Software Requirements Specification

for

Web API for Control

(WAPIC)

Version 0.77

Prepared by Scott St. Cyr

Cyrious Software

7/7/2014

Table of Contents

1. Introduction

1.1 Overview

The Web API for Control (WAPIC) is an interface that allows external access into the Control database. It will be implemented in two phases.

Phase 1 of WAPIC will provide access to the following areas:
- Customers and Contacts
- Addresses, Phone Number, Email, etc.
- Calendar Activities
- Timeclock

Phase 2 of WAPIC will provide access to the following areas:
- Orders and Line Items
- Parts and Inventory
- Production Information

This document outlines this service as well as the installation, use, and best practice for WAPIC.

1.2 User Stories

In Phase I, WAPIC is designed to satisfy the following user stories concerning interaction with the Control database:

- A web site/developer can pull a list of customers.
- A web site/developer can update customers' information.
- A web site/developer can create customers information.

- A web site/developer can pull the contact information from the Control database.
- A web site/developer can update contact information in the Control database.
- A web site/developer can create new contacts for a customer in the Control database.

- A web site/developer can load the Calendar Activities for a date range for an employee.
- A web site/developer can update Calendar Activities for an employee.
- A web site/developer can create Calendar Activities for an employee.

- A web site/developer can load the current timeclock status for one or more employees.
- A web site/developer can clock a user in or out.

- A web site/developer can change a user's station.

In Phase II, WAPIC is designed to satisfy the following user stories concerning interaction with the Control database:

- A web site/developer can retrieve orders.
- A web site/developer can edit existing orders.
- A web site/developer can create new orders.

- A web site/developer can retrieve parts on an order.
- A web site/developer can void parts on an order.
- A web site/developer can add new parts to an order.

- A web site/developer can add usage to a part on an order.

- A web site/developer can change the status of an order.
- A web site/developer can change the station of a line item.

- A web site/developer can add artwork to an order.

- A web site/developer can retrieve shipments for an order.
- A web site/developer can edit existing shipments for an order.
- A web site/developer can create new shipments for an order.

1.3 Intended Audience and Reading Suggestions

This document is written for **web developers** that are interacting with the WAPIC service. It is *not* intended as a guide for the non-technical person.

2. WAPIC Overview

2.1 API Structure

WAPIC is built on a number of different Microsoft technologies. Microsoft SQL Server is the backend database for Control. WAPIC uses Microsoft Entity Framework to connect to the database and save and retrieve data.

The API uses an object model approach, rather than a linked table approach. The web models, therefore, do not match the structure of the database. The database models are assembled, simplified, cajoled, and otherwise transformed into the appropriate web models within WAPICs core.
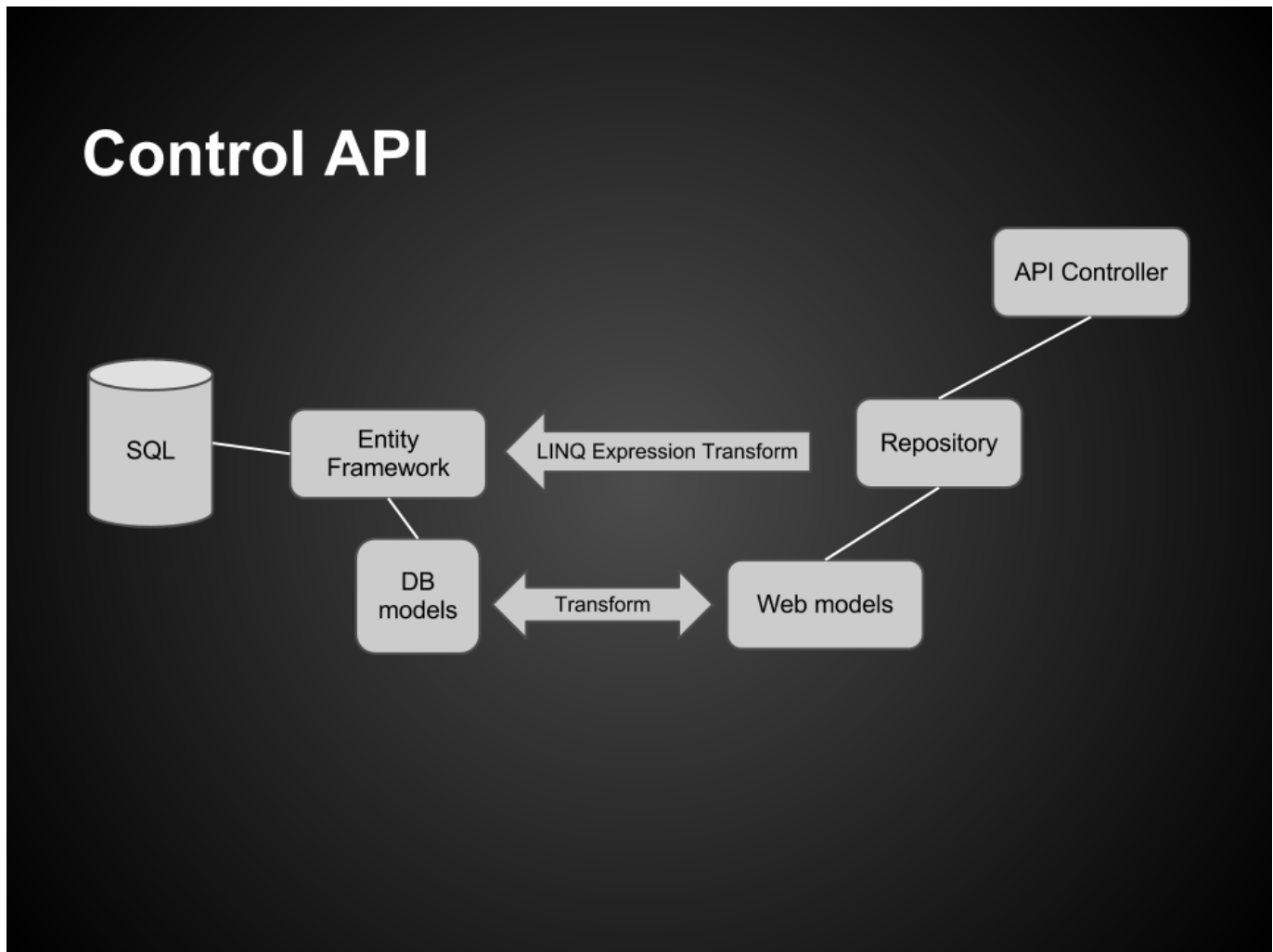
3. Accessing WAPIC Resources

3.1 oData Syntax

The REST standard indicates how communication between the client and server take place. It does nothing to standardize the structure of the communication. The Open Data organization created the oData Protocol as a way to standardize the *syntax* and *grammar* of REST communication. Tutorials, sample code, and reference material on the oData syntax can be found at http://www.odata.org.

The WAPIC is based on the oData protocol though it is not a full implementation of oData (which is very extensive). The general structure of a Cyrious request is:

//<<server>>[:port]/wapic/<<resource path>> [? <<query option>> [& <<query option>> …]] [1]

Where

- **Server** is the URI of the hosting server.
- **Port** is the port number the service is running on. If not provided, this defaults to 443 for HTTPS requests and port 80 for HTTP requests.
- **WAPIC** is the actual oData service handling requests.
- **Resource path** is the path to the object or objects being requested or worked with, such as Companies(1345) or Orders(1053)/Company/CompanyName. It may also refer to a command, such as Order(1053)/DoSomething.
- **Query Options** specify three types of information: System Query Options, Custom Query Options, and Service Operation Parameters.

A few example queries will illustrate how easy it is to use the WAPIC.

3.1.1 System Query String Operations

System Query Options are query string parameters a client may specify to control the amount and order of the data that an OData service returns for the resource identified by the URI. The names of all System Query Options are prefixed with a "$" character.

Operation	Meaning
$filter	The filter operation filters the selected data so only certain information is returned. See Section 3.1.2, **Filter Query Option ($filter)** for more information.

---

[1] See http://www.odata.org/developers/protocols/uri-conventions for more on oData URI structures.

Operation	Meaning
$orderby	**Orderby System Query Option ($orderby)**. The $orderby System Query Option specifies an expression for determining what values are used to order the collection returned.
$top	**Top System Query Option ($top)**. The $top System Query Option identifies a subset of the requested results. This subset is formed by selecting only the first N items of the set, where N is a positive integer specified by this query option. The value must be >= 0. If a request contains a $top query option, it should always contain an $orderby query option or there is no guarantee that the data returned in any query will be consistent as the order from request to request will not necessarily be the same.
$skip	**Skip System Query Option ($skip)**. The $skip System Query Option identifies a subset of the request results. That subset is defined by seeking N Entries into the Collection and selecting only the remaining Entries (starting with Entry N+1). N is a positive integer as specified by this query option. If a request contains a $skip query option, it should always contain an $orderby query option or there is no guarantee that the data returned in any query will be consistent as the order from request to request will not necessarily be the same.
$expand	**Expand System Query Option ($expand)**. The $expand System Query Option indicates that sub-objects associated with the Resource Path must be represented inline (i.e. eagerly loaded). For example, if you want to identify a category and its products, you could use two URIs (and execute two requests), one for /Categories(1) and one for /Categories(1)/Products. The '$expand' option allows you to identify related objects with a single URI. The syntax of a $expand query option is a comma-separated list of Navigation Properties. Additionally each Navigation Property can be followed by a forward slash and another Navigation Property to enable identifying a multi-level relationship.
$select	**Select System Query Option ($select)**. The $select System Query Option specifies that a response from an OData service should return a subset of the Properties. The value of a $select System Query Option is a comma-separated list of selection clauses. Each selection clause may be a Property name, Navigation Property name, or the "*" character.
$inlinecount	$inlinecount=allpages returns the number of records in the total results, not just the number returned in this group. This will differ from the returned count when the return results are limited to a maximum number.

3.1.2 Filter Query Option ($filter)

The $filter System Query Option selects a subset of the results identified by the Resource Path. The subset is determined by selecting only the objects that satisfy the expression specified by the query option.

The expression language that is used in $filter operators supports references to properties and literals. The literal values can be strings enclosed in single quotes, numbers and boolean values (true or false) or any of the additional literal representations shown in the Abstract Type System section.

3.1.2.1 $Filter Operators

The operators supported in the expression language are shown in the following table:

Operator	Meaning	Sample Usage

ge	Greater Than or Equal To	BalanceDue **ge** 100
gt	Greater Than	BalanceDue **gt** 0
		ModfiedDate gt DateTime'2011-05-01'
le	Less Than or Equal To	BalanceDue **le** 99.99
		ModfiedDate le DateTime'2011-05-01'
lt	Less Than	BalanceDue **lt** 100.00
		ModfiedDate **lt** DateTime'2011-05-01'
eq	Equal To	CompanyName **eq** 'Lowes'
		Company/PrimaryContact/LastName eq 'Smith'
ne	Not Equal To	FirstName ne 'Bob'
and	Join two independent filters where both must match	FirstName eq 'Bob' and LastName eq 'Smith'
or	Join two independent filters where either must match	Company eq 'Lowes' or Company eq 'Home Depot'
( )	Grouping – Used to make the logic unambiguous	(Company eq 'Lowes') and (BalanceDue gt 0.00 or SaleDate gt '1/1/2013')
Not	Negate the condition	Not (Company eq 'Lowes')
Add	Add values	BalanceDue gt (CreditLimit add 100.00)
Sub	Subtract values	
Mul	Multiple Values	
Div	Divide Values	
Mod	Modulus Division of Values	(JulianDate Mod 7) eq 0

3.1.2.2 $Filter Functions

Function	Example
String Functions	
bool substringof(string po, string p1)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=substringof('Alfreds', CompanyName) eq true
bool endswith(string p0, string p1)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=endswith(CompanyName, 'Futterkiste') eq true
bool startswith(string p0, string p1)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=startswith(CompanyName, 'Alfr') eq true
int length(string p0)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=length(CompanyName) eq 19
int indexof(string p0, string p1)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=indexof(CompanyName, 'lfreds') eq 1
string replace(string p0, string find, string replace)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=replace(CompanyName, ' ', '') eq 'AlfredsFutterkiste'
string substring(string p0, int pos)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=substring(CompanyName, 1) eq 'lfreds Futterkiste'
string substring(string p0, int pos, int length)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=substring(CompanyName, 1, 2) eq 'lf'
string tolower(string p0)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=tol

	ower(CompanyName) eq 'alfreds futterkiste'
string toupper(string p0)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=toupper(CompanyName) eq 'ALFREDS FUTTERKISTE'
string trim(string p0)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=trim(CompanyName) eq 'Alfreds Futterkiste'
string concat(string p0, string p1)	http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=concat(concat(City, ', '), Country) eq 'Berlin, Germany'
Date Functions	
int day(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=day(BirthDate) eq 8
int hour(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=hour(BirthDate) eq 0
int minute(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=minute(BirthDate) eq 0
int month(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=month(BirthDate) eq 12
int second(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=second(BirthDate) eq 0
int year(DateTime p0)	http://services.odata.org/Northwind/Northwind.svc/Employees?$filter=year(BirthDate) eq 1948
Math Functions	
double round(double p0)	http://services.odata.org/Northwind/Northwind.svc/Orders?$filter=round(Freight) eq 32
double floor(double p0)	http://services.odata.org/Northwind/Northwind.svc/Orders?$filter=filter=round(Freight) eq 32
double ceiling(double p0)	http://services.odata.org/Northwind/Northwind.svc/Orders?$filter=ceiling(Freight) eq 33
Type Functions	
bool IsOf(type p0)	http://services.odata.org/Northwind/Northwind.svc/Orders?$filter=isof('NorthwindModel.Order')
bool IsOf(expression p0, type p1)	http://services.odata.org/Northwind/Northwind.svc/Orders?$filter=isof(ShipCountry, 'Edm.String')

3.1.3 Retrieving Inner Data

To retrieve child data, use of the $expand attribute.

3.1.4 Retrieving Specific Fields

To limit the returned values to specific fields, use the $select attribute. Doing so can greatly reduce the transmission time for information and speed up the results dramatically.

3.2 Retrieving Records (GET)

Objects are retrieved by executing an HTTP or HTTPS GET request against the Collection containing the object. In the WAPIC, all objects are accessed by top level collections that contain (potentially) all objects of that type. For example, the following are some of the collections accessible through the GET operation when WAPIC is fully implemented:

> GET http://(localhost)/wapic /Companies/
> GET http://(localhost)/wapic/Orders/
> GET http://(localhost)/wapic/Estimates/
> GET http://(localhost)/wapic/Payments/
> GET http://(localhost)/wapic/CalendarActivities/
> GET http://(localhost)/wapic/Business/
> GET http://(localhost)/wapic/Products/
> GET http://(localhost)/wapic/Parts/
> …

The maximum that will be returned from one GET operation is 200 records at one time. The request is, therefore, almost always filtered to return only a limited number (or one) record of that type using a **$FILTER** operation.

3.3 Adding Records (POST)

New objects are created by executing an HTTP or HTTPS POST request against the URI of the Collection where the object is to be created. The POST request includes the new object in its body. The request indicates both the content-type of the request body. The server processes the request by creating the resource, assigning default values to all the properties not indicated in the request that are optional, and sending the final state of the resource back to the client in the response.

Embedded Objects and Links

The object being created may contain links to other objects in the service. If that is the case the server is expected to create the object and the appropriate Links.

For example, to create a new product in a product category a client would execute a POST request against the wapic/Products collection with a product object containing a Link to the Product Category Object using any URI that resolves to that resource.

Alternatively a client can create and link an object to a related object by leveraging the addressing scheme if the server supports addressing related items. For example, if the address …/ProductCategories(10)/Products points at all the products in the specified category, when a POST request is issued against that products collection (instead of the top-level products collection) the server will create the new product object and automatically Link it to the parent category.

When a client needs to create multiple related objects it can do so as independent operations or they can perform a single POST with a tree of Objects. The tree is formed by using inline expansion as described in [OData-Atom] and [OData-JSON]. All expanded Objects are considered new. Servers process a request with

inline Objects by creating individual Objects and then linking them in the same way Linking would have happened in an independent request.

3.4 Updating Entire Records (PUT)

A PUT operation in oData syntax is used to replace the entire contents of one entity with new data. Generally, a PUT operation is not desireable because completely changing a record created new IDs and has other implications. Most of the time, developers will favor of the use of the PATCH operation, which allows specific fields to be changed but does not completely replace the existing entity.

3.5 Updating Partial Records (PATCH)

Often it is desirable to update some data without replacing all the contents of an Entry. In order to avoid overloading the meaning of PUT, OData uses the custom HTTPS method PATCH[2] for this scenario. A PATCH request updates only the properties indicated in the request body, and leaves untouched anything not mentioned in its current state.

While there is a distinction between PUT and PATCH for properties, the links of an object are not directly considered part of the structured data portion of an Entry and thus are not reset on PUT. WAPIC uses the PATCH semantics for links for both PUT and PATCH requests.

3.6 Deleting Records (DELETE)

Where allowed, objects are deleted by executing a DELETE request against a URI that points at the Object. If the DELETE operation executes successfully, the server returns 200 (OK) with no response body. If the object does not exist (presumably because it has already been deleted), the server interprets this as a duplicate request already carried out in accordance with the idempotency guidelines for REST, and *still* returns 200 (OK) with no response body.

If the delete is allowed on a trunk object, all branch objects off that trunk will also be deleted. For example, if you delete a prospective company it will delete all contacts, addresses, and phone numbers associated with that company. Activities for that company would remain, but the company and contact information associated with them would be reset to null.

3.7 ETags (Sequence IDs)

Cyrious uses ETags (a.k.a. Sequence IDs) for optimistic concurrency control. An ETag is a key that changes each time the object is updated. Whenever an update occurs, the current ETag must be included in the update request or the update will fail. This ensures that the changes are based on the latest copy of the data and prevents one user from overwriting the changes of another.

A few special considerations apply for ETags:

---

[2]oData syntax currently specifies the use of the verb MERGE to update only certain properties of a record. However, PATCH is expected to be the web standard for this term soon and the oData organization has stated their intent to adopt the term at that time. The Cyrious API will support either term, though programmers are encouraged to use PATCH in new development as the term MERGE will eventually be deprecated.

Software Requirements Specification for WAPIC   *Page 12*

inline Objects by creating individual Objects and then linking them in the same way Linking would have happened in an independent request.

3.4 Updating Entire Records (PUT)

A PUT operation in oData syntax is used to replace the entire contents of one entity with new data. Generally, a PUT operation is not desireable because completely changing a record created new IDs and has other implications. Most of the time, developers will favor of the use of the PATCH operation, which allows specific fields to be changed but does not completely replace the existing entity.

3.5 Updating Partial Records (PATCH)

Often it is desirable to update some data without replacing all the contents of an Entry. In order to avoid overloading the meaning of PUT, OData uses the custom HTTPS method PATCH[2] for this scenario. A PATCH request updates only the properties indicated in the request body, and leaves untouched anything not mentioned in its current state.

While there is a distinction between PUT and PATCH for properties, the links of an object are not directly considered part of the structured data portion of an Entry and thus are not reset on PUT. WAPIC uses the PATCH semantics for links for both PUT and PATCH requests.

3.6 Deleting Records (DELETE)

Where allowed, objects are deleted by executing a DELETE request against a URI that points at the Object. If the DELETE operation executes successfully, the server returns 200 (OK) with no response body. If the object does not exist (presumably because it has already been deleted), the server interprets this as a duplicate request already carried out in accordance with the idempotency guidelines for REST, and *still* returns 200 (OK) with no response body.

If the delete is allowed on a trunk object, all branch objects off that trunk will also be deleted. For example, if you delete a prospective company it will delete all contacts, addresses, and phone numbers associated with that company. Activities for that company would remain, but the company and contact information associated with them would be reset to null.

3.7 ETags (Sequence IDs)

Cyrious uses ETags (a.k.a. Sequence IDs) for optimistic concurrency control. An ETag is a key that changes each time the object is updated. Whenever an update occurs, the current ETag must be included in the update request or the update will fail. This ensures that the changes are based on the latest copy of the data and prevents one user from overwriting the changes of another.

A few special considerations apply for ETags:

---

[2]oData syntax currently specifies the use of the verb MERGE to update only certain properties of a record. However, PATCH is expected to be the web standard for this term soon and the oData organization has stated their intent to adopt the term at that time. The Cyrious API will support either term, though programmers are encouraged to use PATCH in new development as the term MERGE will eventually be deprecated.

Copyright © 2012 by Cyrious Software.

- The ETag value is always included in the data.
- During processing of successful POST, PUT and PATCH the server computes a new ETag and returns it in a response header, regardless of whether the response has a body with the actual Entry information.
- When issuing a PUT, PATCH or DELETE request, clients need to indicate an ETag in the If-Match HTTPS request header. If the If-Match header is omitted, it will be assumed to be the value of the ETag of the content object (for PUT, PATCH).
- A few PUT and PATCH modifications, typically involving the addition of extra information in a sub-collection, are allowed without regard to the previous ETag.

3.8 PUT, PATCH, and DELETE Tunnelling through POST

In many scenarios client browsers are limited to the HTTPS GET and POST methods only. In order to help work-around this limitation, the WAPIC can support method tunneling through POST. The methods that can be executed through tunneling are PATCH, PUT and DELETE.

To issue a request with method tunneling, set up a request with body and headers as needed but use POST as the HTTPS method instead of the actual required one. Then add one more header, "X-HTTP-Method-Override", and gives it the value PATCH, PUT or DELETE. For example:

        X-HTTP-Method-Override: PATCH

3.9 Cross Domain Requests

The WAPIC service supports cross domain access via JSONP or CORS.

4. WAPIC Resources

The Web API for Control (WAPIC) is a REST API for accessing information from Cyrious Control.

4.1 Phase I - Data Object Resources

4.1.1 REST Endpoints Summary

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
Company[]	../Companies	X	X		
Company	../Companies(id)	X		X	
CompanyAddress[]	../Companies(id)/Addresses	X			
CompanyAddress	../Companies(id)/Addresses(id)	X			
CompanyAddressLink[]	../Companies(id)/AddressLinks	X	X		
CompanyAddressLink	../Companies(id)/AddressLinks(id)	X		X	X
Phone[]	../Companies(id)/Phones	X	X		
Phone	../Companies(id)/Phones(id)	X		X	
*StateTaxExemptionLink[]	../Companies(id)/StateTaxExemptionLinks	X			
*StateTaxExemptionLink	../Companies(id)/StateTaxExemptionLinks(id)	X			
CalendarActivity[]	../Companies(id)/CalendarActivities	X	X		
CalendarActivity	../Companies(id)/CalendarActivities(id)	X		X	X
UDF[]	../Companies(id)/UDF	X			
UDF	../Companies(id)/UDF(name)	X		X	
Contact[]	../Companies(id)/Contacts	X	X		
Contact	../Companies(id)/Contacts(id)	X		X	
ContactAddress[]	../Companies(id)/Contacts(id)/Addresses	X			
ContactAddress	../Companies(id)/Contacts(id)/Addresses(id)	X			
ContactAddressLink[]	../Companies(id)/Contacts(id)/AddressLinks	X	X		
ContactAddressLink	../Companies(id)/Contacts(id	X		X	X

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
	)/AddressLinks(id)				
Phone[]	../Companies(id)/Contacts(id)/Phones	X	X		
Phone	../Companies(id)/Contacts(id)/Phones(id)	X		X	
CalendarActivity[]	../Companies(id)/Contacts(id)/CalendarActivities	X	X		
CalendarActivity	../Companies(id)/Contacts(id)/CalendarActivities(id)	X		X	X
UDF[]	../Companies(id)/Contacts(id)/UDF	X			
UDF	../Companies(id)/Contacts(id)/UDF(name)	X		X	
Address[]	../Addresses	X			
Address	../Addresses(id)	X		X	
Phone[]	../Phones	X			
Phone	../Phones(id)	X		X	
CalendarActivity[]	../CalendarActivities	X	X		
CalendarActivity	../CalendarActivities(id)	X		X	
Contact[]	../Contacts	X			
Contact	../Contacts(id)	X		X	
ContactAddress[]	../Contacts(id)/Addresses	X			
ContactAddress	../Contacts(id)/Addresses(id)	X			
ContactAddressLink[]	../Contacts(id)/AddressLinks	X	X		
ContactAddressLink	../Contacts(id)/AddressLinks(id)	X		X	X
Phone[]	../Contacts(id)/Phones	X	X		
Phone	../Contacts(id)/Phones(id)	X		X	
CalendarActivity[]	../Calendars(id)/CalendarActivities	X	X		
CalendarActivity	../Calendars(id)/CalendarActivities(id)	X		X	X
UDF[]	../Contacts(id)/UDF	X			
UDF	../Contacts(id)/UDF(name)	X		X	
Employee[]	../Employees	X			
Employee	../Employees(id)	X			
EmployeeAddressLink[]	../Employees(id)/AddressLinks	X	X		
EmployeeAddressLink	../Employees(id)/AddressLinks(id)	X		X	X

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
CalendarActivity[]	../Employees(id)/CalendarActivities	X	X		
CalendarActivity	../Employees(id)/CalendarActivities(id)	X		X	X
Calendar[]	../Calendars	X			
CalendarActivity[]	../Calendars(id)/CalendarActivities	X	X		
CalendarActivity	../Calendars(id)/CalendarActivities(id)	X		X	X
CompanyAddressLink[]	../CompanyAddressLinks	X	X		
CompanyAddressLink	../CompanyAddressLinks(id)	X		X	X
ContactAddressLink	../ContactAddressLinks	X	X		
ContactAddressLink[]	../ContactAddressLinks(id)	X		X	X
EmployeeAddressLink[]	../EmployeeAddressLinks	X	X		
EmployeeAddressLink	../EmployeeAddressLinks(id)	X		X	X
CompanyAddress[]	../CompanyAddresses	X			
ContactAddress[]	../ContactAddresses	X			
EmployeeAddress[]	../EmployeeAddresses	X			

4.2 Phase I – Command Resources

4.2.1 REST Endpoints Summary

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
TimeClock	../TimeClock	X			
	../TimeClock(id)/ClockIn		X		
	../TimeClock(id)/ClockIn		X		
	../TimeClock(id)/ClockOnStation('stationName')		X		

4.3 Phase II - Data Object Resources

4.3.1 Transaction REST Endpoints Summary

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
Orders[]	../Orders	X	X		
Order	../Orders(id)	X		X	
OrderLineItem[]	../Orders(id)/Items	X			
OrderLineItem	../Orders(id)/Items(id)	X		X	
TransPart[]	../Orders(id)/Parts	X			
TransPart	../Orders(id)/Parts(id)	X			
TransVariable[]	../Orders(id)/Items(id)/Variables	X			
TransVariable	../Orders(id)/Items(id)/Variables('name')	X			
TransMod[]	../Orders(id)/Items(id)/Modifiers	X			
TransMod	../Orders(id)/Items(id)/Modifiers('name')	X			
TransPart[]	../Orders(id)/Items(id)/Parts	X			
TransPart	../Orders(id)/Items(id)/Parts(id)	X			
ProductionActivities[]	../Order(id)/Items(id)/ProductionActivities	X			
ProductionActivity	../Order(id)/Items(id)/ProductionActivities(id)	X			
UDF[]	../Orders(id)/UDFs	X			
UDF	../Orders(id)/UDFs('id')	X		X	
Shipment[]	../Orders(id)/Shipments	X	X		
Shipment	../Orders(id)/Shipments(id)	X		X	
PaymentActivity[]	../Orders(id)/Payments	X			
PaymentActivity	../Orders(id)/Payments(id)	X			
CalenderActivity[]	../Orders(id)/CalendarActivities	X	X		
CalenderActivity	../Orders(id)/CalendarActivities(id)	X		X	
ProductionActivity[]	../Orders(id)/ProductionActivities	X			
ProductionActivity	../Orders(id)/ProductionActivities(id)	X			
OrderActivity[]	../Orders(id)/OrderActivities	X			
OrderActivity	../Orders(id)/OrderActivities(id)	X			
TransDetailTaxItem[]	../Orders(id)/Taxes	X			
Estimate[]	../Estimates	X	X		

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
Estimate	../Estimates(id)	X		X	
UDF[]	../Estimates(id)/UDFs	X			
UDF	../Estimates(id)/UDFs(id)	X		X	
CalenderActivity[]	../Estimates(id)/CalendarActivities	X			
CalenderActivity	../Estimates(id)/CalendarActivities(id)	X		X	
EstimateActivity[]	../Estimates(id)/EstimateActivities	X			
EstimateActivity	../Estimates(id)/EstimateActivities(id)	X			
EstimateVariation[]	../Estimates(id)/Variations	X	X		
EstimateVariation	../Estimates(id)/Variations(id)	X		X	
EstimateLineItem[]	../Estimates(id)/Variations(id)/Items	X	X		
EstimateLineItem	../Estimates(id)/Variations(id)/Items(id)	X		X	
TransPart[]	../Estimates(id)/Variations(id)/Parts	X			
TransPart	../Estimates(id)/Variations(id)/Parts(id)	X			
TransPart[]	../Estimates(id)/Parts	X			
TransPart	../Estimates(id)/Parts(id)	X			
EstimateLineItem[]	../Estimates(id)/Items	X	X		
EstimateLineItem	../Estimates(id)/Items(id)	X		X	
TransVariable[]	../Estimates(id)/Items(id)/Variables	X			
TransVariable	../Estimates(id)/Items(id)/Variables(name)	X			
TransMod[]	../Estimates(id)/Items(id)/Modifiers	X			
TransMod	../Estimates(id)/Items(id)/Modifiers(name)	X			
TransPart[]	../Estimates(id)/Items(id)/Parts	X			
TransPart	../Estimates(id)/Items(id)/Parts(id)	X			
TransDetailTaxItem[]	../Estimates(id)/Taxes	X			
EstimateLineItem[]	../EstimateLineItems	X			
EstimateLineItem	../EstimateLineItems(id)	X		X	
TransVariable[]	../EstimateLineItems(id)/Variables	X			
TransVariable	../EstimateLineItems(id)/Variables(name)	X			
TransMod[]	../EstimateLineItems(id)/Modifiers	X			
TransMod	../EstimateLineItems(id)/Modifiers(name)	X			
TransPart[]	../EstimateLineItems(id)/Parts	X			

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
TransPart	../EstimateLineItems(id)/Parts(id)	X			
OrderLineItem[]	../OrderLineItems	X			
OrderLineItem	../OrderLineItems(id)	X		X	
TransVariable[]	../OrderLineItems(id)/Variables	X			
TransVariable	../OrderLineItems(id)/Variables(name)	X			
TransMod[]	../OrderLineItems(id)/Modifiers	X			
TransMod	../OrderLineItems(id)/Modifiers(name)	X			
TransPart[]	../OrderLineItems(id)/Parts	X			
TransPart	../OrderLineItems(id)/Parts(id)	X			
ProductionActivities[]	../OrderLineItems(id)/ProductionActivities	X			
ProductionActivity	../OrderLineItems(id)/ProductionActivities(id)	X			
Shipments[]	../Shipments	X	X		
Shipment	../Shipments(id)	X		X	
ShipmentItems[]	../Shipments(id)/Items	X			
ShipmentItem	../Shipments(id)/Items(id)	X			

4.3.2 Pricing Definition REST Endpoints Summary

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
ProductCategory[]	../ProductCategory	X			
ProductCategory	../ProductCategory(id)	X			
ProductDefinitions[]	../ProductCategory(id)/ProductDefinitions	X			
ProductDefinition	../ProductDefinitions(id)	X			
ModifierDefinition[]	../ProductDefinitions(id)/Modifiers	X			
PartDefinition[]	../ProductDefinitions(id)/Parts	X			
ProductDefinition[]	../ProductDefinitions(id)/AssemblyItems	X			
VariableDefinition[]	../ProductDefinitions(id)/Variables	X			
UDF[]	../ProductDefinitions(id)/UDFs	X			
PartDefinition[]	../PartDefinitions	X			

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
PartDefinitions[]	../PartDefinitions/InventoryItems	X			
Unit[]	../PartDefinitions/ValidUnits	X			
ProductDefinition[]	../PartDefinitions/Products	X			
ModifierDefinition[]	../PartDefinitions/Modifiers	X			
UDF[]	../PartDefinitions/UDFs	X			
ModifierCategory[]	../ModifierCategory	X			
ModifierCategory	../ModifierCategory(id)	X			
ModifierDefinition[]	../ModifierCategory(id)/Modifiers	X			
ModifierDefinition	../ModifierCategory(id)/Modifiers(id)	X			
ProductDefinition[]	../ModifierCategory(id)/Modifiers(id)/Products	X			
PartDefinition[]	../ModifierCategory(id)/Modifiers(id)/Parts	X			
SelectionListCategory[]	../SelectionListCategories	X			
SelectionListCategory	../SelectionListCategories(id)	X			
SelectionList[]	../SelectionListCategories(id)/SelectionLists	X			
SelectionLists	../SelectionListCategories(id)/SelectionLists(id)	X			
SelectionList[]	../SelectionLists	X			
SelectionListItem[]	../SelectionLists(id)/Items	X			
SelectionListItem	../SelectionLists(id)/Items(id)	X			
PartDefinition[]	../SelectionLists(id)/Items(id)/Parts	X			
PartDefinition	../SelectionLists(id)/Items(id)/Part(id)	X			
SelectionListItem[]	../SelectionListItems	X			
SelectionListItem	../SelectionListItems(id)	X			
PartDefinition[]	../SelectionListItems(id)/Parts	X			
PartDefinition	../SelectionListItems(id)/Parts(id)	X			
Warehouse[]	../Warehouses	X			
Warehouse	../Warehouses(id)	X			
InventoryItem[]	../Warehouses(id)/InventoryItems	X			
InventoryItem	../Warehouses(id)/InventoryItems(id)	X			

Accessed Resource	REST Call Endpoint	GET	POST	PUT, PATCH	DELETE
Warehouse[]	../Warehouses(id)/Warehouses	X			
Warehouse	../Warehouses(id)/Warehouses(id)	X			
InventoryItem[]	../Warehouses(id)/Warehouses(id)/InventoryItems	X			
InventoryItem	../Warehouses(id)/Warehouses(id)/InventoryItems(id)	X			

4.4 Phase II - Subscription Resources …/wapic/subscription

Subscription resources manage the subscription of clients to topics in the pub-sub system.

4.4.1 Supported REST operations

RESOURCE	SUPPORTED CALLS
Subscription	**GET …/wapic/subscription** Returns an oData queryable of all current subscriptions. **POST …/wapic/subscription** Create new subscriptions. Subscriptions are contained within the content of the post. **POST …/wapic/subscription?client=x&callback=y&topic=z** Create a single new subscription. **PUT …/WAPIC/subscription/id** Refresh a subscription. Subscriptions expire after 24 hours if they have not been refreshed. **DELETE …/WAPIC/subscription/id** Delete a subscription. **DELETE …/WAPIC/subscription?client=x [& topic=z]** Delete subscriptions belonging to client *x*. If *z* is specified, only subscriptions to that topic AND for that client will be deleted.

4.4.2 Subscription Structures

The following structures are used for requests and responses in requests to the Subscription controller:

Request Data Structure for

    POST …/WAPIC/subscription

```
{[
     { "client": "00000000-0000-0000-0000-000000000000", "uri":
       "http://tempuri.org/1", topic="bears" },
     { "client": "00000000-0000-0000-0000-000000000000", "uri":
       "http://tempuri.org/2", topic="chairs" },
     { "client": "00000000-0000-0000-0000-000000000000", "uri":
       "http://tempuri.com/1", topic="hairs" }
]}
```

Request Data Structure for

    PUT …/WAPIC/subscription

```
{[
     { "id": "10000000-0000-0000-0000-000000000000" },
     { "id": "20000000-0000-0000-0000-000000000000" },
     { "id": "30000000-0000-0000-0000-000000000000" }
]}
```

4.5 Phase II - PubSubMessage Resources …/WAPIC/PubSubMessage

4.5.1 Overview

- CHAPI relay?
- Socket based alternative for clients?
- What is exposed?

4.5.2 Supported REST operations

RESOURCE	SUPPORTED CALLS
PubSubMessage	**GET …/WAPIC/pubsubmessage** Returns an oData queryable of all PubSubMessages **POST …/WAPIC/pubsubmessage [?client=x]** Create and publish a new PubSubMessage. Messages to publish are contained within the content of the post. If client is supplied, all messages being published will be associated with the client and not sent back to the client if that client is subscribed to the topic of the message. **POST …/WAPIC/pubsubmessage [?client=x]** Create and publish a single PubSubMessage. If client is supplied, the message will not be rebroadcast to that client.

4.5.3 PubSubMessage Structures

POST Structure for .../WAPIC/pubsubmessage

```
{[
    { "topic" : "mytopic",
      "content" : "mymessage hello!" },
    { "topic" : "myothertopic",
      "content" : { "somekey" : "somevalue" } }
]}
```

GET Structure (also used for publishing to a subscribed URI)

```
{[
    { "topic" : "mytopic",
      "content" : "mymessage hello!",
      "messageid" : "10000000-0000-0000-0000-000000000000",
      "publisheddatetime" : "4\/24\/2012 15:37:54"
    },
    { "topic" : "myothertopic",
      "content" : { "somekey" : "somevalue" }
      "messageid" : "20000000-0000-0000-0000-000000000000",
      "publisheddatetime" : "4\/24\/2012 15:38:44"
    }
]}
```

5. WAPIC Data Object Models

The data objects models represent each of the data objects that can be retrieved in WAPIC. **Root models** are considered the "top level" and are referenced in other models, but not included. **Non-root models**, also called **child models**, are contained within another model and may be retrieved in full and saved within their root model.

5.1 Sample Resources Calls

Each resource returns the object model specified in this document. The general structure of the call is as follows:

RESOURCE	SUPPORTED CALLS
Company	**GET ../wapic/Companies [? oData filter, expand options]** Retrieves and reserves company records based on the specified filter. If not filter is applied, all records (up to the maximum) are returned. A maximum of 100 records per query are returned. Expand options can be used to include the full copy of child item data instead of a resource link to them (the default). **POST ../wapic/Companies** Adds a new company record to the database. Contact, address, and phone data should be imbedded in the company information, but can be saved separately. The IDs are assigned and any new record processing is completed by WAPIC. The saved record, with IDs, is returned in the response. **GET ../wapic/Companies({id}) [? oData select, expand options]** Retrieves and reserves the company record for the ID specified. If the ID is not found, a 4xx result is returned. An optional oData select statement can be used to only return specific fields. **PUT|PATCH ../wapic/Companies({id})** Updates an existing company record in the database. When PUT is used, the existing company record is re-initialized before being updated with the contents of the request message. All fields in the record should be resubmitted. When PATCH is used, the existing record is updated with the contents of the request message. Only the changes fields need to be submitted. For both PUT and PATCH, the updated (full) record is returned in the response.

5.2 Address

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Address.ID (int)	IDcCT format
DateCreated	DateTime	R/O	Address.DateCreated	
DateModified	DateTime	R/O	Address.ModifiedDate	
eTag	int	R/O	Address.SeqID (int)	
IsActive	boolean	R/W	Address.IsActive	
Name	string	R/W	Address.	
AddressText	string	R/O	Address.FormattedText	Multi-Line
Street1	string	R/W	Address.StreetAddress1	
Street2	string	R/W	Address.StreetAddress2	
City	string	R/W	Address.City	
State	string	R/W	Address.State	
County	string	R/W	Address.County	
PostalCode	string	R/W	Address.PostalCode	
Country	string	R/W	Address.Country	
IsValidated	boolean	R/W	Address.IsValidated	
ValidatedAddressText	string	R/O	Address.ValidatedAddress	Multi-Line

5.3 Calendar

5.3.1 Calendar Web Object

The calendar object represents the master calendars enabled in Control. It is a combination of Employees with calendars and equipment calendar instances.

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Calendar.ID (int)	IDcCT format
DateCreated	DateTime	R/O		
DateModified	DateTime	R/O	Calendar.ModifiedDate	
eTag	int	R/O	Calendar.SeqID (int)	
IsActive	boolean	R/O	Calendar.IsActive	
Name	String	R/O	Calendar.Name	
ParentID	Int	R/O	Calendar.ParentID	IDcCT format
ParentText	String	R/O	Calendar.ParentText	

API Property Name	Data Type	Access	Control Table.Field	Notes

5.3.2 Calendar View

The following view (read/only) was used to create the Calendar from the Employee and Equiment Parts tables:

```sql
create view Calendar as

select PE.ID, PE.ClassTypeID,
   PE.ModifiedDate as ModifiedDate,
   PE.SeqID as SeqID,
   CAST(1 as Bit) as IsActive,
   PE.ElementName as Name,
   PE.ParentID, PE.ParentClassTypeID,
   Part.ItemName as ParentText
from PricingElement PE
join Part on Part.ID = PE.ParentID
where PE.ClassTypeID = 12075
  and PE.IsActive = 1
  and Part.IsActive = 1

union

select ID, ClassTypeID,
   ModifiedDate as ModifiedDate,
   SeqID as SeqID,
   CAST(1 as Bit) as IsActive,
   (FirstName + ' ' + LastName) as Name,
   ID as ParentID, ClassTypeID as ParentClassTypeID,
   (FirstName + ' ' + LastName) as ParentText
from Employee
where ShowOnActivityManager = 1 and IsActive = 1
```

This view is read-only and may not be modified except by modifying the underlying Employee or Part data.

5.4 Calendar Activity

Calendar Activities consist of the following types of activities in Control:
- Calls
- Appointments
- Meetings
- ToDos (Tasks)
- Notes

In the Control database, Calendar Activities are mixed with many types of activities, and represented with 2 database tables (Journal and ContactActivity). The Web API presents this as a separate collection of objects and uses updatable views to read and write the relevant data.

5.4.1 Calendar Activity Object

The calendar activity object represents the contact activities in Control..

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	CA.ID	IDcCT format
DateCreated	DateTime	R/O	CA.CreatedDateTime	
DateModified	DateTime	R/O	CA.ModifiedDate	
eTag	int	R/O	CA.SeqID	
IsActive	boolean	R/W	CA.IsActive	
Description	string	R/W	CA.Description	
Employee	string	R/W	Link via CA.EmployeeID	
IsPrivate	boolean	R/W	CA.PrivateEvent	
Notes	string	R/W	CA.Notes	
StartDate	DateTime	R/W	CA.StartDate	
EndDate	DateTime	R/W	CA.EndDate	
TotalTime	Time	R/W	CA.TotalTime	
IsAllDay	boolean	R/W	CA.AllDayEvent	
IsDateOnly	boolean	R/O	CA.IsTimeless	
ReminderDate Also sets CA.ReminderPrompt	DateTime	R/W	CA.ReminderDateTime	Null is no reminder.
Location	string	R/W	CA.Location	
Result	string	R/W	CA.Result	
?	string	W1	CA.ContactCallTypeText	Also sets CA.ContactCallType
CompletedBy	Employee	R/W	Linked by CA.CompletedByID	
CompletedDate	DateTime	R/W	CA.CompletedDateTime	
IsAutoRolledOver	boolean	R/W	CA.AutoRollOver	
AccountID	string	R/W	CA.AccountID	IDcCT format
ContactID	string	R/W	CA.ContactID	IDcCT format
TransactionID	string	R/W	TransactionID	IDcCT format
OrderNumber	int	R/W	link via CA.TransactionID where TransactionType in (1,6)	
EstimateNumber	int	R/W	link via CA.TransactionID where TransactionType = 2	
Calendars	Calendars[]		linked via CA.HasCalendarLinks	
CreatedBy	Employee	W1	linked by CA.CreatedByID	
			CA.JournalActivityText	Also update JournalActivityType
			CA.ActivityTypeText	also updates CA.ActivityType

5.4.2 Calendar Activity View

The following view (read/only) was used to create the Calendar from the Journal and ContactActivity tables:

```
CREATE VIEW [dbo].[CalendarActivity] AS

SELECT  J.ID, J.ClassTypeID, J.ModifiedByUser, J.ModifiedByComputer,
       J.ModifiedDate, J.SeqID, J.IsSystem, J.IsActive, J.EmployeeID,
       J.Description, J.Notes, J.StartDateTime AS StartDate, J.EndDateTime AS
       EndDate, J.TotalTime, J.ScheduledDateTime, J.CompletedByID,
       J.CompletedDateTime, J.AccountID, J.ContactID, J.TransactionID,
       J.ReminderDateTime, J.ReminderPrompt, J.ActivityType,
       J.ActivityTypeText, J.IsBillable, J.BillableDateTime, J.UseActualTime,
       J.BillingNotes, J.BillingType, J.TotalBilledTime,
       J.RecurringActivityID, J.DivisionID, J.HasCalendarLinks,

       CA.AllDayEvent, CA.AutoRollOver, CA.AutoRun, CA.Color,
       CA.ContactCallType,  CA.ContactCallTypeText, CA.CreatedByID,
       CA.CreatedDateTime, CA.IsFromRecurring, CA.IsTimeless, CA.Location,
       CA.OriginalScheduledDateTime, CA.Priority, CA.PriorityText,
       CA.PrivateEvent, CA.RESULT, CA.ValidStatus

FROM dbo.Journal AS J
JOIN dbo.ContactActivity AS CA ON J.ID = CA.ID

WHERE (J.ClassTypeID = 21100)
```

This view supports read-write updates and deletes and modifies the underlying data. To achieve this, the following INSTEAD OF triggers must be added to the view.

```
-- =============================================
-- Author:  Cyrious Software
-- Description: CalendarActivity View Delete
-- =============================================
CREATE TRIGGER [dbo].[CalendarActivity_Delete] ON [dbo].[CalendarActivity]
INSTEAD OF DELETE
AS
BEGIN
  DELETE FROM Journal
  WHERE ID IN (SELECT ID FROM DELETED);

  DELETE FROM ContactActivity
  WHERE ID IN (SELECT ID FROM DELETED);
END
```

```sql
-- =============================================
-- Author:Cyrious Software
-- Description:CalendarActivity View Insert
-- =============================================
CREATE TRIGGER [dbo].[CalendarActivity_Insert] ON [dbo].[CalendarActivity]
INSTEAD OF INSERT
AS
BEGIN
  -- Add to the Journal
  INSERT INTO Journal
  (
    ID, StoreID, ClassTypeID, ModifiedByUser, ModifiedByComputer,
      ModifiedDate, SeqID, IsSystem, IsActive, EmployeeID,
      JournalActivityType, JournalActivityText, Description, Notes,
      StartDateTime, EndDateTime, TotalTime, ScheduledDateTime,
      CompletedByID, CompletedDateTime, IsSummary, IsDetail, SummaryID,
      SummaryClassTypeID, SummaryAmount, DetailAmount, StartGLGroupID,
      EndGLGroupID, AccountID, AccountClassTypeID, ContactID,
      ContactClassTypeID, TransactionID, TransactionClassTypeID, IsVoided,
      VoidedDateTime, VoidedEntryID, VoidedEntryClassTypeID, VoidedReason,
      QueryStartDateTime, QueryEndDateTime, ReminderDateTime,
      ReminderPrompt, PartID, ActivityType, ActivityTypeText, IsBillable,
      BillableDateTime, UseActualTime, BillingNotes, BillingType,
      TotalBilledTime, RecurringActivityID, LinkID, LinkStoreID,
      LinkClassTypeID, SpecialCode, DivisionID, HasCalendarLinks,
      TipRecipientID, PartClassTypeID, RecurringClassTypeID, StationID,
      StationClassTypeID, CurrentState, StageID, StageClassTypeID
  )
  SELECT
    INSERTED.ID, -1, INSERTED.ClassTypeID, INSERTED.ModifiedByUser,
      INSERTED.ModifiedByComputer, INSERTED.ModifiedDate, INSERTED.SeqID, 0,
      INSERTED.IsActive, INSERTED.EmployeeID, 10, 'Contact Activity',
      INSERTED.Description, INSERTED.Notes, INSERTED.StartDate,
      INSERTED.EndDate, INSERTED.TotalTime, INSERTED.ScheduledDateTime,
      INSERTED.CompletedByID, INSERTED.CompletedDateTime, 1, 1, NULL, NULL,
      0, 0, NULL, NULL, INSERTED.AccountID, 2000, INSERTED.ContactID, 3000,
      INSERTED.TransactionID, 10000, 1, NULL, NULL, NULL, NULL,
      INSERTED.StartDate, INSERTED.EndDate, INSERTED.ReminderDateTime,
      INSERTED.ReminderPrompt, NULL, INSERTED.ActivityType,
      INSERTED.ActivityTypeText, INSERTED.IsBillable,
      INSERTED.BillableDateTime, INSERTED.UseActualTime,
      INSERTED.BillingNotes, INSERTED.BillingType, INSERTED.TotalBilledTime,
      INSERTED.RecurringActivityID, NULL, NULL, NULL, NULL,
      INSERTED.DivisionID, INSERTED.HasCalendarLinks, NULL, NULL, NULL,
      NULL, NULL, NULL, NULL, NULL
  FROM INSERTED
;
  --
  -- Add to the ContactActivity Table
  --
  INSERT INTO ContactActivity
  (
```

```sql
        ID, StoreID, ClassTypeID, ModifiedByUser, ModifiedByComputer,
          ModifiedDate, SeqID, IsSystem, IsActive, AllDayEvent, AutoRollOver,
          AutoRun, Color, ContactActivityType, ContactActivityTypeText,
          ContactCallType, ContactCallTypeText, CreatedByID, CreatedDateTime,
          ImageID, IsFromRecurring, IsTimeless, Location,
          OriginalScheduledDateTime, Priority, PriorityText, PrivateEvent,
          RecurringActivityID, RunWithoutPrompting, ScheduledEndDateTime,
          ScheduledStartDateTime, RESULT, ValidStatus, TransPartID,
          TransPartStoreID, TransPartClassTypeID, PartInstanceID, IsLocked,
          LocationID, LocationStoreID, LocationClassTypeID
      )
    SELECT
        INSERTED.ID, -1, INSERTED.ClassTypeID, INSERTED.ModifiedByUser,
          INSERTED.ModifiedByComputer, INSERTED.ModifiedDate, INSERTED.SeqID, 0,
          INSERTED.IsActive, INSERTED.AllDayEvent, INSERTED.AutoRollOver,
          INSERTED.AutoRun, INSERTED.Color, INSERTED.ActivityType,
          INSERTED.ActivityTypeText, INSERTED.ContactCallType,
          INSERTED.ContactCallTypeText, INSERTED.CreatedByID,
          INSERTED.CreatedDateTime, NULL, INSERTED.IsFromRecurring,
          INSERTED.IsTimeless, INSERTED.Location,
          INSERTED.OriginalScheduledDateTime, INSERTED.Priority,
          INSERTED.PriorityText, INSERTED.PrivateEvent,
          INSERTED.RecurringActivityID, 0, INSERTED.EndDate, INSERTED.StartDate,
          INSERTED.RESULT, INSERTED.ValidStatus, NULL, NULL, NULL, NULL, 0,
          NULL, NULL, NULL
    FROM INSERTED
END



-- =============================================
-- Author:        Cyrious Software
-- Description:   CalendarActivity View Update
-- =============================================
CREATE TRIGGER [dbo].[CalendarActivity_Update] ON [dbo].[CalendarActivity]
INSTEAD OF UPDATE
AS
BEGIN
  -- Update the Journal
  UPDATE J
  SET
    J.ID                    = INSERTED.ID                      ,
--  J.StoreID               = -1                               ,
    J.ClassTypeID           = INSERTED.ClassTypeID             ,
    J.ModifiedByUser        = INSERTED.ModifiedByUser          ,
    J.ModifiedByComputer    = INSERTED.ModifiedByComputer      ,
    J.ModifiedDate          = INSERTED.ModifiedDate            ,
    J.SeqID                 = INSERTED.SeqID                   ,
--  J.IsSystem              = 0                                ,
    J.IsActive              = INSERTED.IsActive                ,
    J.EmployeeID            = INSERTED.EmployeeID              ,
    J.JournalActivityType   = 10                               ,
    J.JournalActivityText   = 'Contact Activity'               ,
```

```
        J.Description              = INSERTED.Description        ,
        J.Notes                    = INSERTED.Notes              ,
        J.StartDateTime            = INSERTED.StartDate          ,
        J.EndDateTime              = INSERTED.EndDate            ,
        J.TotalTime                = INSERTED.TotalTime           ,
        J.ScheduledDateTime        = INSERTED.ScheduledDateTime  ,
        J.CompletedByID            = INSERTED.CompletedByID      ,
        J.CompletedDateTime        = INSERTED.CompletedDateTime  ,
--      J.IsSummary                = 1                           ,
--      J.IsDetail                 = 1                           ,
--      J.SummaryID                = NULL                        ,
--      J.SummaryClassTypeID       = NULL                        ,
--      J.SummaryAmount            = 0                           ,
--      J.DetailAmount             = 0                           ,
--      J.StartGLGroupID           = NULL                        ,
--      J.EndGLGroupID             = NULL                        ,
        J.AccountID                = INSERTED.AccountID          ,
        J.AccountClassTypeID       = 2000                        ,
        J.ContactID                = INSERTED.ContactID          ,
        J.ContactClassTypeID       = 3000                        ,
        J.TransactionID            = INSERTED.TransactionID      ,
        J.TransactionClassTypeID   = 10000                       ,
--      J.IsVoided                 = 1                           ,
--      J.VoidedDateTime           = NULL                        ,
--      J.VoidedEntryID            = NULL                        ,
--      J.VoidedEntryClassTypeID   = NULL                        ,
--      J.VoidedReason             = NULL                        ,
        J.QueryStartDateTime       = INSERTED.StartDate          ,
        J.QueryEndDateTime         = INSERTED.EndDate            ,
        J.ReminderDateTime         = INSERTED.ReminderDateTime   ,
        J.ReminderPrompt           = INSERTED.ReminderPrompt     ,
--      J.PartID                   = NULL                        ,
        J.ActivityType             = INSERTED.ActivityType       ,
        J.ActivityTypeText         = INSERTED.ActivityTypeText   ,
        J.IsBillable               = INSERTED.IsBillable         ,
        J.BillableDateTime         = INSERTED.BillableDateTime   ,
        J.UseActualTime            = INSERTED.UseActualTime      ,
        J.BillingNotes             = INSERTED.BillingNotes       ,
        J.BillingType              = INSERTED.BillingType        ,
        J.TotalBilledTime          = INSERTED.TotalBilledTime    ,
        J.RecurringActivityID      = INSERTED.RecurringActivityID ,
--      J.LinkID                   = NULL                        ,
--      J.LinkStoreID              = NULL                        ,
--      J.LinkClassTypeID          = NULL                        ,
--      J.SpecialCode              = NULL                        ,
        J.DivisionID               = INSERTED.DivisionID         ,
        J.HasCalendarLinks         = INSERTED.HasCalendarLinks
--      J.TipRecipientID           = NULL                        ,
--      J.PartClassTypeID          = NULL                        ,
--      J.RecurringClassTypeID     = NULL                        ,
--      J.StationID                = NULL                        ,
--      J.StationClassTypeID       = NULL                        ,
```

```sql
--   J.CurrentState              = NULL                                   ,
--   J.StageID                   = NULL                                   ,
--   J.StageClassTypeID          = NULL
   FROM Journal J
     JOIN INSERTED ON J.ID = INSERTED.ID
;
   --
   -- Update the ContactActivity Table
   --
   UPDATE CA
   SET
     CA.ID                        = INSERTED.ID                           ,
--   CA.StoreID                   = -1                                    ,
     CA.ClassTypeID               = INSERTED.ClassTypeID                  ,
     CA.ModifiedByUser            = INSERTED.ModifiedByUser               ,
     CA.ModifiedByComputer        = INSERTED.ModifiedByComputer           ,
     CA.ModifiedDate              = INSERTED.ModifiedDate                 ,
     CA.SeqID                     = INSERTED.SeqID                        ,
--   CA.IsSystem                  = 0                                     ,
     CA.IsActive                  = INSERTED.IsActive                     ,
     CA.AllDayEvent               = INSERTED.AllDayEvent                  ,
     CA.AutoRollOver              = INSERTED.AutoRollOver                 ,
     CA.AutoRun                   = INSERTED.AutoRun                      ,
     CA.Color                     = INSERTED.Color                        ,
     CA.ContactActivityType       = INSERTED.ActivityType                ,
     CA.ContactActivityTypeText   = INSERTED.ActivityTypeText            ,
     CA.ContactCallType           = INSERTED.ContactCallType             ,
     CA.ContactCallTypeText       = INSERTED.ContactCallTypeText         ,
     CA.CreatedByID               = INSERTED.CreatedByID                 ,
     CA.CreatedDateTime           = INSERTED.CreatedDateTime             ,
--   CA.ImageID                   = NULL                                  ,
     CA.IsFromRecurring           = INSERTED.IsFromRecurring             ,
     CA.IsTimeless                = INSERTED.IsTimeless                  ,
     CA.Location                  = INSERTED.Location                    ,
--   CA.OriginalScheduledDateTime = INSERTED.OriginalScheduledDateTime,
     CA.Priority                  = INSERTED.Priority                    ,
     CA.PriorityText              = INSERTED.PriorityText                ,
     CA.PrivateEvent              = INSERTED.PrivateEvent                ,
     CA.RecurringActivityID       = INSERTED.RecurringActivityID         ,
--   CA.RunWithoutPrompting       = 0                                     ,
     CA.ScheduledEndDateTime      = INSERTED.EndDate                     ,
     CA.ScheduledStartDateTime    = INSERTED.StartDate                   ,
     CA.RESULT                    = INSERTED.RESULT                      ,
     CA.ValidStatus               = INSERTED.ValidStatus
--   CA.TransPartID               = NULL                                  ,
--   CA.TransPartStoreID          = NULL                                  ,
--   CA.TransPartClassTypeID      = NULL                                  ,
--   CA.PartInstanceID            = NULL                                  ,
--   CA.IsLocked                  = 0                                     ,
--   CA.LocationID                = NULL                                  ,
--   CA.LocationStoreID           = NULL                                  ,
--   CA.LocationClassTypeID       = NULL
```

```
    FROM ContactActivity CA
       JOIN INSERTED ON CA.ID = INSERTED.ID
    END
```

5.5 Company

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Account.ID (int)	IDcCT format
DateCreated	DateTime	R/O	Account.DateCreated	
DateModified	DateTime	R/O	Account.ModifiedDate	
eTag	int	R/O	Account.SeqID (int)	
IsActive	boolean	R/W	Account.IsActive	
Name	string	R/W	Account.CompanyName	
CalendarActivities	CalendarActivity[]			
Addresses	CompanyAddress[]	R/O		
AddressLinks	CompanyAddressLink[]	R/W		
Contacts	Contact[]			
Division	string	R/W	link via Account.DivisionID	
Flags	string	R/W	Account.Flags	
Industry	string	R/W	link via Account.IndustryID	
IsCustomer	boolean	R/W	Account.IsClient	
IsPrivate	boolean	R/W	Account.IsPersonal	
IsVendor	boolean	R/W	Account.IsVendor	
Notes	string	R/W	Account.Notes	
Origin	string	R/W	linked via Account.OriginID	
Parent	Company	R/W		
Phones	Phone[]			
Region	string	R/W	linked via Account.___	
Salespersons	Employee[]		linked via SalespersonIDx	
UDFs	UDF[]		from AccountUserField	
URL	string	R/W	Account.___	
CustomerCreditAccountID	string	R/W	Account.CreditNumber	
CustomerCreditBalance	float	R/O	Account.CreditBalance	
CustomerCreditLimit	float	R/W	Account.CreditLimit	
CustomerDateTaxExpiration	Date	R/W	Account.TaxNumberExpDate	
CustomerDiscountLevel	float	R/W	Account.DiscountLevel	
CustomerHasCreditAccount	boolean	R/W	Account.HasCreditAccount	
CustomerIsPORequired	boolean	R/W	Account.PONumberRequired	
CustomerIsTaxExempt	boolean	R/W	Account.TaxExempt	

API Property Name	Data Type	Access	Control Table.Field	Notes
CustomerNumber	int	R/W	Account.AccountNumber	
CustomerPaymentTerms	string	R/W		
CustomerPONumberDefault	string	R/W	Account.PONumber	
CustomerPricingLevel	float	R/W	Account.PricingLevel	
CustomerPricingPlan	string	R/W	linked via Account.PricingPlanID	
CustomerTaxClass	string	R/W	linked via Account.TaxClassID	
CustomerTaxNumber	string	R/W	Account.TaxNumber	
VendorCreditBalance	float	R/O	Account.VendorCreditBalance	
VendorIs1099	boolean	R/W	Account.Is1099Vendor	
VendorNumber	int	R/W	Account.MyAccountNumber	

5.5.1 Additional notes for company data model:

When posting a new company, the AddressLinks collection may be filled out in order to create new
CompanyAddressLink records and corresponding Address records.

```
POST
{"Name":"NewCompanyName",
 "AddressLinks":[{
     "AddressName":"Shipping",
     "Address":{
         "Street1":"123 ABC St."}
     },{
     "AddressName":"Billing",
     "Address":{
         "Street1":"123 ABC St."}
     }]
}
```

The logic for determining the taxability status of a company is as follows:

If StateTaxExemptionLinks are present, **ignore** CustomerDateTaxExpiration and CustomerTaxNumber
and use the Links.

If CustomerTaxNumber is NULL and StateTaxExemptionLinks are NULL, Customer is Taxable in All
States.

If StateTaxExemptionLinks Is NULL and CustomerTaxNumber/CustomerDateTaxExpiration is present,
then those values apply to all states.

5.6 Contact

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	AccountContact.ID (int)	IDcCT format
DateCreated	DateTime	R/O	AccountContact.DateCreated	
DateModified	DateTime	R/O	AccountContact.ModifiedDate	
eTag	int	R/O	AccountContact.SeqID (int)	
IsActive	boolean	R/W	AccountContact.IsActive	
CompanyID	string	R/O	AccountContact.AccountID + "c2000"	IDcCT format
FirstName	string	R/W	AccountContact.FirstName	
MiddleName	string	R/W	AccountContact.MiddleName	
LastName	string	R/W	AccountContact.LastName	
DisplayName	string	R/O	AccountContact.FirstName+ " " +AccountContact.LastName	
ImageID	string	R/?	AccountContact.ImageID	IDcCT format
Position	string	R/W	AccountContact.Position	
Birthday	datetime	R/W	AccountContact.BirthDateMonth + AccountContact.BirthDateDay	
Gender	string	R/W	AccountContact.GenderType (int)	
IsPrimary	boolean	R/W	AccountContact.IsPrimaryContact	
IsBilling	boolean	R/W	AccountContact.IsAccountingContact	
CarrierDefault	string	R/W	AccountContact.ShippingCarrier	
CarrierAccountNumber	string	R/W	AccountContact.ShippingAccountNumber	
CarrierRefPostalCode	string	R/W	AccountContact.ShippingAccountPostalCode	
Division	string	R/O	link via Account.DivisionID	
Notes	string	R/W	AccountContact.Notes	
Addresses	Address[]			
Phones	Phone[]			
Email	String	R/W	AccountContact.EmailAddress	
CalendarActivities	CalendarActivity[]			
UDFs	UDF[]		from AccountContactUserField	

5.6.1 Additional notes for contact data model:

When posting a new company, the AddressLinks collection may be filled out in order to create new ContactAddressLink records and corresponding Address records.

5.7 Products and Parts

5.7.1 Division

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Division.ID (int)	IDcCT format
DateCreated	DateTime	R/O	Division.DateCreated	
DateModified	DateTime	R/O	Division.ModifiedDate	
eTag	int	R/O	Division.SeqID (int)	
IsActive	boolean	R/W	Division.IsActive	
Name	string	R/W	Division.GroupName	
Addresses	Address[]		Create list of 1 using only Division. AddressID	
Phones	Phone[]		Build list from Division.MainPhoneNumberID, Division.MainFaxNumberID and/or Division.PrimaryNumber, Division.SecondaryNumber, Division.ThirdNumber	
Email	string	R/W	Division.EmailAddress	
ImageID	Integer	R/?	Division.	Field not in database yet

5.7.2 Employee

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Employee.ID (int)	IDcCT format
DateCreated	DateTime	R/O	Employee.DateCreated	
DateModified	DateTime	R/O	Employee.ModifiedDate	
eTag	int	R/O	Employee.SeqID (int)	
IsActive	boolean	R/W	Employee.IsActive	
FirstName	string	R/W	Employee.FirstName	
MiddleName	string	R/W	Employee.MiddleName	
LastName	string	R/W	Employee.LastName	

API Property Name	Data Type	Access	Control Table.Field	Notes
DisplayName	string	R/O	Employee.FirstName+ " " +Employee.LastName	
ImageID	string	R/?	Employee.ImageID	IDcCT format
Greeting	string	R/W	Employee.Title	
GroupID	String	R/O	Employee.GroupID	
Department	string	R/W	Employee.Department	
Position	string	R/W	Employee.Position	
IsOnTimeclock	boolean	R/W	Employee.ShowOnTimeclock	
IsSalesperson	boolean	R/W	Employee.IsSalesperson	
PricingLevel	Float	R/O	Employee.PricingLevel	
Division	string	R/O	link via EmployeeGroup.DivisionID	Optional. Omit if Difficult
Notes	string	R/W	Employee.Notes	
Address	Address[]		Create list of 1 using only Employee. AddressID	
Email	string	R/W	Employee.EmailAddress	
Phones	Phone[]		Build list from Employee.MainPhoneNumberID, Employee.MainFaxNumberID and/or Employee.PrimaryNumber, Employee.SecondaryNumber, Employee.ThirdNumber	Get clarification from GS
CalendarActivities	CalendarActivity[]			

5.7.3 Employee Group

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	EmployeeGroup.ID (int)	IDcCT format
DateCreated	DateTime	R/O	EmployeeGroup.DateCreated	
DateModified	DateTime	R/O	EmployeeGroup.ModifiedDate	
eTag	int	R/O	EmployeeGroup.SeqID (int)	
IsActive	boolean	R/W	EmployeeGroup.IsActive	
Name	string	R/W	EmployeeGroup.GroupName	
DisplayName	string	R/W	EmployeeGroup.PrinatAsName	
GroupID	string	R/O	EmployeeGroup.ParentID	
Division	string	R/O	link via IsDivision or EmployeeGroup.DivisionID	Optional. Omit if Difficult

API Property Name	Data Type	Access	Control Table.Field	Notes
Notes	string	R/W	EmployeeGroup.Notes	
Employees	Employee[]			

5.7.4 Industry

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	MarketingListItem.ID (int)	IDcCT format
DateCreated	DateTime	R/O	MarketingListItem.DateCreated	
DateModified	DateTime	R/O	MarketingListItem.ModifiedDate	
eTag	int	R/O	MarketingListItem.SeqID (int)	
IsActive	boolean	R/W	MarketingListItem.IsActive	
Name	string	R/W	MarketingListItem.GroupName	
Description	string	R/W	MarketingListItem.Description	
GroupOnly	boolean	R/W	= not MarketingListItem.CanSelect	
GroupID	Int	R/W	MarketingListItem.ParentID	
Code1	string	R/W	MarketingListItem.Code1	SIC Classification
Code2	string	R/W	MarketingListItem.Code2	NAICS Classification

5.7.5 Origin, Region

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	MarketingListItem.ID (int)	IDcCT format
DateCreated	DateTime	R/O	MarketingListItem.DateCreated	
DateModified	DateTime	R/O	MarketingListItem.ModifiedDate	
eTag	int	R/O	MarketingListItem.SeqID (int)	
IsActive	boolean	R/W	MarketingListItem.IsActive	
Name	string	R/W	MarketingListItem.GroupName	
Description	string	R/W	MarketingListItem.Description	
GroupOnly	boolean	R/W	= not MarketingListItem.CanSelect	
GroupID	Int	R/W	MarketingListItem.ParentID	

5.7.6 Phone

Phones are not accessible by ID, but only as an element within another object.

API Property Name	Data Type	Access	Control Table.Field	Notes
Name	string	R/W	PhoneNumber.PhoneNumberTypeText	Parse to fill in PhoneNumberTypeID
Number	string	R/W	PhoneNumber.FormattedText	API parses when changed and writes CountryCode, AreaCode, PhoneNumber, and Extention
Index	Integer	R/W	PhoneNumber.Index	Index for use

5.7.7 CompanyAddressLink

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	CompanyAddressLink.ID (int)	IDcCT format
DateModified	DateTime	R/O	CompanyAddressLink.ModifiedDate	
eTag	int	R/O	CompanyAddressLink.SeqID (int)	
IsActive	boolean	R/W	CompanyAddressLink.IsActive	
IsMaster	string	R/W	CompanyAddressLink.IsMaster	
ParentId	string	R/W	CompanyAddressLink.ParentID	IDcCT format
AddressTypeID	string	R/W	CompanyAddressLink.AddressTypeID	
AddressId	string	R/O	CompanyAddressLink.AddressID	IDcCT format
AddressName	string	R/?	CompanyAddressLink.AddressName	
IsOneTimeCompany	string	R/W	CompanyAddressLink.IsOneTimeCompany	
CompanyName	datetime	R/W	CompanyAddressLink.CompanyName	
ContactName	string	R/W	CompanyAddressLink.ContactName	
Address	Address	R/W	Linked via AddressId	

5.7.8 ContactAddressLink

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	ContactAddressLink.ID (int)	IDcCT format

API Property Name	Data Type	Access	Control Table.Field	Notes
DateModified	DateTime	R/O	ContactAddressLink.ModifiedDate	
eTag	int	R/O	ContactAddressLink.SeqID (int)	
IsActive	boolean	R/W	ContactAddressLink.IsActive	
IsMaster	string	R/W	ContactAddressLink.IsMaster	
ParentId	string	R/W	ContactAddressLink.ParentID	IDcCT format
AddressTypeID	string	R/W	ContactAddressLink.AddressTypeID	
AddressId	string	R/O	ContactAddressLink.AddressID	IDcCT format
AddressName	string	R/?	ContactAddressLink.AddressName	
IsOneTimeCompany	string	R/W	ContactAddressLink.IsOneTimeCompany	
CompanyName	datetime	R/W	ContactAddressLink.CompanyName	
ContactName	string	R/W	ContactAddressLink.ContactName	
Address	Address	R/W	Linked via AddressId	

5.7.9 EmployeeAddressLink

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	EmployeeAddressLink.ID (int)	IDcCT format
DateModified	DateTime	R/O	EmployeeAddressLink.ModifiedDate	
eTag	int	R/O	EmployeeAddressLink.SeqID (int)	
IsActive	boolean	R/W	EmployeeAddressLink.IsActive	
IsMaster	string	R/W	EmployeeAddressLink.IsMaster	
ParentId	string	R/W	EmployeeAddressLink.ParentID	IDcCT format
AddressTypeID	string	R/W	EmployeeAddressLink.AddressTypeID	
AddressId	string	R/O	EmployeeAddressLink.AddressID	IDcCT format
AddressName	string	R/?	EmployeeAddressLink.AddressName	
IsOneTimeCompany	string	R/W	EmployeeAddressLink.IsOneTimeCompany	
CompanyName	datetime	R/W	EmployeeAddressLink.CompanyName	
ContactName	string	R/W	EmployeeAddressLink.ContactName	
Address	Address	R/W	Linked via AddressId	

5.7.10 CompanyAddress

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	CompanyAddress.ID (int)	IDcCT format
DateModified	DateTime	R/O	CompanyAddress.ModifiedDate	
eTag	int	R/O	CompanyAddress.SeqID (int)	
IsActive	boolean	R/O	CompanyAddress.IsActive	
Name	string	R/O	CompanyAddress.Name	
AddressText	string	R/O	CompanyAddress.AddressText	
Street1	string	R/O	CompanyAddress.Street1	
Street2	string	R/O	CompanyAddress.Street2	
City	string	R/O	CompanyAddress.City	
State	string	R/O	CompanyAddress.State	
County	string	R/O	CompanyAddress.County	
PostalCode	datetime	R/O	CompanyAddress.PostalCode	
Country	string	R/O	CompanyAddress.Country	
IsValidated	boolean	R/O	CompanyAddress.IsValidated	
ValidatedAddress	string	R/O	CompanyAddress.ValidatedAddress	
IsMaster	boolean	R/O	CompanyAddress.IsMaster	
ParentId	string	R/O	CompanyAddress.ParentId	IDcCT format
AddressTypeId	int	R/O	CompanyAddress.AddressTypeId	
IsOneTimeCompany	boolean	R/O	CompanyAddress.IsOneTimeCompany	
CompanyName	string	R/O	CompanyAddress.CompanyName	
ContactName	string	R/O	CompanyAddress.ContactName	

5.7.11 ContactAddress

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	ContactAddress.ID (int)	IDcCT format
DateModified	DateTime	R/O	ContactAddress.ModifiedDate	
eTag	int	R/O	ContactAddress.SeqID (int)	
IsActive	boolean	R/O	ContactAddress.IsActive	
Name	string	R/O	ContactAddress.Name	

API Property Name	Data Type	Access	Control Table.Field	Notes
AddressText	string	R/O	ContactAddress.AddressText	
Street1	string	R/O	ContactAddress.Street1	
Street2	string	R/O	ContactAddress.Street2	
City	string	R/O	ContactAddress.City	
State	string	R/O	ContactAddress.State	
County	string	R/O	ContactAddress.County	
PostalCode	datetime	R/O	ContactAddress.PostalCode	
Country	string	R/O	ContactAddress.Country	
IsValidated	boolean	R/O	ContactAddress.IsValidated	
ValidatedAddress	string	R/O	ContactAddress.ValidatedAddress	
IsMaster	boolean	R/O	ContactAddress.IsMaster	
ParentId	string	R/O	ContactAddress.ParentId	IDcCT format
AddressTypeId	int	R/O	ContactAddress.AddressTypeId	
IsOneTimeCompany	boolean	R/O	ContactAddress.IsOneTimeCompany	
CompanyName	string	R/O	ContactAddress.CompanyName	
ContactName	string	R/O	ContactAddress.ContactName	

5.7.12 EmployeeAddress

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	EmployeeAddress.ID (int)	IDcCT format
DateModified	DateTime	R/O	EmployeeAddress.ModifiedDate	
eTag	int	R/O	EmployeeAddress.SeqID (int)	
IsActive	boolean	R/O	EmployeeAddress.IsActive	
Name	string	R/O	EmployeeAddress.Name	
AddressText	string	R/O	EmployeeAddress.AddressText	
Street1	string	R/O	EmployeeAddress.Street1	
Street2	string	R/O	EmployeeAddress.Street2	
City	string	R/O	EmployeeAddress.City	
State	string	R/O	EmployeeAddress.State	
County	string	R/O	EmployeeAddress.County	

API Property Name	Data Type	Access	Control Table.Field	Notes
PostalCode	datetime	R/O	EmployeeAddress.PostalCode	
Country	string	R/O	EmployeeAddress.Country	
IsValidated	boolean	R/O	EmployeeAddress.IsValidated	
ValidatedAddress	string	R/O	EmployeeAddress.ValidatedAddress	
IsMaster	boolean	R/O	EmployeeAddress.IsMaster	
ParentId	string	R/O	EmployeeAddress.ParentId	IDcCT format
AddressTypeId	int	R/O	EmployeeAddress.AddressTypeId	
IsOneTimeCompany	boolean	R/O	EmployeeAddress.IsOneTimeCompany	
CompanyName	string	R/O	EmployeeAddress.CompanyName	
ContactName	string	R/O	EmployeeAddress.ContactName	

5.7.13 ProductCategory

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ProductCategory.ID	IDcCT format
DateCreated	DateTime	R/O	(view) ProductCategory.DateCreated	
DateModified	DateTime	R/O	(view) ProductCategory.ModifiedDate	
eTag	int	R/O	(view) ProductCategory.SeqID	
IsActive	boolean	R/O	(view) ProductCategory.IsActive	
Name	string	R/O	(view) ProductCategory.Name	
Category	string (ProductCategory)	R/O		
Products	ProductDefinitions[]	R/O	via links to ProductDefinitions	

5.7.14 PartCategory

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) PartCategory.ID	IDcCT format
DateCreated	DateTime	R/O	(view)	

API Property Name	Data Type	Access	Control Table.Field	Notes
			PartCategory.DateCreated	
DateModified	DateTime	R/O	(view) PartCategory.ModifiedDate	
eTag	int	R/O	(view) PartCategory.SeqID	
IsActive	boolean	R/O	(view) PartCategory.IsActive	
Name	string	R/O	(view) PartCategory.Name	
Category	string (PartCategory)	R/O		
Parts	PartDefinitions[]	R/O	via links to PartDefinitions	

5.7.15 ModifierCategory

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ModifierCategory.ID	IDcCT format
DateCreated	DateTime	R/O	(view) ModifierCategory.DateCreated	
DateModified	DateTime	R/O	(view) ModifierCategory.ModifiedDate	
eTag	int	R/O	(view) ModifierCategory.SeqID	
IsActive	boolean	R/O	(view) ModifierCategory.IsActive	
Name	string	R/O	(view) ModifierCategory.Name	
Category	string (ModifierCategory)	R/O		
Modifiers	ModifierDefinitions[]	R/O		

5.7.16 ProductDefinition

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ProductDefinition.ID	IDcCT format
DateCreated	DateTime	R/O	(view) ProductDefinition.DateCreated	

API Property Name	Data Type	Access	Control Table.Field	Notes
DateModified	DateTime	R/O	(view) ProductDefinition.ModifiedDate	
eTag	int	R/O	(view) ProductDefinition.SeqID	
IsActive	boolean	R/O	(view) ProductDefinition.IsActive	
Name	string	R/O	(view) ProductDefinition.Name	
Category	string (ProductCategory)	R/O		
PricingPlan	string (PricingPlan)	R/O	linked to ProductPricingPlan	*Key Field along with ID
DisplayName	string	R/O		
TaxabilityCode	string (TaxabilityCode)	R/O		
BarCode	string	R/O		
DefaultStation	string (Station)	R/O	Product.StartingStation	
PricingFamily	string	R/O	linked to PricingFamily	
IsChildByDefault	boolean	R/O		
IsChildOnly	boolean	R/O		
IsExludedFromPromotions	boolean	R/O		
IsExemptFromPriceLock	boolean	R/O		
IsExemptFromPricingLevel	boolean	R/O		
IsAssembly	boolean	R/O		
Description	string	R/O		
Flags	string	R/O		
OnlineHelp	string	R/O		
InternalNotes	string	R/O		
InvoiceNotes	string	R/O		
Image	Image	R/O		
BaseMinimum	currency	R/O		

API Property Name	Data Type	Access	Control Table.Field	Notes
SubTotalMinimum	currency	R/O		
PerPieceMinimum	currency	R/O		
UnitVariable	Unit	R/O		
IncomeAccount	string (GLAccount)	R/O		
GLDepartment	string (Station)	R/O		
Variables	VariableDefinitions[]	R/O		
Modifiers	ModifierDefinitions[]	R/O		
Parts	PartDefinitions[]	R/O		
Tags	StringList	R/O		
UDFs	UDFs[]	R/O		

5.7.17 ModifierDefinition

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ModifierDefinition.ID	ID
DateCreated	DateTime	R/O	(view) ModifierDefinition.DateCreated	
DateModified	DateTime	R/O	(view) ModifierDefinition.ModifiedDate	
eTag	int	R/O	(view) ModifierDefinition.SeqID	
IsActive	boolean	R/O	(view) ModifierDefinition.IsActive	
Name	string	R/O	(view) ModifierDefinition.Name	
Category	string (ModifierCategory)	R/O		
DisplayName	string	R/O		
Question	string	R/O		

API Property Name	Data Type	Access	Control Table.Field	Notes
SortLevel	integer	R/O		
EntryType	{checkbox, number, list}	R/O		
SelectionList	string (SelectionList)	R/O		
Notes	string	R/O	linked to PricingFamily	
Products	ProductDefinitions[]	R/O		
Parts	PartDefinitions[]	R/O		
Tags	StringList	R/O		

5.7.18 PartDefinition

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) PartDefinition.ID	
DateCreated	DateTime	R/O	(view) PartDefinition.DateCreated	
DateModified	DateTime	R/O	(view) PartDefinition.ModifiedDate	
eTag	int	R/O	(view) PartDefinition.SeqID	
IsActive	boolean	R/O	(view) PartDefinition.IsActive	
Name	string	R/O	(view) PartDefinition.Name	
Category	string (PartCategory)	R/O		
PartType	{material, labor, equipment, outsource, other, freight}	R/O		
Description	string	R/O		
SKU	string	R/O		
BarCode	string	R/O		

API Property Name	Data Type	Access	Control Table.Field	Notes
DefaultStation	string (Station)	R/O		
InternalNotes	string	R/O		
OnlineHelp	string	R/O		
IsTrackingCosts	boolean	R/O		
IsTrackingInventory	boolean	R/O		
IsAccuringCosts	boolean	R/O		
InventoryUnitType	{ '', None, Each, Weight, Length, Area, Volume, Time, Rate}	R/O		
InventoryUnit	string	R/O		
IsScrapAdded	boolean	R/O		
ScrapPercentage	float	R/O		
Rounding	string	R/O		
DisplayUnitType	{ '', None, Each, Weight, Length, Area, Volume, Time, Rate}	R/O		
DisplayUnit	string	R/O		
DisplayUnitFactor	string	R/O		Formated as x/y
ValidUnits	Units[]	R/O		
GLExpenseDepartment	string (Station)	R/O		
GLExpenseAccount	string (GLAccount)	R/O		
GLInventoryAccount	string (GLAccount)	R/O		
IsFixedCost	boolean	R/O		
FixedCost	currency	R/O		

API Property Name	Data Type	Access	Control Table.Field	Notes
Products	ProductDefinitions[]	R/O		
Modifiers	ModifierDefinitions[]	R/O		
UDFs	UDFs[]	R/O		
Tags	StringLIst	R/O		
InventoryItems	InventoryItems[]	R/O		

5.7.19 Warehouse

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	Warehouse.ID	IDcCT format
DateCreated	DateTime	R/O	Warehouse.DateCreated	
DateModified	DateTime	R/O	Warehouse.ModifiedDate	
eTag	int	R/O	Warehouse.SeqID	
IsActive	boolean	R/O	Warehouse.IsActive	
Name	string	R/O	Warehouse.Name	
Group	string (Warehouse)	R/O		
WarehouseType	{Standard, Group, Stockroom, Kanban}	R/O		
InventoryItems	InventoryItems[]	R/O		
Warehouses	Warehouses[]	R/O		

5.7.20 InventoryItem

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	InventoryItem.ID	IDcCT format
DateCreated	DateTime	R/O	InventoryItem.DateCreated	
DateModified	DateTime	R/O	InventoryItem.ModifiedDate	

API Property Name	Data Type	Access	Control Table.Field	Notes
eTag	int	R/O	InventoryItem.SeqID	
IsActive	boolean	R/O	InventoryItem.IsActive	
Warehouse	Warehouse	R/O		
Part	Part	R/O		
QuantityBilled	float	R/O		
QuantityReceived	float	R/O		
QuantityOnHand	float	R/O		
QuantityReserved	float	R/O		
QuantityAvailable	float	R/O		
QuantityOnOrder	float	R/O		
QuantityExpected	float	R/O		
LastInventoried	DateTime	R/O		
GroupID	string (InventoryItem ID)	R/O		IDcCT format
Items	InventoryItem[]	R/O		

5.7.21 SelectionListCategory

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) SelectionListCategory.ID	IDcCT format
DateCreated	DateTime	R/O	(view) SelectionListCategory.DateCreated	
DateModified	DateTime	R/O	(view) SelectionListCategory.ModifiedDate	
eTag	int	R/O	(view) SelectionListCategory.SeqID	
IsActive	boolean	R/O	(view) SelectionListCategory.IsActive	
Name	string	R/O	(view)	

API Property Name	Data Type	Access	Control Table.Field	Notes
			SelectionListCategory.Name	
SelectionLists	SelectionList[]	R/O		
ID	string	R/O	(view) SelectionListCategory.ID	IDcCT format
DateCreated	DateTime	R/O	(view) SelectionListCategory.DateCreated	
DateModified	DateTime	R/O	(view) SelectionListCategory.ModifiedDate	
eTag	int	R/O	(view) SelectionListCategory.SeqID	
IsActive	boolean	R/O	(view) SelectionListCategory.IsActive	
Name	string	R/O	(view) SelectionListCategory.Name	
SelectionLists	SelectionList[]	R/O		

5.7.22 SelectionList

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) SelectionList.ID	IDcCT format
DateCreated	DateTime	R/O	(view) SelectionList.DateCreated	
DateModified	DateTime	R/O	(view) SelectionList.ModifiedDate	
eTag	int	R/O	(view) SelectionList.SeqID	
IsActive	boolean	R/O	(view) SelectionList.IsActive	
Name	string	R/O	(view) SelectionList.Name	
GroupID	string (SelectionLi	R/O		IDcCT format

API Property Name	Data Type	Access	Control Table.Field	Notes
	stCategory ID)			
Items	SelectionLis tItems[]	R/O		
IsDynamic	boolean	R/O		true if it contains Dynamic Part Links

5.7.23 SelectionListItem

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) SelectionListItem.ID	IDcCT format
DateCreated	DateTime	R/O	(view) SelectionListItem.DateCreated	
DateModified	DateTime	R/O	(view) SelectionListItem.Modified Date	
eTag	int	R/O	(view) SelectionListItem.SeqID	
IsActive	boolean	R/O	(view) SelectionListItem.IsActive	
Name	string	R/O	(view) SelectionListItem.Name	Manually Construct "Blank" and "Default" values where ID = SelectionList.ID
IsDefault	boolean	R/O		
IsBlank	boolean	R/O		
IsDynamic	boolean	R/O		true if created through a dynamic part link
SelectionList	SelectionLis t	R/O		
Parts	Parts[]	R/O		
Filters	Dictionary	R/O		Included for discussion. Not mandatory.

API Property Name	Data Type	Access	Control Table.Field	Notes
Properties	Dictionary	R/O		Included for discussion. Not mandatory.
ImagePaths	Dictionary	R/O		Included for discussion. Not mandatory.

5.7.24 PricingPlan

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ID	IDcCT format
DateCreated	DateTime	R/O	(view) DateCreated	
DateModified	DateTime	R/O	(view) ModifiedDate	
eTag	int	R/O	(view) SeqID	
IsActive	boolean	R/O	(view) IsActive	
Name	string	R/O	(view) Name	
ID	string	R/O	(view) ID	IDcCT format
DateCreated	DateTime	R/O	(view) DateCreated	
DateModified	DateTime	R/O	(view) ModifiedDate	
eTag	int	R/O	(view) SeqID	
IsActive	boolean	R/O	(view) IsActive	
Name	string	R/O	(view) Name	

5.7.25 Variable

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string	R/O	(view) ID	IDcCT format
DateCreated	DateTime	R/O	(view) DateCreated	
DateModified	DateTime	R/O	(view) ModifiedDate	
eTag	int	R/O	(view) SeqID	
IsActive	boolean	R/O	(view) IsActive	
Name	string	R/O	(view) Name	
ProductID	string (Product ID)			IDcCT format

API Property Name	Data Type	Access	Control Table.Field	Notes
PricingPlan	string (PricingPlan)			
VariableType	{Number, Text, Boolean, List, Table, Image, DateTime}			
Notes				
AnswerRequired	{true, false, conditional}			
DefaultValue				
UnitType	{}			
EntryUnits	string (Units)			
DisplayUnits	string (Units)			
IsResetOnClone	boolean			
IsLimitedToList	boolean			
DefaultSelectionList	string (SelectionList)			blank if not applicable
SelectionListFilter	string			
SelectionListImageLink	string			
UsesDependentList	boolean			
DependentListVariable	string (Variable)			
DependentListName	string (SelectionList)			
TableName	string			

5.7.26 Order

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		(view)Order.ID	IDcCT format
DateCreated	DateTime		(view)Order.OrderDateCreated	
DateModified	DateTime		(view)Order.ModifiedDate	
eTag	int		(view)Order.SeqID	
IsActive	boolean		IsActive	

API Property Name	Data Type	Access	Control Table.Field	Notes
OrderNumber	int		OrderNumber	
InvoiceNumber	int		InvoiceNumber	
OriginalEstimateNumber	int		EstimateNumber	
TransactionType	{1=Order, 2=Estimate, 3=Template, 4=CreditMemo, 6=ServiceTicket, 7=PO, 8=Bill, 9=RD, 10=VendorCreditMemo}		TransactionType	Only 1,4,6 are valid for Orders
Status	{1=WIP, 2=Built, 3=Sale, 4=Closed, 9=Voided}		StatusID or StatusText	
Station	string		StationID	
Description	string		Description	
CompanyID	string (Account.ID)		(view)Order.AccountID + ctID	IDcCT format
ContactID	string (AccountContact.ID)		(view)Order.ContactID + ctID	IDcCT format
Division	string		(Name)linked via (view)Order.DivisionID	
ProductionDivision	string		linked via ProductionDivisionID	
ProductionWarehouse	string		linked via WarehouseID	
EstimatingWarehouse	string		linked via EstimatingWarehouseID	
InvoiceCompanyID	string (Account.ID)		InvoiceCompanyID + ctID	
InvoiceContactID	string (AccountContact.ID)		InvoiceContactID + ctID	
InvoiceAddress	Address		linked via InvoiceAddressID	Do we need the address links here? (probably not)

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
EnteredByID	string (Employee.ID)		EnteredByID + ctID	IDcCT format
Salesperson[]	Employee[]			Handled same as in Company
Items[]	LineItem[]			Includes ALL line items, not just top level
Parts[]	TransPart[]		Pulled from TransPart	
Payments[]	Payment[]			
OrderActivities[]	OrderActivity[]			
ProductionActivities[]	ProductionActiv ity[]			
CalendarActivities[]	CalendarActivit y[]			
CustomerPONumber	string		PONumber	Renamed to avoid confusion with PurchaseOrderNum ber
DateDue	DateTime		DueDate + DueTime	
IsFirmDateDue	boolean		IsFirmDueDate	
DateProofDue	DateTime		ProofDate	
TaxClass	string (TaxClass)		linked via TaxClassID	
IsTaxExempt	boolean		IsTaxExempt	
TaxExemptionNumber	string		TaxNumber	
Taxes[]	Taxes[]		built from TaxItems	
PricingLevel	Float		PricingLevel	
DiscountLevel	Float		DiscountLevel	
Price	OrderPrice (Struct)			
AmountPaid	Float		PaymentTotal	
AmountWrittenOff	Float		WriteOffAmount	
AmountCreditMemo	Float		CreditMemoAmount	
AmountDue	Float		BalanceDue	
CreditMemoNumber	int		{select OrderNumber from TransHeader where ID =	

API Property Name	Data Type	Access	Control Table.Field	Notes
			this.CreditMemoOrderID}	
Notes	string		OrderNotes	
ProductionNotes	string		ProductionNotes	
IsShipped	boolean		derived from (ShipmentType == 0) ?	constructed in view
DefaultShipToCompanyID	string (Account.ID)		ShippingCompanyID + ctID	
DefaultShipToContactID	string (AccountContact.ID)		ShippingContactID + ctID	
DefaultShipToAddress	Address		linked via ShippingAddressID	Do we need the address links here?
DefaultShippingAccountNumber	string		ShippingAccountNumber	
DefaultShippingCarrier	string		link via ShippingCarrierID	
DefaultShippingAccountPostalCode	string		ShippingAccountPostalCode	
ShipFromCompanyID	string (Account.ID)		ShipFromCustomerID	
ShipFromContactID	string (AccountContact.ID)		ShipFromID	When ShipFromID = 3000
ShipFromAddress	Addresss			
ShipFromPhoneNumber	string		link via ShipFromPhoneNumberID	
ShipFromEmail	string		ShipFromEmailAddress	
DateBuilt	DateTime		BuiltDate	
DateSale	DateTime		SaleDate	
DateClosed	DateTime		ClosedDate	
DateVoided	DateTime		VoidedDate	

5.7.27 Estimate

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		ID	IDcCT format

API Property Name	Data Type	Access	Control Table.Field	Notes
DateCreated	DateTime		OrderDateCreated	
DateModified	DateTime		ModifiedDate	
eTag	int		SeqID	
IsActive	boolean		IsActive	
EstimateNumber	int		EstimateNumber	
ConvertedOrderNumber	int		OrderNumber	
TransactionType	{1=Order, 2=Estimate, 3=Template, 4=CreditMemo, 6=Order, 7=PO, 8=Bill, 9=RD, 10=VendorCreditMemo}		TransactionType	Only 2 is valid for Estimates
Status	{1=WIP, 2=Built, 3=Sale, 4=Closed, 9=Voided}		StatusID or StatusText	
Station	string		StationID	
Description	string		Description	
CompanyID	string (Account.ID)		AccountID + ctID	IDcCT format
ContactID	string (AccountContact.ID)		ContactID + ctID	IDcCT format
Division	string		linked via DivisionID	
ProductionDivision	string		linked via ProductionDivisionID	
ProductionWarehouse	string		linked via WarehouseID	
EstimatingWarehouse	string		linked via EstimatingWarehouseID	
InvoiceCompanyID	string (Account.ID)		InvoiceCompanyID + ctID	
InvoiceContactID	string (AccountContact.ID)		InvoiceContactID + ctID	
InvoiceAddress	Address		linked via InvoiceContactID	Do we need the

API Property Name	Data Type	Access	Control Table.Field	Notes
				address links here?
EnteredByID	string (Employee.ID)		EnteredByID + ctID	IDcCT format
Salesperson[]	Employee[]			Handled same as in Company
VariationCount	int			Contains a count of the number of variations
Variations[]	EstimateVariation[]			Contains the links to the variations
Items[]	LineItem[]			Contains the links to the PRIMARY variation. Includes ALL line items, not just top level
Parts[]	TransParts[]			Contains the links to the PRIMARY variation Parts
EstimateActivities[]	EstimateActivity[]			
CalendarActivities[]	CalendarActivity[]			
CustomerPONumber	string		PONumber	Renamed to avoid confusion with PurchaseOrderNumber
DateDue	DateTime		DueDate + DueTime	
IsFirmDateDue	boolean		IsFirmDueDate	
DateProofDue	DateTime		ProofDate	
TaxClass	string (TaxClass)		linked via TaxClassID	
IsTaxExempt	boolean		IsTaxExempt	
TaxExemptionNumber	string		TaxNumber	
Taxes[]	Taxes[]		built from TaxItems	
PricingLevel	Float		PricingLevel	
DiscountLevel	Float		DiscountLevel	
Price	OrderPrice			

API Property Name	Data Type	Access	Control Table.Field	Notes
	(Struct)			
AmountDue	Float		BalanceDue	
Notes	string		OrderNotes	
ProductionNotes	string		ProductionNotes	
IsShipped	boolean		derived from (ShipmentType == 0) ?	
DefaultShipToCompanyID	string (Account.ID)		ShippingCompanyID + ctID	
DefaultShipToContactID	string (AccountContact.ID)		ShippingContactID + ctID	
DefaultShipToAddress	Address		linked via ShippingAddressID	Do we need the address links here?
DefaultShippingAccountNumber	string		ShippingAccountNumber	
DefaultShippingCarrier	string		link via ShippingCarrierID	
DefaultShippingAccountPostalCode	string		ShippingAccountPostalCode	
ShipFromCompanyID	string (Account.ID)		ShipFromCustomerID	
ShipFromContactID	string (AccountContact.ID)		ShipFromContactID	
ShipFromAddress	Addresss			
ShipFromPhoneNumber	string		link via ShipFromPhoneNumberID	
ShipFromEmail	string		ShipFromEmailAddress	
DateConverted	DateTime		ConvertedDate	
DateRejected	DateTime		RejectedDate	
DateVoided	DateTime		VoidedDate	

5.7.28 EstimateVariation

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		ID	IDcCT format

API Property Name	Data Type	Access	Control Table.Field	Notes
DateCreated	DateTime		EstimateCreatedDate	
DateModified	DateTime		ModifiedDate	
eTag	int		SeqID	
IsActive	boolean		IsActive	
EstimateNumber	int		linked via ParentID	
VariationName	string		VariationName	
Items[]	LineItem[]			Includes ALL line items, not just top level
Parts[]	TransParts[]		Pulled from TransPart	Contains the links to the PRIMARY variation Parts
Price	OrderPrice (Struct)			
AmountDue	Float		BalanceDue	

5.7.29 OrderPrice (Struct)

API Property Name	Data Type	Access	Control Table.Field	Notes
SubTotal	Float		SubTotalPrice or MeAndSonsSubTotalPrice	
Taxable	Float		TaxablePrice or ...	
Taxes	Float		TaxesPrice or ...	
Total	Float		TotalPrice or ...	

5.7.30 OrderLineItem

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		ID	IDcCT format
DateCreated	DateTime			
DateModified	DateTime		ModifiedDate	
eTag	int		SeqID	
IsActive	boolean		IsActive	

API Property Name	Data Type	Access	Control Table.Field	Notes
LineItemRefID	string		TransHeaderTransNumber+ "-"+LineItemNumber	Can be used as a "key" field to identify the record
OrderNumber	int		TransHeaderTransNumber	
LineItemNumber	string		LineItemNumber	
IsTopLevel	boolean		(ParentClassTypeID == 10000)	
SortIndex	float		= fx(LineItemNumber)	
TransactionType	{1=Order, 2=Estimate, 3=Template, 4=CreditMemo, 6=Order, 7=PO, 8=Bill, 9=RD, 10=VendorCreditMemo}		TransactionType	
Product	string		GoodsItemCode	
Quantity	float		UnitParamValue	
Units	string		UnitParamName	
Description	string		Description	
Station	string		linked via StationID	
SummaryHTMLLayout	string		HTMLShortFormat	
ProductionHTMLLayout	string		HTMLLongFormat	
PricingLevel	Float		PricingLevel	
DiscountLevel	Float		DiscountLevel	
Price	LineItemPrice			constructed in transformation
PriceWithChildren	LineItemPrice			constructed in transformation
Taxes[]	Taxes[]		extracted from TaxItems	
Priority	string		Linked via PriorityID	
ProductionWarehouse	string		linked via WarehouseID	
Parts[]	TransPart[]		extracted from PartStr	
Modifiers[]	TransMod[]		extracted from ModifierStr	
Variables[]	TransVariable[]		extracted from ParameterStr	
DateDue	DateTime		DueDate + DueTime	

API Property Name	Data Type	Access	Control Table.Field	Notes
ProductionNotes	string		InternalNotes	
IsProofApproved	boolean		ProofApproved	
ProofApprovedContactID	string (AccountContact.ID)		ProofApprovedByID	
DateProofApproved	DateTime		ProofApprovedDate	
IsComplete	boolean		IsComplete	
PercentComplete	Float		PercentComplete	
AssignedToID	string (Employee.ID)		AssignToID	
TaxClass	string (TaxClass)		linked via TaxClassID	
IsTaxExempt	boolean		IsTaxExempt	is this a field in TransDetail?
TaxExemptionNumber	string		TaxNumber	

5.7.31 EstimateLineItem

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		ID	IDcCT format
DateCreated	DateTime			
DateModified	DateTime		ModifiedDate	
eTag	int		SeqID	
IsActive	boolean		IsActive	
LineItemRefID	string		EstimateNumber+"-"+LineItemNumber + "-" + VariationName	Can be used as a "key" field to identify the record
EstimateNumber	int		EstimateNumber	
LineItemNumber	string		LineItemNumber	
IsTopLevel	boolean		(ParentClassTypeID == 10000)	
SortIndex	float		= fx(LineItemNumber)	
TransactionType	{1=Order, 2=Estimate, 3=Template,		TransactionType	

API Property Name	Data Type	Access	Control Table.Field	Notes
	4=CreditMemo, 6=Order, 7=PO, 8=Bill, 9=RD, 10=VendorCreditMemo}			
Product	string		GoodsItemCode	
Quantity	float		UnitParamValue	
Units	string		UnitParamName	
Description	string		Description	
Station	string		linked via StationID	
SummaryHTMLLayout	string		HTMLShortFormat	
ProductionHTMLLayout	string		HTMLLongFormat	
PricingLevel	Float		PricingLevel	
DiscountLevel	Float		DiscountLevel	
Price	LineItemPrice			constructed in transformation
PriceWithChildren	LineItemPrice			constructed in transformation
Taxes[]	Taxes[]		extracted from TaxItems	use entity TransDetailTaxItem
Priority	string		Linked via PriorityID	
ProductionWarehouse	string		linked via WarehouseID	
Parts[]	TransPart[]		extracted from PartStr	
Modifiers[]	TransMod[]		extracted from ModifierStr	
Variables[]	TransVariable[]		extracted from ParameterStr	
DateDue	DateTime		DueDate + DueTime	
ProductionNotes	string		InternalNotes	
IsProofApproved	boolean		ProofApproved	
ProofApprovedContactID	string (AccountContact.ID)		ProofApprovedByID	
DateProofApproved	DateTime		ProofApprovedDate	
AssignedToID	string (Employee.ID)		AssignToID	
TaxClass	string (TaxClass)		linked via TaxClassID	

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
IsTaxExempt	boolean		IsTaxExempt	is this a field in TransDetail?
TaxExemptionNumber	string		TaxNumber	

5.7.32 LineItemPrice (Struct)

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
Base	Float		BasePrice or MeAndSonsBasePrice	
Discount	Float		DiscountPrice or ...	
Modifier	Float		ModifierPrice or ...	
Rounding	Float		RoundingPrice or ...	
SubTotal	Float		SubTotalPrice or ...	
Taxable	Float		TaxablePrice or ...	
Taxes	Float		TaxesPrice or ...	
Total	Float		TotalPrice or ...	

5.7.33 TransTax

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
ID	int		Row_Number()	
TaxAccountID	string (IDcCT)		TaxAccountID	Need to parse (id, ctid) int IDcCT
Rate	float		TaxRate	
Cap	float		TaxCap	suppress if value = 0
Taxes	float		TaxAmount	
Code	string		TaxCode	suppress if blank

5.7.34 TransVariable

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	int		Row_Number()	
VariableID	string (IDcCT)		VariableID	Need to parse (id, ctid) int IDcCT
Variable	string		VariableName	
Value	string		ValueAsString or ValueAsDateTime	Use ValueAsDateTime if ValueType=7, else use ValueAsString
Unit	string		Link from UnitID	suppress if value = (-1, 300)
IsOverridden	boolean		Overridden	
ValueType	{enum}		based on ValueType	Boolean = 11, DateTime= 7, Double = 5, Int64 = 20, Integer = 3, Null = 1, String = 256, Variant = 12, Currency= 6, Byte = 17
ListID	string (IDcCT)		ListID	suppress if blank

5.7.35 TransMod

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	int		Row_Number()	
ModifierID	string (IDcCT)		GoodsItemID	Need to parse (id, ctid) int IDcCT
ModifierCode			ModifierCode	
Value	string		ValueAsString or ValueAsDateTime	Use ValueAsDateTime if ValueType=7, else use ValueAsString
Unit	string		Link from UnitID	suppress if value = (-1, 300)

API Property Name	Data Type	Access	Control Table.Field	Notes
IsOverridden	boolean		ValueOverridden	
ValueType	{enum}		based on ValueType	Boolean = 11, DateTime= 7, Double = 5, Int64 = 20, Integer = 3, Null = 1, String = 256, Variant = 12, Currency= 6, Byte = 17
ListID	string (IDcCT)		ListID	suppress if blank
DiscountLevel	float			
PricingLevel	float			
Price	float			
IsPriceOverridden	boolean			

5.7.36 TransPart

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		ID	Need to parse (id, ctid) int IDcCT
ReferenceNumber	integer		ReferenceNumber	
PartCode	string		PartCode	
PartID	string (Part.ID)		PartID	Need to parse (id, ctid) int IDcCT
IsIncluded	boolean		Include	Possibly filter this so only included appear
SortIndex	Integer		SortOrder	
TransHeaderID	string (TransDetail.ID)		TransheaderID	Need to parse (id, ctid) int IDcCT
TransDetailID	string (TransDetail.ID)		TransDetailID	Need to parse (id, ctid) int IDcCT
ProductID	string (GoodsItem.ID)		GoodsItemID	Need to parse (id, ctid) int IDcCT
ProductCode	string		GoodsItemCode	
StationName	string		link via StationID	Suppress if

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
				StationID = (-1, -1)
VariableName	string		link via VariableID	Suppress if VariableID = (-1, -1)
WarehouseName	string		link via WarehouseID	
InventoryID	string (Inventory.ID)		InventoryID	Need to parse (id, ctid) int IDcCT
Unit	string		UnitText	
Description	string		Description	
Quantity	TransPartQuanti ty (struct)			
Cost	TransPartCost (struct)			
Price	TransPartPrice (struct)			
IsManuallyIncluded	boolean		not (IsAutomatic)	
ListPartIndex	int		PartIndex	suppress if value = -1
IsAttached	boolean		IsAttachedToVTrans	
AttachedPODetailID	string (VendorTransD etail.ID)		AttachedPODetailID	Need to parse (id, ctid) int IDcCT. Suppress if value = (-1, -1)
AttachedBillDetailID	string (VendorTransD etail.ID)		AttachedBillDetailID	Need to parse (id, ctid) int IDcCT. Suppress if value = (-1, -1)

5.7.37 TransPartQuantity

API Property Name	Data Type	Acc ess	Control Table.Field	Notes
Calculated			CalculatedQuantity	
Estimated			EstimatedQuantity	
Actual			ActualQuantity	
IsEstimatedOV			EstimatedOverridden	

5.7.38 TransPartCost

API Property Name	Data Type	Access	Control Table.Field	Notes
Calculated			CalculatedCost	
Estimated			EstimatedCost	
Actual			ActualCost	
Unit			UnitCost	
IsEstimatedOV			EstimatedCostOverridden	

5.7.39 TransPartPrice

API Property Name	Data Type	Access	Control Table.Field	Notes
SuggestedPrice				
CostMultiplier				
FixedCostMultiplier				
CostMultiplierType				

5.7.40 PaymentActivity

API Property Name	Data Type	Access	Control Table.Field	Notes
ID	string		(view)PaymentActivity.ID	IDcCT format
DateCreated	DateTime		CompletedDateTime	
DateModified	DateTime		ModifiedDate	
eTag	int		SeqID	
IsActive	boolean		IsActive	
EmployeeID	string (EmployeeID)		EmployeeID	
Description	string		Description	
Notes	string		Notes	
TenderType	{Cash, Check,		TenderType	0=Cash, 1=Check,

API Property Name	Data Type	Access	Control Table.Field	Notes
	CreditCard, DebitCard, Other, EFT, System, WireTransfer}			2=CreditCard, 3=DebitCard, 4=Other, 5=EFT, 6-System, 7=WireTransfer
PaymentAccount	string		PaymentAccount	
Amount	float		Amount	
IsPartOfGroupPay	boolean		(ID <> TotalPaymentID)	
GroupPaymentAmount	float		TotalPaymentAmount	
GroupID	int		TotalPaymentID	ID only, not IDcCT
CompletedDateTime	DateTime		CompletedDateTime	
BankAccount	string		BankAccount	
IsVoided	boolean		IsVoided	
VoidedDateTime	DateTime		VoidedDateTime	suppress if IsVoided = 0
VoidedReason	string		VoidedReason	suppress if IsVoided = 0
AccountID	string (Account.ID)		AccountID	IDcCT format
ContactID	string (AccountContact.ID)		ContactID	IDcCT format
OrderNumber	integer		linked via TransactionID	
Division	string		link via DivisionID	
NameOnCard	string		NameOnCard	
DisplayNumber	string		DisplayNumber	
ExpirationDate	DateTime		ExpirationDate	
IsDeposited	boolean		= not Undeposited	

6. Non-Model Resources (Commands)

Some *commands* are issues against a resource maanger, such as a logging service or a time-clock. The commands do not pass or return object models, but may act on a model or on the program as a whole.

6.1 Time Clock Resource

The time clock endpoint on the web api will support listing of the current employees with their clock-in status. There are additional actions for clock-in and clock-out including the option of a station.

6.1.1 Supported REST operations

RESOURCE	SUPPORTED CALLS
TimeClock	**GET ../wapic/TimeClock**
	Retrieves the current list of employees along with their current time clock status (clocked in/clocked out).
	POST ../wapic/TimeClock(####c3500)/ClockIn
	Clocks-in the employee with the supplied Id. If that employee is not found this will return a 404. If the employee is not active this will return a 403.
	POST ../wapic/TimeClock(####c3500)/ClockOut
	Clocks-out the employee with the supplied Id. If that employee is not found this will return a 404. If the employee is not active this will return a 403.
	POST ../wapic/TimeClock(####c3500)/ClockOnStation('stationName')
	Clocks the employee with the supplied Id. Additionally sets the current station of the employee to the supplied stationname. If that employee is not found this will return a 404. If the employee is not active this will return a 403

6.1.2 Data Format

6.1.2.1 *Example Requests and Responses*
The username and password for authorization in these examples is a standard base64 encode. In these examples username:password is the text being encoded. The clientid cookie is returned from the server upon valid request and subsequent requests will need to include it. For the clock-in and -out examples, the base state of the

data is that of the response of the first example. Additionally, there is an employee with Id = 1050c3500 but he is marked inactive, and will therefore not show on the list.

6.1.2.1.1 Get timeclock employee list example:

Request

```
GET https://127.0.0.1:5321/wapic/TimeClock HTTP/1.1
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
```

Response

```
HTTP/1.1 200 OK
Content-Length: xxxx
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Set-Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
Date: Tue, 21 Aug 2012 15:04:10 GMT
[
{"FirstName":"Account","MiddleName":null,"LastName":"_House","DisplayName":"Account
_House","Id":"10c3500", "IsClockedIn":true, "CurrentStation":null},
{"FirstName":"Tommy","MiddleName":"Lee","LastName":"Jones","DisplayName":"Tommy Lee
Jones","Id":"1001c3500", "IsClockedIn":false, "CurrentStation":null},
{"FirstName":"Magic","MiddleName":null,"LastName":"Johnson","DisplayName":"Magic
Johnson","Id":"1002c3500", "IsClockedIn":true, "CurrentStation":"Out to Lunch"}
]
```

6.1.2.1.2 Valid Post clock in example:

Request

```
POST https://127.0.0.1:5321/wapic/TimeClock(1001c3500)/ClockIn HTTP/1.1
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
Accept-Type: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 21 Aug 2012 15:19:48 GMT
Content-Type: application/json
Connection: keep-alive
Content-Length: XX

{"FirstName":"Tommy","MiddleName":"Lee","LastName":"Jones","DisplayName":"Tommy Lee
Jones","Id":"1001c3500", "IsClockedIn":true, "CurrentStation":null}
```

6.1.2.1.3 Valid Post clock out example:

Request

```
POST https://127.0.0.1:5321/wapic/TimeClock(10c3500)/ClockIn HTTP/1.1
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
Accept-Type: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 21 Aug 2012 15:19:48 GMT
Content-Type: application/json
Connection: keep-alive
Content-Length: XX


{"FirstName":"Account","MiddleName":null,"LastName":"_House","DisplayName":"Account
_House","Id":"10c3500", "IsClockedIn":false, "CurrentStation":null},
```

6.1.2.1.4 Employee Id Not Found clock in example:

Request
```
POST https://127.0.0.1:5321/wapic/TimeClock(1001c3500)/ClockIn HTTP/1.1
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
Accept-Type: application/json
```

Response
```
HTTP/1.1 404 Not Found
Date: Tue, 21 Aug 2012 15:36:18 GMT
Content-Type: text/plain
Connection: keep-alive
Content-Length: X
```

6.1.2.1.5 Employee inactive clock in example:

Request
```
POST https://127.0.0.1:5321/wapic/TimeClock(1050c3500)/ClockIn HTTP/1.1
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Cookie: clientid=886F9006-7199-429B-9BFA-8A829C7FBDAD
Accept-Type: application/json
```

Response
```
HTTP/1.1 403 Forbidden
Server: nginx
Date: Tue, 21 Aug 2012 15:36:22 GMT
Content-Type: text/plain
Connection: keep-alive
Content-Length: 13

Employee is inactive and cannot be clocked in/out
```

7. Security Model

7.1 Overview

Security in WAPIC consists of two distinct parts – secure authentication and authorization security.

Authentication restricts the basic access to the system to those users that can be identified. WAPIC relies on basic authentication over encrypted IP connections (SSL) for authentication security.

Authorization involves restricts known users to the areas they are permitted to see. Authorization in WAPIC is implemented

No user security options can be configured via the API. All configuration must be done from within Control or the database directly. Any changes made to these rights will apply at the next request made by the affected login.

7.2 Authentication

Authentication is implemented with basic authentication over SSL.

7.3 Authorization

Once a user is authenticated, specific access to models and commands are controlled via user rights. For each model, a user may have rights to:

- Get (read)
- Post (create)
- Put/Merge (edit)
- Delete

All rights are at the model level. There are no rights at field levels. (For example, if a user has a right to edit a model, that user can edit all values.)

7.3.1 AuthorizationItem Table

The AuthorizationItem table, in the Control database, contains these user rights for the API. It consists of: an identity id (ID: int), a model name (RightName: nvarchar(200)), whether the model is a class (IsClass: bool, currently should be set to `true`), the class type id (ClassTypeID: int, needs to be `225311`, will be used by Control), whether the model can be retrieved (CanGet: bool), whether the model can be modified (CanPut: bool), whether the model can be created (CanPost: bool), whether the model can be deleted (CanDelete: bool), and the version of the value (SeqID: int, will iterate as changes occur to the record).

7.3.2 SecurityRole Table

The SecurityRole table represents role templates for users. It consists of: ID (int), an identity generated by the database, ClassTypeID (int), used by Control and needs to be `225310`, Name (nvarchar(50)), the name of the security role, Discriminator (nvarchar(128)), currently this value needs to be `UserSecurityRole`, and SeqID (int), which will iterate as changes occur to the record.

7.3.3 AuthorizationItemSecurityRole Table

The AuthorizationItemSecurityRole table links the AuthorizationItem records with the SecurityRole records by id. It constists of the AuthorizationItem_ID (int), and SecurityRole_ID (int).

7.3.4 UserNameUserSecurityRole Table

The UserNameUserSecurityRole table links the SecurityRole records with existing Control UserName records. It consists of the UserName_ID (int), and UserSecurityRole_ID (int).

7.4 Login Limitations

The credentials used in the authentication step are the the user's Control username and password, Base64 encoded as `username:password`. On the first attempt to login for a given user through the API, a `clientid` cookie value will be returned. The `clientid` cookie will need to be re-sent on subsequent requests. This clientid will remain active for 15 minutes

8. Logging and Health Monitor System

WAPIC's health monitoring system (HMS) tracks internal events and metrics. HMS records are passed to CHAPI for recording. Records are then saved in two places by chapi

- The internal logging system. These records may be accessed via the ../CHAPI/ interface and are stored in the log table in the configuration database.
- The windows event log. These records may be accessed via the Windows management console.

The following events are recorded by the HMS in the Windows [Application] event viewer and the WAPIC log table:

- Start-up and Shut-down events.
- Authorization failures.
- Database connection events.
- Database failure events.
- Configuration change events.

The following events are recorded by the HMS in the WAPIC log table but not the Windows event viewer:

- User authorization failures.
- License authorization events (successful or failure).
- Any unhandled exception (5xx response).

The following metrics are recorded by the HMS in 5 minute increments. (The increments may be changed in the configuration file, though 5 minutes is the default.) These events are only recorded to the WAPIC log tables and not to the Windows event viewer.

- Statistics for user requests (includes all requests to ../WAPIC/refresh).
    - o Number of requests.
    - o Average response time for requests.
    - o Number of failed responses (4xx or 5xx).

9. Technical Appendix

9.1 Service Specifications

Language	C# .NET 4.5
Application Mode	Windows Service
Permissions	chapi user (by default)
Connection	REST
Data Format	JSON
Protocol	HTTPS
Port	5321 (by default)
Login Credentials	Basic Authentication