the actual item name + the time period)

Mapping transaction adjustments to Sage 50™

When the transaction type of a billing item is equal to “Recurring Service Adjustment” (or “Recurring Service Bundle Adjustment”), the Autotask PSAXML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
Item_ID	Set the Item_ID =allocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to "AUTOTASK"	allocation_code_external_number
Unit_Price	1:1	service_unit_price
Amount	1:1	extended_price
Quantity	1:1	service_units
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Type=Tax_Type	Item_ID (Sage 50)
SalesOrder-DistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ID to obtain GL_Inventory_Account. Set GL_Inventory_Account=GL_Inventory_Account	Item_ID (Sage 50)
Description	1. Sage 50 Query using Item_ID to obtain item Description 2. If	Item_ID (Sage 50) RECURRINGSERVICEADJUSTMENTCOMMENT

Sage 50	Integration Logic	Autotask PSA XML Field
	RECURRINGSERVICEADJUSTMENTCOMMENT is blank, then Description = [Sage 50 item description] + [item_name] 3. Otherwise, Description = [Sage 50 item description] + [RECURRINGSERVICEADJUSTMENTCOMMENT]	(Setup) item_name

Mapping billing milestones to Sage 50™

When the transaction type of a billing item is equal to Milestone, the Autotask PSA XML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
Item_ID	Set the Item_ID =allocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to "AUTOTASK"	allocation_code_external_number
Unit_Price	1:1	service_unit_price
Amount	1:1	extended_price
Quantity	1:1	quantity
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Type=Tax_Type	Item_ID (Sage 50)
SalesOrderDistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ID to obtain GL_Inventory_Account. Set GL_Inventory_Account=GL_Inventory_	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSA XML Field
	Account	
Description	1. Sage 50 Query using Item_ID to obtain item Description 2. If MILESTONECOMMENT is blank, then Description = [Sage 50 item description] + "Milestone:" + [item_name] + "Contract:" + [contract_name] 3. Otherwise, Description = [Sage 50 item description] + [MILESTONECOMMENT]	Item_ID (Sage 50) MILESTONECOMMENT (Setup) item_name contract_name

Mapping subscription items to Sage 50™

When the transaction type of a billing item is equal to Subscription, the Autotask PSA XML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
Item_ID	Set the Item_ID =alloocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to “AUTOTASK”	allocation_code_external_number
Unit_Price	1:1	service_unit_price
Amount	1:1	extended_price
Quantity	1:1	service_units
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Typee=Tax_Type	Item_ID (Sage 50)
SalesOrderDistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ID to obtain GL_Inventory_Account. Set GL_Inventory_Account=GL_Inventory_Account	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSA XML Field
Description	1. Sage 50 Query using Item_ID to obtain item Description 2. If SUBSCRIPTIONCOMMENT is blank, then Description = "Subscription:" + [item_name] 3. Otherwise, Description = [Sage 50 item description] + [SUBSCRIPTIONCOMMENT]	Item_ID (Sage 50) SUBSCRIPTIONCOMMENT (Setup) item_name

Mapping expense items to Sage 50™

All expense items transferred from Autotask PSA will use the same Sage 50 item number. This item number is entered into the Boomi Integration tool.

When the transaction type of a billing item is equal to Expenses, the Autotask PSAXML fields are mapped to Sage 50 PAW_Invoice\SalesLine fields as follows:

Sage 50	Integration Logic	Autotask PSA XML Field
Item_ID	Set the Item_ID =allo-ocation_code_external_number. If the resultant Item_ID cannot be matched in Sage 50, then set the Item_ID to “AUTOTASK”	allocation_code_external_number
Unit_Price	1:1	service_unit_price
Amount	1:1	extended_price
Quantity	1:1	service_units
GL_Account	Sage 50 Query using Item_ID to obtain GL_Sales_Account. Set GL_Sales_Account = GL_Account	Item_ID (Sage 50)
Tax_Type	Sage 50 Query using Item_ID to obtain Tax_Type. Set Tax_Type-e=Tax_Type	Item_ID (Sage 50)
SalesOrder-DistributionNumber	1:1	Invoice_item_id
GL_Inventory Account	Sage 50 Query using Item_ID to obtain GL_Inventory_Account. Set GL_Inventory_Accoun-t=GL_Inventory_	Item_ID (Sage 50)

Sage 50	Integration Logic	Autotask PSA XML Field
	Account	
Description	1. Sage 50 Query using Item_ID to obtain item Description 2. If EXPENSECOMMENT is blank, then Description = "Expense:" + [item_name] + [expense_type] 3. Otherwise, Description = [Sage 50 item description] + [EXPENSECOMMENT]	Item_ID (Sage 50) EXPENSECOMMENT (Setup) item_name expense_type

Workflow of billable items in Autotask

The following is an overview of the workflow of billable items from creation in Autotask PSA to the export of the XML file Boomi imports to Sage 50.

Create the billing item

Create a charge

Charges (billing items for products) can be associated with a contract, project or ticket. The process of adding a charge is identical. Refer to Tracking Project Charges.

Create an expense

When you create an expense Report, you can make all or selected expenses billable to a selected company. After the expense report is approved by the expense report approver, billable expenses are invoiced to the customer. Refer to Add and Edit Expenses and Approve or Reject Expense Reports.

Enter time on a task or ticket to create a billable labor item

All time entries on tasks and tickets that are under a time & materials contract or no contract and have a billable work type are directly billed to the customer. Refer to Add a Ticket Time Entry and Enter Regular Time and Project Time.

Create a milestone billing item

Billing Milestones are creates when a fixed price contract is set up. Each billing milestone is associated with a required Milestone Code. The Milestone Code is mapped to the Sage 50 ITEMNMBR field. Refer to Create a fixed price contract.

Create recurring services on contracts

Recurring Services and Recurring Service Bundles are added to a recurring service contract on the **Services** page. Refer to Add and Remove Services. Making adjustments to services is also a billing event. Refer to Adjust the Number of Units and Price of a Service.

Create a subscription billing item for a product

Subscriptions are billing items associated with configuration items. Each subscription item is associated with a required Billing Code, which is mapped to the Sage 50 ITEMNMBR field. Refer to Add or Edit a Billing Subscription for a Configuration Item.

Approve and post all billing items

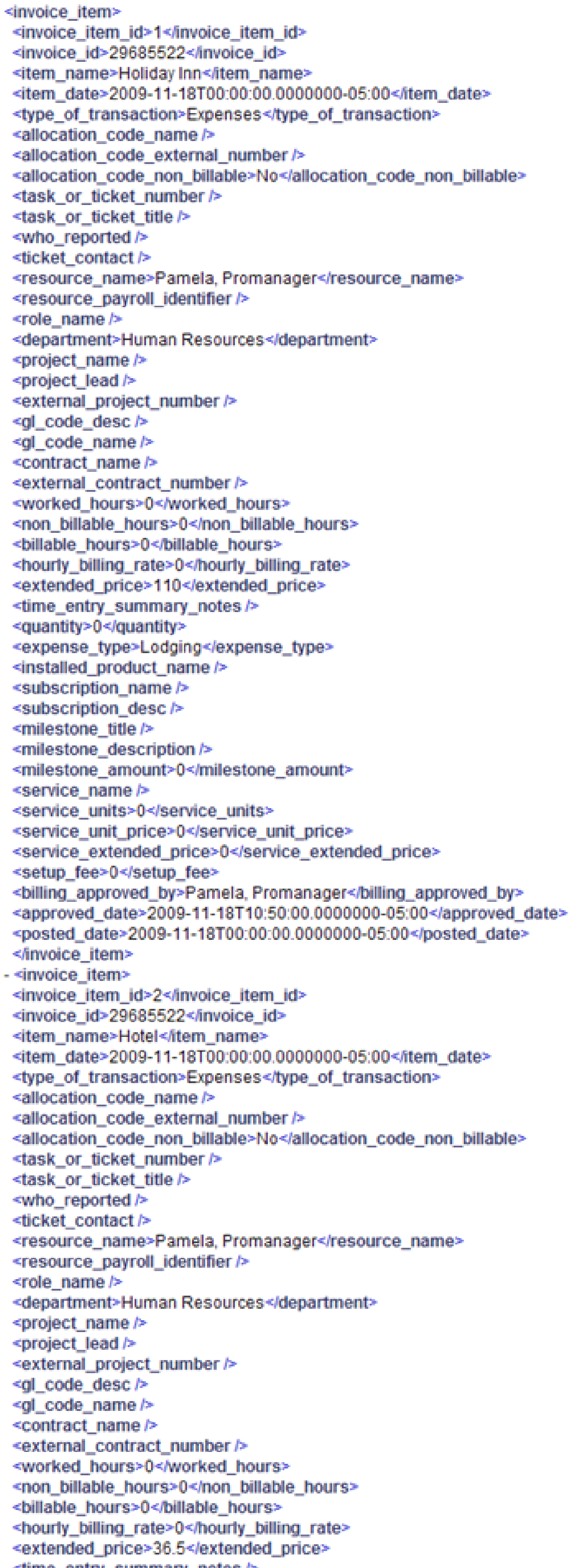
All billing item types must go through a financial approval step next. Refer to Approve and Post Billing Items.

Create invoice and XML file

Once approved and posted, the billing items will be available for invoicing on Items to Invoice.

When you process the invoice, you must select the Processing Action "Create Autotask Invoices and XML file". You can also change the Invoice Date, Invoice Date Range, Purchase Order Number, and Invoice Notes.

When you select **Process Invoices**, a printable invoice is created in Autotask PSA. At the same time, an XML file is generated:



```
<invoice_item>
 <invoice_item_id>1</invoice_item_id>
 <invoice_id>29685522</invoice_id>
 <item_name>Holiday Inn</item_name>
 <item_date>2009-11-18T00:00:00.0000000-05:00</item_date>
 <type_of_transaction>Expenses</type_of_transaction>
 <allocation_code_name />
 <allocation_code_external_number />
 <allocation_code_non_billable>No</allocation_code_non_billable>
 <task_or_ticket_number />
 <task_or_ticket_title />
 <who_reported />
 <ticket_contact />
 <resource_name>Pamela, Promanager</resource_name>
 <resource_payroll_identifier />
 <role_name />
 <department>Human Resources</department>
 <project_name />
 <project_lead />
 <external_project_number />
 <gl_code_desc />
 <gl_code_name />
 <contract_name />
 <external_contract_number />
 <worked_hours>0</worked_hours>
 <non_billable_hours>0</non_billable_hours>
 <billable_hours>0</billable_hours>
 <hourly_billing_rate>0</hourly_billing_rate>
 <extended_price>110</extended_price>
 <time_entry_summary_notes />
 <quantity>0</quantity>
 <expense_type>Lodging</expense_type>
 <installed_product_name />
 <subscription_name />
 <subscription_desc />
 <milestone_title />
 <milestone_description />
 <milestone_amount>0</milestone_amount>
 <service_name />
 <service_units>0</service_units>
 <service_unit_price>0</service_unit_price>
 <service_extended_price>0</service_extended_price>
 <setup_fee>0</setup_fee>
 <billing_approved_by>Pamela, Promanager</billing_approved_by>
 <approved_date>2009-11-18T10:50:00.0000000-05:00</approved_date>
 <posted_date>2009-11-18T00:00:00.0000000-05:00</posted_date>
 </invoice_item>
- <invoice_item>
 <invoice_item_id>2</invoice_item_id>
 <invoice_id>29685522</invoice_id>
 <item_name>Hotel</item_name>
 <item_date>2009-11-18T00:00:00.0000000-05:00</item_date>
 <type_of_transaction>Expenses</type_of_transaction>
 <allocation_code_name />
 <allocation_code_external_number />
 <allocation_code_non_billable>No</allocation_code_non_billable>
 <task_or_ticket_number />
 <task_or_ticket_title />
 <who_reported />
 <ticket_contact />
 <resource_name>Pamela, Promanager</resource_name>
 <resource_payroll_identifier />
 <role_name />
 <department>Human Resources</department>
 <project_name />
 <project_lead />
 <external_project_number />
 <gl_code_desc />
 <gl_code_name />
 <contract_name />
 <external_contract_number />
 <worked_hours>0</worked_hours>
 <non_billable_hours>0</non_billable_hours>
 <billable_hours>0</billable_hours>
 <hourly_billing_rate>0</hourly_billing_rate>
 <extended_price>36.5</extended_price>
 <time_entry_summary_notes />
```

NOTE Best Practice: We recommend that you save all XML export files into the same folder, using a naming convention such as Year_Month_Day_Invoices. When it comes time to run the **AT to Sage 50** Invoices process, copy the files you want to process into the specified Disk Connector directory removing any left over files there from the last run.

Index