



Interface Specification

WaiterPad System

Revision History		
5 th September 2008	1.00	Initial Revision of Document
13 th January 2009	1.10	Version 6.08 revision
15 th July 2011	1.20	Updated revision created
22 nd February 2012	1.22	Added Card line type
3 rd March 2012	1.24	Fixed line type value for Card line being incorrect
8 th March 2012	1.26	Added Text Stamp line type
23 rd May 2012	1.27	Corrected: PLU Sale and Custom Product Price field was noted as in cents, this is not correct, it should be decimal. Eg 10.25
16 th April 2014	1.28	Added Adjustment line type to doco
17 th November 2016	1.29	Correction to packet header fields to match the examples.

Table of Contents

1. DISCLAIMER AND COPYRIGHT	3
2. DATA TYPE DEFINITIONS.....	4
3. GLOSSARY OF TERMS	5
4. TCP COMMUNICATION (NC/POS) PACKET FORMATS.....	8
4.1. WaiterPad Orders (one-way).....	8
4.1.1 Communication Flow Example.....	8
4.1.2 Packet Header.....	9
4.1.3 Order Header.....	10
4.1.4 PLU Sale	11
4.1.5 Custom Product.....	12
4.1.6 Custom Message.....	13
4.1.7 Card	14
4.1.8 Adjustment (Discount / Surcharge)	15
4.1.9 Text Stamp	16
4.1.10 FastPay Payment	17
4.1.11 Order Footer	18
4.1.12 Comms Status Response	19
4.2. XML Formats.....	20
4.2.1 PLU's.....	20
4.2.2 Minor Groups (Printing / Reporting)	23
4.2.3 Main Groups	23
4.2.4 Sale Modifiers.....	24
4.2.5 Terminals.....	25
4.2.6 Table Ranges	26

1. Disclaimer and Copyright

Document

This document is designed to give TriniTEQ staff and dealers succinct and useful details relating to our software and hardware products.

It should be noted that only information of valuable note is included in this document and if it is felt that information is missing we welcome feedback. Please email your suggestions to techsupport@triniteq.com subject: TriniTEQ Document Feedback.

TriniTEQ strives to continue being a leader in providing solutions for the hospitality and service industry. In order to achieve this we will:

- Continue to deliver the highest level of consistent quality in all our products and services
- Ensure that outstanding service turns our customers into advocates of our products
- Research, understand and anticipate our markets, customers and competitors and respond with innovative solutions
- Act efficiently, honestly and fairly at all times

DISCLAIMER

TriniTEQ will not be held accountable for any errors or omission in this document. If any errors or omissions are found in this document, please alert our technical support staff by emailing techsupport@triniteq.com subject: TriniTEQ Document Feedback.

Copyright – Commercial in Confidence

This document, contents and ideas are the property of TriniTEQ International Pty Ltd and are strictly confidential.

This document is subject to Australian and International copyright laws. Commercial use, publication or distribution of all or part of this document, images, logos and graphics is strictly prohibited without the prior express authorisation from TriniTEQ International Pty Ltd.

All images and product representations within this document are for illustration purposes only.

© TriniTEQ 2013

TriniTEQ International Pty Ltd
ABN 81 092 943 649

2. Data Type Definitions

Type Name	Equivilant c data type	Description
UInt8	unsigned char	8 bit unsigned
UInt16	unsigned short	16 bit unsigned
UInt32	unsigned long	32 but unsigned
Int8	signed char	8 bit signed
Int16	signed short	16 bit signed
Int32	signed long	32 but signed
String	char[??]	Specified length character array
PString	UInt8 (datalen) char[] (data)	Variable Length String, uses a UInt8 followed by a String of the length specified by the UInt8.
ZString	Char[] (null)	Variable Length String, uses a Null (0) terminating character to mark the end of the text

3. Glossary of Terms

Term	Description						
Big Endian	<p>Refers to byte ordering in a system. PC's use big endian format (also called network byte order). The network controller ncd database files are stored in little endian format. All TCP communications are performed in big endian format.</p> <p>For example, a 32 bit value of 0x10203040 would be stored as follows</p> <p style="text-align: right;"><i>Increasing memory address →</i></p> <table> <tr> <td>Big Endian (network byte order)</td><td>[0x10] [0x20] [0x30] [0x40]</td></tr> <tr> <td>Little Endian</td><td>[0x40] [0x30] [0x20] [0x10]</td></tr> </table>	Big Endian (network byte order)	[0x10] [0x20] [0x30] [0x40]	Little Endian	[0x40] [0x30] [0x20] [0x10]		
Big Endian (network byte order)	[0x10] [0x20] [0x30] [0x40]						
Little Endian	[0x40] [0x30] [0x20] [0x10]						
Chase Note	<p>A chase note, or chase status is a status applied to a sale item on the WaiterPad which indicates the item was previously ordered but has not been delivered yet. It is basically used to query the status of an item with the kitchen. Chase items are designed to be printed to on the kitchen dockets, but not to be stored on the actual order.</p>						
Child Product	<p>A child product is a classification for a PLU. Child product plus automatically attach to the nearest parent product above it. They are considered a sub component of the parent product.</p> <p>For Example</p> <table> <tr> <td>T-Bone Steak</td><td>← parent product</td></tr> <tr> <td>Medium Rare</td><td>← child product</td></tr> <tr> <td>Pepper Sauce</td><td>← child product</td></tr> </table>	T-Bone Steak	← parent product	Medium Rare	← child product	Pepper Sauce	← child product
T-Bone Steak	← parent product						
Medium Rare	← child product						
Pepper Sauce	← child product						
Cooking Instruction	<p>A cooking instruction is a sub classification for a PLU. It is considered part of the child product classification. Cooking instruction PLUs are intended not to have a price. Examples would be PLUs specifying how to cook a steak eg: 'Medium Rare', 'Rare', 'Well Done' etc.</p>						
Condiment	<p>A condiment is a sub classification for a PLU. It is considered part of the child product classification. Condiment PLUs may have a price. Examples would be PLUs specifying what sauces to have with a steak eg: 'BBQ', 'Hickory', 'Pepper' etc.</p>						
Covers	<p>Covers refer to the number of guests/patrons on a table. A table with 2 covers simply means there are 2 people sitting at that table.</p>						
Custom Message	<p>A custom message is a sub classification for a sale item. It is considered part of the child product classification. Custom Message sales do not have a price, and are not actually PLUs. They are custom created messages from the WaiterPad. These are usually used when a PLU does not exist to satisfy the required message.</p>						
Custom Product	<p>A custom product is a sub classification for a sale item. It is considered part of the parent product classification. Custom Product sales are not actually PLUs. They are custom sale items from the WaiterPad. The price, group, tax and printer options are specified at the time the custom product is created. These are usually used when a PLU does not exist to satisfy the required message, or perhaps a special meal is being made that does not exist on the menu.</p>						
KP Docket	<p>Refers to an order docket printed in the kitchen which specifies what the staff are required to prepare. This does not exclusively refer to orders for the kitchen, but generally encompasses any orders prepared by staff in the kitchen, bar or elsewhere.</p>						
KP Printer	<p>Refers to a printer used to print KP dockets.</p>						
Little Endian	<p>Refers to byte ordering in a system. The network controller ncd database files are stored in little endian format. PC's use big endian format (also called network byte order). All TCP communications are performed in big endian format.</p> <p>For example, a 32 bit value of 0x10203040 would be stored as follows</p> <p style="text-align: right;"><i>Increasing memory address →</i></p>						

	<p>Big Endian (network byte order) [0x10] [0x20] [0x30] [0x40] Little Endian [0x40] [0x30] [0x20] [0x10]</p>														
Main Group	<p>Main group is the top level classification for groups. Printing/Reporting groups belong to a main group. For Example</p> <table> <tr> <td>Food</td><td>← Main Group</td></tr> <tr> <td>Entrees</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Mains</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Desserts</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Beverage</td><td>← Main Group</td></tr> <tr> <td>Beers</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Cold Drinks</td><td>← Printing/Reporting Group</td></tr> </table>	Food	← Main Group	Entrees	← Printing/Reporting Group	Mains	← Printing/Reporting Group	Desserts	← Printing/Reporting Group	Beverage	← Main Group	Beers	← Printing/Reporting Group	Cold Drinks	← Printing/Reporting Group
Food	← Main Group														
Entrees	← Printing/Reporting Group														
Mains	← Printing/Reporting Group														
Desserts	← Printing/Reporting Group														
Beverage	← Main Group														
Beers	← Printing/Reporting Group														
Cold Drinks	← Printing/Reporting Group														
Message	<p>A message is a sub classification for a PLU. It is considered part of the child product classification. Message PLUs do not have a price. Examples would be PLUs specifying what how to serve a steak eg: 'Serve As Main', 'No Onion' etc</p>														
Parent Product	<p>A parent product is a classification for a PLU. Parent product plus are considered the base/top level product. Child products attach to parent products. For Example</p> <table> <tr> <td>T-Bone Steak</td><td>← parent product</td></tr> <tr> <td>Medium Rare</td><td>← child product</td></tr> <tr> <td>Pepper Sauce</td><td>← child product</td></tr> </table>	T-Bone Steak	← parent product	Medium Rare	← child product	Pepper Sauce	← child product								
T-Bone Steak	← parent product														
Medium Rare	← child product														
Pepper Sauce	← child product														
PLU	<p>PLUs are the core sale items in the system. PLUs are classified into product classes and contain all the price, group, printing and configuration options for the sale.</p>														
Printer Map	<p>A printer map is a method or routing printers from one to another. For example, let's say we have a restaurant with two floors, and a kitchen on each floor. The menu for both floors is the same, however each respective kitchen is responsible for making the meals for the floor it is on. A printer map could be used so that waiters who are serving tables on the top floor will have their kitchen dockets printed to the top floor kitchen, while waiters on the ground floor will have their orders printed to the lower kitchen.</p>														
Printing Group	<p>Printing group is the second level classification for groups. Each printing group belongs to a parent Main Group. Printing Groups and Reporting Groups are from the same set of minor groups, however printing group refers is the group name printed on kitchen dockets for the sale, whereas reporting group refers to the group that reporting figures are recorded against for the sale. For Example</p> <table> <tr> <td>Food</td><td>← Main Group</td></tr> <tr> <td>Entrees</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Mains</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Desserts</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Beverage</td><td>← Main Group</td></tr> <tr> <td>Beers</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Cold Drinks</td><td>← Printing/Reporting Group</td></tr> </table>	Food	← Main Group	Entrees	← Printing/Reporting Group	Mains	← Printing/Reporting Group	Desserts	← Printing/Reporting Group	Beverage	← Main Group	Beers	← Printing/Reporting Group	Cold Drinks	← Printing/Reporting Group
Food	← Main Group														
Entrees	← Printing/Reporting Group														
Mains	← Printing/Reporting Group														
Desserts	← Printing/Reporting Group														
Beverage	← Main Group														
Beers	← Printing/Reporting Group														
Cold Drinks	← Printing/Reporting Group														
Product Class	<p>Product Class refers to the sub-classification of a PLU item. Product class controls how PLU sale items are linked, printed and generally processed. The following are product classes.</p> <ul style="list-style-type: none"> • Standard Product • Side Dish • Condiment • Cooking Instruction • Message • Custom Product (assigned to custom plus created during ordering phase) • Custom Message (assigned to custom messages created during ordering phase) 														
Reporting Group	<p>Reporting group is the second level classification for groups. Each reporting group belongs to a parent Main Group. Reporting and Printing Groups are from the same set of minor groups, however reporting group refers to the group that reporting figures are</p>														

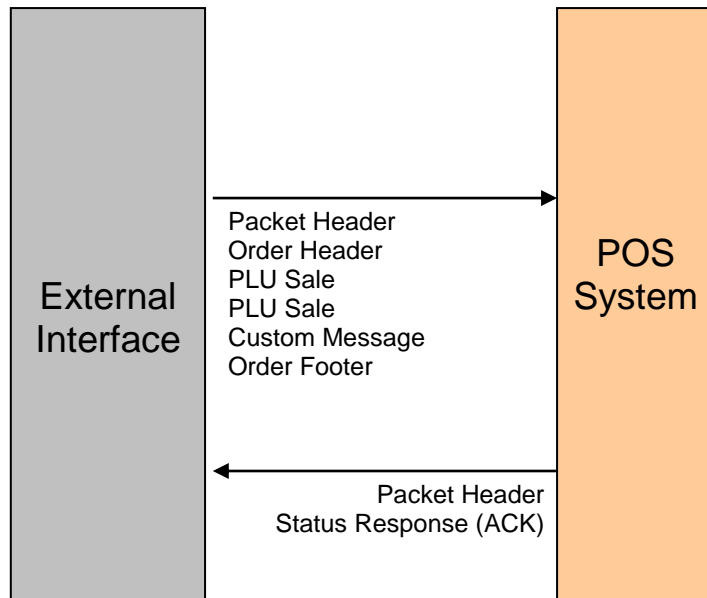
	<p>recorded against for the sale whereas printing group refers is the group name printed on kitchen dockets for the sale.</p> <p>For Example</p> <table> <tr> <td>Food</td><td>← Main Group</td></tr> <tr> <td> Entrees</td><td>← Printing/Reporting Group</td></tr> <tr> <td> Mains</td><td>← Printing/Reporting Group</td></tr> <tr> <td> Desserts</td><td>← Printing/Reporting Group</td></tr> <tr> <td>Beverage</td><td>← Main Group</td></tr> <tr> <td> Beers</td><td>← Printing/Reporting Group</td></tr> <tr> <td> Cold Drinks</td><td>← Printing/Reporting Group</td></tr> </table>	Food	← Main Group	Entrees	← Printing/Reporting Group	Mains	← Printing/Reporting Group	Desserts	← Printing/Reporting Group	Beverage	← Main Group	Beers	← Printing/Reporting Group	Cold Drinks	← Printing/Reporting Group
Food	← Main Group														
Entrees	← Printing/Reporting Group														
Mains	← Printing/Reporting Group														
Desserts	← Printing/Reporting Group														
Beverage	← Main Group														
Beers	← Printing/Reporting Group														
Cold Drinks	← Printing/Reporting Group														
Sale Modifier	<p>A sale modifier refers to PLU sale modifiers that change the serving type of a sale item. For example, converting a normal glass of beer into a Pint of beer, or a normal shot of spirits into a double shot. A sale modifier generally only changes the description and price of a sale, but can also change the printing group and product class.</p>														
Service Location	<p>A service location is a classification for a type of order and/or range of that type. A single service location can be made up of multiple types and ranges, however there cannot be any overlap between different service locations.</p> <p>Some examples of service locations are</p> <p>Inside Tables – Tables 1 – 65, and Tables 80-90 Outside Tables – Tables 101-150 Bar Orders – All bar orders</p> <p>Service locations are commonly used when configuring various areas of functionality and reporting with POS systems.</p>														
Side Dish	<p>A side dish is a sub classification for a PLU. It is considered part of the child product classification. Side Dish PLUs may have a price. Examples would be PLUs specifying what sides to have with a steak eg: 'Salad', 'Veges' etc</p>														
Standard Product	<p>A standard product is a sub classification for a PLU. It is considered part of the parent product classification. Standard products are the most common form of PLU. These are items that are in themselves a standalone sale. They may or may not have child items attached. A few examples would be 'T-Bone Steak', 'Corona', 'Garlic Bread'.</p>														

4. TCP Communication (NC/POS) Packet Formats

4.1. WaiterPad Orders (one-way)

4.1.1 Communication Flow Example

Assuming the WaiterPad has an order of two PLUs to send to the POS, it would send the following.



4.1.2 Packet Header

All TCP communications start with a packet header. The header format is as follows:

Data Type	Data (ascii code)	Description
UInt8	0	Sync Bytes 1 of 4
UInt8	255	Sync Bytes 2 of 4
UInt8	0	Sync Bytes 3 of 4
UInt8	255	Sync Bytes 4 of 4
UInt16	LineType*	Header Type (see defines below)
UInt8	64	Internal
UInt8	64	Internal
UInt8	0	Internal
UInt8	DeviceType*	Device Type (see defines below)
UInt16	DataLen	Length of data block to follow

LineTypes

SVR_NC_DATA 12001 (0x2E 0xE1) *Use this for messages you send to POS*
SVR_POS_DATA 12002 (0x2E 0xE2) *Expect this for messages you receive from POS*

DeviceType

DEVICE_EXTERNAL 9

The packet header above preceeds a series of smaller records which make up the data block. The datalen field of the packet header specifies the length of the entire data block (which contains the series of smaller records)

As the data length is part of the packet header, if the TCP library chooses to internally break the overall data into smaller packets, this will be handled without fault, as the packet will not be considered complete until the data length + header length (12 bytes) has been reached.

For all Packet Headers that you generate as part of a message you are sending to the POS terminal, you must use Line Type 12001 (0x2E, 0xE1).

All Packets Headers contained within messages that you receive back from the POS terminal will contain Line Type 12002 (0x2E, 0xE2).

4.1.3 Order Header

All sale based packets (ie when ordering PLU items) must be encapsuated by an Order Header and Order Footer.

Data Type	Data (ascii code)	Description
UInt8	01	lineType for a Header Record
UInt16	18	Block length of header data to follow
UInt8	Comms Version*	Communications Protocol Version Number for Orders
UInt16	Sequence*	Order Sequence Number (0-65000)
UInt16	Clerk ID	ID of clerk placing the order
UInt8	Order Type	The order type (1 = table, 2 = tab)
UInt16	Table / Tab Number	Table / Tab Number
UInt8	Bill Number	Bill Number the order is for (0=default bill, 1+ = bill number)
UInt16	Covers	Number of covers/guests on the table
UInt8	Status*	See defines below (bit flags)
UInt8	Printer Map ID	Printer Map ID to apply (0 = no printer map)
UInt8	Auto Price Receipt	Print a receipt (0 = false, 1 = true)
UInt16	Printer Number	Printer ID to print receipt too, if applicable
UInt16	WP Device ID	WaiterPad Device ID (0-65535)

* Sequence number must be unique per order header. Once sequence reaches 65000 limit, it should loop back to 1. If no response is received for a packet containing an order, the sequence number is used to check for duplicates if it has been resent. Sequence number will only ever come through as 0 when the network controller has been reset. The POS must only perform sequence number validation on sequence numbers greater than 0.

* Comms Version represents the current communications format revision for WaiterPad Orders. Note: This is a separate version number applicable to orders only, the two-way comms version is separate. If a different comms version is detected, it means an incompatible version of the WaiterPad handheld or Network Controller software is currently being used.

The current WaiterPad Order communications version number is 1.

Status (bit flags)

No Status	00000000b
Reserved	00000001b
Reserved	00000010b
Reserved	00000100b
Print Order To KPs	00001000b

4.1.4 PLU Sale

Specifies a PLU sale as part of an order.

Data Type	Data (ascii code)	Description
UInt8	02	lineType for a PLU Record
UInt16	* <variable>	Block length of the data to follow
UInt8	Product Class*	The product class of the sale (parent / child)
UInt32	PLU	PLU Number (1 - 4,000,000,000)
UInt8	Quantity	Quantity of sale
UInt8	Price Level	Price level of sale
UInt8	Modifier ID	Sale Modifier ID applied to sale (0 = no modifier, 1+ = modifier ID)
UInt8	Seat	Seat Number of sale (if applicable)
UInt8	Group ID	Printing Group override
Pstring	Price	Price override (decimal)
UInt16	Status*	See Below (bit flags)
UInt32	wplIndexNum	Unique record identifier for sale. (must be unique for the life of the table)

*The length of the packet varies depending on the size of the Pstring(s).

* Product Class: This specifies the type of PLU item. Valid values are as follows. Child PLU types attach to parent PLU types.

- 1 – Standard Product (parent)
- 2 – Side Dish (child)
- 4 – Condiment (child)
- 5 – Cooking Instruction (child)
- 7 – Message PLU (child)

Status (bit flags)

Voided Item	00000001b	// item is voided
Chase Item	00000010b	// item is chased (not actually ordered)
Weighed Item	00000100b	// needs a weight docket
Reserved	00001000b	// reserved
Don't Print	00010000b	// don't print item to KPs
Price Override	00100000b	// has a price override
Group Override	01000000b	// has a group override

4.1.5 Custom Product

Specifies a custom sale (no plu) with a price as part of an order.

Data Type	Data (ascii code)	Description
UInt8	03	lineType for a PLU Record
UInt16	* <variable>	Block length of the data to follow
UInt8	Quantity	Quantity of sale
UInt8	Seat	Seat Number of sale (if applicable)
UInt8	Group ID	Printing Group
Pstring	Price	Price (decimal)
Pstring	Product Name	Name of the Custom Product
UInt16	Status	See Below (bit flags)
UInt16	KP Printers*	KP Printer mask (use bit number for printer number)
UInt32	wplIndexNum	Unique record identifier for sale. (must be unique for the life of the table)

*The length of the packet varies depending on the size of the Pstring(s).

* KP Printers is a bit mask representing which kp printers the item is to be printed too. For example, a value of 11 (1011b) would be printers 1, 2 and 4.

Status (bit flags)

Voided Item	00000001b	// item is voided
Chase Item	00000010b	// item is chased (not actually ordered)
Weighed Item	00000100b	// needs a weight docket
Reserved	00001000b	// reserved
Don't Print	00010000b	// don't print item to KPs
Reserved	00100000b	// reserved
Reserved	01000000b	// reserved

4.1.6 Custom Message

Specifies a custom message as part of an order.

Data Type	Data (ascii code)	Description
UInt8	04	lineType for a PLU Record
UInt16	* <variable>	Block length of the data to follow
UInt8	Quantity	Quantity of sale
UInt8	Seat	Seat Number of sale (if applicable)
Pstring	Message Text	Custom Message Text
UInt16	Status	See Below (bit flags)
UInt16	KP Printers*	KP Printer mask (use bit number for printer number)
UInt32	wplIndexNum	Unique record identifier for sale. (must be unique for the life of the table)

*The length of the packet varies depending on the size of the Pstring(s).

* KP Printers is a bit mask representing which kp printers the item is to be printed too. For example, a value of 11 (1011b) would be printers 1, 2 and 4.

Status (bit flags)

Voided Item	00000001b	// item is voided
Chase Item	00000010b	// item is chased (not actually ordered)
Weighed Item	00000100b	// needs a weight docket
Reserved	00001000b	// reserved
Don't Print	00010000b	// don't print item to KPs
Reserved	00100000b	// reserved
Reserved	01000000b	// reserved

4.1.7 Card

Specifies a card to be attached to the order. Must be part of an order (ie a record that is encapsulated by an order header and order footer).

Data Type	Data (ascii code)	Description
UInt8	5	lineType for a card record
UInt16	...	Block length of the data to follow (2 + CardLen)
UInt8	Card Type	Type of card (2 = member card)
UInt8	Card Len	Length of the card string data to follow
String[CardLen]	Card Data	Card track data

Member Card Format

<six zeros>11<six digit member number><two zeros>

Card Data Example: 0000001100000200

would represent member 000002

Card Data Example: 0000001100123400

would represent member 001234

4.1.8 Adjustment (Discount / Surcharge)

Specifies an adjustment as part of an order.

Data Type	Data (ascii code)	Description
UInt8	06	lineType for an Adjustment record
UInt16	*<variable>	Block length of the data to follow
UInt16	ID	Adjustment ID
UInt32	wplIndexNum*	WaiterPads Unique record identifier for the selected item sale (if applicable).
UInt32	posIndexNum*	POS's unique index identifier for the selected item sale (if applicable).
Pstring	posTerminalID*	POS's terminal id for the selected item sale (if applicable).
UInt8	Watch Override Count	The number of adjustment watch overrides to follow
{ repeat Watch Override Count times		
<i>UInt16</i>	<i>Watch ID*</i>	<i>The ID of the adjustment watch</i>
<i>Pstring</i>	<i>RewardAmount*</i>	<i>Reward value of the watch (blank if not applicable)</i>
<i>Pstring</i>	<i>RewardPercentage*</i>	<i>Percentage value of the watch (blank if not applicable)</i>
}		

*The length of the packet varies depending on the size of the Pstring(s).

*wplIndexNum refers to the wplIndexNum of the selected item used by the adjustment (if applicable). Use 0 (null) if its not a selected item adjustment. (See concepts section on unique item identifiers for more information)

*posIndexNum refers to the posIndexNum of the selected item used by the adjustment (if applicable). Use 0 (null) if its not a selected item adjustment. (See concepts section on unique item identifiers for more information)

*posTerminalID refers to the posTerminalID of the selected item used by the adjustment (if applicable). Use blank string (2 spaces) if its not a selected item adjustment. (See concepts section on unique item identifiers for more information)

*WatchID refers to the unique watch identifier as specified in the XML specification for adjustment watches. This should be 0 if the adjustment uses adjustment level rewards, or the watch ID if the adjustment uses requirement level rewards. (see concepts section on adjustments for more information).

*Reward Amount specifies the fixed amount of an adjustment as determined by the WaiterPad (if applicable). This value is expected as a string representing the amount, therefore a value of \$1.00 would be sent as the string 1.00

*Reward Percent specifies the adjustment percentage entered by the user on the WaiterPad (if applicable). This value is expected as a string representing the percentage, therefore a value of 10% would be sent as the string 10, a value of 8.25% would be sent as the string 8.25.

4.1.9 Text Stamp

Specifies a text stamp as part of an order.

Data Type	Data (ascii code)	Description
UInt8	07	lineType for a Text Stamp Record
UInt16	*<variable>	Block length of the data to follow
UInt32	ID	Text Stamp ID
PString	Message Text	User entered text for the text stamp.

*The length of the packet varies depending on the size of the Pstring(s).

4.1.10 FastPay Payment

Specifies a payment. If the payment is being sent as part of an order (fastPay), then it must be encapsulated by an order header and order footer. There is no verification of payment value vs order value for fastPay.

If the payment is being sent separately, it does not require an order header and footer, however it will now be treated as a payment request and as such will follow request transmission rules (two-way comms).

Data Type	Data (ascii code)	Description
UInt8	08	lineType for fastpay payment record
UInt16	* <variable>	Block length of the data to follow
UInt8	Media Number	Media ID that payment is being made with
Pstring	Payment Amount*	Amount of the payment (Decimal)
Pstring	Tip Amount*	Extra tip amount to be added (if applicable) (decimal)
Pstring	Subtotal*	Assumed subtotal of the order (decimal)
UInt16	Status (bit flag)	See below
UInt8	Printer Num	Printer number for receipt (0 = pos default receipt printer)

*The length of the packet varies depending on the size of the Pstring(s).

The payment Pstrings are in decimal format. For example, to a payment amount of 123.45 is sent as (pstring would look like)

```

UInt8      6                // len of the string data being sent
String    123.45           // string data

```

Status (bit flags)

```

PrintBill      0x01  00000001  Print a receipt
PrintPaymentReceipt 0x02  00000010  Print a payment slip

```

4.1.11 Order Footer

All order based packets (ie sending PLU orders) must be encapsuated by an Order Header and Order Footer.

Data Type	Data (ascii code)	Description
UInt8	15	lineType for the order footer
UInt16	0	Block length of the data to follow

4.1.12 Comms Status Response

Specifies a status response as an acknowledgement of an order. POS will always return a status response to any order based packet.

Data Type	Data (ascii code)	Description
UInt8	Response Value	Response Value (see below)
UInt16	0	Reserved

Valid response values from POS for orders from the WaiterPad

ACK	242	// packet was received successfully
NAK	243	// an unknown error occurred with the packet data

If no response is received from the POS within 15 seconds, a timeout will occur.

4.2. XML Formats

4.2.1 PLU's

```
<?xml version="1.0" encoding="windows-1252" ?>
```

```
<atp xmlns:atp="http://xml.palmteq.com/atp">
```

```
<transaction>
```

```
<table name="POS:Plus">
```

```
<row id="1">
```

```
<col name="LongDescription">Garlic Bread</col>
```

```
<col name="TillDescription">Garlic Bread</col>
```

```
<col name="ReportingGroup">1</col>
```

```
<col name="PrintingGroup">1</col>
```

```
<col name="Class">1</col>
```

```
<col name="CourseBreakID">1</col>
```

```
<col name="LookupWindowID">1</col>
```

```
<col name="OrderNote">Baked garlic bread!</col>
```

```
<col name="Printers">00000009</col>
```

```
<col name="Flags1">00000000</col>
```

```
<col name="Flags2">00000000</col>
```

```
<col name="StockCountdown">False</col>
```

```
<col name="TaxableServiceLocation">0</col>
```

```
</row>
```

```
</table>
```

```
<table name="POS:PluPrices">
```

```
<row id="1">
```

```
<col name="PLU">1</col>
```

```
<col name="PriceSchemeID">0</col>
```

```
<col name="ModifierID">0</col>
```

```
<col name="PriceLevel">1</col>
```

```
<col name="Price">500</col>
```

```
</row>
```

```
</table>
```

```
</transaction>
```

```
</atp>
```

Database Table

PLU Number

Long Description (32 char max) [1]

Till Description (16 char max) [2]

Reporting Group Number [3]

Printing Group Number [3]

Product Class [4]

Course Break ID

ID of Lookup Window [11]

Order Note (512 char max) [5]

Printers (bit mask) [6]

Flags 1 (bit mask) [7]

Flags 2 (bit mask) [8]

Is this a Stock Countdown PLU?

Taxable Service Location [9]

Database Table

Internal Record ID (increment per record)

The PLU who's price is being specified

Price Scheme (ID) being used [10]

Modifier this price applies to [10]

Price Level for this price [10]

Actual Price (cents)

1. Long Description (32 Character Max) – This is the text displayed on the ordering window on the WaiterPOS as well as on the printed receipts.
2. Till Description (16 Character max) – This is the text displayed on the Pole Display if one is used.
3. Reporting Group is the group number under which the PLU figures are reported. Printing group is the group name that the PLU is printed under on the KP dockets. These must match existing group numbers in the Minor Groups configuration.
4. Product Class is the type of product the item is. This affects how the item is printed and whether it can have a price. Child products are linked to the Parent product above it when items are orderd on the POS or WP.
 - 1 – Standard Product (Parent Product)
 - 2 – Side Dish (Child Product)
 - 4 – Condiment (Child Product)
 - 5 – Cooking Instruction (Child Product)
 - 7 – Message (Child Product, no Price)
5. Order Note is a description of the product which is accessible on the Waiterpad by clicking the [note] icon next to the product, or by using the product enquiry key on the WaiterPOS. This field is usually only used on occasion to describe a certain cocktail or special of the day.

6. Printers is a bitmask of which printer(s) the PLU prints to. The value stored in the XML printer field is the 32 bit hex value of the printer code. Eg **00000009** is printer 1 and 4 only.
- 0 = No printers (0000000000000000b) 00000000h
 - 1 = Printer 1 only (0000000000000001b) 00000001h
 - 2 = Printer 2 only (0000000000000010b) 00000002h
 - 3 = Printer 1 and 2 only (0000000000000011b) 00000003h
 - 4 = Printer 3 only (0000000000000100b) 00000004h
 - 4 = Printer 1 and 4 only (0000000000000101b) 00000009h
 - Etc up to printer 16 max
7. **Flags 1** is the hex value of the PLU flags group 1 to 4. (See flags below)
- Format = 0000(pos flag 2)(pos flags 1)
 - Example = **00004001** = pluAffectsTax1 and pluSendToNC
8. **Flags 2** is the hex value of the PLU flags group 5 to 8. (See flags below)
- Format = 0000(wp flag 1)(nc flag 1)
 - Example = **0000C020** = plu6PromptForPrice, plu6PromptForQuantity and plu1UseThisGroup
9. Taxable Service Location is the override service location ID that the order tax calculation should use if this PLU is ordered. (0 = no override)

Flags Section	Binary	Hex	Possetup Flag Description
POS flag 1			
pluAdjustment1	10000000b	80h	Adjustment 4 Item
pluAdjustment2	01000000b	40h	Adjustment 3 Item
pluAdjustment3	00100000b	20h	Adjustment 2 Item
pluAdjustment4	00010000b	10h	Adjustment 1 Item
Not Used	00001000b	08h	
Not Used	00000100b	04h	
Not Used	00000010b	02h	
Not Used	00000001b	01h	
POS flag 2			
pluPromptForPrice	10000000b	80h	PLU Prompts for Price
pluSendToNC	01000000b	40h	PLU is sent to Network Controller
pluRollPriceOnReceipts	00100000b	20h	Child prices combine to parent on receipts
pluPriceByWeight	00010000b	10h	Price is calculated by weight
pluSpecialOfTheDay	00001000b	08h	Special of the Day
pluPromptForQty	00000100b	04h	Prompts for Quantity
pluAllowNegativePrice	00000010b	02h	Allow Negative Price
pluBarStockPLU	00000001b	01h	Use as BarStock PLU?
POS flag 3			
Reserved	10000000b	80h	
pluDontPromptForSeat	01000000b	40h	Don't prompt for seat number
Reserved	00100000b	20h	
pluScalePLU	00010000b	10h	Reads scale to determine weight
pluDontAllowSaleAtZeroPrice	00001000b	08h	Can't be sold for zero value
pluNotPartOfCourseBreaks	00000100b	04h	PLU is not part of any course break
pluNotUsed	00000010b	02h	
pluNonExportablePLU	00000001b	01h	Non Exportable PLU
NC flag 1			
plu1TicketPLU	10000000b	80h	Ticket PLU

plu1WeightDocket	01000000b	40h	Print Separate Price by Weight docket
plu1UseThisGroup	00100000b	20h	Modify Parents printing group
plu1PrintModifier	00010000b	10h	Print Modifier Order Text
plu1UseOwnPrinters	00001000b	08h	Use own Printer Code (Child products only)
pluNotUsed	00000100b	04h	Not used (was formerly Product Countdown)
plu1ParentUsesMyPrinter	00000010b	02h	Parent uses childs printers
plu1DontConsolidate	00000001b	01h	PLU doesnt get consolidated for printing

WP flag 1

plu6PromptForPrice	10000000b	80h	Prompts for Price?
plu6PromptForQuantity	01000000b	40h	Prompts for Quantity?
plu6InactivePLU	01000000b	20h	PLU is inactive

10. Prices are defined for each PLU per Price Level, per Modifier, per Price Scheme. A price Scheme of 0 is default, and a modifier ID of 0 is "unmodified". Price levels supported are 1 through 9. As such, a single PLU can be expected to contain many of these records defining its price at the various price levels, per modifier, per price scheme.

Price Level 0 = PLU Cost Price

Price Level 1 – 9 = PLU Prices 1 to 9

Price Level 10 = PLU Set Price (used as base price for price by weight items.)

4.2.2 Minor Groups (Printing / Reporting)

<pre> <?xml version="1.0" encoding="windows-1252" ?> <atp xmlns:atp="http://xml.palmteq.com/atp"> <transaction> <table name="POS:Groups"> <row id="1"> <col name="Name">Entrees</col> <col name="MainGroup">1</col> </row> </table> </transaction> </atp> </pre>	<i>Database Table</i> Group ID Group Name [1] Main Group ID
--	--

4.2.3 Main Groups

<pre> <?xml version="1.0" encoding="windows-1252" ?> <atp xmlns:atp="http://xml.palmteq.com/atp"> <transaction> <table name="POS:MainGroups"> <row id="1"> <col name="Name">Food</col> </row> </table> </transaction> </atp> </pre>	<i>Database Table</i> Main Group ID Main Group Name
---	---

4.2.4 Sale Modifiers

<pre> <?xml version="1.0" encoding="windows-1252" ?> <atp xmlns:atp="http://xml.palmteq.com/atp"> <transaction> <table name=" POS:Modifiers"> <row id="1"> <col name="Description">Double Shot</col> <col name="Prefix">DBL</col> <col name="StockMultiplier">2</col> <col name="PriceMode">0</col> <col name="PriceValue">0</col> <col name="AdjustmentGroupID">8</col> <col name="PrivilegeCardID">0</col> <col name="ReportingGroup">0</col> <col name="PrintingGroup">1</col> <col name="ProductClass">0</col> <col name="Flags1">00000000h</col> </row> </table> </transaction> </atp> </pre>	<div> <div>Database Table</div> <div> <div>Unique Modifier ID</div> <div>Modifier Description</div> <div>Modifier Prefix [1]</div> <div>Stock Multiplier</div> <div>Price Mode [2]</div> <div>Price value for price mode</div> <div>Adjustment group requirement [3]</div> <div>Privilege card requirement [4]</div> <div>Reporting Group Override [5]</div> <div>Printing Group Override [6]</div> <div>Product Class Override [7]</div> <div>Flags 1 [8]</div> </div> </div>
--	--

1. Modifier Prefix – This specifies the text that will be added to the front of the PLU order text.
2. Price Mode – This specifies how the price is calculated for modifier items
 - 0 = Use Modifier Price Table (from POS:PLUPrices table for matching modifier id)
 - 1 = Multiply Price – Multiply by the value specified in the PriceValue field
 - 2 = Price Adjustment – Adjust the price (+ / -) by the value specified in the PriceValue field
 - 3 = Set Price – Set the price to the value specified in the PriceValue field
 - 4 = PLU Modifier – Change the PLU number by the offset specified in the PriceValue field.
3. Adjustment Group Requirement – Only PLU sale items that belong to the specified adjustment group can have the modifier applied to them. (0 = no requirement)
4. Privilege Card Requirement – The specified privilege card ID must be linked to the order before the sale modifier can be applied to any items. (0 = no requirement)
5. Reporting Group Override – Currently not implemented, set as 0.
6. Printing Group Override – Changes the printing group of the item that the modifier is applied to. (0 = no override)
7. Product Class Override – Changes the product class of the PLU item the modifier is applied to. (0 = no override)
8. Modifier Flags

Flag	Binary (MSB \leftrightarrow LSB)	Flag Description
ApplyToSelectedItem	00000000000000000000000000000001b	Apply to child item if applicable

4.2.5 Terminals

<pre> <?xml version="1.0" encoding="windows-1252" ?> <atp xmlns:atp="http://xml.palmteq.com/atp"> <transaction> <table name=" POS:Terminals"> <row id="P1"> <col name="Name">POS Terminal 1</col> <col name="HostName">pos1</col> <col name="HostIP">192.168.100.101</col> <col name="Flag1">00000000h</col> </row> </table> </transaction> </atp> </pre>	<p><i>Database Table</i></p> <p>Unique Terminal ID (2 digits)</p> <p>Terminal Description</p> <p>Host Name of the terminal</p> <p>IP Address of the terminal</p> <p>Terminal flags [1]</p>
---	--

<i>Flag</i>	<i>Binary (MSB \leftrightarrow LSB)</i>	<i>Flag Description</i>
BackofficePC	000000000000000000000000000001	Terminal is a backoffice PC and cannot receive orders

The Backoffice PC flag indicates to menuedit that the terminal entry should not be synchronised to the network controller. This means the network controller will never try to connect to that terminal to send incoming orders to.

4.2.6 Table Ranges

<pre> <?xml version="1.0" encoding="windows-1252" ?> <atp xmlns:atp="http://xml.palmteq.com/atp"> <transaction> <table name=" ME:TableRanges"> <row id="1"> <col name="Min"> 1</col> <col name="Max">100</col> <col name="Location">1</col> </row> </table> </transaction> </atp> </pre>	<p><i>Database Table</i></p> <p>Unique Record ID</p> <p>Low limit for table range</p> <p>High limit for table range</p> <p>Service Location ID</p>
--	--

Note: The min to max range for each table range should not overlap with any other table range record.