



RESELLONE.NET

**ResellOne.net Provisioning System (RPS)
TPP API Specification**

February, 2006

Table of Contents

REVISIONS AND UPDATES	4
INTRODUCTION	6
GENERIC REQUEST KEY HIERARCHY.....	6
GENERIC RESPONSE KEY HIERARCHY.....	7
COMMON CONTACT DATA.....	7
THE ORDER OBJECT	8
PARAMETER DEFINITIONS.....	8
RESPONSE PARAMETER DEFINITIONS.....	12
THE CREATE OPERATION.....	17
THE UPDATE OPERATION.....	18
THE PROCESS OPERATION.....	20
THE CANCEL OPERATION.....	21
THE QUERY OPERATION.....	21
THE QUERY OPERATION - DEPRECATE.....	24
	25
	25
RESPONSE PARAMETER DEFINITIONS.....	26
THE CREATE OPERATION.....	27
THE UPDATE OPERATION.....	29
THE UPDATE OPERATION – DEPRECATE.....	30
THE CHECK OPERATION.....	31
THE DELETE OPERATION.....	34
THE RECOVER PASSWORD OPERATION.....	36
THE QUERY OPERATION - DEPRECATE.....	38
	40
	40
REQUEST PARAMETER DEFINITIONS.....	40
RESPONSE PARAMETER DEFINITIONS.....	42
THE CREATE OPERATION.....	42
THE UPDATE OPERATION.....	44
THE DELETE OPERATION.....	45
THE QUERY OPERATION - DEPRECATE.....	46
THE SESSION OBJECT	48
REQUEST PARAMETER DEFINITIONS.....	48
RESPONSE PARAMETER DEFINITIONS.....	48
THE CREATE OPERATION.....	49
THE QUERY OBJECT	51
CONTACTS BY USER ID.....	53
QUERY INVENTORY ITEM BY ID.....	56
QUERY INVENTORY ITEMS BY USER ID.....	57
QUERY INVENTORY ITEMS BY DESCRIPTION.....	58

Revisions and Updates

December 15, 2005

- Incremented document version number to 2.9.4.
- Added error message number 350 in section *Error Codes*
- The *Query Queued Order Items* and *Acknowledge queued items* commands now support the Digital Certificates, Managed DNS, Email Defense, and Website Builder services.

Horizon test environment availability: December 8, 2005

Live: December 15, 2005.

November 15, 2005

- In section *Error Codes*, added a description of the error code hierarchy, which is relevant when there are multiple parts to a request.

October 25, 2005

- Corrected example and description for Update Order command to include required "user_id" parameter in the "attributes" array.

September 8, 2005

- Incremented document version number to 2.9.3.
- Enhanced provisioning of TPP products to include additional synchronous processes. Some processes cannot be synchronous, and so API commands have been added to handle queuing functionality.

Horizon test environment availability: October 13, 2005

Live: October 19, 2005.

June 22, 2005

- Incremented document version number to 2.9.1.
- Added the list of response codes and response messages returned for generic TPP commands; these responses do not apply to specific services such as Email, DNS, Digital Certificates, etc.

April 15, 2005

- Incremented document version number to 2.8.9.
- Correction: removed mention that transfers are possible on a 'user' object. Users cannot be transferred.

January 27, 2005

- Incremented document version number to 2.8.8.
- Added *Query User by Credentials* to *The Query Object* section.

January 11, 2005

- Incremented document version number to 2.8.7.
- Removed ID parameter from *The User Object* section.

October 5, 2004

- Incremented document version number to 2.8.6.

- Added error codes to end of document.
- Monthly Billing functionality enhanced; as a result, a new parameter, "expiry_date", can be specified for new orders and exiting order items. The "expiry_date" parameter is used with the *Order Object > Create Operation* and *Order Object > Update Operation*. It can also be used with the *Inventory_Item Object > Update Operation*.
- Added new parameter "renewal_msg_flag", which can be used to activate or deactivate a renewal reminder that is sent to end users when a service is about to expire. This parameter is used with the *Inventory_Item Object > Update Operation*.
- Added section *The Inventory_Item Object*, which includes overviews of how to update, suspend, reactivate, and delete inventory items.
- Added the *User > Check Operation*, which checks to see if a username is available for use in the ResellOne.net Provisioning System (RPS). This operation includes the new parameters "is_available".

July 21, 2004

- The RWI (ResellOne.net Web Interface) has a new setting that impacts the TPP API. In the RWI, Resellers can set a default for how their orders are processed: either processed immediately, or saved to pending. If the 'handling' variable is not specified in the API, the default as set in the RWI will be used. If the RWI order-processing-setting was not specified, and a TPP order command does not include the 'handling' variable, the default, which is now 'process' rather than 'save', is used.
- Replaced *Query Contacts by User ID* example.
- Added new command *Query Inventory Items by Description* to *Query Object*.

May 10, 2004

- Character limit for RWI 2 username and respective password has changed. Usernames must be 1 – 256 characters; passwords must be 3 – 256.
- RWI 2 user passwords cannot contain special characters. Creation of a new user will fail if you attempt to submit "!", "@", or "#" in the password.
- Updated *Order Object > Query command*; the previous version of this command was deprecated.

April 20, 2004

- Replaced examples for *User Object > Update Operation*.

September 11, 2003

- Added XML request and response examples throughout this document.

June 30, 2003

- Added *Delete User Operation* to *User Object*.
- Added *Recover User Password Operation* to *User Object*.
- Added *Delete Contact Operation* to *Contact Object*.
- Changed *Update User Operation* in *User Object* to support the updating of multiple users in the same API call.

Introduction

The TPP is the API protocol used for provisioning Managed DNS, Website Builder, and other services. The TPP protocol provides abilities such as querying products, creating contacts, and processing orders. Reseller systems will be able to create, update, query, process and cancel orders via TPP – allowing for more integration with reseller CRM and reseller order management systems.

TPP data is classified into **actions**, **objects**, **services** and **object_types**. The following section details the general structure of a TPP request.

Generic Request Key Hierarchy

```
version (string)
protocol
action (string)
object
requestor => { (associative array)
  username
}
attributes => {

object_type
}

~ also ~

    (string)
    (string)
        (associative array)
    (string)
}
        (associative array)
...
create_items => [ (array of associative arrays)
{
    (string)
        (string)
}
]
}
```

These four keys allow the ResellOne.net server to properly process and dispatch the request to the appropriate subsystem (e.g. Email, Digital Certificates, etc.).

Generic Response Key Hierarchy

action (*string*)
object (*string*)
is_success (*boolean*)
response_code (*integer*)
response_text (*string*)
attributes (*associative array*)

TPP commands are sent to the ResellOne.net servers using the same host and port as with the original XCP client protocol. In order for the ResellOne.net servers to distinguish between TPP and XCP, the "protocol" header must be provided. Previously, this value was not required as all commands were directed to the ResellOne.net Domain system. With a value of "TPP" for protocol, the ResellOne.net servers can properly direct the requests to the appropriate provisioning subsystem.

The following **objects** exist in TPP:

- orders
- contacts
- users
- products
- sessions

The following lists all possible **actions** in TPP:

- check
- query
- create
- update
- delete
- process
- cancel
- reverse
- renew
- transfer

Not all actions are available for all objects and object types, so check each object and object type definition in this document to see which actions are valid for each object.

All TPP commands include a "requestor" associative array in the header. This identifies to ResellOne.net which user is requesting this operation. Normally, this associative array contains the username that your reseller account uses to log into the RWI.

Common Contact Data

ResellOne.net uses contact data for customer relation management. For every type of contact in ResellOne.net, there is a set of common contact data, such as last name, a single street address, country, etc. Some products specify additional required contact fields. Please see the descriptions and definitions of individual services for additional contact specifications.

The Order Object

The Order object facilitates order management. Orders are used to provision billable products such as email accounts and digital certificates. They are also used to perform billable changes to active products. An example of this would be an upgrade of storage capacity to an email account.

Orders may contain information regarding the registrant, the contact(s) for the products, and the products themselves. Orders can contain multiple order items up to a defined system limit. This permits the client to order multiple and diverse products in a single order (e.g. a domain, a web certificate and 5 email accounts).

Orders permit the following actions to be performed:

- create
- update
- process
- cancel
- query

Request Parameter Definitions

client_reference	<p><i>Parameter Name:</i> client_reference <i>Type:</i> String <i>Description:</i> Reseller-supplied reference id for correlation with reseller-based customer and data management systems. <i>Obligation:</i> Optional <i>Location:</i> This key is permitted in the attributes associative array. <i>Allowed Value(s):</i> Free text. Maximum 64 characters.</p>
handling	<p><i>Parameter Name:</i> handling <i>Type:</i> String <i>Description:</i> Tells ResellOne.net how to process the new order. Using 'process' avoids having to send a separate order-process command. Failed processing or validation will result in the order and order items (where possible) being saved. If not submitted, the processing setting, as set in the ResellOne.net Web Interface, will be used. If the processing setting in the RWI has not been set, the default, 'process' will be used. <i>Obligation:</i> Optional, defaults to 'process' <i>Location:</i> This key is permitted in the attributes associative array. <i>Allowed Value(s):</i> One of 'save' or 'process'.</p>
username	<p><i>Parameter Name:</i> username <i>Type:</i> String <i>Description:</i> The username of the registrant for which this order is being created. If the username and the password are matched with an existing record within the requestor's namespace, then the new order is linked to this user. If the username matches but the password doesn't, then the order will be rejected and not saved. <i>Obligation:</i> Required <i>Location:</i> This key is permitted in the attributes associative array. <i>Allowed Value(s):</i> Minimum length of 4 characters.</p>

password	<p><i>Parameter Name:</i> password</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The password for the registrant. See the description of the username key for more information.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the attributes associative array.</p> <p><i>Allowed Value(s):</i> Minimum 4 characters.</p>
contacts	<p><i>Parameter Name:</i> contacts</p> <p><i>Type:</i> Array of associative arrays</p> <p><i>Description:</i> The array of contacts to be used for this order. Both new and existing contacts can be specified here. The server will look for an "id" key. If it exists, this signals the reuse of an existing contact. Without the "id" key, the server will create a new contact with the given data.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the attributes associative array.</p> <p><i>Allowed Value(s):</i> See the description of the common contact data earlier in this document.</p>
create_items	<p><i>Parameter Name:</i> create_items</p> <p><i>Type:</i> Array of associative arrays</p> <p><i>Description:</i> The list of items for the order. The contents of each item depend on the values of the service, object_type and orderitem_type.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the attributes associative array.</p> <p><i>Allowed Value(s):</i> Values are variable. The keys service, object_type, orderitem_type, period, product_data and contact_set must be provided for all items.</p>
service	<p><i>Parameter Name:</i> service</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The ResellOne.net service which will be provisioning the product.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the "create_items" associative array.</p> <p><i>Allowed Value(s):</i> One of 'cert', 'email', 'dns', or 'emaildefense'</p>
object_type	<p><i>Parameter Name:</i> object_type</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The type of object to be provisioned by the service.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the "create_items" associative array.</p> <p><i>Allowed Value(s):</i> Dependant on the service above. See the individual service descriptions for more information.</p>
orderitem_type	<p><i>Parameter Name:</i> orderitem_type</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The type of the order item.</p> <p><i>Obligation:</i> Optional. Defaults to 'new' if not supplied.</p> <p><i>Location:</i> This key is permitted in the "create_items" associative array.</p> <p><i>Allowed Value(s):</i> One of 'new' (default), 'upgrade', 'renew' or 'transfer'</p>
contact_set	<p><i>Parameter Name:</i> contact_set</p> <p><i>Type:</i> Associative array</p> <p><i>Description:</i> The contacts to be used for this order item and the resulting product, if applicable. If successfully processing the order item means creating a new product in the system, then these contacts are used for the product. Otherwise, if the order item upgrades or renews an existing product, then the contact data from the order item are not propagated to the product.</p> <p>Typical contacts types are owner, admin, billing and tech. Refer to each product description to see which contacts type are required.</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the "create_items" associative array.</p> <p><i>Allowed Value(s):</i> Values of the contact types supplied (e.g. Admin) refer to array indexes in the contacts array in the attributes associative array. The array index numbering starts at 0 (zero).</p>

product_data	<p><i>Parameter Name:</i> product_data <i>Type:</i> Associative array <i>Description:</i> The product data for the order item <i>Obligation:</i> Required <i>Location:</i> This key is permitted in the "create_items" associative array. <i>Allowed Value(s):</i> Variable. See the individual product descriptions for the list of product data keys.</p>
expiry_date	<p><i>Parameter Name:</i> expiry_date <i>Type:</i> String <i>Description:</i> When used with the "create_items" associative array, this parameter allows you to set an expiry date for a new order item. When used with the "update_items" associative array, this parameter allows you to add, change, or remove an expiry date on an existing order item. Some services, i.e. Digital Certificates, are sold for a fixed term and therefore do not support expiration dates. To set or change an expiry date The date specified must be a future date, in the following format: - the day must be 2 characters, i.e. 01-31 - the month must be the first three characters of the month name, i.e. Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec - the year must be in four character format, e.g. 2005 - e.g. 27-Aug-2008 To remove an expiry date: Submit '-1' <i>Obligation:</i> Optional - If not specified, the order item will not expire, and will be billed to your Reseller account each month, indefinitely or until the item is canceled, suspended, or an expiry date is set. <i>Location:</i> Permitted in the "create_items", "update_items", and "inventory_items" associative arrays. <i>Allowed Value(s):</i> Future date in valid format, e.g. '27-Aug-2008', or '-1' if removing an expiry date.</p>
order_id	<p><i>Parameter Name:</i> order_id <i>Type:</i> Integer <i>Description:</i> The integer ResellOne.net order ID, which was returned in the response to, the successful order_create. <i>Obligation:</i> Required. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Valid order ID</p>
contacts	<p><i>Parameter Name:</i> contacts <i>Type:</i> Array of associative arrays <i>Description:</i> The list of contact data to be used for this command. Contact data must be supplied for all new items in create_items. Contact data is only required for the update_items if contact type assignments are changing (e.g. owner contact data is not being changed, but rather, the entire contact is switched with another). To update contact data specifically, use the update-contact command described later in this document. As with the create-order command, contact id's can be supplied to signal to the server to reuse an existing Resellone.net contact. <i>Obligation:</i> Optional. See description above. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See the contact data description in the create-order section.</p>

create_items	<p><i>Parameter Name:</i> create_items <i>Type:</i> Array of associative arrays <i>Description:</i> The list of new order items to add to this ResellOne.net order. See the description of create_items from the create-order command earlier for information regarding keys and values in each element of this array. <i>Obligation:</i> Optional. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i></p>
update_items	<p><i>Parameter Name:</i> update_items <i>Type:</i> Array of associative arrays <i>Description:</i> The list of items to be updated. See below for a list of generic keys and values. <i>Obligation:</i> Optional. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i></p>
cancel_items	<p><i>Parameter Name:</i> cancel_items <i>Type:</i> Array of associative arrays <i>Description:</i> The list of order items to canceled. The only key required and permitted in the element associative arrays is the "item_id". <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i></p>
reverse_items	<p><i>Parameter Name:</i> reverse_items <i>Type:</i> Array of associative arrays <i>Description:</i> The list of order items to reversed. The only key required and permitted in the element associative arrays is the "item_id". <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i></p>
item_id	<p><i>Parameter Name:</i> item_id <i>Type:</i> Integer <i>Description:</i> The ResellOne.net id for this order item. <i>Obligation:</i> Required <i>Location:</i> This key is located in the "update_items" associative array. <i>Allowed Value(s):</i></p>
contact_set	<p><i>Parameter Name:</i> contact_set <i>Type:</i> Associative array <i>Description:</i> The contacts to be used for the. This associative array permits the client to change which contacts are associated with the order item. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See the description of "contact_set" for create_items in the order-create command.</p>
product_data	<p><i>Parameter Name:</i> product_data <i>Type:</i> Associative array <i>Description:</i> The product data for the order item. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Variable. Some data is not changeable once the order item is created. See the individual product descriptions for the list of product data keys.</p>
user_id	<p><i>Parameter Name:</i> user_id <i>Type:</i> Integer <i>Description:</i> The ID of the user for the request <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Valid user ID</p>

data	<p><i>Parameter Name:</i> data</p> <p><i>Type:</i> String</p> <p><i>Description:</i> Indicates to ResellOne.net whether a full or brief response should be returned.</p> <p><i>Obligation:</i> Optional. Defaults to 'full'.</p> <p><i>Location:</i> This key is located in the attributes associative array.</p> <p><i>Allowed Value(s):</i> One of 'brief' or 'full'.</p>
-------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Response Parameter Definitions

order_id	<p><i>Parameter Name:</i> order_id</p> <p><i>Type:</i> Integer</p> <p><i>Description:</i> The integer identifier of the new ResellOne.net order</p> <p><i>Obligation:</i> Always present.</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i> An integer.</p>
-----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

client_reference	<p><i>Parameter Name:</i> client_reference</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The client's reference to the order, repeated back from the request.</p> <p><i>Obligation:</i> Only present if supplied in the request</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i> Free text. Maximum 64 characters.</p>
-------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

status	<p><i>Parameter Name:</i> status</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The new ResellOne.net order's status.</p> <p><i>Obligation:</i> Always present.</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i> One of : 'charged' = order completed 'pending-process' = order was created and saved but not processed 'provisioning-requested' = order is in progress 'declined' 'refunded' 'cancelled'</p>
---------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

price	<p><i>Parameter Name:</i> price</p> <p><i>Type:</i> Integer (in US cents)</p> <p><i>Description:</i> The new ResellOne.net order's total price.</p> <p><i>Obligation:</i> Present only if the order is valid and has valid order items.</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i></p>
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

contacts	<p><i>Parameter Name:</i> contacts</p> <p><i>Type:</i> Array of associative arrays</p> <p><i>Description:</i> The results of processing the contacts from the request (only contact creates are possible here).</p> <p><i>Obligation:</i> Only present if supplied in the request .</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i> See description below</p>
-----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

create_items	<p><i>Parameter Name:</i> create_items</p> <p><i>Type:</i> Array of associative arrays</p> <p><i>Description:</i> The individual results of creating each order item.</p> <p><i>Obligation:</i> Only present if supplied in the request.</p> <p><i>Location:</i> This key is located in the response message.</p> <p><i>Allowed Value(s):</i> See description below</p>
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

id	<p><i>Parameter Name:</i> id</p> <p><i>Type:</i> Integer</p> <p><i>Description:</i> The ResellOne.net internal contact id.</p>
-----------	--------------------------------------------------------------------------------------------------------------------------------

Obligation: Only present if the contact already exists, or if the contact was successfully created.
Location: This key is located in the response message.
Allowed Value(s):

major_code *Parameter Name:* major_code
Type: Integer
Description: The major result code of processing the contact's data.
Obligation: Only present if the contact was created.
Location: This key is located in the response message.
Allowed Value(s):

major_text *Parameter Name:* major_text
Type: String
Description: The major result text of processing the contact's data.
Obligation: Only present if the contact was created.
Location: This key is located in the response message.
Allowed Value(s):

item_id *Parameter Name:* item_id
Type: Integer
Description: The ResellOne.net internal order item id.
Obligation: Required
Location: This key is located in the response message.
Allowed Value(s):

price *Parameter Name:* price
Type: Integer (in US cents)
Description: The order item's price.
Obligation: Present if order item is valid.
Location: This key is located in the response message.
Allowed Value(s): Order items, which contain errors, have no price.

major_code *Parameter Name:* major_code
Type: Integer
Description: The major result code of the order item's processing.
Obligation: Required
Location: This key is located in the response message.
Allowed Value(s):

major_text *Parameter Name:* major_text
Type: String
Description: The major result text of the order item's processing.
Obligation: Required
Location: This key is located in the response message.
Allowed Value(s):

product_item *Parameter Name:* product_item
Type: Associative array
Description: An echo of the complete item from the request.
Obligation: Required
Location: This key is located in the response message.
Allowed Value(s):

order_id *Parameter Name:* order_id
Type: Integer
Description: The integer identifier of the new ResellOne.net order
Obligation: Always present.
Location: This key is located in the response message.
Allowed Value(s): An integer.

client_reference *Parameter Name:* client_reference
Type: String
Description: The client's reference to the order, repeated back from the request.

Obligation: Only present if supplied in the request
Location: This key is located in the response message.
Allowed Value(s): Free text. Maximum 64 characters.

price

Parameter Name: price
Type: Integer (in US cents)
Description: The new ResellOne.net order's total price.
Obligation: Present only if the order is valid and has valid order items.
Location: This key is located in the response message.
Allowed Value(s):

contacts

Parameter Name: contacts
Type: Array of associative arrays
Description: The results of processing the contacts from the request (only contact creates are possible here).
Obligation: Only present if supplied in the request.
Location: This key is located in the response message.
Allowed Value(s): See description below

create_items

Parameter Name: create_items
Type: Array of associative arrays
Description: The individual results of creating each order item.
Obligation: Only present if supplied in the request.
Location: This key is located in the response message.
Allowed Value(s): See description below

id

Parameter Name: id
Type: Integer
Description: The ResellOne.net internal contact id.
Obligation: Only present if the contact already exists, or if the contact was successfully created.
Location: This key is located in the response message.
Allowed Value(s):

major_code

Parameter Name: major_code
Type: Integer
Description: The major result code of processing the contact's data.
Obligation: Only present if the contact was created.
Location: This key is located in the response message.
Allowed Value(s):

major_text

Parameter Name: major_text
Type: String
Description: The major result text of processing the contact's data.
Obligation: Only present if the contact was created.
Location: This key is located in the response message.
Allowed Value(s):

item_id

Parameter Name: item_id
Type: Integer
Description: The ResellOne.net internal order item id.
Obligation: Required
Location: This key is located in the response message.
Allowed Value(s):

term

Parameter Name: term
Type: Integer
Description: Duration of the item
Obligation: In response from query order
Location: This key is located in the response message.
Allowed Value(s):

price	<p><i>Parameter Name:</i> price <i>Type:</i> Integer (in US cents) <i>Description:</i> The order item's price. <i>Obligation:</i> Present if order item is valid. <i>Location:</i> This key is located in the response message. <i>Allowed Value(s):</i> Order items, which contain errors, have no price.</p>
major_code	<p><i>Parameter Name:</i> major_code <i>Type:</i> Integer <i>Description:</i> The major result code of the order item's processing. <i>Obligation:</i> Required <i>Location:</i> This key is located in the response message. <i>Allowed Value(s):</i></p>
major_text	<p><i>Parameter Name:</i> major_text <i>Type:</i> String <i>Description:</i> The major result text of the order item's processing. <i>Obligation:</i> Required <i>Location:</i> This key is located in the response message. <i>Allowed Value(s):</i></p>
product_item	<p><i>Parameter Name:</i> product_item <i>Type:</i> Associative array <i>Description:</i> An echo of the complete item from the request. <i>Obligation:</i> Required <i>Location:</i> This key is located in the response message. <i>Allowed Value(s):</i></p>
order_id	<p><i>Parameter Name:</i> order_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the ResellOne.net order <i>Obligation:</i> Always present <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> An integer</p>
client_reference	<p><i>Parameter Name:</i> client_reference <i>Type:</i> String <i>Description:</i> The client's reference to the order. <i>Obligation:</i> Only present if supplied when the order was created. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Free text.</p>
price	<p><i>Parameter Name:</i> price <i>Type:</i> Integer (in US cents) <i>Description:</i> The order's price. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Can be 0 (zero) if the order no longer has priced items.</p>
contacts	<p><i>Parameter Name:</i> contacts <i>Type:</i> Array of associative arrays <i>Description:</i> The results of processing the contacts. <i>Obligation:</i> Only present if supplied in the update request. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See description below.</p>
create_items	<p><i>Parameter Name:</i> create_items <i>Type:</i> Array of associative arrays <i>Description:</i> The individual results of creating each order item <i>Obligation:</i> Only present if supplied in the update request <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See description of create_items in the create-order response for a list of keys and values.</p>

update_items	<p><i>Parameter Name:</i> update_items <i>Type:</i> Array of associative arrays <i>Description:</i> The results of processing order item updates. <i>Obligation:</i> Only present if supplied in the update request. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See description of create_items in the create-order response for a list of keys and values.</p>
cancel_items	<p><i>Parameter Name:</i> cancel_items <i>Type:</i> Array of associative arrays <i>Description:</i> The individual results of canceling order items. <i>Obligation:</i> Only present if supplied in the update request <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See description of create_items in the create-order response for a list of keys and values. Canceled items do not have a price.</p>
reverse_items	<p><i>Parameter Name:</i> reverse_items <i>Type:</i> Array of associative arrays <i>Description:</i> The individual results of reversing order items. <i>Obligation:</i> Only present if supplied in the update request <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> See description of create_items in the create-order response for a list of keys and values. Refunded items do not have a price.</p>
order_id	<p><i>Parameter Name:</i> order_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the ResellOne.net order. <i>Obligation:</i> Required <i>Location:</i> This key is located in the response attributes associative array. <i>Allowed Value(s):</i> An integer.</p>
client_reference	<p><i>Parameter Name:</i> client_reference <i>Type:</i> String <i>Description:</i> The client's reference to the order <i>Obligation:</i> This key is located in the response attributes associative array. <i>Location:</i> This key is located in the response message. <i>Allowed Value(s):</i> Free text.</p>
order_id	<p><i>Parameter Name:</i> order_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the ResellOne.net order. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> An integer <i>Allowed Value(s):</i> An integer</p>
client_reference	<p><i>Parameter Name:</i> client_reference <i>Type:</i> String <i>Description:</i> The client's reference to the order. <i>Obligation:</i> Only present if supplied when the order was created. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Free text</p>

The Create Operation

The create operation for orders permits the creation of orders in the ResellOne.net Provisioning System (RPS). An order is bound to a user in the system, so one must already exist or be created first using the user-create command. As a convenience, order_create allows the reseller to specify new or existing contacts for the order. The ResellOne.net Provisioning System (RPS) will automatically create the contacts for the registrant user as required.

Order_create permits the creation of empty orders for later editing. The minimum data required for create-order is the username and password of the customer. The reseller can store the returned order id for later use in update-order commands.

Request Example

The format of a typical order create request is as follows:

```
protocol => 'TPP',
action => 'create',
object => 'order',
requestor => {
  username => '',
},
attributes => {
  client_reference => '',
  handling => '',
  username => '',
  password => '',
  contacts => [
    {
      id => '',
      first_name => '',
      ...
    },
    ...
  ],
  create_items => [
    {
      service => '',
      object_type => '',
      expiry_date => '',
      contact_set => {
        ...
      },
      product_data => {
        ...
      },
    },
    ...
  ],
},
}
```

Response Example

The format of a typical order create response:

```
      :reply',
object => 'order',
is_success => 1|0,
response_code => xxx,
response_text => '',
attributes => {
  order_id => '',
  client_reference => '',
  status => '',
  price => '',
}
```

```

contacts => [
  {
    id => '',
    major_code => xxx,
    major_text => ''
  },
  ...
],
create_items => [
  {
    item_id => '',
    client_reference => '',
    status => '',
    price => '',
    major_code => '',
    major_text => '',
    product_item => {
      ...
    },
  },
  ...
],
},
}

```

The Update Operation

The update operation for orders permits the modification of orders and order items in the ResellOne.net Provisioning System (RPS). Order items can be created, modified, canceled or reversed. Please note that status dictates whether an order item can be modified or canceled. Also, some products and order items do not permit reversal of completed order items. Reversals, when permitted by the product, are time-sensitive and are only permitted on recently processed order items.

The format of a typical order update request is as follows:

The format of a typical order update response is as follows:

The Process Operation

This command allows the client to instruct ResellOne.net servers to start processing an order.

The format of a typical process order request is as follows:

The format of a typical process order response is as follows:

The Cancel Operation

This command allows the client to cancel an entire ResellOne.net order. If at least one order item cannot be canceled, then the entire cancel order will fail.

The Query Operation

Query-Order permits the client to retrieve data pertaining to an order. ResellOne.net will return minimal data for each order item (item id, client reference, price and status).

PERL

XML

```
<?xml version='1.0' encoding='UTF-8'?>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key="version">1.4</item>
        <item key="attributes">
          <dt_assoc>
            <item key="data">full</item>
            <item key="order_id">14763</item>
          </dt_assoc>
        </item>
        <item key="object">order</item>
        <item key="requestor">
          <dt_assoc>
            <item key="password">secret</item>
            <item key="username">purple</item>
          </dt_assoc>
        </item>
        <item key="action">query</item>
        <item key="protocol">TPP</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

PERL

```
{
  'protocol' => 'TPP',
  'version' => '1.4.0',
  'action' => 'QUERY:REPLY',
  'object' => 'ORDER',
  'is_success' => '1',
  'session_id' => 'OSRS-5579179',
  'response_text' => 'Request completed successfully',
  'response_code' => '200',
  'attributes' => {
    'contacts' => [ ],
    'client_reference' => '',
    'status' => 'cancelled',
    'order_id' => '14763',
    'price' => '0',
    'items' => [
      {
        'product_item' => {
          'product_data' => {
            ...
          },
          ...
        },
        'contact_set' => {
          ...
        },
        ...
      }
    ],
  },
};
```

XML

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<OPS_envelope>
  <header>
```

```

    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key="response_code">200</item>
        <item key="version">1.4.0</item>
        <item key="action">QUERY:REPLY</item>
        <item key="is_success">1</item>
        <item key="session_id">OSRS-5579179</item>
        <item key="protocol">TPP</item>
        <item key="object">ORDER</item>
        <item key="response_text">Request completed successfully</item>
        <item key="attributes">
          <dt_assoc>
            <item key="status">cancelled</item>
            <item key="order_id">14763</item>
            <item key="price">0</item>
            <item key="client_reference"/>
            <item key="items">
              <dt_array>
                <item key="0">
                  <dt_assoc>
                    <item key="status">cancelled</item>
                    <item key="product_item">
                      <dt_assoc>
                        <item key="product_data">
                          <dt_assoc>
                            ...
                          </dt_assoc>
                        </item>
                      ...
                    </dt_assoc>
                  </item>
                ...
              </dt_array>
            </item>
          </dt_assoc>
        </item>
      </dt_array>
    </item>
    <item key="contacts">
      <dt_array/>
    </item>
  </dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

Please see order-create and order-update for additional descriptions of response data.

- DEPRECATE

WARNING: This object was deprecated in version 1.1 of the TPP protocol.

Query-Order permits the client to retrieve data pertaining to an order in ResellOne.net. ResellOne.net will return minimal data for each order item (item id, client reference, price and status).

The format of a typical order query request is as follows:

The format of a typical order query response is as follows:

Please see order-create and order-update for descriptions of response data.

The User object in ResellOne.net is quite generic. A user is an entity with privileges to perform actions on objects that it owns. Resellone.net resellers are users, and so are their customers. Resellers will want to manipulate users for their customers' profiles, for example, in creating orders, managing provisioned products, etc.

Users permit the following actions to be performed:

- check
- create
- update
- delete

Request Parameter Definitions

username	<p><i>Parameter Name:</i> username</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The username of the new user</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the attributes array</p> <p><i>Allowed Value(s):</i> 1 – 256 characters</p>
password	<p><i>Parameter Name:</i> password</p> <p><i>Type:</i> String</p> <p><i>Description:</i> The password for the new user</p> <p><i>Obligation:</i> Required</p> <p><i>Location:</i> This key is permitted in the attributes array</p> <p><i>Allowed Value(s):</i> 3 – 256 characters</p>
description	<p><i>Parameter Name:</i> description</p> <p><i>Type:</i> String</p> <p><i>Description:</i> A free text description or note for the user account</p> <p><i>Obligation:</i> Optional</p> <p><i>Location:</i> This key is permitted in the attributes array</p> <p><i>Allowed Value(s):</i> Maximum 255 characters</p>
users	<p><i>Parameter Name:</i> users</p> <p><i>Type:</i> Associative array</p> <p><i>Description:</i> Used with the "Check" operation. Lists the usernames to check for availability.</p> <p><i>Obligation:</i> Optional</p> <p><i>Location:</i> This key is permitted in the attributes array</p> <p><i>Allowed Value(s):</i> "name" attribute</p>
name	<p><i>Parameter Name:</i> name</p> <p><i>Type:</i> String</p> <p><i>Description:</i> Used with the "Check" operation. The username to check for availability.</p> <p><i>Obligation:</i> Optional</p> <p><i>Location:</i> This key is permitted in the attributes array</p> <p><i>Allowed Value(s):</i> 1 – 256 characters</p>

Response Parameter Definitions

user_id	<i>Parameter Name:</i> user_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the new Resellone.net user. <i>Location:</i> This key is part of the response attributes associative array. <i>Allowed Value(s):</i> Integer
name	<i>Parameter Name:</i> name <i>Type:</i> String <i>Description:</i> Used with the "Check" operation. The username checked for availability. <i>Location:</i> This key is permitted in the attributes associative array <i>Allowed Value(s):</i> 1 – 256 characters
is_available	<i>Parameter Name:</i> is_available <i>Type:</i> Int <i>Description:</i> Used with the "Check" operation. Indicates if the username is available. <i>Location:</i> This key is permitted in the attributes associative array <i>Allowed Value(s):</i> '1' = is available '0' = is not available

description	<i>Parameter Name:</i> description <i>Type:</i> String <i>Description:</i> The user's description. <i>Obligation:</i> Optional <i>Location:</i> This is part of the response attributes associative array. <i>Allowed Value(s):</i>
name	<i>Parameter Name:</i> name <i>Type:</i> String <i>Description:</i> The user name. <i>Obligation:</i> Required <i>Location:</i> This is part of the response attributes associative array. <i>Allowed Value(s):</i>
customer_id	<i>Parameter Name:</i> customer_id <i>Type:</i> String <i>Description:</i> The numerical ID for the customer <i>Obligation:</i> Required <i>Location:</i> This is part of the response attributes associative array. <i>Allowed Value(s):</i>
active	<i>Parameter Name:</i> active <i>Type:</i> Boolean <i>Description:</i> True if user is active <i>Obligation:</i> Optional <i>Location:</i> This is part of the response attributes associative array. <i>Allowed Value(s):</i> True if user is active
contact_set	<i>Parameter Name:</i> contact_set <i>Type:</i> String <i>Description:</i> A set of typed contact links belonging to the user <i>Obligation:</i> Required <i>Location:</i> This is part of the response attributes associative array. <i>Allowed Value(s):</i>

The Create Operation

User creation is very simple. All that's required is the user's username and password.

Request Example (PERL)

Request Example (XML)

```

<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>

```

```

<data_block>
  <dt_assoc>
    <item key='attributes'>
      <dt_assoc>
        <item key='password'>luda</item>
        <item key='username'>user-1061928949449</item>
      </dt_assoc>
    </item>
    <item key='object'>user</item>
    <item key='requestor'>
      <dt_assoc>
        <item key='password'>secret</item>
        <item key='username'>purple</item>
      </dt_assoc>
    </item>
    <item key='action'>create</item>
    <item key='protocol'>TPP</item>
  </dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

```

{
  protocol => 'TPP',
  action => 'create:reply',
  object => 'user',
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
  attributes => {
    user_id => '',
  }
}

```

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='is_success'>1</item>
        <item key='session_id'>OSRS-5472199</item>
        <item key='protocol'>TPP</item>
        <item key='object'>USER</item>
        <item key='response_text'>Request completed successfully</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='user_id'>7591</item>
            <item key='description' />
            <item key='password_last_updated' />
            <item key='active'>1</item>
            <item key='reseller'>25742</item>
            <item key='username'>user-1061928949449</item>
            <item key='response_text' />
            <item key='password_change_next_logon'>0</item>
            <item key='response_code'>200</item>
            <item key='primary_group'>18003</item>
            <item key='last_ip' />
            <item key='password_change_allowed'>0</item>
          </dt_assoc>
        </item>
        <item key='response_code'>200</item>
        <item key='action'>CREATE:REPLY</item>
      </dt_assoc>
    </data_block>
  </body>

```

```
</OPS_envelope>
```

ResellOne.net permits the modification of user objects. One or more users may be updated with this operation. This operation will only work with TPP version 1.2 or greater.

```
{
  'protocol' => 'TPP',
  'object' => 'user',
  'version' => '1.3',
  'action' => 'update',
  'requestor' => {
    'password' => 'secret',
    'username' => 'myresellerusername'
  },
  'attributes' => {
    'contacts' => [
      {
        'contact_id' => '227041'
      }
    ],
    'users' => [
      {
        'password' => 'aokfME',
        'contact_set' => {
          'tech' => '0',
          'billing' => '0'
        },
        'user_id' => '61528',
        'username' => 'user-1082488216749'
      }
    ]
  }
};
```

```
<?xml version='1.0' encoding='UTF-8'?>
<OPS_envelope>
<header>
  <version>0.9</version>
</header>
<body>
  <data_block>
    <dt_assoc>
      <item key="version">1.3</item>
      <item key="attributes">
        <dt_assoc>
          <item key="contacts">
            <dt_array>
              <item key="0">
                <dt_assoc>
                  <item key="contact_id">227041</item>
                </dt_assoc>
              </item>
            </dt_array>
          </item>
          <item key="users">
            <dt_array>
              <item key="0">
                <dt_assoc>
                  <item key="user_id">61528</item>
                  <item key="password">aokfME</item>
                  <item key="contact_set">
                    <dt_assoc>
                      <item key="tech">0</item>
                    </dt_assoc>
                  </item>
                </dt_assoc>
              </item>
            </dt_array>
          </item>
        </dt_assoc>
      </item>
    </dt_assoc>
  </data_block>
</body>
</OPS_envelope>
```

```

        <item key="billing">0</item>
      </dt_assoc>
    </item>
    <item key="username">user-1082488216749</item>
  </dt_assoc>
</item>
</dt_array>
</item>
</dt_assoc>
</item>
<item key="object">user</item>
<item key="requestor">
  <dt_assoc>
    <item key="password">secret</item>
    <item key="username">myresellerusername</item>
  </dt_assoc>
</item>
<item key="action">update</item>
<item key="protocol">TPP</item>
</dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

The format of a typical user update response is as follows. A separate response text/code is given for each user you attempt to update.

```

{
  'protocol' => 'TPP',
  'object' => 'USER',
  'session_id' => 'OSRS-3391250',
  'response_text' => 'Request completed successfully',
  'version' => '1.3.0',
  'action' => 'UPDATE:REPLY',
  'response_code' => '200',
  'attributes' => {},
  'is_success' => '1'
};

```

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<OPS_envelope>
<header>
  <version>0.9</version>
</header>
<body>
  <data_block>
    <dt_assoc>
      <item key="is_success">1</item>
      <item key="session_id">OSRS-3391250</item>
      <item key="protocol">TPP</item>
      <item key="object">USER</item>
      <item key="response_text">Request completed successfully</item>
      <item key="attributes">
        <dt_assoc/>
      </item>
      <item key="response_code">200</item>
      <item key="version">1.3.0</item>
      <item key="action">UPDATE:REPLY</item>
    </dt_assoc>
  </data_block>
</body>
</OPS_envelope>

```

The Update Operation – DEPRECATE

WARNING: The following version of the User Update Operation works only with TPP version 1.1. See above for the new version of this operation, which works with TPP version 1.2 and greater.

ResellOne.net permits the modification of user objects.

The format of a typical user update request is as follows:

The format of a typical user update response is as follows:

The Check Operation

The Check operation allows you to determine if a user ID is available in the ResellOne.net Provisioning System (RPS) (i.e. not already taken). This command is useful as all usernames must be unique.

The Delete Operation

The Delete operation allows one or more users to be deleted. Users are identified by their **user_id**.

The format of a typical user delete request is as follows:

```

    .
    .
    .
    { user_id => ''}
  ]
}

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='attributes'>
          <dt_assoc>
            <item key='users'>
              <dt_array>
                <item key='0'>
                  <dt_assoc>
                    <item key='user_id'>7591</item>
                  </dt_assoc>
                </item>
              </dt_array>
            </item>
          </dt_assoc>
        </item>
        <item key='object'>user</item>
        <item key='requestor'>
          <dt_assoc>
            <item key='password'>secret</item>
            <item key='username'>purple</item>
          </dt_assoc>
        </item>
        <item key='action'>delete</item>
        <item key='protocol'>TPP</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

The format of a typical user delete response is as follows. For each user deletion that was requested, a separate response text/code is returned.

```

{
  protocol => 'TPP',
  action => 'delete:reply',
  object => 'user',
  session_id => xxxx
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
  attributes => {
    users => [
      {
        response_text => ' ',
        response_code => xxx,
        user_id => '',
      },

      {
        response_text => ' ',
        response_code => xxx,
        user_id => '',
      }
    ]
  }
}

```

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='is_success'>1</item>
        <item key='session_id'>OSRS-5472202</item>
        <item key='protocol'>TPP</item>
        <item key='object'>USER</item>
        <item key='response_text'>Request completed successfully</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='users'>
              <dt_assoc>
                <item key='user_id'>7591</item>
                <item key='response_text' />
                <item key='response_code'>200</item>
              </dt_assoc>
            </item>
          </dt_assoc>
        </item>
        <item key='response_code'>200</item>
        <item key='action'>DELETE:REPLY</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>

```

The Recover Password Operation

ResellOne.net allows the password for a user object to be "recovered".

The format of a typical user recover password request is as follows. One or more users may specified in this request.

The format of a typical user password recover response is as follows. A separate response text/code is given for each username you request a password recovery on.

```
                .PASSWORD</item>
<item key='response_text'>Request completed successfully</item>
<item key='attributes'>
  <dt_assoc>
    <item key='user_id'>7592</item>
    <item key='recipient_email'>jmith@ResellOne.net</item>
    <item key='response_text'>Request completed successfully</item>
    <item key='response_code'>200</item>
  </dt_assoc>
</item>
<item key='response_code'>200</item>
<item key='action'>RECOVER:REPLY</item>
</dt_assoc>
</data_block>
</body>
</OPS_envelope>
```

WARNING: This object was deprecated in version 1.1 of the TPP protocol.

ResellOne.net permits the querying of user objects.

The format of a typical query user request:

```
{
  protocol => 'TPP',
  action => 'query',
  object => 'user',
  requestor => {
    username => '',
  },
  attributes => {
    username => '',
  },
}
```

The format of a typical user query request:

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='version'>1.1</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='username'>user-1061929190162</item>
          </dt_assoc>
        </item>
        <item key='object'>user</item>
        <item key='requestor'>
          <dt_assoc>
            <item key='password'>secret</item>
            <item key='username'>purple</item>
          </dt_assoc>
        </item>
        <item key='action'>query</item>
        <item key='protocol'>TPP</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

The format of a typical query user response:

```
{
  protocol => 'TPP',
  action => 'query:reply',
  object => 'user',
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
  attributes => {
    id => '',
    description => '',
  },
}
```

The format of a typical query user response:

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='is_success'>1</item>
        <item key='session_id'>OSRS-5472209</item>
        <item key='protocol'>TPP</item>
        <item key='object'>USER</item>
        <item key='response_text'>Request completed successfully</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='user_id'>7593</item>
            <item key='description' />
            <item key='password_last_updated'>26-Aug-2003 16:16:10</item>
            <item key='active'>1</item>
            <item key='reseller'>25742</item>
            <item key='username'>user-1061929190162</item>
            <item key='response_text' />
            <item key='id'>7593</item>
            <item key='password_change_next_logon'>0</item>
            <item key='response_code'>200</item>
            <item key='primary_group'>18003</item>
            <item key='last_ip' />
            <item key='password_change_allowed'>1</item>
          </dt_assoc>
        </item>
        <item key='response_code'>200</item>
        <item key='action'>QUERY:REPLY</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

The Contact object is an improvement over the previous ResellOne.net domain contact, which was treated more as an attribute of the domain, rather than an independent object with a relation to the domain. As an independent object, the contact can be independently updated and the reseller can trust that this update will be propagated to all services where the particular contact is referenced.

Contacts permit the following actions to be performed:

- create
- update
- delete
- transfer (not yet available)

first_name	<i>Parameter Name:</i> first_name <i>Type:</i> String <i>Description:</i> The contact's first name. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 64
last_name	<i>Parameter Name:</i> last_name <i>Type:</i> String <i>Description:</i> The contact's last name. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 64
org_name	<i>Parameter Name:</i> org_name <i>Type:</i> String <i>Description:</i> The name of the contact's organization <i>Obligation:</i> Optional. <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 64
title	<i>Parameter Name:</i> title <i>Type:</i> String <i>Description:</i> The title of the contact (e.g. "Mr.", "Ms."). <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 64
Address1	<i>Parameter Name:</i> address1 <i>Type:</i> String <i>Description:</i> The first line of the contact's postal address. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 100
Address2	<i>Parameter Name:</i> address2 <i>Type:</i> String <i>Description:</i> The second line of the contact's postal address. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 100

Address3	<p><i>Parameter Name:</i> address3 <i>Type:</i> String <i>Description:</i> The third line of the contact's postal address. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 100</p>
city	<p><i>Parameter Name:</i> city <i>Type:</i> String <i>Description:</i> The postal address city. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 64</p>
state	<p><i>Parameter Name:</i> state <i>Type:</i> String <i>Description:</i> The postal state. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 32</p>
postal_code	<p><i>Parameter Name:</i> postal_code <i>Type:</i> String <i>Description:</i> The address postal/zip code. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 32</p>
country	<p><i>Parameter Name:</i> country <i>Type:</i> String <i>Description:</i> The postal country. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> 2-character ISO country code.</p>
phone	<p><i>Parameter Name:</i> phone <i>Type:</i> String <i>Description:</i> The contact's telephone number. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Preferably in the format "+999.1111x2222". Where "999" is the country's telephone code (e.g. "1" for North America, "44" for UK, etc.), "1111" is the actual phone number and "x2222" is an optional extension, up to 4 digits. Maximum length: 20</p>
fax	<p><i>Parameter Name:</i> fax <i>Type:</i> String <i>Description:</i> The contact's fax number. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Same format as the phone number above.</p>
email	<p><i>Parameter Name:</i> email <i>Type:</i> String <i>Description:</i> The contact's e-mail address. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 255</p>

url	<i>Parameter Name:</i> url <i>Type:</i> String <i>Description:</i> The contact's URL (i.e. web address). <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Maximum length: 255
duns	<i>Parameter Name:</i> duns <i>Type:</i> String <i>Description:</i> Numerical identifier for US locations. <i>Obligation:</i> Optional <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> A nine-digit number
Contact_id	<i>Parameter Name:</i> contact_id <i>Type:</i> Integer <i>Description:</i> The contact's unique ID. <i>Obligation:</i> Required if contact set provided <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i>

The create operation permits the reseller to create contacts

The format of a typical contact create request is as follows:

```

{
  protocol => 'TPP',
  version => '1.1'
  action => 'create',
  object => 'contact',
  requestor => {
    username => '',
  },
  attributes => {
    username => '',
    user_id => '',
    contacts => [
      {
        first_name => '',
        last_name => '',
        title => '',
        org_name => '',
        address1 => '',
        address2 => '',
        address3 => '',
        city => '',
        state => '',
        postal_code => '',
        country => '',
        phone => '',
        fax => '',
        email => '',
        url => '',
        duns => '',
      }
    ]
  },
}

```

For a description of the values for each field, please see the standard list of contact data fields described earlier in this document.

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='version'>1.1</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='user_id'>5577</item>
            <item key='contacts'>
              <dt_array>
                <item key='0'>
                  <dt_assoc>
                    <item key='address3'>Address 3</item>
                    <item key='state'>NY</item>
                    <item key='duns' />
                    <item key='address2'>Address2</item>
                    <item key='url'>www.ResellOne.net</item>
                    <item key='address1'>Address 1</item>
                    <item key='country'>US</item>
                    <item key='fax'>+1.4165350124</item>
                    <item key='org_name'>Orgname</item>
                    <item key='city'>City</item>
                    <item key='first_name'>Firstname</item>
                    <item key='phone'>+1.4165350125</item>
                    <item key='email'>jmith@ResellOne.net</item>
                    <item key='title'>Title</item>
                    <item key='postal_code'>m2r2a6</item>
                    <item key='last_name'>Lastname</item>
                  </dt_assoc>
                </item>
              </dt_array>
            </item>
            <item key='username'>user-1061928949449</item>
          </dt_assoc>
        </item>
        <item key='object'>contact</item>
        <item key='requestor'>
          <dt_assoc>
            <item key='password'>secret</item>
            <item key='username'>purple</item>
          </dt_assoc>
        </item>
        <item key='action'>create</item>
        <item key='protocol'>TPP</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

Format of the basic contact create response:

```
{
  protocol => 'TPP',
  action => 'update:reply',
  object => 'contact',
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
  attributes => {
    contacts => [
```

```

        {
        contact_id => '',
        response_code => xxx,
        response_text => '',
        }
    ]
}

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='protocol'>TPP</item>
        <item key='action'>CREATE:REPLY</item>
        <item key='object'>CONTACT</item>
        <item key='is_success'>1</item>
        <item key='session_id'>OSRS-5472200</item>
        <item key='response_code'>200</item>
        <item key='response_text'>Request completed successfully</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='user_id'>5577</item>
            <item key='contacts'>
              <dt_array>
                <item key='0'>
                  <dt_assoc>
                    <item key='country'>US</item>
                    <item key='last_name'>Lastname</item>
                    <item key='postal_code'>m2r2a6</item>
                    <item key='client_reference' />
                    <item key='duns'>www.ResellOne.net</item>
                    <item key='phone'>+1.4165350125</item>
                    <item key='org_name'>Orgname</item>
                    <item key='first_name'>Firstname</item>
                    <item key='state'>NY</item>
                    <item key='title'>Title</item>
                    <item key='address1'>Address 1</item>
                    <item key='address2'>Address2</item>
                    <item key='address3'>Address 3</item>
                    <item key='city'>City</item>
                    <item key='fax'>+1.4165350124</item>
                    <item key='url'>www.ResellOne.net</item>
                    <item key='contact_id'>18339</item>
                    <item key='response_text' />
                    <item key='response_code'>200</item>
                    <item key='email'>jmith@ResellOne.net</item>
                  </dt_assoc>
                </item>
              </dt_array>
            </item>
          </dt_assoc>
        </item>
      </data_block>
    </body>
  </OPS_envelope>

```

The update operation permits the reseller to modify contacts that were created by their customers. All contact data can be modified.

The format of a typical contact update request is as follows:

```

{
  protocol => 'TPP',
  action => 'update',
  object => 'contact',
  requestor => {
    username => '',
  },
  attributes => {
    contact_id => '',
    first_name => '',
    last_name => '',
    title => '',
    org_name => '',
    address1 => '',
    address2 => '',
    address3 => '',
    city => '',
    state => '',
    postal_code => '',
    country => '',
    phone => '',
    fax => '',
    email => '',
    url => '',
  },
}

```

For a description of the values for each field, please see the standard list of contact data fields described earlier in this document.

The format of the basic contact update response is as follows:

```

{
  protocol => 'TPP',
  action => 'update:reply',
  object => 'contact',
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
}

```

The Delete operation allows one or more contacts for a specific user to be deleted.

The format of a typical contact delete request is as follows:

```

{
  protocol => 'TPP',
  action => 'delete',
  object => 'contact',
  requestor => {
    username => '',
  },
  attributes => {
    user_id => '',
    contacts => [
      { contact_id => ''},
    ],
  },
}

```

```

    ]
  }
}

```

The format of a typical contact delete response is as follows. For each contact that is deleted for the user, a separate response text/code is returned.

```

{
  protocol => 'TPP',
  action => 'delete:reply',
  object => 'user',
  session_id => xxxx
  is_success => 1|0,
  response_code => xxx,
  response_text => '',
  attributes => {
    contacts => [
      {
        response_text => ' ',
        response_code => xxx,
        contact_id => '',
      },

      {
        response_text => ' ',
        response_code => xxx,
        contact_id => '',
      }
    ]
  }
}

```

WARNING: This object was deprecated in version 1.1 of the TPP protocol.

ResellOne.net Contacts can be queried using their ID. This will have the server return the standard set of contact data.

The format of a typical contact query request is as follows:

```

{
  protocol => 'TPP',
  action => 'query',
  object => 'contact',
  requestor => {
    username => '',
  },
  attributes => {
    id => '',
    username => '',
    password => ''
  },
}

```

The format of the basic contact update response is as follows:

```
{
  protocol => 'TPP',
  action => 'query:reply',
  object => 'contact',
  is_success => 1|0,
  attributes => {
    id => '',
    first_name => '',
    last_name => '',
    org_name => '',
    address1 => '',
    address2 => '',
    address3 => '',
    city => '',
    state => '',
    postal_code => '',
    country => '',
    phone => '',
    fax => '',
    email => '',
    url => ''
  },
}
```

For a description of the values for each field, please see the standard list of contact data fields described earlier in this document.

The Session Object

The session object is used to establish a persistent session for a user in the system. Please note that is not the same as the socket connection established by the reseller with the ResellOne.net servers. Multiple user sessions can peacefully coexist in a single reseller connection. All that is required is to properly manage the sessions IDs for each interacting user.

Sessions permit the following actions to be performed:

- create
- update (not yet available)
- delete (not yet available)

username	<i>Parameter Name:</i> username <i>Type:</i> String <i>Description:</i> The username of the user. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i>
password	<i>Parameter Name:</i> password <i>Type:</i> String <i>Description:</i> The password for the user. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i>
session_id	<i>Parameter Name:</i> session_id <i>Type:</i> String <i>Description:</i> The ID of the new ResellOne.net session. <i>Obligation:</i> Required <i>Location:</i> This key is located in the response message <i>Allowed Value(s):</i>
user_id	<i>Parameter Name:</i> user_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the user which requests the operation <i>Obligation:</i> Required <i>Location:</i> This key is located in the response attributes associative array. <i>Allowed Value(s):</i>
requestor_id	<i>Parameter Name:</i> requestor_id <i>Type:</i> Integer <i>Description:</i> The integer ID of the owner of the user <i>Obligation:</i> Required <i>Location:</i> This key is located in the response attributes associative array. <i>Allowed Value(s):</i>

This lets you create a session. The server validates the username and password provided. If both match up, then a session is created. Otherwise, an error response is returned.

Request Example (PERL)

The format of a typical session create request is as follows:

```
{
  protocol => 'TPP',
  action => 'create',
  object => 'session',
  requestor => {
    username => '',
  },
  attributes => {
    username => '',
    password => '',
  },
}
```

Request Example (XML)

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='version'>1.1</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='password'>${conf_tulips_user_password}</item>
            <item key='username'>purple_user</item>
          </dt_assoc>
        </item>
        <item key='object'>session</item>
        <item key='requestor'>
          <dt_assoc>
            <item key='password'>secret</item>
            <item key='username'>purple</item>
          </dt_assoc>
        </item>
        <item key='action'>create</item>
        <item key='protocol'>TPP</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

The format of a typical session create response is as follows:

```
{
  protocol => 'TPP',
  is_success => '1',
  action => 'create:reply',
  object => 'session',
  response_code => '200',
  response_text => 'Request completed successfully',
  session_id => 'OSRS-31933',
  requestor_id => '5209',
  attributes => {
    user_id => '',
  },
}
```

```
    },  
  }  
}
```

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>  
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>  
<OPS_envelope>  
  <header>  
    <version>0.9</version>  
  </header>  
  <body>  
    <data_block>  
      <dt_assoc>  
        <item key='is_success'>1</item>  
        <item key='session_id'>OSRS-5472184</item>  
        <item key='protocol'>TPP</item>  
        <item key='object'>SESSION</item>  
        <item key='requestor_id'>7418</item>  
        <item key='response_text'>Request completed successfully</item>  
        <item key='attributes'>  
          <dt_assoc>  
            <item key='user_id'>7436</item>  
          </dt_assoc>  
        </item>  
        <item key='response_code'>200</item>  
        <item key='action'>CREATE:REPLY</item>  
      </dt_assoc>  
    </data_block>  
  </body>  
</OPS_envelope>
```

The Query Object

This type of request fulfils the need for a simple API query for each of the existing objects, as well as the need for a more complex and advanced reporting mechanism.

query_name	<p><i>Parameter Name:</i> query_name <i>Type:</i> String <i>Description:</i> Name of query or report available in the system. <i>Obligation:</i> Required <i>Location:</i> This key is located in the 'attributes' associative array. <i>Allowed Value(s):</i> One of 'contacts.by_user_id', 'inventory_item.by_id', 'inventory_items.created.by_user_id', or 'queued.order_items'</p>
start_index	<p><i>Parameter Name:</i> start_index <i>Type:</i> Int <i>Description:</i> Tells the system which record you want to start with in the result list. This MUST be '1' the first time you run the query. On subsequent runs, you can set this to which record you want the result set to start with. For example, if your page size is set to 45, the results array will show 45 records starting at the 'start_index' record. <i>Obligation:</i> Optional <i>Location:</i> Within the 'attributes' associative array <i>Allowed Value(s):</i> Int</p>
page_size	<p><i>Parameter Name:</i> page_size <i>Type:</i> Int <i>Description:</i> Tells the system how many records to return to you in the list. If you do not include this parameter, a default page size of 50 will be used. The maximum page size allowed is 50. <i>Obligation:</i> Optional <i>Location:</i> Within the 'attributes' associative array <i>Allowed Value(s):</i> Int</p>
conditions	<p><i>Parameter Name:</i> conditions <i>Type:</i> Array of associative arrays <i>Description:</i> Parameters for the query to be executed. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Values are variable; see below</p>
type	<p><i>Parameter Name:</i> type <i>Type:</i> String <i>Description:</i> Type of condition <i>Obligation:</i> Required <i>Location:</i> This key is located in the "conditions" associative array. <i>Allowed Value(s):</i> One of 'simple' or 'link'</p>
operand	<p><i>Parameter Name:</i> operand <i>Type:</i> String <i>Description:</i> Indicates which operand to use, e.g. "eq", "between", etc. <i>Obligation:</i> Required <i>Location:</i> This key is located in the "conditions" associative array. <i>Allowed Value(s):</i> eq = equal to the given value neq = not equal to the given value leq = less than or equal to the given value geq = greater than or equal to the given value between = between the given values</p>

field

Parameter Name: field
Type: String
Description: Name of field being used in the query, e.g. 'description', 'state', 'service', 'username', 'submission_date', 'completion_date', 'acknowledged'
Obligation: Required
Location: This key is located in the "conditions" array of associative arrays.
Allowed Value(s): Any valid field name.

link

Parameter Name: link
Type: String
Description: Determines how the conditions are used together
Obligation: Required
Location: This key is located in the "conditions" associative array.
Allowed Value(s): One of 'and' or 'or'

Query Contacts by User ID

This query lets you search for all contacts linked to a particular user.

- PERL

Response Example - PERL

```
'action' => 'EXECUTE:REPLY'
'object' => 'QUERY',
'response_code' => '200',
'response_text' => 'Request completed successfully',
'attributes' => {
  'result' => [
    {
      'last_updated' => '30-Jun-2004 13:45:11',
      'country' => 'CA',
      'last_name' => 'Smith',
      'postal_code' => 'M6K3M1',
      'client_reference' => undef,
      'duns' => undef,
      'phone' => '+1.4165350123',
      'org_name' => undef,
      'user_id' => '71047',
      'first_name' => 'Vince',
      'state' => 'ON',
      'title' => undef,
      'address1' => '96 Mowat Ave.',
      'expiry_date' => undef,
      'address2' => undef,
      'address3' => undef,
      'city' => 'Toronto',
      'fax' => undef,
      'contact_id' => '358592',
      'parent_id' => '0',
      'email' => 'support@resellone.net'
    },
    {
      'last_updated' => '30-Jun-2004 13:46:14',
      'country' => 'CA',
      'last_name' => 'Jones',
      'postal_code' => 'M6K3M1',
      'client_reference' => undef,
      'duns' => undef,

```

```
'phone' => '+1.4165350123',
'org_name' => 'Resellone.net',
'user_id' => '71047',
'first_name' => 'Vince',
'state' => 'ON',
'title' => undef,
'address1' => '96 Mowat Ave.',
'expiry_date' => undef,
'address2' => undef,
'address3' => undef,
'city' => 'Toronto',
'fax' => undef,
'contact_id' => '358595',
'parent_id' => '0',
'email' => 'support@resellone.net'
}
],
'result_control' => {
  'start_index' => '0',
  'report_instance_id' => '380421',
  'record_count' => '2',
  'page_size' => '50'
}
},
};
```

Request Example – XML

Response Example – XML

This query lets you search for all inventory items of a particular type.

Request Example

Response Example

This query lets you search for all inventory items created by a particular user.

Request Example

Response Example

This query enables you to retrieve an inventory item ID using the 'description' parameter, using multiple conditions. The returned value for 'description' depends on the service queried. For Digital Certificates, DNS, and Email Defense, 'description' is a domain name; for Email, 'description' is an email address; and for Website Builder, 'description' is the Website Builder account name.

The parameters 'state' (state of the service sold item) and 'service' (type of the service) are optional and can be used to narrow down the list of items that you want to retrieve. For example, you can query for 'active' (state) 'email' (service) items.

Request Example, PERL

```
        'like' => '*erdomai*'
      },
      'type' => 'simple',
      'field' => 'description'
    },
    {
      'link' => 'and',
      'type' => 'link'
    },
    {
      'operand' => {
        'eq' => 'active'
      },
      'type' => 'simple',
      'field' => 'state'
    },
    {
      'link' => 'and',
      'type' => 'link'
```

```

    },
    {
      'operand' => {
        'eq' => 'email'
      },
      'type' => 'simple',
      'field' => 'service'
    }
  ]
}
};

```

Response Example, PERL

```

{
  'protocol' => 'TPP',
  'object' => 'query',
  'session_id' => 'OSRS-676178',
  'response_text' => 'Request completed successfully',
  'version' => '1.3.0',
  'action' => 'execute:reply',
  'attributes' => {
    'result_control' => {
      'page_size' => '10',
      'record_count' => '3',
      'start_index' => '1',
      'report_instance_id' => '625'
    },
    'result' => [
      {
        'original_inventory_item_id' => '0',
        'object_type' => 'email',
        'description' => '2345@testerdomain.com',
        'service' => 'email',
        'expiry_date' => '',
        'inventory_item_id' => '460372',
        'state' => 'active',
        'creation_date' => '28-Apr-2004 16:09:26',
        'renewal_ctl_mask' => '1',
        'contact_set' => {},
        'user_id' => '13634',
        'start_date' => ''
      },
      {
        'original_inventory_item_id' => '0',
        'object_type' => 'email',
        'description' => '123a@testerdomain.com',
        'service' => 'email',
        'expiry_date' => '',
        'inventory_item_id' => '460373',
        'state' => 'active',
        'creation_date' => '28-Apr-2004 16:19:02',
        'renewal_ctl_mask' => '1',
        'contact_set' => {},
        'user_id' => '13634',
        'start_date' => ''
      },
      {
        'original_inventory_item_id' => '0',
        'object_type' => 'email',
        'description' => '123b@testerdomain.com',
        'service' => 'email',
        'expiry_date' => '',
        'inventory_item_id' => '460378',
        'state' => 'active',
        'creation_date' => '30-Apr-2004 08:48:00',
        'renewal_ctl_mask' => '1',
        'contact_set' => {},
        'user_id' => '13634',
        'start_date' => ''
      }
    ]
  },
  'response_code' => '200',
}

```

```
'is_success' => '1'
};
```

Request Example, XML

```
<?xml version='1.0' encoding="UTF-8" standalone="no" ?>
<OPS_envelope>
<header>
  <version>0.9</version>
</header>
<body>
  <data_block>
    <dt_assoc>
      <item key="version">1.3</item>
      <item key="protocol">TPP</item>
      <item key="object">QUERY</item>
      <item key="action">EXECUTE</item>
      <item key="requestor">
        <dt_assoc>
          <item key="username">user5000</item>
        </dt_assoc>
      </item>
      <item key="attributes">
        <dt_assoc>
          <item key="start_index">1</item>
          <item key="query_name">inventory_items.by_description</item>
          <item key="conditions">
            <dt_array>
              <item key="0">
                <dt_assoc>
                  <item key="operand">
                    <dt_assoc>
                      <item key="like">*erdomai*</item>
                    </dt_assoc>
                  </item>
                  <item key="field">description</item>
                  <item key="type">simple</item>
                </dt_assoc>
              </item>
              <item key="1">
                <dt_assoc>
                  <item key="type">link</item>
                  <item key="link">and</item>
                </dt_assoc>
              </item>
              <item key="2">
                <dt_assoc>
                  <item key="operand">
                    <dt_assoc>
                      <item key="eq">active</item>
                    </dt_assoc>
                  </item>
                  <item key="field">state</item>
                  <item key="type">simple</item>
                </dt_assoc>
              </item>
              <item key="3">
                <dt_assoc>
                  <item key="type">link</item>
                  <item key="link">and</item>
                </dt_assoc>
              </item>
              <item key="4">
                <dt_assoc>
                  <item key="operand">
                    <dt_assoc>
                      <item key="eq">email</item>
                    </dt_assoc>
                  </item>
                  <item key="field">service</item>
                  <item key="type">simple</item>
                </dt_assoc>
              </item>
            </dt_array>
          </item>
        </dt_assoc>
      </item>
    </dt_assoc>
  </data_block>
</body>
</OPS_envelope>
```

```

        </item>
        <item key="report_instance_id"/>
        <item key="page_size">10</item>
    </dt_assoc>
</item>
</dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

Response Example, XML

```

<?xml version='1.0' encoding='UTF-8'?>
<OPS_envelope>
<header>
    <version>0.9</version>
</header>
<body>
    <data_block>
        <dt_assoc>
            <item key="response_code">200</item>
            <item key="session_id">OSRS-676178</item>
            <item key="action">execute:reply</item>
            <item key="object">query</item>
            <item key="attributes">
                <dt_assoc>
                    <item key="result">
                        <dt_array>
                            <item key="0">
                                <dt_assoc>
                                    <item key="state">active</item>
                                    <item key="creation_date">28-Apr-2004 16:09:26</item>
                                    <item key="renewal_ctl_mask">1</item>
                                    <item key="original_inventory_item_id">0</item>
                                    <item key="contact_set">
                                        <dt_assoc/>
                                    </item>
                                    <item key="expiry_date"/>
                                    <item key="start_date"/>
                                    <item key="service">email</item>
                                    <item key="inventory_item_id">460372</item>
                                    <item key="object_type">email</item>
                                    <item key="description">2345@testerdomain.com</item>
                                    <item key="user_id">13634</item>
                                </dt_assoc>
                            </item>
                            <item key="1">
                                <dt_assoc>
                                    <item key="state">active</item>
                                    <item key="creation_date">28-Apr-2004 16:19:02</item>
                                    <item key="renewal_ctl_mask">1</item>
                                    <item key="original_inventory_item_id">0</item>
                                    <item key="contact_set">
                                        <dt_assoc/>
                                    </item>
                                    <item key="expiry_date"/>
                                    <item key="start_date"/>
                                    <item key="service">email</item>
                                    <item key="inventory_item_id">460373</item>
                                    <item key="object_type">email</item>
                                    <item key="description">123a@testerdomain.com</item>
                                    <item key="user_id">13634</item>
                                </dt_assoc>
                            </item>
                            <item key="2">
                                <dt_assoc>
                                    <item key="state">active</item>
                                    <item key="creation_date">30-Apr-2004 08:48:00</item>
                                    <item key="renewal_ctl_mask">1</item>
                                    <item key="original_inventory_item_id">0</item>
                                    <item key="contact_set">
                                        <dt_assoc/>
                                    </item>
                                    <item key="expiry_date"/>
                                </dt_assoc>
                            </item>
                        </dt_array>
                    </item>
                </dt_assoc>
            </item>
        </dt_assoc>
    </data_block>
</body>
</OPS_envelope>

```

```

        <item key="start_date"/>
        <item key="service">email</item>
        <item key="inventory_item_id">460378</item>
        <item key="object_type">email</item>
        <item key="description">123b@testerdomain com</item>
        <item key="user_id">13634</item>
    </dt_assoc>
</item>
</dt_array>
</item>
<item key="result_control">
    <dt_assoc>
        <item key="start_index">1</item>
        <item key="report_instance_id">625</item>
        <item key="page_size">10</item>
        <item key="record_count">3</item>
    </dt_assoc>
</item>
</dt_assoc>
</item>
<item key="is_success">1</item>
<item key="version">1.3.0</item>
<item key="protocol">TPP</item>
<item key="response_text">Request completed successfully</item>
</dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

This query enables you to retrieve an end-user's user_id using their username.

Request Example, PERL

```
{
  protocol=> TPP
  version=> 1.4
  action=> execute
  object=> query
  requestor=>
    {
      ipaddress=> 123
      username=> orange
    }
  attributes=>
    {
      query_name=> user.by_credentials
      conditions=>
        [
          {
            operand=>
              {
                eq=> aaaa
              }
            type=> simple
            field=> username
          }
        ]
    }
}
```

Response Example, PERL

```
{
  protocol=> TPP
  version=> 1.4.0
  action=> EXECUTE:REPLY
  object=> QUERY
  is_success=> 1
  response_code=> 200
  response_text=> Request completed successfully
  session_id=> OSRS-12337
  attributes=>
    {
      result=>
        [
          {
            reseller=> 3
            user_id=> 1003
            active=> 1
            contact_set=>
              {
                primary=> 102
              }
            username=> aaaa
          }
        ]
      result_control=>
        {
          start_index=> 0
          report_instance_id=> 0
          page_size=> 0
          record_count=> 0
        }
    }
}
```

Request Example, XML

```
<?xml version='1.0' encoding='UTF-8'?>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='protocol'>TPP</item>
        <item key='version'>1.4</item>
        <item key='action'>execute</item>
        <item key='object'>query</item>
        <item key='requestor'>
          <dt_assoc>
            <item key='ipaddress'>123</item>
            <item key='username'>orange</item>
          </dt_assoc>
        </item>
        <item key='attributes'>
          <dt_assoc>
            <item key='query_name'>user.by_credentials</item>
            <item key='conditions'>
              <dt_array>
                <item key='0'>
                  <dt_assoc>
                    <item key='operand'>
                      <dt_assoc>
                        <item key='eq'>aaaa</item>
                      </dt_assoc>
                    </item>
                    <item key='type'>simple</item>
                    <item key='field'>username</item>
                  </dt_assoc>
                </item>
              </dt_array>
            </item>
          </dt_assoc>
        </item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

Response Example, XML

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<!DOCTYPE OPS_envelope SYSTEM 'ops.dtd'>
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key='protocol'>TPP</item>
        <item key='version'>1.4.0</item>
        <item key='action'>EXECUTE:REPLY</item>
        <item key='object'>QUERY</item>
        <item key='is_success'>1</item>
        <item key='session_id'>OSRS-12337</item>
        <item key='response_code'>200</item>
        <item key='response_text'>Request completed successfully</item>
        <item key='attributes'>
          <dt_assoc>
            <item key='result'>
              <dt_array>
                <item key='0'>
                  <dt_assoc>
                    <item key='reseller'>3</item>
                    <item key='user_id'>1003</item>
                    <item key='contact_set'>
                      <dt_assoc>
                        <item key='primary'>102</item>
                      </dt_assoc>
                    </item>
                    <item key='username'>aaaa</item>
                    <item key='active'>1</item>
                  </dt_assoc>
                </item>
              </dt_array>
            </item>
            <item key='result_control'>
              <dt_assoc>
                <item key='start_index'>0</item>
                <item key='report_instance_id'>0</item>
                <item key='record_count'>0</item>
                <item key='page_size'>0</item>
              </dt_assoc>
            </item>
          </dt_assoc>
        </item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>
```

Note: As of December 15, 2005, this command supports the Digital Certificates, Managed DNS, Email Defense, and Website Builder services. The Email service currently is supported.

Items that were processed asynchronously are placed in a queue. To list the items in the queue, execute the Query Queued Order Items command.

The queue is intended to help Resellers manage the asynchronously processed order items that they need to act on. If you've taken action on an order item, e.g. contacted your customer, you can acknowledge it, which will remove it from the queue summary. See the [Acknowledge Queued Items command](#).

Note:

- The query_name parameter can use either "queued.order_items" or "queued_inventory_item" for its value.
- All conditions are optional and can be used in any combination that is meaningful to you.

Examples

Request example, PERL – Query queued order items

```
{
  'protocol' => 'TPP',
  'version' => '1.4.0',
  'action' => 'EXECUTE',
  'object' => 'QUERY',
  'requestor' => {
    'username' => 'default_user'
  },
  'attributes' => {
    'page_size' => '10',
    'query_name' => 'queued.order_items',
    'start_index' => '1',
    'conditions' => [
      {
        'operand' => {
          'between' => {
            'end' => '02-aug-2005 00:00:00',
            'start' => '01-feb-2005 12:00:00'
          }
        },
        'type' => 'simple',
        'field' => 'submission_date'
      },
      {
        'link' => 'and',
        'type' => 'link'
      },
      {
        'operand' => {
          'between' => {
            'end' => '01-jul-2005 00:00:00',
            'start' => '01-jan-2005 00:00:00'
          }
        },
        'type' => 'simple',
        'field' => 'completion_date'
      },
      {
        'link' => 'and',
        'type' => 'link'
      },
      {
        'operand' => {
```

```

        'eq' => 'email'
    },
    'type' => 'simple',
    'field' => 'service'
},
{
    'link' => 'and',
    'type' => 'link'
},
{
    'operand' => {
        'like' => 'test*.com'
    },
    'type' => 'simple',
    'field' => 'description'
}
]
}
};

```

Response example, PERL – Query queued order items

```

{
    'protocol' => 'TPP',
    'session_id' => 'OSRS-5710738',
    'object' => 'query',
    'version' => '1.4.0',
    'response_code' => '200',
    'is_success' => '1',
    'response_text' => 'Request completed successfully',
    'action' => 'execute:reply',
    'attributes' => {
        'result_control' => {
            'page_size' => '10',
            'record_count' => '9',
            'start_index' => '1',
        },
        'result' => [
            {
                'completion_date' => '28-Jul-2005 17:42:06',
                'acknowledged' => '0',
                'service' => 'email',
                'description' => 'user1@jul28c.com',
                'queued_item_id' => '2411',
                'action' => 'process',
                'object_id' => '3096479',
                'submission_date' => '28-Jul-2005 17:28:52',
                'result' => 'completed'
            },
            {
                'completion_date' => '28-Jul-2005 17:45:03',
                'acknowledged' => '0',
                'service' => 'email',
                'description' => 'user3@jul28b.com',
                'queued_item_id' => '2405',
                'action' => 'process',
                'object_id' => '3096476',
                'submission_date' => '28-Jul-2005 17:20:02',
                'result' => 'completed'
            },
            {
                'completion_date' => '22-Jul-2005 10:30:13',
                'acknowledged' => '0',
                'service' => 'email',
                'description' => 'user1@jul21c.com',
                'queued_item_id' => '774',
                'action' => 'process',
                'object_id' => '3096287',
                'submission_date' => '21-Jul-2005 14:28:18',
                'result' => 'completed'
            }
        ]
    }
}

```

```
};
```

Request example, XML – Query queued order items

```
<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key="protocol">TPP</item>
        <item key="version">1.4.0</item>
        <item key="action">EXECUTE</item>
        <item key="object">QUERY</item>
        <item key="requestor">
          <dt_assoc>
            <item key="username">default_user</item>
          </dt_assoc>
        </item>
        <item key="attributes">
          <dt_assoc>
            <item key="query_name">queued.order_items</item>
            <item key="start_index">1</item>
            <item key="page_size">10</item>
            <item key="conditions">
              <dt_array>
                <item key="0">
                  <dt_assoc>
                    <item key="field">submission_date</item>
                    <item key="operand">
                      <dt_assoc>
                        <item key="between">
                          <dt_assoc>
                            <item key="start">01-feb-2005 12:00:00</item>
                            <item key="end">02-aug-2005 00:00:00</item>
                          </dt_assoc>
                        </item>
                      </dt_assoc>
                    </item>
                    <item key="type">simple</item>
                  </dt_assoc>
                </item>
                <item key="1">
                  <dt_assoc>
                    <item key="link">and</item>
                    <item key="type">link</item>
                  </dt_assoc>
                </item>
                <item key="2">
                  <dt_assoc>
                    <item key="field">completion_date</item>
                    <item key="operand">
                      <dt_assoc>
                        <item key="between">
                          <dt_assoc>
                            <item key="start">01-jan-2005 00:00:00</item>
                            <item key="end">01-jul-2005 00:00:00</item>
                          </dt_assoc>
                        </item>
                      </dt_assoc>
                    </item>
                    <item key="type">simple</item>
                  </dt_assoc>
                </item>
                <item key="3">
                  <dt_assoc>
```

```

        <item key="link">and</item>
        <item key="type">link</item>
    </dt_assoc>
</item>
<item key="4">
    <dt_assoc>
        <item key="operand">
            <dt_assoc>
                <item key="eq">email</item>
            </dt_assoc>
        </item>
        <item key="field">service</item>
        <item key="type">simple</item>
    </dt_assoc>
</item>
<item key="5">
    <dt_assoc>
        <item key="link">and</item>
        <item key="type">link</item>
    </dt_assoc>
</item>
<item key="6">
    <dt_assoc>
        <item key="operand">
            <dt_assoc>
                <item key="like">test*.com</item>
            </dt_assoc>
        </item>
        <item key="field">description</item>
        <item key="type">simple</item>
    </dt_assoc>
    <dt_assoc>
        <item key="operand">
            <dt_assoc>
                <item key="eq">1</item>
            </dt_assoc>
        </item>
        <item key="field">acknowledged</item>
        <item key="type">simple</item>
    </dt_assoc>
</item>
</dt_array>
</item>
</dt_assoc>
</item>
</dt_assoc>
</data_block>
</body>
</OPS_envelope>

```

Response example, XML – Query queued order items

```

<OPS_envelope>
  <header>
    <version>0.9</version>
  </header>
  <body>
    <data_block>
      <dt_assoc>
        <item key="protocol">TPP</item>
        <item key="version">1.4.0</item>
        <item key="action">execute:reply</item>
        <item key="object">query</item>
        <item key="session_id">OSRS-5710738</item>
        <item key="is_success">1</item>
        <item key="response_text">Request completed successfully</item>
        <item key="response_code">200</item>
      </dt_assoc>
    </data_block>
  </body>
</OPS_envelope>

```

```

<item key="attributes">
  <dt_assoc>
    <item key="result_control">
      <dt_assoc>
        <item key="record_count">9</item>
        <item key="page_size">10</item>
        <item key="start_index">1</item>
        <item key="report_instance_id">0</item>
      </dt_assoc>
    </item>
    <item key="result">
      <dt_array>
        <item key="0">
          <dt_assoc>
            <item key="submission_date">28-Jul-2005 17:28:52</item>
            <item key="action">process</item>
            <item key="acknowledged">0</item>
            <item key="completion_date">28-Jul-2005 17:42:06</item>
            <item key="object_id">3096479</item>
            <item key="description">user1@jul28c.com</item>
            <item key="result">completed</item>
            <item key="service">email</item>
            <item key="queued_item_id">2411</item>
          </dt_assoc>
        </item>
        <item key="1">
          <dt_assoc>
            <item key="submission_date">28-Jul-2005 17:20:02</item>
            <item key="action">process</item>
            <item key="acknowledged">0</item>
            <item key="completion_date">28-Jul-2005 17:45:03</item>
            <item key="object_id">3096476</item>
            <item key="description">user3@jul28b.com</item>
            <item key="result">completed</item>
            <item key="service">email</item>
            <item key="queued_item_id">2405</item>
          </dt_assoc>
        </item>
        <item key="2">
          <dt_assoc>
            <item key="submission_date">21-Jul-2005 14:28:18</item>
            <item key="action">process</item>
            <item key="acknowledged">0</item>
            <item key="completion_date">22-Jul-2005 10:30:13</item>
            <item key="object_id">3096287</item>
            <item key="description">user1@jul21c.com</item>
            <item key="result">completed</item>
            <item key="service">email</item>
            <item key="queued_item_id">774</item>
          </dt_assoc>
        </item>
      </dt_array>
    </item>
  </dt_assoc>
</item>
</data_block>
</body>
</OPS_envelope>

```

The Inventory_Item Object

The inventory_item object is used to modify existing order items; it cannot be used to create orders.

This is the general structure for using the inventory_item object. Each of the different services (e.g. DNS, Digital Certificates, Website Builder) use the inventory_item object differently; please refer to the service-specific API documents for details.

inventory_items	<p><i>Parameter Name:</i> inventory_items <i>Type:</i> Array of associative arrays <i>Description:</i> Array which lists inventory items to be acted upon <i>Obligation:</i> Required <i>Location:</i> This key is permitted in the "attributes" associative array. <i>Allowed Value(s):</i> Values are variable; some attributes are listed below.</p>
service	<p><i>Parameter Name:</i> service <i>Type:</i> String <i>Description:</i> The service provisioning the product <i>Obligation:</i> Required <i>Location:</i> Within the "inventory_items" array <i>Allowed Value(s):</i> Valid service, e.g. 'dns', 'email'</p>
inventory_item_id	<p><i>Parameter Name:</i> inventory_item_id <i>Type:</i> Int <i>Description:</i> The ID of the inventory item that you want to modify <i>Obligation:</i> Required <i>Location:</i> Within the "inventory_items" array <i>Allowed Value(s):</i> Valid inventory item ID</p>
expiry_date	<p><i>Parameter Name:</i> expiry_date <i>Type:</i> String <i>Description:</i> Allows you to set, change, or remove an expiry date for an inventory_item (i.e. a processed order item). Some services, i.e. Digital Certificates, are sold for a fixed term and therefore do not support expiration dates. To set or change an expiry date The date specified must be a future date, in the following format: - the day must be 2 characters, i.e. 01-31 - the month must be the first three characters of the month name, i.e. Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec - the year must be in four character format, e.g. 2005 - e.g. 27-Aug-2008 To remove an expiry date: Submit '-1' <i>Obligation:</i> Optional <i>Location:</i> Within the "inventory_items" array <i>Allowed Value(s):</i> Future date in valid format, e.g. '27-Aug-2008', or '-1' if removing an expiry date.</p>
renewal_msg_flag	<p><i>Parameter Name:</i> renewal_msg_flag <i>Type:</i> Int <i>Description:</i> This flag activates or deactivates the renewal message sent to end users to notify them that their service is about to expire. If activated, renewal reminders are sent 10 days prior to expiry date. This flag can only be applied to active services; it cannot be applied to services that are expired or in suspended state. <i>Obligation:</i> Optional <i>Location:</i> Within the "inventory_items" array <i>Allowed Value(s):</i> 1 0 where 1=activate renewal reminder, 0=deactivate renewal reminder</p>

The update operation for `inventory_item` permits the modification of provisioned services, i.e. completed orders and order items. Updates vary according to service (e.g. DNS, Email Defense, Website Builder); please refer to the API specification for each service for details.

Request Example, PERL

The format of a typical `inventory_item` update request:

```
'expiry_date' => '01-oct-2005'
```

Response Example, PERL

The Suspend Operation

The suspend operation for inventory_item permits the suspension of provisioned services, i.e. completed orders and order items. This operation is not available for all services. As of October 5, 2004, this operation is not available for either Digital Certificates, or Email Defense. Please refer to the API specification for each service for the exact structure.

The format of a typical inventory_item suspend request:

Request Example, PERL

Response Example, PERL

Request Example, XML

Response Example, XML

The Activate Operation

The activate operation for inventory_item permits the reactivation of provisioned services that have been suspended.

If the "expiry_date" parameter is not submitted or specified, one year will be added to the service, from the activation date.

This operation is not available for all services. As of October 5, 2004, this operation is not available for either Digital Certificates, or Email Defense. Please refer to the API specification for each service for the exact structure.

Request Example, PERL

Response Example, PERL

Request Example, XML

Response Example, XML

The Delete Operation

The delete operation for `inventory_items` permits the deletion of provisioned services, i.e. completed orders and order items. The format of an `inventory_item` delete request varies according to service (e.g. DNS, Email Defense, Website Builder); please refer to the API specification for each service for the exact structure.

Request Example, PERL

The format of a typical `inventory_item` delete request:

Response Example, PERL

The Product Object – DEPRECATE

WARNING: This object was deprecated in version 1.1 of the TPP protocol.

The Product object in ResellOne.net provides access to information about products provisioned in ResellOne.net. Examples of products in ResellOne.net are E-mail accounts, E-mail domain names, Digital Certificates, etc.

ResellOne.net uses four key values to classify products. First, there's the standard "action" and "object" found in all TPP commands. Additionally, there is a "service" which identifies the provisioning sub-system that provides services for the product. And finally, there is the "object_type" key, which identifies the product within the service.

Products permit the following actions to be performed:

- check
- create
- query
- update
- delete
- transfer (not available yet)

Service	<i>Parameter Name:</i> service <i>Type:</i> String <i>Description:</i> The ResellOne.net service which is responsible for the provisioning of the product. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> One of "certs" or "email"
object_type	<i>Parameter Name:</i> object_type <i>Type:</i> String <i>Description:</i> The type of object within the service. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Values vary depending on the selected service.
product_data	<i>Parameter Name:</i> product_data <i>Type:</i> Associative Array <i>Description:</i> The data specific to the product. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Keys and values vary depending on the service and object_type.
Service	<i>Parameter Name:</i> service <i>Type:</i> String <i>Description:</i> The ResellOne.net service responsible for the provisioning of the product. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> One of "certs" or "email"
object_type	<i>Parameter Name:</i> object_type <i>Type:</i> String <i>Description:</i> The type of object within the service. <i>Obligation:</i> Required <i>Location:</i> This key is located in the attributes associative array. <i>Allowed Value(s):</i> Values vary depending on the selected service.

product_data

Parameter Name: product_data
Type: Associative Array
Description: The data specific to the product.
Obligation: Required
Location: This key is located in the attributes associative array.
Allowed Value(s): Keys and values vary depending on the service and object_type

The check operation allows the client to check for the availability of a named product.

Request Example

The format of a typical product check request is as follows:

Response Example

The format of the basic product check response is as follows:

The common response keys are the standard ones. The variable data for the command's response is contained in the attributes' "product_data" associative array.

The create operation allows the client to create/provision objects which are free of charge. Objects that have an associated cost must be provisioned via an ResellOne.net order.

Request Example

The format of a typical product create request is as follows:

Response Example

The format of the basic product create response is as follows:

The common response keys are the standard ones. The variable data for the command's response is contained in the attributes' "product_data" associative array.

The query operation allows the client to retrieve data regarding a provisioned object in ResellOne.net.

The generic request and response formats will not be repeated here as they are the same as with the check and create operations. Of course, the "action" value is "query".

The update command allows the client to change non-billable properties of an ResellOne.net-provisioned object.

The generic request and response formats will not be repeated here as they are the same as with the check and create operations. Of course, the "action" value is "update".

The delete command allows the client to delete a provisioned object. Note that some restrictions may apply, such as when the requested object is associated with another object. Usually, all associations and dependencies must be removed before an object can be deleted.

The generic request and response formats will not be repeated here as they are the same as with the check and create operations. Of course, the "action" value is "delete".

The transfer operation will allow the client to change the ownership of a provisioned object in ResellOne.net. Object will include change of customer and/or user. Only non-billable transfers will be handled with this command. Transfers that are accompanied by a financial cost are processed through ResellOne.net orders.

The generic request and response formats will not be repeated here as they are the same as with the check and create operations. Of course, the "action" value is "transfer".

Acknowledge queued items

Note: As of December 15, 2005, this command supports the Digital Certificates, Managed DNS, Email Defense, and Website Builder services. The Email service currently is supported.

Items that were processed asynchronously are placed in a queue; you can execute a query that will list items in the queue. See the [Query Queued Order Items command](#).

The queue is intended to help Resellers manage the asynchronously processed order items that they need to act on. If you've taken action on an order item, e.g. contacted your customer, you can acknowledge it, which will remove it from the queue summary.

Note: If you acknowledge an already acknowledged item, a success response will be returned.

queued_items	<i>Parameter Name:</i> queued_items <i>Type:</i> associative array <i>Description:</i> Lists the queued items to acknowledge <i>Obligation:</i> Required <i>Location:</i> In the 'attributes' associative array. <i>Allowed Value(s):</i> "queued_item_id" parameter
queued_item_id	<i>Parameter Name:</i> queued_item_id <i>Type:</i> String <i>Description:</i> The ID of the queued item to acknowledge <i>Obligation:</i> Required <i>Location:</i> This key is located in the 'queued_items' associative array. <i>Allowed Value(s):</i> Valid item ID

Examples

Request example, PERL – Acknowledge queued items

Response example, PERL – Acknowledge queued items

Request example, XML – Acknowledge queued items

Response example, XML – Acknowledge queued items

Error Codes

Some error codes have multiple parts. A hierarchy exists for these error codes. The following examples illustrate several situations and the rules governing each one.

If there is only one item in the order, then the item's error code/message is also the top level (order level) error code/message.

If there are multiple items in the order, the first failed item's error code/message is also the top level (order level) error code/message.

If there are multiple items in the order and the command contains two commands, the first failed item's error code/message is also the top level (order level) error code/message but all failed items returned will contain an error code/message so that the error may be remedied.

Error Code	Error Message
4508	Cannot set expiry date on this type of product. This message is returned if trying to set an expiry date for a service whose expiry date is automatically defined by its term, e.g., trial orders or Digital Certificates. Digital Certificates can only be purchased for one-year terms and expire after that one-year.
Success / failure errors	
200	SUCCESS
300	ORDER_REQUESTED
310	In progress. Supplier is unavailable. (order queued)
320	In progress. System is busy. (order queued)
330	In progress. Action is temporarily unavailable. (order queued)
340	In progress. Unexpected error to be resolved. (order queued)
350	In progress. Request successfully submitted and placed in the queue.
400	FAILURE
500	FATAL_SERVER_ERROR
510	Supplier is unavailable. (applies to all non-billable APIs)
520	System is busy. (applies to all non-billable APIs)
530	Action is unavailable. (applies to all non-billable APIs)
540	Unexpected error. (applies to all non-billable APIs)
1200	PARTIAL_SUCCESS
1400	PARTIAL_FAILURE
1500	INCOMPLETE_REQUEST
1600	SERVER_LIFECYCLE_ERROR
Protocol errors	
1700	TPP_PROTOCOL_ERROR
1701	TPP_VERSION_ERROR
1702	TPP_UNSUPPORTED_OPERATION
1703	TPP_INVALID_VALUE
1800	PEP_PROTOCOL_ERROR

1900	OPS_PROTOCOL_ERROR
Server lifecycle authorization errors	
2000	AUTHORIZATION_FAILURE
2100	AUTHENTICATION_ERROR
2101	AUTHENTICATION_ERROR_PASSWORD_MISSING
2102	AUTHENTICATION_ERROR_ROLE_MISSING
Request general errors	
3001	DB_INCONSISTENT_ERROR
3002	OBJECT_NOT_FOUND
3003	MISSING_PRODUCT_DATA
3003	TRANSACTION_ROLLBACKED
3004	In progress. Unexpected error to be resolved.
Sub-system failure 3500-3999	
3500	SUB_SYSTEM_FAILURE
3501	SUB_SYSTEM_COMM_FAILURE
User system failure 4000-4499	
Contract system failure 4500-4999	
4500	CONTRACT_ERROR_UNKNOWN
4501	CONTRACT_CREATE_ERROR
4502	CONTRACT_UPDATE_ERROR
4503	CONTRACT_GET_ERROR
4504	CONTRACT_REMOVE_ERROR
4505	CONTRACT_CHARGE_ERROR
4506	CONTRACT_REFUND_ERROR
4507	CONTRACT_INVALID_STATE
4508	CONTRACT_INVALID_TYPE
4601	PROJ_BILL_TRANS_CREATE_ERROR
4602	PROJ_BILL_TRANS_GET_ERROR
4603	PROJ_BILL_TRANS_UPDATE_ERROR
4604	PROJ_BILL_TRANS_TYPE_ERROR
4605	PROJ_BILL_TRANS_STATE_ERROR
4606	PROJ_BILL_TRANS_PERIOD_ERROR
Order system failure 5000-5499	
5000	ORDER_CREATE_SUCCESS
5051	ORDER_MANAGER_FAILURE
5051	ORDER_VALIDATION_FAILURE
5052	ORDER_ITEM_STATE_INVALID
5053	ORDER_ITEM_INVALID_CONTACT
5054	ORDER_FAILURE
5055	ORDER_ERROR
5056	ORDER_CANNOT_CHARGE

5057	ORDER_CANNOT_RESERVE_MONEY
5058	ORDER_CANNOT_CANCEL_RESERVED
5059	ORDER_CANNOT_REFUND
5060	ORDER_ITEM_CANNOT_PROCESS
5061	ORDER_CONTAINS_NO_ACTIVE_ITEMS
5062	ORDER_CANNOT_FULFILL
5063	ORDER_CANNOT_CANCEL
5064	ORDER_CANNOT_REVOKE
5065	ORDER_CANNOT_DECLINE
5066	ORDER_ITEM_TYPE_INVALID
5067	ORDER_ITEM_INVALID_ITEM
5068	ORDER_ITEM_INVALID_STRATEGY
5069	ORDER_CANNOT_RECOVER
5070	ORDER_CANNOT_MANAGE
Provisioing system failure 5500-5699	
5500	PROVISIONING_MANAGER_FAILURE
5501	PROVISIONING_ERROR_CREATE
5502	PROVISIONING_ERROR_DELETE
5503	PROVISIONING_ERROR_UPDATE
5004	PRODUCT_ITEM_VALIDATION_ERROR
5005	PRODUCT_ITEM_CONTACT_LINK_VALIDATION_ERROR
5005	PRODUCT_ITEM_AUTHORIZATION_ERROR
5006	PROVISIONING_SYNC_ERROR
5507	PROVISIONING_PLUGIN_ERROR
5508	PROVISIONING_COMMUNICATION_ERROR
Inventory system failure 5700 -5999	
5701	INVENTORY_GENERAL_FAILURE
5702	INVENTORY_PERSISTENCE_FAILURE
5703	INVENTORY_ITEM_STATE_INVALID
5704	PRODUCT_ATTRIBUTE_NOT_EXIST
5705	PRODUCT_DELETE_ATTRIBUTE_FAILURE
5706	PRODUCT_GET_SKU_FAILURE
5707	PRODUCT_ATTRIBUTE_ALREADY_EXIST
5708	PRODUCT_VALIDATE_FAILURE
5709	PRODUCT_DELETE_SKU_FAILURE
5710	PRODUCT_VALUE_NOT_EXIST
5711	INVENTORY_ITEM_EXPIRY_DATE_INVALID This error is returned if the date specified in the 'expiry_date' parameter is not a future date. The expiry date must be in the future.
Contact system failure 6000-6499	
6000	CONTACT_MANAGER_FAILURE
6001	CONTACT_VALIDATION_FAILURE

6002	CONTACT_NOT_FOUND
6003	CONTACT_UPDATE_ERROR
6004	CONTACT_DELETE_ERROR
6005	CONTACT_EXIST
6006	CONTACT_GET_FAILURE
6007	CONTACT_CREATE_FAILURE
6008	CONTACT_OWNERSHIP_FAILURE
Workflow system error 6500-6999	
6500	WORKFLOW_MANAGER_FAILURE
6501	WORKFLOW_PLUGIN_ERROR
Pricing adapter codes 7000-7499	
7000	PRICING_ERROR
7001	PRICING_FAILURE
7002	PRICING_WRONG_PRODUCT_PARAMETER
Billing adapter codes 7500-7999	
7500	BILL_FAILURE
7501	BILL_ERROR
7502	BILL_INSUFFICIENT_BALANCE
Customer system codes 8000-8499	
8000	CUSTOMER_MANAGEMENT_FAILRURE
8001	CUSTOMER_VALIDATION_FAILRURE
8002	CUSTOMER_NOT_FOUND
8003	CUSTOMER_PARENT_ID_CHANGE_ERROR
8004	CUSTOMER_ALREADY_EXIST
8005	CUSTOMER_DOES_NOT_EXIST
8006	CUSTOMER_CONTACT_LINK_ERROR
8007	CUSTOMER_DATA_MISSING
8008	CUSTOMER_CREATE_ERROR
8009	CUSTOMER_UPDATE_ERROR
8010	CUSTOMER_DELETE_ERROR
8011	CUSTOMER_GET_ERROR