

Profits Plus Web Services v1.4 for HTTP(1.1)

Updated April 18, 2016

Introduction

This document describes how a programmer can interface with Profits Plus Web Services using HTTP messages. Web Services is a Software Basic program that listens to a communication port on the Profits Plus server for HTTP messages requesting information or actions. It replies back to the sender with another HTTP message containing the requested information or a confirmation that a requested action has taken place. The server can launch up to 10 different and simultaneous instances of RUN to service up to 10 clients at once. Each client will be serviced by its own dedicated instance of RUN. Each client and RUN pair will communicate to each other via a separate TCP/IP socket connection.

A web client controls the Profits Plus web server by sending it http POST requests for specific pre-defined functions. These POST functions are:

- ORDER.CREATE
- ORDER.MODIFY.HEADER
- ORDER.MODIFY.DETAIL
- ORDER.SUBMIT
- ORDER.CANCEL
- ITEM.GET.INFO
- CONNECTION.CLOSE

Most of the above web server functions will return an http response message back to the client.

The rest of this document explains how the client should format the http request messages and how to interpret the http responses messages sent back from the web server.

1.1 ORDER.CREATE

1.1.1 Description

This client request creates an empty order in Profits Plus. The server returns an order number.

1.1.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
divisionNumber	CHAR(2)	??	No	01-99. Default value assigned during system setup. OEH_WHS_NUM
customerNumber	CHAR(5)	None	Yes	OEH_CUS_NUM
altShipNumber	CHAR(5)	00000	No	If not provided, then default to 00000 OEH_SHP_NUM
headerMemo	TEXT	Null	No	OEH_MEMO_PTR
transactionType	CHAR(1)	1	No	1=sale, 2=quote, 3=credit memo OEH_TRX_TYPE
purchaseOrder	CHAR(20)	None	No	OEH_PO
customerShipNumber	CHAR(20)	None	No	OEH_CUS_SHP
orderDate	CHAR(6)	Today	No	OEH_ORD_DTE mmddyy
invoiceDate	CHAR(6)	Null	No	OEH_INVOICE_DTE mmddyy
generalInfo	CHAR(16)	Null	No	OEH_GEN_1
salespersonNumber	CHAR(4)	Null	No	OEH_SLM_NUM

The fields are located in the oehtrx_dat table. The database table field names above are shown in uppercase (example: OEH_WHS_NUM), and this is how the field names are spelled in the SQL database schema.

Example http messages sent from client.

```
POST order.create HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 114[CRLF]
[CRLF]
companyNumber=01&customerNumber=12345&altShipNumber=00000&headerMemo=Deliver
ies+are+in+the+back.&transactionType=1[no CRLF]
```

1.1.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, display InfoText to user. 1 = Failure. Display “An error occurred ...” text to user and send an error email containing InfoText to the web admin. 2= Request denied. Display InfoText to user, allow user to try again.
orderNumber	NUMBER(6)	None	Yes	If order is created, then order number will be > 0. If 0, then look at infoText for a user readable message
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application.

Example http messages sent back to client.

```
HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 60[CRLF]
[CRLF]
successFlag=0&orderNumber=123456&infoText=New+order+created.[no CRLF]
```

1.1.4 Implementation Notes

This function is a simple implementation of the order creation process in Profits Plus. Any header fields that need to be set in the creation of an order will be set by default based on system parameters. These would be salesperson, department, division, etc.

1.2 ITEM.GET.INFO

1.2.1 Description

This client request returns item information specific for a particular customer.

1.2.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
customerNumber	CHAR(5)	None	Yes	
altShipNumber	CHAR(5)	00000	No	If not provided, then default to 00000
itemNumber	CHAR(18)	None	Yes	

Example http messages sent from client.

```
POST item.get.info HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 73[CRLF]
[CRLF]
companyNumber=01&customerNumber=12345&altShipNumber=00000&itemNumber=1001[no
CRLF]
```

1.2.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, display InfoText to user. 1 = Failure. Display “An error occurred ...” text to user and send an error email containing InfoText to the web admin. 2= Request denied. Display InfoText to user, allow user to try again.
itemNumber	CHAR(18)	None	Yes	See implementation notes below.
itemDescription	CHAR(30)	None	No	If no item description is retrieved, then look at infoText for answers. Contains original item # for swapped special orders. See implementation notes below,
itemPrice	NUMBER	None	No	
itemAvailability	CHAR(1)	None	No	A = Available, N = Out of Stock, U = Unknown, O = On Order, S = Special Order only
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application. Example: Quantity discount breaks at: 6, 12, 18

Example http messages sent back to client.

```

HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 142[CRLF]
[CRLF]
successFlag=0&itemDescription=GASKET+KIT+A,+TEST+OLD+ITEM&itemPrice=145.54&itemAvailability=A&infoText=Quantity+discount+breaks+at:+6,+12,+18.[no CRLF]

```

1.2.4 Implementation Notes

This function will use the current logic of the INVCAT program that prints out a price list. This program has the functionality to show prices for a particular customer for a particular item. The program also shows quantity discount break points.

Special handling for items that are not on file.

If itemNumber is not-on-file then the mvs server will respond with success but substitute the not-on-file item # with a generic item number for special orders that *will* be on file. As of 05/02/06 on the mvs machine the generic special order # is “WEBSPECIALORDER”.

The response you get back will have these special characteristics:

- The response itemNumber field will contain the swapped special order item #
- The itemDescription field will be “For item # xxxxxxxxxx” where xxxxxxxxxx is the original not-on-file item #.
- The itemAvailability will be “s”
- The itemPrice will be zero.

Example:

If you send an item.get.info request for item NOF1 (which is not on file) then you will get a successful response back like this:

```
successFlag=0&itemNumber=WEBSPECIALORDER&itemDescription=For+item+#+NOF1&itemPrice=0.00&itemAvailability=S
```

Of course if the generic special order # is not on file either then the above substitution can not occur and you will get a “successFlag=1” failure response with “infoText=Item # xxxxxxxxxx is not on file.” And the itemNumber field returned will be the same not-on-file number you requested.

1.3 ORDER.MODIFY.HEADER

1.3.1 Description

This client request changes the value of certain fields on the header record for a specific order.

1.3.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
orderNumber	NUMBER(6)	None	Yes	OEH_ORD_NUM
altShipNumber	CHAR(5)	00000	No	OEH_SHP_NUM If not provided, then default to 00000.
shipDate	CHAR(6)	000000	No	OEH_SHP_DTE Change if not 000000. Will already be in MMDDYY format
purchaseOrder	CHAR(20)	None	No	OEH_PO
shipViaCode	CHAR(1)	None	No	OEH_SHP_CODE Code in shptbl_dat table.
headerMemo	TEXT	Null	No	OEH_MEMO_PTR Foreign table = oehtrx_mmo
transactionType	CHAR(1)	None	No	1=Sale, 2=Quote, 3=CR Memo OEH_TRX_TYPE
customerShipNumber	CHAR(20)	None	No	OEH_CUS_SHP
invoiceDate	CHAR(6)	Today	No	OEH_INVOICE_DTE mmddyy
orderDate	CHAR(6)	Today	No	OEH_ORD_DTE mmddyy
generalInfo	CHAR(16)	Null	No	OEH_GEN_1
salespersonNumber	CHAR(4)	Null	No	OEH_SLM_NUM
orderTaxStatus	CHAR(1)	??	No	Default assigned by customer. T=taxable N=exempt, All other values mean non-taxable. OEH_TAX_FLG
Year	CHAR(2)	Null	No	OEH_YEAR
Make	CHAR(8)	Null	No	OEH_MAKE
Model	CHAR(15)	Null	No	OEH_MODEL
Meter	NUMBER(3)	Null	No	OEH_METER
serialNumber	CHAR(18)	Null	No	OEH_SER_ENG

These fields are located in the oehtrx_dat table. The database table field names above are shown in uppercase (example: OEH_WHS_NUM), and this is how the field names are spelled in the SQL database schema.

Example http messages sent from client.

```
POST order.modify.header HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 144[CRLF]
[CRLF]
companyNumber=01&orderNumber=123456&altShipNumber=00000&shipDate=000000&purchaseOrder=MYPO#&shipViaCode=A&headerMemo=Deliveries+are+in+the+back.[no CRLF]
```

1.3.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, display InfoText to user. 1 = Failure. Display “An error occurred ...” text to user and send an error email containing InfoText to the web admin. 2= Request denied. Display InfoText to user, allow user to try again.
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application. This text will only be meaningful in the case of technical success Examples: Sorry, this order has already been sent to shipping. We cannot change the shipping information at this time. Order details successfully changed.

Example http messages sent back to client.

```
HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 51[CRLF]
[CRLF]
successFlag=0&infoText=Your+order+has+been+changed.[no CRLF]
```


1.3.4 Implementation Notes

This function will encapsulate some business rules that will govern whether the order header can be updated or not.

1.4 ORDER.MODIFY.DETAIL

1.4.1 Description

This client request changes the value of certain fields on a detail line-item for a specific order.

1.4.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
divisionNumber	CHAR(2)	??	No	01-99. Default value assigned at system setup. OED_WHS_NUM
orderNumber	NUMBER(6)	None	Yes	OED_ORD_NUM
itemNumber	CHAR(18)	None	Yes	OED_ITM_NUM
sequenceNumber	NUMBER(4)	0	No	OED_SEQ_NUM If 0, then this is a new item that should be added to the order
newQty	NUMBER	None	Yes	OED_QTY_ORD 0 if the item is to be removed from the order
itemQtyType	CHAR(1)	None	No	OED_QTY_UNT S=single (sell-by) B=bulk (stock-by)
lineItemMemo	VARCHAR	null	No	Null if memo is to be removed. OED_MEMO_PTR Foreign table = oedtrx_mmo
itemCost	CHAR(15)	Null	No	OED_ITM_CST Assumed to contain decimal point. "1.00"
itemPrice	CHAR(15)	Null	No	OED_ITM_PRC Assumed to contain decimal point. "1.00"
itemDescription	CHAR(35)	Null	No	OED_DESCR
itemTaxable	CHAR(1)	Null	No	True or False, Y or N, T or N OED_TXFLAG
size	NUMBER	Null	No	Dimensional size OED_DLEN
length	NUMBER	Null	No	Dimensional length OED_WIDTH
height	NUMBER	Null	No	Dimensional height OED_WIDTH
width	NUMBER	Null	No	Dimensional width OED_DLEN
originals	NUMBER	Null	No	Dimensional originals OED_DORIG
sets	NUMBER	Null	No	Dimensional sets OED_DSETS
copies	NUMBER	Null	No	Dimensional copies OED_DSETS

These fields are located in the oedtrx_dat table. The database table field names above are shown in uppercase (example: OED_WHS_NUM), and this is how the field names are spelled in the SQL database schema.

Example http messages sent from client.

```
POST order.modify.detail HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 152[CRLF]
[no CRLF]
companyNumber=01&orderNumber=123456&itemNumber=1001&sequenceNumber=0&newQty=
1&lineItemMemo=This+is+line+one+of+the+memo.%0DThis+is+line+two+of+the+memo.
[no CRLF]
```

1.4.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, display InfoText to user. 1 = Failure. Display “An error occurred ...” text to user and send an error email containing InfoText to the web admin. 2= Request denied. Display InfoText to user, allow user to try again.
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application. This text will only be meaningful in the case of technical success Examples: Sorry, this order has already been sent to shipping. We cannot change the item information at this time. Item details successfully changed.

Example http messages sent back to client.

```
HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 51[CRLF]
[no CRLF]
successFlag=0&infoText=Your+order+has+been+updated. [no CRLF]
```

1.4.4 Implementation Notes

Dimensional items:

Dimensional items are identified by the INVMAS.INV_DIM_FLG field CHAR(1).

There are three values that mean the item is *not* dimensional:

- null
- N = not dimensional
- R = Repro-no warehouse quantities

All other values of INV_DIM_FLG will mean the item is dimensional in some way and, in addition to the newQty field, can accept optional dimensional fields in the order.modify.detail API call.

- size, length, or height
- width
- originals
- sets, or copies

Decimals are accepted.

Note: Your web client application will still have to compute and set the newQty API field in the correct unit of measure, even when supplying dimensional fields. The web server will not compute newQty for you based on the given dimensions. The dimensional fields are used more like supporting detail than quantities.

It is up to the web client to know which dimensional fields are appropriate for a specific item's value of INV_DIM_FLG. The web server will accept these optional dimensional fields "as is" and will not generate an error if they are omitted or zero even when they are expected.

Compose the API call using the same labels that are prompted on the order entry line item screen for that dimensional item. Size, length, height, width, originals, sets or copies.

See the Profits Plus Inventory Control task "32/17/18. Dimensional types" for a complete list of possible values defined for INVMAS.INV_DIM_FLG.

1.5 ORDER.SUBMIT

1.5.1 Description

This client request sends an order to shipping after all items have been added

1.5.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
orderNumber	NUMBER(6)	None	Yes	
emailAddress	CHAR(255)	None	No	Provide an email address to this call if you want the system to send you a copy of the work order as a PDF attachment This address will come from the Users table in the eCommerce system database.

Example http messages sent from client.

```
POST order.submit HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 68[CRLF]
[no CRLF]
companyNumber=01&orderNumber=123456&emailAddress=myname@mydomain.com
```

1.5.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, display InfoText to user. 1 = Failure. Display “An error occurred ...” text to user and send an error email containing InfoText to the web admin. 2= Request denied. Display InfoText to user, allow user to try again.
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application. This text will only be meaningful in the case of technical success Examples: Your order has been sent to shipping. The expected shipping date is <MM/DD/YYYY>. Shipping method on the order is <ship method>. An email confirmation has been sent to you as well.

Example http messages sent back to client.

```
HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 200[CRLF]
[no CRLF]
successFlag=0&infoText=Your+order+has+been+sent+to+shipping.++The+expected+s
hipping+date+is+10/15/2005.++Shipping+method+on+the+order+is+OUR+TRUCK.++An+
email+confirmation+has+been+sent+to+you+as+well. [no CRLF]
```

1.5.4 Implementation Notes

This function “SHOPS” the order. In addition, this function also generates a PDF of the order and sends the emailAddress in the call an email with this PDF attached.

Email text could be:

Thank you for placing your order with Detroit Hitch Company. Please find attached a copy of the work-order for your records.

If you have any questions please email us at dhsales@detroithitch.com or call 800-424-5232

Thank you for your business!

Detroit Hitch Company.

1.6 ORDER.CANCEL

1.6.1 Description

This client request cancels an order. This function does not verify if the order really belongs to the client. It will blindly cancel any order that is currently in the oehtrx_dat table.

1.6.2 Inputs

This is the format of the body of the http message sent from the client.

Name	Type	Default	Required	Comments
companyNumber	CHAR(2)	01	Yes	
orderNumber	NUMBER(6)	None	Yes	

Example http messages sent from client.

```
POST order.submit HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 35[CRLF]
[no CRLF]
companyNumber=01&orderNumber=123456
```

1.6.3 Outputs

This is the format of the body of the http message sent back to the client.

Name	Type	Default	Required	Comments
successFlag	NUMBER(1)	0	Yes	0 = Success, 1 = Failure. This indicates technical success or failure.
infoText	CHAR(255)	NULL	No	This text can be directly shown to the user of the web application. This text will only be meaningful in the case of technical success Examples: Your order # 123456 has been cancelled. We're sorry, order # 123456 is not on file.

Example http messages sent back to client.

```
HTTP/1.1 200 OK[CRLF]
Server: TCPSRV/1.5 (A) [CRLF]
Date: Wed, Aug 24 2005 06:38:43 GMT[CRLF]
Content-Type: application/x-www-form-urlencoded[CRLF]
Content-Length: 62[CRLF]
[no CRLF]
successFlag=0&infoText=Your+order+#+123456+has+been+cancelled.
```


1.7 CONNECTION.CLOSE

1.7.1 Description

This client request tells the server to close the current socket connection.

1.7.2 Inputs

There are no input parameters. There is no body to the message. The request consists only of headers. The most important header is the “Connection: close” header. For example:

Example http messages sent from client.

```
POST connection.close HTTP/1.1[CRLF]
From: somebody@mysite.com[CRLF]
User-Agent: HTTPTool/1.1[CRLF]
Connection: close [CRLF]          ←-----This header is required
[CRLF]
```

1.7.3 Outputs

There is no high level response back to the client. However there might be an acknowledgement sent back to the client by the low level socket software. The server will immediately close the current socket connection.

1.7.4 Implementation Notes

Normally the client maintains a “persistent” socket connection which means the server responds to a client request and then waits for another request without closing the connection. When the client user logs out the client software should send the server a connection.close request so the server can release resources related to servicing the client. Even with persistent connections the server will wait only 10 minutes with no activity before closing the connection anyway.

Connection timeout issues

1.7.5 Description

The mvs web server is HTTP 1.1 compliant which means it is capable of handling multiple client/server messages in a single connection session. This is called a persistent connection. The client can open a connection to the mvs web port number and issue multiple requests and receive multiple responses before closing the connection. The final request from the client should contain a "Connection: close" header line so the server will know to automatically close the connection after sending the final response back to the client. If the client wants the server to keep the connection open for multiple requests then the client should not send the "Connection: close" header until the final request.

As of 05/10/06 the mvs server will permit the client in a persistent connection to remain idle for up to 10 *minutes* before it gives up and closes the connection. This server timeout period is a parameter and it can be adjusted if necessary.

For non-persistent connections (where the client includes the "Connection: close" header in the initial request) the server will wait only 5 *seconds* between each line of the request before closing the connection.