

---

# Command Alkon

*Solutions To Build On™*

---



**COMMAND**  
*series*™

---

## **Industry Standard Batch Control Interface Specification (COMMANDseries to Third-Party Batch Control)**

15-Jan-02



© 2001 Command Alkon Incorporated. All rights reserved.

Command Alkon Incorporated believes the statements contained herein are accurate as of the date of publication of this document. HOWEVER, COMMAND ALKON INCORPORATED HEREBY DISCLAIMS ALL WARRANTIES EITHER EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANT OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event will Command Alkon Incorporated be liable for any damages, including any lost profits, lost savings, or other incidental or consequential damage arising out of the use of or inability to use any information provided through this publication, even if Command Alkon Incorporated has been advised of the possibility of such damages, or for any claim by any other party. Some states do not allow the limitation or exclusion of liability or consequential damages, so the above limitation may not apply.

This information is not intended to be an assertion of future action. The contents of this document are for informational purposes only and are subject to change without notice. Command Alkon Incorporated expressly reserves the right to change or withdraw current products that may or may not have the same characteristics listed in this publication. Should Command Alkon Incorporated modify its products in a way that may affect the information contained in this publication, Command Alkon Incorporated assumes no obligation whatever to inform any user of the modification.

This publication may contain technical inaccuracies or typographical errors. Command Alkon Incorporated assumes no responsibility for any errors that may appear in this manual. This publication is intended only for the direct benefit of users of Command Alkon Incorporated products. This manual may not be used for any purposes other than those for which it is provided. Information disclosed herein was originated by and is the property of Command Alkon Incorporated, and, except for rights granted by written consent, such information shall not be disclosed, disseminated, or duplicated in whole or in part. Command Alkon Incorporated reserves all patent, proprietary, design, manufacturing, use, and reproduction rights.

### **Command Alkon Incorporated**

1800 International Park Drive, Suite 400  
Birmingham, AL 35243-4232  
(205) 879-3282

605 East Safari Parkway, Suite C-4  
Grand Prairie, TX 75050-2325  
(972) 262-2692

5168 Blazer Parkway  
Dublin, OH 43017-1339  
(614) 799-6650

[www.commandalkon.com](http://www.commandalkon.com)

Microsoft and Windows are trademarks of Microsoft Corporation of Redmond, Washington.

# Contents

---

<b>CONTENTS .....</b>	<b>2</b>
<b>HARDWARE REQUIREMENTS.....</b>	<b>4</b>
OVERVIEW .....	4
GENERAL CONSIDERATIONS .....	4
PHYSICAL CONSIDERATIONS .....	5
<i>Connector .....</i>	<i>5</i>
<i>Configuration .....</i>	<i>5</i>
ELECTRICAL CONSIDERATIONS .....	6
BINARY CRC TO ASCII CRC .....	7
<b>SOFTWARE REQUIREMENTS.....</b>	<b>8</b>
TRANSMISSION PROTOCOL – MODIFIED BISYSNC.....	8
<i>CAI Dispatch Computer to BCC.....</i>	<i>8</i>
<i>BCC to CAI Dispatch Computer.....</i>	<i>9</i>
<i>Status Codes of BCC .....</i>	<i>10</i>
<i>Parity Checks .....</i>	<i>13</i>
MESSAGE LEVEL .....	13
EXCEPTIONS.....	13
<b>TICKET FORMAT.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
OVERVIEW .....	15
THE COMMUNICATIONS CHANNEL.....	15
SYNC TRANSMISSION .....	16
TICKET TRANSMISSION .....	16
<i>Non-recoverable errors include:.....</i>	<i>18</i>
<i>Ticket Header Fields.....</i>	<i>18</i>
<i>Typical Ticket Script.....</i>	<i>19</i>
<i>Ticket Data Fields Layout.....</i>	<i>20</i>
<i>Extra Products - Sent If On Ticket.....</i>	<i>21</i>
<i>Minimum Load Charge Fields.....</i>	<i>22</i>
<i>Price Totals – COD Only.....</i>	<i>23</i>
<i>General Purpose Fields .....</i>	<i>23</i>
<i>Admix/Water Trims.....</i>	<i>24</i>

<i>General Purpose Fields</i> .....	24
<i>General Purpose Fields (cont'd)</i> .....	27
<i>New Standard Fields</i> .....	28
MIX TRANSMISSION .....	30
<i>Mix Transmission Headers</i> .....	31
<i>Mix Transmission Fields Layout</i> .....	32
RETURN BATCH RESULTS .....	33
<i>Return Batch Results</i> .....	34
<i>Typical Brief Batch Results Script</i> .....	34
<i>Brief Batch Results Data Fields Layout</i> .....	35
<i>Typical Extended Batch Results Script</i> .....	36
MATERIAL RECEIPTS .....	38
<i>Typical Return Material Inventory Script</i> .....	39
<i>Return Material Inventory Data Fields Layout</i> .....	39
PURGE BATCH RESULTS .....	40
SEND NEXT DELIVERY .....	41
<i>How Send Next Delivery Data is Packaged</i> .....	42
<i>Typical Send Next Delivery Script</i> .....	42
<i>Layout of Send Next Delivery Data Fields</i> .....	43
<b>APPENDIX A. TCP/IP INTERFACE .....</b>	<b>44</b>
INTRODUCTION – SOURCE NOTES .....	44
SOFTWARE SPECIFICATION .....	44
<i>Transmission Protocol for TCP/IP (See Note 3)</i> .....	44
<i>Dispatch Computer To BCC</i> .....	44
<i>BCC To Dispatch Computer</i> .....	45
LOG EXAMPLES .....	46
TABLE OF NONPRINTABLE CHARACTERS .....	50
EXAMPLE C CODE .....	51

# Hardware Requirements

---

## Overview

This specification defines the hardware and software requirements placed on an interface to a Batch Control Computer. This interface is designed to support communications from Command Alkon's dispatching computer to any batching computer capable of handling this protocol as defined. Typical information sent would include batching instructions, ticket printing information, or mix designs. The interface is also capable of supporting the return of each batch's actual weights to Command Alkon's dispatching computer for analysis and inventory control.

The interface can be implemented in two parts:

1. A one-way interface can be created to support ticket download only.
2. A two-way interface can be created to support ticket download, as well as batch mix design downloads and weight uploads.

## General Considerations

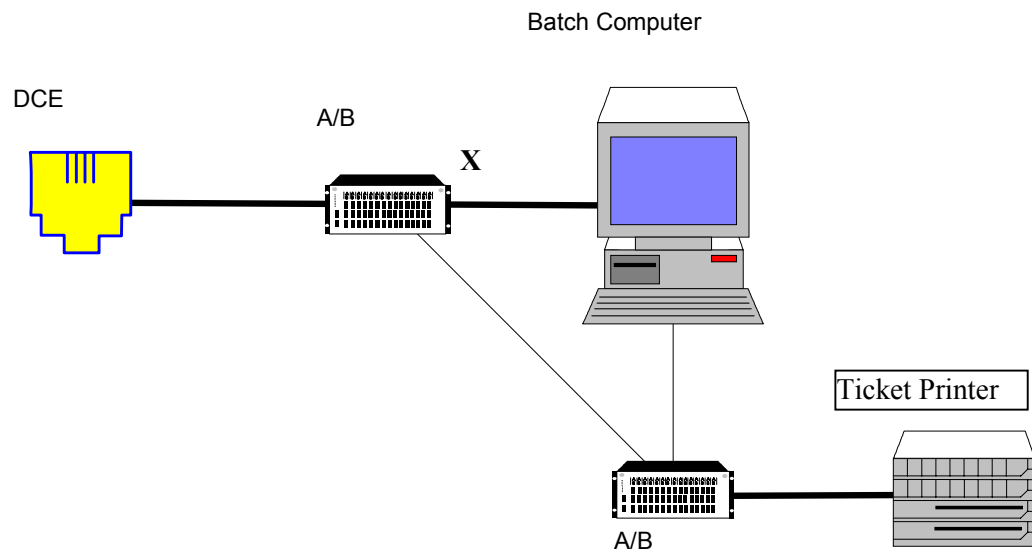
- The interface does not specify the data communication equipment, but it should be comprehensive enough to interface with most modern telecommunications devices.
- The interface will support direct connections between Command Alkon's dispatch computer and the batch computer as long as the 'rules' for a null modem are used.
- The interface will support point-to-point connections for two-way interfaces, or multi-drop connections for one-way interfaces. In either case, Command Alkon's dispatching computer is considered the 'MASTER' and the batch computer is considered the 'SLAVE'.
- The interface specifies a full-duplex communication channel and a half-duplex communications protocol (modified BISYNC). This was done to avoid delays and software requirements of a half-duplex communication channel.
- The Batch Control Computer may also be referenced as BCC. The CAI Dispatch System Computer may also be referenced as DSP.

# Physical Considerations

## Connector

A 25-pin dataphone connector (DB-25) with female contacts. This is the connector on the batch computer itself.

## Configuration



DCE is specified as:

- **Null modem**
- **Communications multiplexer**
- **Modem**

A/B is a multi-contact two position switch and is optional.

**X** marks the interface specification point.

## Electrical Considerations

The electrical interface is as specified in EIA RS-232C. The following interchange circuits are supported with the batch computer acting as the Data Terminal Equipment (DTE):

Interchange Circuit	Pin #	Description	Direction
AA	1	Protective Ground	N/A
AB	7	Signal Ground	N/A
BA	2	Transmitted Data	To DCE
BB	3	Received Data	From DCE
CA	4	Request to Send	To DCE
CB	5	Clear to Send	From DCE
CF	8	Data Carrier Detect	From DCE
CD	20	Data Terminal Ready	To DCE

Please note that this is not a complete implementation of the RS-232C specification. These signals were chosen to provide support for data communication channels which were dial-up, leased multi-drop, or leased point-to-point connections.

- Baud Rate     **2400-28.8 bps**
- Timing        **Asynchronous**
- Line           **Full-duplex 2 wire, dial-up or leased line**
- Line Level     **8 data bits - Non-ASCII CRC, 1 start bit, 1 stop bit**

Beginning with COMADS Version BC5.D, there is support for an ASCII representation of the 2 CRC bytes. The CRC value and ordering remains the same, only it will now be 4 ASCII bytes long.

**Note:** This is recommended when using dial-up connections or any communications equipment that uses X-on / X-off flow control protocol.



## Binary CRC to ASCII CRC

Binary CRC to ASCII CRC is used to avoid embedding possible control characters in the data stream which potentially affects intermediate automatic switching devices (i.e. the SAS box).

- Example

A block of data has a CRC of 0X0304 (304 HEX). This is a 16-bit binary value, but it consists of lower ASCII characters (unprintable) which are typically used for control codes in switch box protocols. If we expand this to a 4-byte ASCII value, prevents the possibility of having unprintable characters in the data stream. The binary value is expanded by substituting an ASCII character for each 4-bit nibble in the 16-bit binary CRC.

0X0304 =	0000	0011	0000	0100	binary
	↓	↓	↓	↓	
ASCII characters	'0'	'3'	'0'	'4'	

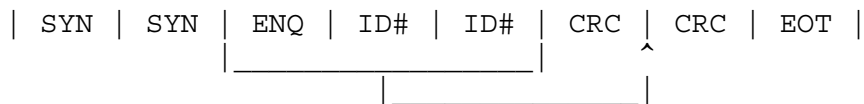
So instead of a 2-byte HEX value of 0X0304, a 4-byte ASCII value 0304 is used.

# Software Requirements

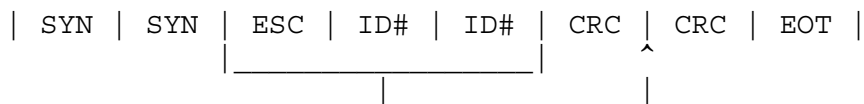
## Transmission Protocol – Modified BISYSNC

### CAI Dispatch Computer to BCC

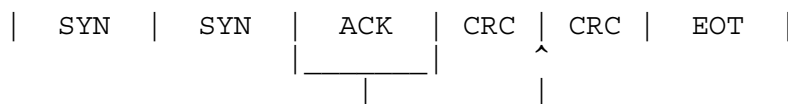
- **WAKE-UP/POLL station**



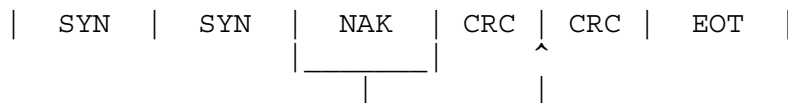
- **Return to IDLE (see Note 1)**



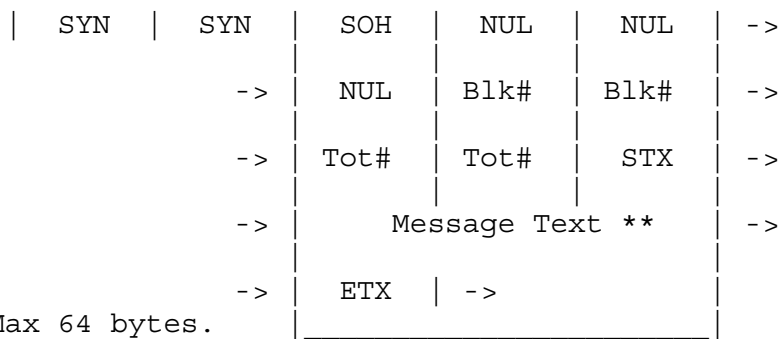
- **Response to Message Block**



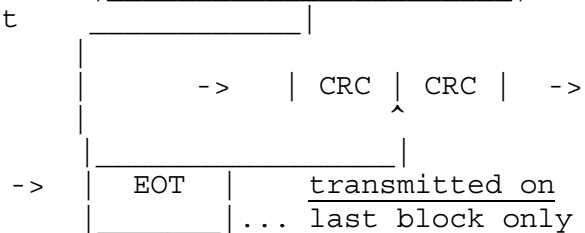
- **Response to transmission error/data error**



- **Message Block**

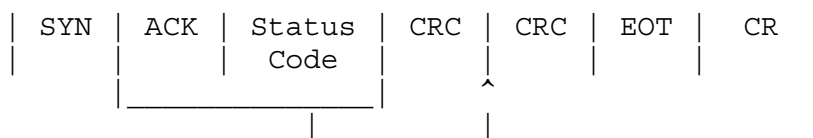


Max 64 bytes.  
Only the last  
block can be  
less than 64  
bytes long.

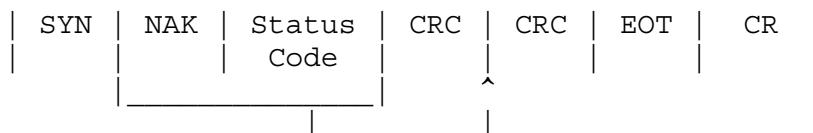


## BCC to CAI Dispatch Computer

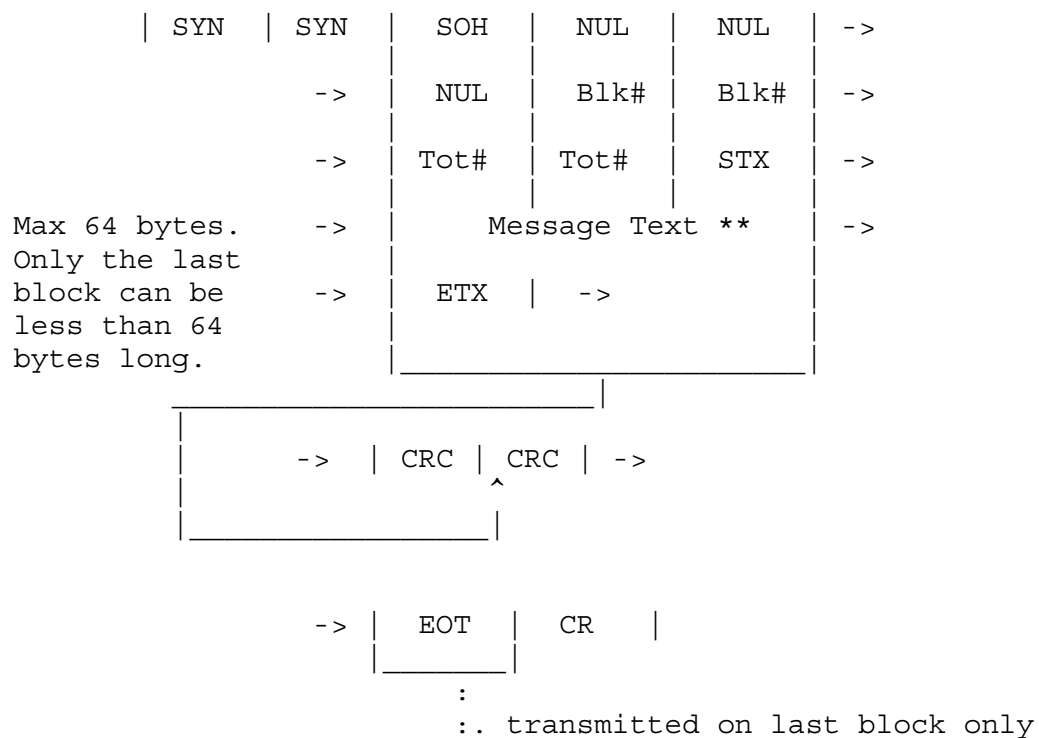
- **Response to WAKE-UP/POLL/Message Block**



- **Response to Transmission Error**



- **Message Block**



**Note:** When the DSP sends a Return to IDLE a BCC that is 'awake', the BCC simply returns to 'sleep' with no transmitted reply.

## Status Codes of BCC

The purpose of the BCC status code is to allow the BCC to give a meaningful response to the latest interaction with the CAI dispatch system. It should allow the dispatch system to quickly determine if there was an error and what action to take in response.

A single character should be used to represent the BCC status code. The statuses should be grouped in two broad categories: non-fatal statuses, representing informative type data to the CAI dispatch system; and fatal statuses, which essentially indicate a failure of the BCC to carry out the last interaction.

In representing these status codes there is a convention of informing the CAI dispatch system of pending batch results held in the BCC. The current status code should be sent in its lower-case equivalent if batch results are pending. For example, if the BCC determines that the current status is 'All OK', then it should send the letter 'A' as a status code. However, if there are batch results pending, then this status code should be the letter 'a', its lower-case equivalent.

On the following pages there is a table of defined status codes, their causes, and suggested error recovery procedures.

### Non-Fatal Status Codes

Status Code	Cause	Recovery
A	All OK	None required.
a	All OK, batch result pending	Poll for batch results after completion of current transmission sequence.

## Fatal Status Codes

Status Code	Cause	Recovery
B	Format/Syntax error	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
b	Format/Syntax error, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
C	Product name mismatch	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
c	Product name mismatch, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
D	BCC buffer full	The DSP should wait five minutes, then retry transmission. If it fails on the second attempt, alert the operator.
d	BCC buffer full, batch results pending	The DSP should wait five minutes, then retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
F	Message too long	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
f	Message too long, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
H	Ticket # already on queue	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
h	Ticket # already on queue, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
I	Batching in progress	The DSP should wait a few moments then retry transmission.
i	Batching in progress, batch results pending	The DSP should wait a few moments, then retry transmission and poll for batch results.

### Fatal Status Codes (cont.)

Status Code	Cause	Recovery
J	Ticket not available	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
j	Ticket not available, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
M	Mix Design error	The DSP should retry transmission. If it fails on the second attempt, the DSP should alert the operator.
m	Mix Design error, batch results pending	The DSP should retry transmission. If it fails on the second attempt, alert the operator, then poll for batch results.
N	Plant not ready	The DSP should wait for a few moments then retry transmission, or signal the operator to contact the plant.
n	Plant not ready, batch results pending	The DSP should wait for a few moments then retry transmission, or signal the operator to contact the plant, then poll for batch results.
P	Block mismatch	The DSP should abort the present sequence and begin over. If it fails on the second attempt, the DSP should alert the operator.
p	Block mismatch, batch results pending	The DSP should abort the present sequence and begin over. If it fails on the second attempt, the DSP should alert the operator, then poll for batch results.
R	Receive error	The DSP should retry transmission. If it fails on four consecutive retries, the DSP should alert the operator.
r	Receive error, batch results pending	The DSP should retry transmission. If it fails on four consecutive retries, the DSP should alert the operator, then poll for batch results.
T	Time-out	The DSP should abort the present sequence and begin over. If it fails on the second attempt, the DSP should alert the operator.
t	Time-out, batch results pending	The DSP should abort the present sequence and begin over. If it fails on the second attempt, the DSP should alert the operator, then poll for batch results.

## Parity Checks

A single parity check occurs on each message transmission. This parity check occurs over the length of the message block, utilizing CRC (Cyclic Redundancy Checking). This checking is based on the CCITT-CRC polynomial,  $X^{16} + X^{12} + X^5 + 1$ . This polynomial is applied to the entire message bit stream to produce a 16-bit checking word. The 16-bit checking word appears as two bytes, low order first, in the message transmission. When ASCII CRC's are selected, these two bytes are expanded to four ASCII bytes. The value and ordering remains the same. For further technical descriptions of this error detection technique, contact Command Alkon's Engineering department. When ASCII CRC's are selected, this 16-bit value is expanded to 4 bytes.

## Message Level

These message formats evolved from current industry standards for Dispatch/Batch Control Interface specifications. Embed these message formats in a modified BISYNC protocol for purposes of transmission error detection. Below are some other aspects of the transmission scheme that are impacted by this protocol.

## Exceptions

- The field header information will only be transmitted once at the beginning of a field segment. It will not be transmitted between each field. For example, once you transmit a T007 field header it will not be necessary to send it again unless another type of field header is sent. This eliminates the need to send this redundant data with each field.
- At the block level, the DSP should have received the BCC ACK/NAK response within 5 seconds. If it fails to receive an anticipated reply within 5 seconds, it must begin an error recovery procedure with the assumption that a NAK was received.
- Before the BCC can handle any ticket transmission from the CAI dispatch computer it must be in the 'awake' mode. To put the BCC into the 'awake' mode, the WAKE-UP/POLL message must be sent to and acknowledged by the BCC. The BCC at this time should be 'awake' and ready to receive ticket transmissions. The BCC should stay 'awake' as long as there is transmission activity. If the BCC fails to receive any transmissions from the dispatch computer for a period of 5 minutes, the BCC should automatically go back 'asleep'. At this point another WAKE-UP/POLL message will be sent to put the BCC back into the 'awake' mode.

- Only active fields should transmit. If a field does not transmit, it should be assumed to contain 'spaces' if text and zeroes if numeric, unless otherwise noted.
- Field lengths must be variable. Decimal places should be sent for numeric fields. All alpha fields should be left justified and all numeric fields will be right justified.
- Only a carriage return (CR) may be used as a field delimiter.
- The CAI dispatch computer generates 'keep-alive' traffic with each BCC. This means that each BCC must handle a transmission from the dispatch computer within 5 minutes or it will alert the batch plant operator that the line is down.
- Once a transmission sequence begins, it must be completely and successfully transmitted before any other type of transmission can begin. If the CAI dispatch computer fails to receive the 'Success Reply' on a ticket transmission, it assumes the ticket failed to be inserted in the BCC ticket queue. At this point, the dispatch computer will re-transmit the entire ticket over in order to assure its placement on the BCC ticket queue.
- As a general rule, anytime there is a failure to receive an anticipated 'ACK', it is assumed to be a received 'NAK' with a CRC error. Similarly, lack of an anticipated 'Success Reply' is construed as a 'Failure Reply'.



# Interface Protocol

---

## Overview

Each morning the Command Alkon dispatch computer sends a WAKEUP message to each BCC site. This ensures that each system is active and online. Additionally, synchronization information is sent at this time. A typical start-up sequence for each BCC could be as follows:

1. Send a WAKEUP message to BCC, ensure ACK message received.
2. Send Sync Transmission with Date/Time information, ensure ACK information received followed by Success/Failure message.
3. Send IDLE message to put BCC back to sleep.

Throughout the day, the Command Alkon dispatch system transmits to the BCC system instructions to load trucks. The information transmitted consists of information required to print the ticket and batch the load. The BCC must queue this information until batch time. The queue should handle a maximum of 10 tickets before being declared full.

Ticket numbers are assigned sequentially by the Command Alkon dispatch system. If several BCC systems are interfaced with a single dispatch system, no attempt will be made to match the ticket number known to the BCC system with the pre-printed numbers. These numbers will be used for stock control purposes, which appear on forms at the batch plants.

## The Communications Channel

The technical aspects of the communications channel have been discussed previously in this document. All transmissions must be imbedded in the BISYNC protocol. The only other points that remain to be discussed are as follows:

- The ID #'s that are transmitted with the WAKE-UP/POLL/IDLE messages from the dispatch allow a unique address to be assigned to each BCC. This selective addressing allows for a multi-drop system. These numbers are equivalent to the Batch Computer ID #'s of the Sync Transmissions.
- If the dispatch system detects that the BCC system is down, the system can be reverted to a manual mode of operation. The A/B switches at the BCC site should be switched to allow the data

transmissions to be routed to the ticket printer. The dispatch system then alters its ticket format for direct printout by the ticket printer. The batchman can then read the tickets as they arrive and batch the loads manually.

## Sync Transmission

The sync transmission is used to synchronize the time/date of the BCC with the dispatch system. This function is normally done at the beginning of the day, but it can be performed at any time. When the dispatch sends the sync transmission, it must wait for the reply from the BCC. The ID info field should equal the ID # of the WAKE-UP/POLL/IDLE messages. The '+' means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		W001[Date Time]+	DD-MMM-YY HH:MM ex: 01-Jan-84 13:10	X
	X	W017[StatusCode]000000[ID info]+	Success Reply	
	X	W021[StatusCode]000000[ID info]+	Failure Reply	

## Ticket Transmission

The ticket messages follow the Ticket Transmission Syntax. It is important to start by reviewing this syntax and then trying to understand the mechanics of ticket transmission.

Only one ticket transmission can be active at one time. Only after a ticket transmission has been received and fully acknowledged can transmission of the next ticket begin. A full acknowledgment consists of an ACK for each block and a Success Reply to signal that the entire ticket appears to be good. If the ticket violates the syntax rules, a Failure Reply should be sent.

In the event an ACK is not received on a single block, only that block is re-transmitted. If a Success Reply is not received, the entire ticket re-transmits.

A typical ticket transmission is shown below:

1. The dispatch computer has a ticket ready to send. It sends a WAKE-UP to the BCC. Then it awaits ACK/NAK.
2. The BCC receives the WAKE-UP and responds with ACK.
3. The dispatch computer sends the first block. Then it awaits ACK/NAK.
4. The BCC receives the first block OK. It responds with ACK.

5. The dispatch computer sends the second block. Then it awaits ACK/NAK.
6. The BCC receives the second block with CRC error. It responds with NAK.
7. The dispatch computer re-sends the second block. Then it awaits ACK/NAK.
8. The BCC receives the second block OK. It responds with ACK.
9. The dispatch computer sends the final block. Then it awaits ACK/NAK.
10. The BCC receives the final block OK. It responds with ACK.
11. The dispatch computer awaits Success Reply.
12. The BCC checks entire ticket for validity. If OK, then it sends a Success Reply.
13. The ticket transmission is complete.

Ticket information is transmitted via "T fields". This information is used to batch a load of concrete. This information can appear on the ticket as well as affect the nature of the batch. These "T fields" can appear in any order as long as they respect the requirements for the "begin/end" fields.

Every ticket starts with a "begin" field and ends with an "end" field. Along with each "begin" and "end" marker, there is a ticket number. For any given ticket the "begin" ticket number must match the "end" ticket number.

On the following page is an example ticket transmission. The first field sent is the ticket number. This "begin" ticket number will be compared against the "end" ticket number to ensure that they are identical. If they fail to match, the BCC should request a re-transmission of the entire ticket.

Next follow the T007 fields. These contain data that describes the concrete to be batched. Notice that T007 is sent only once, as opposed to multiple times. Once the field type is set, it remains that until a different one is sent. However the T007 field identifier could be sent with each field, with the penalty of extra transmission time.

If the entire ticket is sent without a non-recoverable error, then the BCC must send a Success Reply (T017). However, if the ticket has been sent with a non-recoverable error, then the BCC should send a Failure Reply (T021).

## Non-recoverable errors include:

- Failure to match "begin" and "end" ticket numbers
- Failure of data to meet type/format specifications
- Re-transmission of a single block more than 4 times
- Failure to write ticket record to BCC ticket file

If a non-recoverable error is detected, the lack of a Success Reply (T017) will cause the dispatch system to re-transmit the entire ticket. This cycle may repeat itself indefinitely; therefore, it is important that the dispatch system enforce error recovery.

After a ticket has been completed and before it has been batched, the interface should allow a dispatcher to cancel a ticket. This is the only change allowed by the dispatch system to a dispatched ticket. These messages are sent to the BCC system for the appropriate action.

## Ticket Header Fields

The following fields represent the "header" type fields in a ticket transmission. The "+" means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		T002[Ticket #]+	Begin Ticket	
X		T003[Ticket #]+	End Ticket	X
X		T006[Ticket #]+	Cancel Ticket	X
X		T007[Field #] [Data]+	Ticket Data Field	
	X	T017[Status Code] [Ticket #]+	Success Reply	
	X	T021[Status Code] [Ticket #]+	Failure Reply	

## Typical Ticket Script

The following is a typical example of ticket transfer. The modified BISYNC protocol has been omitted for clarity. The "+" means carriage return.

HeaderField#Data	Comments
T00212345678+	Begin Ticket #12345678
T00700101+	Plant #01
00212345678+	Ticket #12345678
003345+	Truck #345
0048.00+	Load Size 8.00 yds
005ABCD1234+	Mix Name ABCD1234
0063000 PSI 3/4+	Mix Description
0072.50+	Slump
008BRIDGE+	Use
00912-Dec-84+	Date
0100+	Customer ID
011ACME CONST+	Customer Name
014777+	Driver #
015A.J.FOYT+	Driver Name
016233 EASY ST+	Delivery Addr
017ANYTOWN TX+	Delivery Addr
018QRT2221+	Purchase Order #QRT2221
01913+	Order #13
0201.50+	Resold Amount 1.50 yds
021KNOCK ON DOOR+	Delivery Instructions
022REAL HARD !!+	Delivery Instructions
0247+	Tax Code
02511:30+	Time Due 11:30 am
0268.00+	Load Size 8.00 yds
0278.00+	Cumulative Qty 8.00 yds
02824.00+	Ordered Qty 24.00 yds
029ABCD1234+	Mix Name ABCD1234
0303000 PSI 3/4"+	Mix Description
031YARD+	Unit of measure
03254.00+	Mix Price \$54.00
033432.00+	Mix Extended \$432.00
03450+	Extra Product Qty 50
03523AAC+	Extra Product Name 23AAC
036REBAR+	Extra Product Desc
037FT+	Extra Product Unit
0380.50+	Extra Product Price \$0.50
03925.00+	Extra Product Amt \$25.00
072457.00+	Subtotal \$457.00
07327.42+	Tax Total \$27.42
0820+	Local Ticket Change Allowed
T00312345678+	End Ticket #12345678

## Ticket Data Fields Layout

Below is a description of each of the allowed entries for data in the T007 type fields. The Max entry is the maximum length of each field. The "\*" character denotes values which directly affect a given batch.

Field #	Field Description	Type/Max	Format	Req
001	Plant ID #	Num/2		X
002	Ticket #	Num/8		X
003	Truck #	Num/4 *		X
004	Load Size	Num/5 *	NN.NN	X
005	Mix Code (Batchbook ID)	AN/8 *		X
006	Mix Description	AN/24		
007	Slump	Num/7	NNNN.NN	X
008	Use	AN/16		
009	Date	AN/9	NN-AAA-NN	
010	Customer ID	AN/8		
011	Customer Name #1	AN/32		
012	Customer Name #2	AN/32		
013	Customer Name #3	AN/32		
014	Driver #	Num/8		
015	Driver Name	AN/32		
016	Delivery Address #1	AN/32		
017	Delivery Address #2	AN/32		
018	Purchase Order #	AN/16		
019	Order #	Num/8		X
020	Quantity on Board (Adjust)	Num/5 *	NN.NN	
021	Delivery Instructions 1	AN/32		
022	Delivery Instructions 2	AN/32		
023	Delivery Instructions 3	AN/32		
024	Tax Code	Num/2		
025	Time Due on Job	AN/5		
026	RESERVED (was Load Size)	AN/8		
027	Cumulative Quantity	Num/8	NNNNN.NN	
028	Ordered Quantity	Num/8	NNNNN.NN	
029	Mix Product Code	AN/8		
030	Mix Product Description	AN/40		
031	Unit of Measure	AN/4		
032	Mix Price	Num/12	NNNNNNNNN.NN	
033	Mix Extended Amount	Num/12	NNNNNNNNN.NN	

## Extra Products or Sundry Charges - Sent If On Ticket

Field #	Field Description	Type/Max	Format
034	Extra Product #1 Load Qty	Num/12	NNNNNNNNNN.NN
035	Extra Product #1 Name	AN/8	
036	Extra Product #1 Desc.	AN/16	
037	Extra Product #1 U/M	AN/4	
038	Extra Product #1 Price	Num/12	NNNNNNNNNN.NN
039	Extra Product #1 Amount	Num/12	NNNNNNNNNN.NN
040	Extra Product #2 Load Qty	Num/12	NNNNNNNNNN.NN
041	Extra Product #2 Name	AN/8	
042	Extra Product #2 Desc.	AN/16	
043	Extra Product #2 U/M	AN/4	
044	Extra Product #2 Price	Num/12	NNNNNNNNNN.NN
045	Extra Product #2 Amount	Num/12	NNNNNNNNNN.NN
046	Extra Product #3 Load Qty	Num/12	NNNNNNNNNN.NN
047	Extra Product #3 Name	AN/8	
048	Extra Product #3 Desc.	AN/16	
049	Extra Product #3 U/M	AN/4	
050	Extra Product #3 Price	Num/12	NNNNNNNNNN.NN
051	Extra Product #3 Amount	Num/12	NNNNNNNNNN.NN
052	Extra Product #4 Load Qty	Num/12	NNNNNNNNNN.NN
053	Extra Product #4 Name	AN/8	
054	Extra Product #4 Desc.	AN/16	
055	Extra Product #4 U/M	AN/4	
056	Extra Product #4 Price	Num/12	NNNNNNNNNN.NN
057	Extra Product #4 Amount	Num/12	NNNNNNNNNN.NN
058	Extra Product #5 Load Qty	Num/12	NNNNNNNNNN.NN
059	Extra Product #5 Name	AN/8	
060	Extra Product #5 Desc.	AN/16	
061	Extra Product #5 U/M	AN/4	
062	Extra Product #5 Price	Num/12	NNNNNNNNNN.NN
063	Extra Product #5 Amount	Num/12	NNNNNNNNNN.NN
064	Extra Product #6 Load Qty	Num/12	NNNNNNNNNN.NN
065	Extra Product #6 Name	AN/8	
066	Extra Product #6 Desc.	AN/16	
067	Extra Product #6 U/M	AN/4	
068	Extra Product #6 Price	Num/12	NNNNNNNNNN.NN
069	Extra Product #6 Amount	Num/12	NNNNNNNNNN.NN
253	Extra Product #7 Load Qty	Num/12	NNNNNNNNNN.NN
254	Extra Product #7 Name	AN/8	
255	Extra Product #7 Desc.	AN/16	
256	Extra Product #7 U/M	AN/4	
257	Extra Product #7 Price	Num/12	NNNNNNNNNN.NN
258	Extra Product #7 Amount	Num/12	NNNNNNNNNN.NN
259	Extra Product #8 Load Qty	Num/12	NNNNNNNNNN.NN
260	Extra Product #8 Name	AN/8	
261	Extra Product #8 Desc.	AN/16	
262	Extra Product #8 U/M	AN/4	
263	Extra Product #8 Price	Num/12	NNNNNNNNNN.NN
264	Extra Product #8 Amount	Num/12	NNNNNNNNNN.NN
265	Extra Product #9 Load Qty	Num/12	NNNNNNNNNN.NN
266	Extra Product #9 Name	AN/8	

267	Extra Product #9 Desc.	AN/16	
268	Extra Product #9 U/M	AN/4	
269	Extra Product #9 Price	Num/12	NNNNNNNNNN.NN
270	Extra Product #9 Amount	Num/12	NNNNNNNNNN.NN
271	Extra Product #10 Load Qty	Num/12	NNNNNNNNNN.NN
272	Extra Product #10 Name	AN/8	
273	Extra Product #10 Desc.	AN/16	
274	Extra Product #10 U/M	AN/4	
275	Extra Product #10 Price	Num/12	NNNNNNNNNN.NN
276	Extra Product #10 Amount	Num/12	NNNNNNNNNN.NN
277	Extra Product #11 Load Qty	Num/12	NNNNNNNNNN.NN
278	Extra Product #11 Name	AN/8	
279	Extra Product #11 Desc.	AN/16	
280	Extra Product #11 U/M	AN/4	
281	Extra Product #11 Price	Num/12	NNNNNNNNNN.NN
282	Extra Product #11 Amount	Num/12	NNNNNNNNNN.NN
283	Extra Product #12 Load Qty	Num/12	NNNNNNNNNN.NN
284	Extra Product #12 Name	AN/8	
285	Extra Product #12 Desc.	AN/16	
286	Extra Product #12 U/M	AN/4	
287	Extra Product #12 Price	Num/12	NNNNNNNNNN.NN
288	Extra Product #12 Amount	Num/12	NNNNNNNNNN.NN

**Note:** Sundry Charges will be sent as Extra Products in fields 034-069. All fields that are send with extra products will be accompanied by Sundry Charges. This includes delivery quantity, product name, product description, product unit of measure, product price, and extended price.

**Note:** Fields 253-288 are not available for the COMMANDeagle interface.



## Minimum Load Charge Fields

Field #	Field Description	Type/Max	Format
070	Minimum Load Charge Prt	AN/24	
071	Minimum Load Charge Amt	Num/12	NNNNNNNNN.NN

## Price Totals – COD Only

Field #	Field Description	Type/Max	Format
072	Ticket Subtotal	Num/12	NNNNNNNNN.NN
073	Ticket Tax Total	Num/12	NNNNNNNNN.NN
074	Order Cum Total	Num/12	NNNNNNNNN.NN

## General Purpose Fields

Field #	Field Description	Type/Max	Format	Req
075	Time Ticket Sent	AN/5		
076	Zone	AN/8		
077	Project Number	AN/16		
078	Ordered By	AN/24		
079	Special Instructions #1	AN/32		
080	Special Instructions #2	AN/32		
081	Operator Message	AN/32		
082	Local Ticket Change	Num/1 ***	N	X
	0 = change allowed			
083	Lot/Block #	AN/8		
084	Project Phone #	AN/12		
085	% Calcium	AN/4 @**		
086	% Air Entrainment	AN/4 @**		
087	% Super Plasticizer	AN/4 @**		
100	Print Weights (Y/N)	AN/1	'Y' or 'N'	
142	Metric (Y/N)	AN/1 *	'Y' or 'N'	

@: descriptive, but not affecting batch values

\* 'Y' = Metric Ticket - all values in ticket are based on Metric units.

\*\* The numbers displayed in 085 and 092, 086 and 093, and 087 and 094 are the same respectively. All trim percent.

\*\*\* Effects values of batched ingredients of these types.

## Admix/Water Trims

These values are sent if the actual batch values are to be entered or changed at the dispatch computer.

Field #	Field Description	Type/Max	Format
092	% Calcium	Num/4 **/**	N.NN
093	% Air Entrainment	Num/3 **/**	NNN
094	% Super Plasticizer	Num/3 **/**	NNN
095	% Hot Water	Num/3 ***	NNN
096	Pounds of Ice	Num/4 ***	NNNN
097	Loads Delivered	Num/4	NNNN
098	Admixture Code (If Batchbook ID)	AN/3	AAA
099	Customer Job Number	AN/16	

**Note:** For some of the above fields the assumed default value of zero is invalid. Below is a list of the exceptions and their defaults.

Field #	Field Description	Default
092	% Calcium	1.00
093	% Air Entrainment	100
094	% Super Plasticizer	100

## Adjusted Mix

Adjusted mixes are used when the mix design needed differs from the mix design held in the local file of the Batch Control Computer. If a variation of the mix design is desired, then the **entire** mix design is sent.

**Note:** The variation in admixes for batching purposes is sent in this format. The printout type data (price, UOM, etc.) is sent in the Extra Product fields.

Field #	Field Description	Type/Max	Format	Req
101	Mix Code (Batchbook ID)	AN/8		
102	Mix Description	AN/24		
103	Agg #1 Product Name	AN/8		
104	Agg #1 Weight	Num/4	NNNN	
105	Agg #2 Product Name	AN/8		
106	Agg #2 Weight	Num/4	NNNN	
107	Agg #3 Product Name	AN/8		
108	Agg #3 Weight	Num/4	NNNN	
109	Agg #4 Product Name	AN/8		
110	Agg #4 Weight	Num/4	NNNN	
111	Cement #1 Product Name	AN/8		
112	Cement #1 Weight	Num/4	NNNN	
113	Cement #2 Product Name	AN/8		
114	Cement #2 Weight	Num/4	NNNN	
115	Water #1 Product Name	AN/8		
116	Water #1 Weight/Gal	Num/6	NNNN.N	
117	Admix #1 Product Name	AN/8		
118	Admix #1 Ozs/%Cement	Num/6	NNNN.N	
119	Admix #2 Product Name	AN/8		
120	Admix #2 Ozs/%Cement	Num/6	NNNN.N	
121	Admix #3 Product Name	AN/8		
122	Admix #3 Ozs/%Cement	Num/6	NNNN.N	
123	Admix #4 Product Name	AN/8		
124	Admix #4 Ozs/%Cement	Num/6	NNNN.N	
125	Admix #5 Product Name	AN/8		
126	Admix #5 Ozs/%Cement	Num/6	NNNN.N	
127	Admix #6 Product Name	AN/8		
128	Admix #6 Ozs	Num/6	NNNN.N	
129	Cement #3 Product Name	AN/8		
130	Cement #3 Weight	Num/4	NNNN	
131	Water #2 Product Name	AN/8		
132	Water #2 Weight/Gal	Num/6	NNNN.N	
133	Mixer Time	Num/3	NNN	
134	Maximum Load Size	Num/5	NN.NN	X
135	Agg #5 Product Name	AN/8		
136	Agg #5 Weight	Num/4	NNNN	
137	Percent Air	Num/5	NN.NN	
138	Slump Target	Num/5	NN.NN	
139	Maximum Dry Load Size	Num/5	NN.NN	
140	Maximum Async Load Size	Num/5	NN.NN	
141	Agg Moisture Reference	Num/1*	N	
143-148	RESERVED			
143-149	Zone Travel Mileage	Num/3	NNN	

\* Agg Moisture Reference: 0 = SSD, 1 = Wet, 2 = Oven Dry

Additionally product quantities are limited to be within the following ranges:

Agg	0 - 9999
Cem	0 - 9999
Water	0 - 9999.9
Admix	0 - 9999.9

## **General Purpose Fields**

The following fields are included to allow a dispatch computer to send a number of printout-only items to the BCC. The information contained in these fields is merely re-printed on the BCC tickets as they are received.

## General Purpose Fields (cont'd)

Field #	Field Description	Type/Max	Format
150	Mix desc (chars 33-40)	AN/32	Left Jus.
151	Weighmaster name	AN/32	Left Jus.
152	Begin unload time	AN/32	Left Jus.
153	User Defined #4	AN/32	Left Jus.
154	User Defined #5	AN/32	Left Jus.
155	User Defined #6	AN/32	Left Jus.
156	User Defined #7	AN/32	Left Jus.
157	User Defined #8	AN/32	Left Jus.
158	Pocket number	AN/32	Left Jus.
159	Cement type	AN/32	Left Jus.
160	User Defined #11	AN/32	Left Jus.
161	Max water	AN/32	NNNNN.NN
162	Max water x delv qty	AN/32	NNNNN.NN
163	Agg water	AN/32	NNNNN.NN
164	Agg water x delv qty	AN/32	NNNNN.NN
165	Plt water - agg water	AN/32	NNNNN.NN
166	(Plt water - agg water) x qty	AN/32	NNNNN.NN
167	Max wtr - plt wtr - agg wtr	AN/32	NNNNN.NN
168	(Max - plt - agg wtrs) x qty	AN/32	NNNNN.NN
169	User Defined #20	AN/32	Left Jus.
170	User Defined #21	AN/32	Left Jus.
171	User Defined #22	AN/32	Left Jus.
172	User Defined #23	AN/32	Left Jus.
173	User Defined #24	AN/32	Left Jus.
174	User Defined #25	AN/32	Left Jus.
175	Slump	AN/32	Right Jus.
176	Weighmaster name	AN/32	Right Jus.
177	Customer name (chars 33-40)	AN/32	Right Jus.
178	Delivery addr (chars 65-96)	AN/32	Right Jus.
179	Delivery addr (chars 97-128)	AN/32	Right Jus.
180	User Defined #31	AN/32	Right Jus.
181	Ship to customer code	AN/32	Right Jus.
182	Ship to customer name	AN/32	Right Jus.
183	Ship to customer address 1	AN/32	Right Jus.
184	Ship to customer address 2	AN/32	Right Jus.
185	Ship to customer postal cd	AN/32	Right Jus.
186	Ship to customer city	AN/32	Right Jus.
187	Reference customer code	AN/32	Right Jus.
188	Project name (chars 1-32)	AN/32	Right Jus.
189	Project name (chars 33-40)	AN/32	Right Jus.
190	Reference customer name	AN/32	Right Jus.
191	Currency code	AN/32	Right Jus.
192	User Defined #43	AN/32	Right Jus.
193	User Defined #44	AN/32	Right Jus.
194	User Defined #45	AN/32	Right Jus.
195	User Defined #46	AN/32	Right Jus.
196	User Defined #47	AN/32	Right Jus.
197	User Defined #48	AN/32	Right Jus.
198	Map page	AN/32	Right Jus.
199	Previous truck code	AN/32	Right Jus.

## New Standard Fields

Field #	Field Description	Type/Max	Format
200	Extra Prod #1 Order Qty	Num/12	NNNNNNNNNN.NN
201	Extra Prod #1 Cum Qty	Num/12	NNNNNNNNNN.NN
202	Extra Prod #1 Long Desc.	AN/40	
203	Extra Prod #1 Price U/M	AN/12	
204	Extra Prod #2 Order Qty	Num/12	NNNNNNNNNN.NN
205	Extra Prod #2 Cum Qty	Num/12	NNNNNNNNNN.NN
206	Extra Prod #2 Long Desc.	AN/40	
207	Extra Prod #2 Price U/M	AN/12	
208	Extra Prod #3 Order Qty	Num/12	NNNNNNNNNN.NN
209	Extra Prod #3 Cum Qty	Num/12	NNNNNNNNNN.NN
210	Extra Prod #3 Long Desc.	AN/40	
211	Extra Prod #3 Price U/M	AN/12	
212	Extra Prod #4 Order Qty	Num/12	NNNNNNNNNN.NN
213	Extra Prod #4 Cum Qty	Num/12	NNNNNNNNNN.NN
214	Extra Prod #4 Long Desc.	AN/40	
215	Extra Prod #4 Price U/M	AN/12	
216	Extra Prod #5 Order Qty	Num/12	NNNNNNNNNN.NN
217	Extra Prod #5 Cum Qty	Num/12	NNNNNNNNNN.NN
218	Extra Prod #5 Long Desc.	AN/40	
219	Extra Prod #5 Price U/M	AN/12	
220	Extra Prod #6 Order Qty	Num/12	NNNNNNNNNN.NN
221	Extra Prod #6 Cum Qty	Num/12	NNNNNNNNNN.NN
222	Extra Prod #6 Long Desc.	AN/40	
223	Extra Prod #6 Price U/M	AN/12	
224	Delivery Instructions 4	AN/32	
225	Delivery Instructions 5	AN/32	
226	Delivery Instructions 6	AN/32	
227	Heat Charge Prt	AN/24	
228	Heat Charge Amt	Num/12	NNNNNNNNNN.NN
229	Ticket Grand Total	Num/12	NNNNNNNNNN.NN
230	Prev. Order Cum. Total	Num/12	NNNNNNNNNN.NN
231	Mix Max Water/Yard	Num/8	NNNNN.NN
232	Mix Max Water/Load	Num/8	NNNNN.NN
233	Sand Water/Yard	Num/8	NNNNN.NN
234	Sand Water/Load	Num/8	NNNNN.NN
235	Plant Water/Yard	Num/8	NNNNN.NN
236	Plant Water/Load	Num/8	NNNNN.NN
237	Allowed Water/Yard	Num/8	NNNNN.NN
238	Allowed Water/Load	Num/8	NNNNN.NN
239	Ticket Misc. 1	AN/8*	
240	Ticket Misc. 2	AN/8*	
241	Order Misc. 1	AN/8*	
242	Order Misc. 2	AN/8*	
243	Order Misc. 3	AN/8*	
244	Mix Class	AN/2	
245	Mix Strength	AN/7	
246	Mix Aggregate Size	AN/5	
247	Sales Type Code	AN/2	
248	Sales Type Short Desc.	AN/8	
249	Sales Type Long Desc.	AN/32	

\* Eleven characters are sent with these fields. These user-defined fields are transmitted with a three-character prefix, which defines the occurrence number. The interface parses out the prefix, leaving the eight-character field as entered.

## Mix Transmission

The purpose of Mix Transmission is to allow the central storage of mix designs in the dispatch computer database. This allows a centralized method of updating the current mix file in all BCC's. Each BCC should have a local storage file of current mix designs that can be updated either by the Batch Operator or through a Mix Transmission. In either case the products called for must be available at the plant site. This means that the product names sent for a given mix design by a dispatch computer must match those contained in the BCC files. If the desired product name does not match those held in the BCC files, that mix design will not successfully send to the BCC.

The field options allow for change/add of a mix design and a purge of the entire local mix design file. If the BCC is busy with batching, it should be attentive to a Mix Transmission when it has the time. Therefore, it is best to reserve Mix Transmissions to times when the BCC is less busy.

Storage of the mix design in the local mix design file will be inhibited if the mix components violate certain format rules. First, the product names of constituents in the mix design must not be used twice on a given mix design. Product names must match valid product names setup in the BCC using uppercase letters only with no spaces. Second, the mix design should not "skip" a mix component. For example, if you call for two aggregate components, they must occupy the Agg #1 and Agg #2 fields. No other combination will be accepted. Finally, the product quantities must be within the specified ranges.

Each Mix Transmission must have a "begin" and an "end" mix name that matches. If these names do not match, this is a non-recoverable error, and the entire Mix Transmission must be repeated. Other non-recoverable errors that demand correction and re-transmission are:

- Failure of data to meet type/format specifications
- Re-transmission of a single block more than 4 times
- Failure to write mix design to BCC storage file
- Mix design with invalid product names

If a non-recoverable error is detected, the lack of a Success Reply (M017) should prompt the CAI dispatching system to re-transmit the entire Mix Transmission. This cycle may repeat itself indefinitely; therefore, the CAI dispatching system enforces error recovery procedures.



## Mix Transmission Headers

The following fields represent the "header" type fields in a Mix Transmission. The "+" means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		M001ALL+	Purge Current Mix File	X
X		M002[Mix Name] +	Begin Mix	
X		M003[Mix Name] +	End Mix	X
X		M007[Field #] [Data] +	Mix Data Field	
	X	M017[Status Code] [Mix Name/ALL] +	Success Reply	
	X	M021[Status Code] [Mix Name/ALL] +	Failure Reply	

**Note:** When a M001 - Purge Current Mix File is issued, the BCC will respond with either a M017 or M021, with the word "ALL" substituted for the Mix Name.

## Mix Transmission Fields Layout

Below is a description of each of the allowed entries for data in the M007 type data fields. Individual product quantities are given for 1 yard load (1 cubic meter load if metric).

Field #	Field Description	Type/Max	Format	Req
001	Mix Code (Batchbook ID)	AN/8		X
002	Mix Description	AN/24		
003	Agg #1 Product Name	AN/8		
004	Agg #1 Weight (lb/kg)	Num/4	NNNN	
005	Agg #2 Product Name	AN/8		
006	Agg #2 Weight (lb/kg)	Num/4	NNNN	
007	Agg #3 Product Name	AN/8		
008	Agg #3 Weight (lb/kg)	Num/4	NNNN	
009	Agg #4 Product Name	AN/8		
010	Agg #4 Weight (lb/kg)	Num/4	NNNN	
011	Cement #1 Product Name	AN/8		
012	Cement #1 Weight (lb/kg)	Num/4	NNNN	
013	Cement #2 Product Name	AN/8		
014	Cement #2 Weight (lb/kg)	Num/4	NNNN	
015	Water #1 Product Name	AN/8		
016	Water #1	Num/6	NNNN.N	
	Weighed Units/Metered Units			
017	Admix #1 Product Name	AN/8		
018	Admix #1 Unit (oz/mL)	Num/6	NNNN.N	
019	Admix #2 Product Name	AN/8		
020	Admix #2 Unit (oz/mL)	Num/6	NNNN.N	
021	Admix #3 Product Name	AN/8		
022	Admix #3 Unit (oz/mL)	Num/6	NNNN.N	
023	Admix #4 Product Name	AN/8		
024	Admix #4 Unit (oz/mL)	Num/6	NNNN.N	
025	Admix #5 Product Name	AN/8		
026	Admix #5 Unit (oz/mL)	Num/6	NNNN.N	
027	Admix #6 Product Name	AN/8		
028	Admix #6 Unit (oz/mL)	Num/6	NNNN.N	
029	Cement #3 Product Name	AN/8		
030	Cement #3 Weight (lb/kg)	Num/4	NNNN	
031	Water #2 Product Name	AN/8		
032	Water #2	Num/6	NNNN.N	
	Weighed Units/Metered Units			
033	Mixer Time	Num/3	NNN	
034	Maximum Load Size	Num/5	NN.NN	X
035	Agg #5 Product Name	AN/8		
036	Agg #5 Weight (lb/kg)	Num/4		
037	Percent Air	Num/5	NN.NN	
038	Slump Target	Num/5	NN.NN	
039	Maximum Dry Load Size	Num/5	NN.NN	
040	Maximum Async Load Size	Num/5	NN.NN	
041	Agg Moisture Reference	Num/1*	N	
042	Metric (Y/N)	AN/1**	'Y' or 'N'	

\* Agg Moisture Reference: 0 = SSD, 1 = Wet, 2 = Oven Dry

\*\* Eagle 8.10 and later.

Additionally, product quantities are limited to be within the following ranges:

	Rev 5.3	Rev 6.0+	Rev 7.33 & Later
<b>Agg</b>	0 - 4095	0 - 9999	0 - 9999
<b>Cem</b>	0 - 999	0 - 999	0 - 9999
<b>Water</b>	0 - 1600.0	0 - 9999.9	0 - 9999.9
<b>Admix</b>	0 - 409.5	0 - 999.9	0 - 9999.9

## Return Batch Results

The purpose of the Return Batch Results is to allow the dispatch computer to receive the latest batch results from the BCC. For each ticket sent to the BCC there should eventually be a batch result or a cancellation. In any event, the BCC should store no more than ten batch results at one time. Therefore, the CAI dispatch computer periodically requests the latest batch results from the BCC. Please note that the status code byte being a lower case character indicates that batch results are available. The dispatch computer requests batch results before sending a ticket, ensuring that the build-up of batch results in the BCC is kept to a minimum.

When the dispatch computer requests batch results it should receive the oldest batch result from the BCC. It will accept only one ticket's batch results per request. If there are no batch results to report, the first "ACK" response from the BCC should contain an upper case Status Code character indicating no batch results pending. The dispatch computer will accept no other transmissions from the BCC at this time.

If there are batch results to report, two types of batch results are available. Both styles should embed results information according to the standard Message Block format, which begins with a "Batch Results" marker. A Brief Batch Results is sent as a fixed length block of 64 bytes. The Extended Batch Results are **variable in length** with a maximum length of 801 bytes. In both cases the order of appearance of each field is fixed and delimited by a carriage return. This means that the Extended Batch Results must send a carriage return even for a blank field.

If the dispatch computer receives the entire batch result block without an error, then it sends an ACK response. If the BCC receives the ACK response, it must send a Success Reply (T017). However, if it fails to receive the ACK response, the BCC should attempt to re-transmit the entire batch result. The BCC should try to re-transmit a single block up to 3 times before sending a Failure Reply (T021).

A typical batch results interaction is as follows:

1. The dispatch computer transmits a Send Batch Result (T009) to a BCC with pending results.
2. The BCC receives the Send Batch Result and responds with an ACK.
3. Then the BCC sends the oldest pending Batch Result block.
4. The dispatch computer responds to the Batch Result block with an ACK.
5. The BCC receives the ACK and sends a Success Reply (T017).

**Note:** The above sequence is for a Brief Batch Results request. To perform an Extended Batch Results request the (T009) is changed to a (T013).

## Return Batch Results

The following fields represent the "header" type fields in a return batch results transmission. The "+" means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		T009+	Send Brief Batch Result	X
	X	T010+	Brief Batch Result	
X		T013+	Send Extended Batch Result	X
	X	T014+	Extended Batch Result	
	X	T017[Status Code][Ticket #]+	Success Reply	
	X	T021[Status Code][Ticket #]+	Failure Reply	

## Typical Brief Batch Results Script

The following is a typical example of a "brief" batch results transfer. The modified BISYNC protocol has been omitted for clarity. The "+" means carriage return. This script should be imbedded in the message block format for transmission by the BCC.

Data	Comments
T010+	Brief Batch Result
12345678+	Ticket #12345678
0345+	Truck #345
08.00+	Load Size 8.00 yds
ABCD1234+	Mix Name ABCD1234
01.50+	Quantity On Board 1.50 yds
14:05:23+	Load Time 14:05:23
Charley B.+	Driver Name Charley B.

## Brief Batch Results Data Fields Layout

Below is a description of each of the entries for data in the Batch Result block. The field numbers are supplied to help reference to printed data. Each item should be stuffed with default data to achieve their maximum lengths. Numeric items should be right-justified with leading zeroes. Alphanumeric items should be left-justified with trailing spaces.

Field #	Field Description	Type/Max	Format	Req
002	Ticket #	AN/8		X
003	Truck #	Num/4		X
004	Load Size	Num/5	NN.NN	X
005	Mix Name	AN/8		X
020	Quantity On Board	Num/5	NN.NN	X
075	Load Time	AN/8	HH:MM:SS	X
015	Driver Name	AN/14		X

## Typical Extended Batch Results Script

The following is a typical example of an "extended" batch results transfer. The modified BISYNC protocol has been omitted for clarity. The "+" means carriage return. This script would be imbedded in the message block format for transmission by the BCC. Numbers in parentheses give index in bytes from start if maximum lengths of all fields are sent. In this case unused values have only carriage returns to mark their fields.

Data	Comments
T014+	Extended Batch Result
12345678+	Ticket #12345678
0345+	Truck #345
08.00+	Load Size 8.00 yds
ABCD1234+	Mix Name ABCD1234
01.50+	Quantity On Board 1.50 yds
14:05:23+	Load Time 14:05:23
Charley B. +	Driver Name Charley B.
1.23+	Specific Gravity of Slurry
21+	% Activity of Slurry
100+	% Substitution of Slurry
Sand +	Agg #1 Name (76)
123456+	Agg #1 Target wt
123456+	Agg #1 Actual wt
lbs +	Agg #1 Units
15.0+	Agg #1 % Moisture
3/4 Rock+	Agg #2 Name (109)
018456+	Agg #2 Target wt
018460+	Agg #2 Actual wt
lbs +	Agg #2 Units
3.0+	Agg #2 % Moisture
+	Agg #3 Name (142)
+	Agg #3 Target wt
+	Agg #3 Actual wt
+	Agg #3 Units
+	Agg #3 % Moisture
+	Agg #4 Name (175)
+	Agg #4 Target wt
+	Agg #4 Actual wt
+	Agg #4 Units
+	Agg #4 % Moisture
+	Agg #5 Name (208)
+	Agg #5 Target wt
+	Agg #5 Actual wt
+	Agg #5 Units
+	Agg #5 % MoistureType I
Type I +	Cem #1 Name (241)
004912+	Cem #1 Target wt
004910+	Cem #1 Actual wt
lbs +	Cem #1 Units
Flyash +	Cem #2 Name (269)
001104+	Cem #2 Target wt

Data	Comments
001120+	Cem #2 Actual wt
lbs +	Cem #2 Units
+	Cem #3 Name (297)
+	Cem #3 Target wt
+	Cem #3 Actual wt
+	Cem #3 Units
Calcium +	Amx #1 Name (325)
001234+	Amx #1 Target wt
001230+	Amx #1 Actual wt
ozs +	Amx #1 Units
SuprPlas+	Amx #2 Name (353)
000128+	Amx #2 Target wt
000128+	Amx #2 Actual wt
ozs +	Amx #2 Units
Air Ent +	Amx #3 Name (381)
000024+	Amx #3 Target wt
000024+	Amx #3 Actual wt
ozs +	Amx #3 Units
+	Amx #4 Name (409)
+	Amx #4 Target wt
+	Amx #4 Actual wt
+	Amx #4 Units
+	Amx #5 Name (437)
+	Amx #5 Target wt
+	Amx #5 Actual wt
+	Amx #5 Units
+	Amx #6 Name (465)
+	Amx #6 Target wt
+	Amx #6 Actual wt
+	Amx #6 Units
Water +	Wat #1 Name (493)
000270+	Wat #1 Target wt
000269+	Wat #1 Actual wt
gals+	Wat #1 Units
+	Wat #2 Name (521)
+	Wat #2 Target wt
+	Wat #2 Actual wt
+	Wat #2 Units
Charles B. Kleppenheimer III +	Long Driver Name
0014+	Temper Water
gals+	Temper Water Units
N+	Nonmetric Ticket; Terminate with CR (594)

## Material Receipts

The purpose of the Return Material Inventory is to allow the dispatch computer to receive the current inventory status of a selected material from the BCC. This process is essentially a query/response for each material, one at a time. Therefore to obtain the complete inventory status of a single BCC, the dispatch computer must initiate the query/response for every material at that BCC.

When the BCC receives a query for a material inventory its response will be a normal 'ACK' and status code. The BCC will then begin sending a fixed length material inventory block. This block will contain material name, material on hand, daily usage, long term usage, units, and daily received quantity. This information will begin with 'Material Inventory' marker and will be embedded in the standard Message Block format. If the dispatch computer requested a non-existent material name, then this block will contain '\*'s in the numeric fields and '?'s in the units field. Also the upcoming Success Reply will contain the status code 'C' or 'c' to reflect the product name mismatch.

If the dispatch computer receives the entire material inventory block without an error, then it sends an ACK response. If the BCC receives the ACK response, then it must send a Success Reply (T017). However, if it fails to receive the ACK response, then the BCC will attempt to retransmit the entire material inventory block. The BCC will try to retransmit a single block up to 4 times before sending a Failure Reply (T021).

A typical material inventory interaction is as follows:

1. The dispatch computer transmits a Return Material Inventory (T011) to a BCC.
2. The BCC receives the Return Material Inventory request and responds with an ACK.
3. Then the BCC sends the requested Material Inventory.
4. The dispatch computer responds to the Material Inventory with an ACK.
5. The BCC receives the ACK and replies a Success Reply (T017).



The following fields represent the 'header' type fields in a return material inventory transmission. The '+' means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		T011[material name]+	Return Material Inventory	X
	X	T012[material name]+	Material Inventory	
	X	T017[Status Code][material name]+	Success Reply	
	X	T021[Status Code][material name]+	Failure Reply	

## Typical Return Material Inventory Script

The following is a typical example of material inventory transfer. The modified BISYNC protocol has been omitted for clarity. The '+' means carriage return. This script would be imbedded in the message block format for transmission by the BCC.

Data	Comments
T012SAND+	Material Inventory - SAND
-0002345.6+	On Hand -2345.6
00000157.3+	Daily Usage 157.3
00007890.1+	Long Term Usage 7890.1
tons+	Units of Measure tons
00000123.4+	Daily Received Quantity
X+	1 spare byte

## Return Material Inventory Data Fields Layout

Below is a description of each of the entries for data in the Material Inventory block. Each item will be stuffed with default data to achieve their maximum lengths. Numeric items will be right-justified with leading zeroes. If the numeric item is negative, then the leading zero will be replaced by a minus sign. Alphanumeric items will be left-justified with trailing spaces. In the event of invalid data, the numeric quantities will become '\*'s and the units will become '?'s.

Field Description	Type/Max	Format	Req
Material Name	AN/8	XX...XX	X
On Hand	Num/10	NNNNNNNN.N	X
Daily Usage	Num/10	NNNNNNNN.N	X
Long Term Usage	Num/10	NNNNNNNN.N	X
Units of Measure*	AN/4	XXXX	X
Daily Received Qty	Num/10	NNNNNNNN.N	X
1 spare	AN/1	X	X

\* Allowed units of measure:      tons =      Short or metric tons  
    gals =      U.S. gallons  
    l     =      liters

Also, any user-defined 4-character units (ie: Mt3, Sacs)

## Purge Batch Results

The purpose of the Purge Batch Results is to allow the dispatch computer to purge or remove any pending batch results from the batch results queue. Since the batch results in many cases are not used by the dispatch computer, and a full queue (maximum of ten batch results) will cause the BCC to refuse any new tickets, this function was added for the dispatch computer to quickly clear out the queue.

When a BCC receives a Purge Batch Results, its response should be a normal ACK and status code. The BCC should remove *ALL* pending batch results from the batch results queue, and then respond with either a Success (T017) or Failure (T021) reply.

A typical Purge Batch Results interaction is as follows:

1. The dispatch computer transmits a Purge Batch Results (T015) to a BCC.
2. The BCC receives the request and responds with an ACK.
3. The BCC purges the batch results queue and transmits a Success reply (T017).

The following fields represent the "header" type fields in a Purge Batch Results transmission. The "+" means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		T015+	Purge Batch Results	X
	X	T017 [Status Code] [Reserved] +	Success Reply	
	X	T021 [Status Code] [Reserved] +	Failure Reply	

## Send Next Delivery

The purpose of the Send Next Delivery message is to allow the dispatch computer to receive delivery ticket data from the BCC. Each delivery ticket entered at the BCC will be made available to the DSP. Once the DSP has acknowledged receipt of a delivery ticket, the BCC will mark that record as sent and will not send it again.

When the dispatch computer requests delivery ticket information it will receive the oldest delivery ticket from the BCC. It will receive only one delivery ticket per request. If there are no delivery tickets to send, the Eagle will respond with a Next Delivery message (T020) with a value of "NONE" in the text block.

If the dispatch computer receives the entire delivery ticket block without an error, then it sends an ACK response. If the BCC receives the ACK response, then it must send a Success Reply (T017). However, if it fails to receive the ACK response, then the BCC will attempt to re-transmit the entire delivery ticket. The BCC will try to re-transmit a single block up to 3 times before sending a Failure Reply (T021).

A typical delivery ticket interaction is as follows:

1. The dispatch computer transmits a Send Next Delivery (T019) to a BCC.
2. The BCC receives the Send Next Delivery and responds with an ACK
3. Then the BCC sends the oldest pending delivery ticket block.
4. The dispatch computer responds to the delivery ticket block with an ACK.
5. The BCC receives the ACK and replies a Success Reply (T017).

## How Send Next Delivery Data is Packaged

The following fields represent the 'header' type fields in a send next delivery transmission. The '+' means carriage return.

From		Syntax	Explanation	WA
DSP	BCC			
X		T019+	Send Next Delivery	X
	X	T020+	Next Delivery	
	X	T017[Status Code][Ticket #]+	Success Reply	
	X	T021[Status Code][Ticket #]+	Failure Reply	

## Typical Send Next Delivery Script

The following is a typical example of a delivery ticket data transfer. The modified BISYNC protocol has been omitted for clarity. The '+' means carriage return. This script would be imbedded in the message block format for transmission by the BCC.

Data	Comments
T020+	Next Delivery
Sand #1 +	Material Name
00055000+	Qty Received
lbs +	Units
Acme Sand & Gravel +	Supplier
00001234+	Delivery Ticket Number
1:27PM 12Dec95 +	Date and Time Entered
ZQ4385 +	Delivery truck number
Wilson Delivery, Inc+	Hauler

If there are no deliveries to transmit, it will look like this:

Data	Comments
T020+	Next Delivery
NONE+	No Deliveries Available

**NOTE:** For Cement and Aggregate, the first character of the Unit of Measure will be examined. If it is a "T", the UOM is tons. Otherwise, the UOM is assumed to be pounds.

## Layout of Send Next Delivery Data Fields

Below is a description of each of the entries for data in the Delivery Data block. Each item will be stuffed with default data to achieve their maximum lengths. Numeric items will be right-justified with leading zeroes. Alphanumeric items will be left-justified with trailing spaces.

Field Description	Type/Max	Format
Material	AN/8	
Qty Received	Num/8	NNNNNNNN
Units	AN/4	
Supplier	AN/20	
Ticket	AN/8	
Date and Time	AN/16	HH:MMXM DDMmmYY
Truck	AN/8	
Hauler	AN/20	

# Appendix A. TCP/IP Interface

---

## Introduction – Source Notes

Visual C++ source code has been created to help test the TCP/IP interface. The Zip file (.zip) contains the following source code:

An example TCP/IP batch panel is server.cpp. When this executable runs, the correct responses will be given to the COMMANDseries software, resulting in a functioning interface. This is a starting template for a batch panel company interfacing to our system.

A client that can run is client.cpp. This client will connect to the server software, send one message to the server software, and shut down the interface.

This software is available for download for a batch panel company to obtain as a first step in interfacing to our TCP/IP system. Contact the Documentation Group for more information.

## Software Specification

### Transmission Protocol for TCP/IP (See Note 3)

#### Dispatch Computer To BCC

##### 1. WAKE-UP/POLL station

| SYN | SYN | ENQ | ID# | ID# | ID # | EOT |

##### 2. Return to IDLE (see Note 1)

| SYN | SYN | ESC | ID# | ID# | ID# | EOT |

##### 3. Message Block

| SYN | SYN | STX | Message Text | ETX | EOT |

Message text is "unlimited." A typical length is 800 to 900 bytes per ticket packet.

## BCC To Dispatch Computer

### 1. Response to WAKE-UP/POLL/Message Block

SYN	ACK	Status	EOT	CR
		Code		

### 2. Message Block

SYN	SYN	STX	Message Text	ETX	EOT	CR
-----	-----	-----	--------------	-----	-----	----

Message text is "unlimited." A typical length might be 800 to 900 bytes per batch weight packet.

**Note 1.** When the DSP sends a Return to IDLE a BCC that is "awake," the BCC simply returns to "sleep" with no transmitted reply.

**Note 2.** See the ASCII equivalents on the final page of this document.

**Note 3.** Note the differences between Modified Bisync Protocol (COMM port protocol) and TCP/IP protocol.

1. CRC error checking is not used/needed.
2. Ack and Nak packets are no longer needed for message packet blocks.
3. Message blocks are "unlimited" in length. (Typical lengths of big message packets are 800-1000 bytes.)
4. Because there is only one message block per transaction, the start of header code at the beginning of each packet is no longer used/necessary.
5. In wake up packet and Ack packet, the 3 byte ID# is the COMMANDconcrete plant code. (i.e. Plant 1 is <space><space>1.) Alpha and alpha-numeric plants are possible here as well.
6. When the time stamp packet is sent, the date is in dd-Mmm-yyyyy format. This is a change from the COMM bisync protocol to support 4 digit years. (Please refer to the TCP/IP log file example.)

## Log Examples

The following is an example log of batch panel transactions. The first column is time stamp. Within [ ] brackets is the packet length in bytes and the direction symbol,

r = COMMANDseries to batch panel

s = batch panel to COMMANDseries.

Within <> brackets is a non-printable ASCII character ( <sy> = ASCII 22 or Hex 0x16) Refer to table 1.

The following is an example of the receipts option turned on, no receipts, and no batch weights available.

```
11:53:44.850 [0007r] <sy><sy><eq> 1<et>
11:53:44.850 [0005s] <sy><ak>A<et><cr>
11:53:45.020 [0010r] <sy><sy><sx>T019<cr><ex><et>
11:53:45.020 [0016s] <sy><sy><sx>T020<cr>NONE<cr><ex><et><cr>
11:53:45.130 [0027r] <sy><sy><sx>W00101-Feb-1999 11:53<cr><ex><et>
11:53:45.130 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
11:53:45.180 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of the receipts option on, no receipts, and batch results returned.

```
11:54:45.930 [0007r] <sy><sy><eq> 1<et>
11:54:45.930 [0005s] <sy><ak>a<et><cr>
11:54:46.040 [0010r] <sy><sy><sx>T019<cr><ex><et>
11:54:46.040 [0016s] <sy><sy><sx>T020<cr>NONE<cr><ex><et><cr>
11:54:46.150 [0010r] <sy><sy><sx>T013<cr><ex><et>
11:54:46.150 [0324s] <sy><sy><sx>T014<cr> 11086<cr>0166<cr>04.00<cr>3000
<cr>00.00<cr>08:57:11<cr>John Birdsong <cr>0.00<cr>00<cr>000<cr>SAND
<cr>004848<cr>004840<cr>Lb <cr>01.0<cr> 12<cr>007347<cr>007360<cr>Lb
<cr>02.0<cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
><cr>TYPE10 <cr>002000<cr>002000<cr>Lb <cr><cr><cr><cr><cr><cr><cr><cr>ACCELER
<cr>000320<cr>000320<cr>Oz <cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
<cr><cr><cr><cr><cr>WATER <cr>000117<cr>000115<cr>Gal <cr><cr><cr><cr><cr>John Birdsong
<cr>0000<cr>Gal
<cr>N<cr><ex><et><cr>
11:54:46.260 [0027r] <sy><sy><sx>W00101-Feb-1999 11:54<cr><ex><et>
11:54:46.260 [0012s] <sy><sy><sx>W017a<cr><ex><et><cr>
11:54:46.310 [0007r] <sy><sy><ec> 1<et>
```



The following is an example of the receipts option on and receipts available.

```
11:56:47.810 [0007r] <sy><sy><eq> 1<et>
11:56:47.810 [0005s] <sy><ak>A<et><cr>
11:56:47.970 [0010r] <sy><sy><sx>T019<cr><ex><et>
11:56:47.970 [0111s] <sy><sy><sx>T020<cr>SAND <cr>00048000<cr>Lb <cr>nmmnm
<cr> 2010031<cr> 7:54AM 18Jan99 <cr> 1<cr>klkl <cr><ex><et><cr>
11:56:48.080 [0007r] <sy><sy><eq> 1<et>
11:56:48.080 [0005s] <sy><ak>A<et><cr>
11:56:48.140 [0010r] <sy><sy><sx>T019<cr><ex><et>
11:56:48.140 [0016s] <sy><sy><sx>T020<cr>NONE<cr><ex><et><cr>
11:56:48.190 [0027r] <sy><sy><sx>W00101-Feb-1999 11:56<cr><ex><et>
11:56:48.190 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
11:56:48.250 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of the receipts option off and no weights.

```
11:58:30.190 [0007r] <sy><sy><eq> 1<et>
11:58:30.190 [0005s] <sy><ak>A<et><cr>
11:58:30.250 [0027r] <sy><sy><sx>W00101-Feb-1999 11:58<cr><ex><et>
11:58:30.250 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
11:58:30.360 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of the receipts option off and weights available.

```
11:59:30.990 [0007r] <sy><sy><eq> 1<et>
11:59:30.990 [0005s] <sy><ak>a<et><cr>
11:59:31.100 [0010r] <sy><sy><sx>T013<cr><ex><et>
11:59:31.100 [0324s] <sy><sy><sx>T014<cr> 11086<cr>0166<cr>04.00<cr>3000
<cr>00.00<cr>08:57:11<cr>John Birdsong <cr>0.00<cr>00<cr>000<cr>SAND
<cr>004848<cr>004840<cr>Lb <cr>01.0<cr> 12<cr>007347<cr>007360<cr>Lb
<cr>02.0<cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
><cr>TYPE10 <cr>002000<cr>002000<cr>Lb <cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>ACCELER
<cr>000320<cr>000320<cr>Oz <cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
<cr><cr><cr><cr><cr>WATER <cr>000117<cr>000115<cr>Gal <cr><cr><cr><cr><cr>John Birdsong
<cr>0000<cr>Gal
<cr>N<cr><ex><et><cr>
11:59:31.160 [0027r] <sy><sy><sx>W00101-Feb-1999 11:59<cr><ex><et>
11:59:31.160 [0012s] <sy><sy><sx>W017a<cr><ex><et><cr>
11:59:31.210 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of the receipts no weights and ticket transmission.

```
12:02:11.210 [0007r] <sy><sy><eq> 1<et>
12:02:11.210 [0005s] <sy><ak>a<et><cr>
12:02:11.270 [0027r] <sy><sy><sx>W00101-Feb-1999 12:02<cr><ex><et>
12:02:11.270 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
12:02:11.380 [0792r] <sy><sy><sx>T00211054<cr>T0070011<cr>00211054<cr>0030001<cr>0040
5.00<cr>00530001<cr>006mix<cr>0075<cr>00901-Feb-99<cr>01010000<cr>011ABC Construction<cr>0123838
North Lyndale<cr>0142<cr>015Ron
Tschannen<cr>016addr<cr>018po<cr>019100<cr>02000.00<cr>02512:15<cr>026
05.00<cr>02700005.00<cr>02800200.00<cr>02930001<cr>030mix<cr>031CYDS<cr>07512:01<cr>0761000<cr>0
78Jim Smith<cr>081Slump
5<cr>0820<cr>083<cr>0850.00<cr>0860.00<cr>0870.00<cr>0921.00<cr>093000<cr>094000<
cr>095100<cr>0970001<cr>099job<cr>100Y<cr>151Charlie
Closenuf<cr>15211:00<cr>16100000.00<cr>16200000.00<cr>16300000.00<cr>16400000.00<cr>16500175.00<
cr>1
6600875.00<cr>16700000.00<cr>16800000.00<cr>1755<cr>176Charlie Closenuf<cr>18110000<cr>182ABC
Construction<cr>1833838 North Lyndale<cr>18535202<cr>186Birmingham<cr>18710000<cr>190ABC
Construction<cr>191USA<cr>23100000.00<cr>23200000.00<cr>23300000.00<cr>23400000.00<cr>23500175.0
0<cr>23600875.00<cr>23700000.00<cr>23800000.00<cr>2453000<cr>2471<cr>248Comme
rc<cr>249Commercial<cr>25200000.00<cr>T00311054<cr><ex><et>
12:02:11.380 [0012s] <sy><sy><sx>T017A<cr><ex><et><cr>
12:02:11.490 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of weights available and ticket transmission.

```
12:03:42.940 [0007r] <sy><sy><eq> 1<et>
12:03:42.940 [0005s] <sy><ak>a<et><cr>
12:03:43.100 [0010r] <sy><sy><sx>T013<cr><ex><et>
12:03:43.100 [0324s] <sy><sy><sx>T014<cr> 11086<cr>0166<cr>04.00<cr>3000
<cr>00.00<cr>08:57:11<cr>John Birdsong <cr>0.00<cr>00<cr>000<cr>SAND
<cr>004848<cr>004840<cr>Lb <cr>01.0<cr> 12<cr>007347<cr>007360<cr>Lb
<cr>02.0<cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
><cr>TYPE10 <cr>002000<cr>002000<cr>Lb <cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>ACCELER
<cr>000320<cr>000320<cr>Oz <cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr><cr>
<cr><cr><cr><cr><cr>WATER <cr>000117<cr>000115<cr>Gal <cr><cr><cr><cr><cr>John Birdsong
<cr>0000<cr>Gal
<cr>N<cr><ex><et><cr>
12:03:43.210 [0027r] <sy><sy><sx>W00101-Feb-1999 12:03<cr><ex><et>
12:03:43.210 [0012s] <sy><sy><sx>W017a<cr><ex><et><cr>
12:03:43.270 [0813r]
<sy><sy><sx>T00211055<cr>T0070011<cr>00211055<cr>0030166<cr>00405.00<cr>00530001<cr>006mix<cr>00
75<cr>00901-Feb-99<cr>01010000<cr>011ABC Construction<cr>0123838 North
Lyndale<cr>01455289784<cr>015John
Birdsong<cr>016addr<cr>018po<cr>019100<cr>02000.00<cr>02512:16<cr>02605.00<cr>02700010.00<cr>028
00200.00<cr>02930001<cr>030mix<cr>031CYDS<cr>07512:02<cr>0761000<cr>078Jim Smith<cr>081Slump
5<cr>0820<cr>083<cr>0850.00<cr>0860.00<cr>0870.00<cr>0921.00<cr>093000<cr>094000<cr>095100<cr>09
70002<cr>099job<cr>100Y<cr>151Charlie
Closenuf<cr>15212:20<cr>16100000.00<cr>16200000.00<cr>16300000.00<cr>16400000.00<cr>16500175.00<
cr>16600875.00<cr>16700000.00<cr>16800000.00<cr>1755<cr>176Charlie
Closenuf<cr>18110000<cr>182ABC Construction<cr>1833838 North
Lyndale<cr>18535202<cr>186Birmingham<cr>18710000<cr>190ABC Construction<cr>191USA<cr>199
1<cr>23100000.00<cr>23200000.00<cr>23300000.00<cr>23400000.00<cr>23500175.00<cr>23600875.00<cr>2
3700000.00<cr>23800000.00<cr>2453000<cr>2471<cr>248Commerc<cr>249Commercial<cr>25200005.00<cr>T0
0311055<cr><ex><et>
12:03:43.320 [0012s] <sy><sy><sx>T017a<cr><ex><et><cr>
12:03:43.490 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of mix download to batch panel.

```
12:04:35.120 [0007r] <sy><sy><eq> 1<et>
12:04:35.120 [0005s] <sy><ak>A<et><cr>
12:04:35.170 [0027r] <sy><sy><sx>W00101-Feb-1999 12:04<cr><ex><et>
12:04:35.170 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
12:04:35.230 [0145r]
<sy><sy><sx>M00230001<cr>M00700130001<cr>002mix<cr>003750<cr>0041200<cr>005715<cr>0061800<cr>011
601<cr>0120500<cr>015850<cr>0160035.0<cr>017805<cr>0180000.5<cr>033030<cr>03410.00<cr>042N<cr>M0
0330001<cr><ex><et>
12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><cr>
12:04:35.280 [0007r] <sy><sy><ec> 1<et>
```

The following is an example of removing all mixes from the batch panel and sending mix 3000 to the batch panel.

```
12:06:04.810 [0007r] <sy><sy><eq> 1<et>
12:06:04.810 [0005s] <sy><ak>A<et><cr>
12:06:04.970 [0013r] <sy><sy><sx>M001ALL<cr><ex><et>
12:06:04.970 [0012s] <sy><sy><sx>M017A<cr><ex><et><cr>
12:06:05.080 [0007r] <sy><sy><ec> 1<et>
12:06:05.630 [0007r] <sy><sy><eq> 1<et>
12:06:05.630 [0005s] <sy><ak>A<et><cr>
12:06:05.800 [0027r] <sy><sy><sx>W00101-Feb-1999 12:06<cr><ex><et>
12:06:05.800 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
12:06:05.960 [0145r]
<sy><sy><sx>M00230001<cr>M00700130001<cr>002mix<cr>003750<cr>0041200<cr>005715<cr>0061800<cr>011
601<cr>0120500<cr>015850<cr>0160035.0<cr>017805<cr>0180000.5<cr>033030<cr>03410.00<cr>042N<cr>M0
0330001<cr><ex><et>
12:06:05.960 [0012s] <sy><sy><sx>M017A<cr><ex><et><cr>
12:06:06.070 [0007r] <sy><sy><ec> 1<et>
```

## Table of Nonprintable Characters

Text	ASCII	Hex	Character Name	Symbol
<nu>	0	0x00	null	■
<sh>	1	0x01	^A	␣
<sx>	2	0x02	^B	␤
<ex>	3	0x03	^C	␣
<et>	4	0x04	^D	␣
<eq>	5	0x05	^E	♣
<ak>	6	0x06	^F	♣
<bl>	7	0x07	^G	•
<bs>	8	0x08	^H	□
<ht>	9	0x09	^I	○
<lf>	10	0x0a	^J	␣
<vt>	11	0x0b	^K	♂
<ff>	12	0x0c	^L	♀
<cr>	13	0x0d	^M	␣
<so>	14	0x0e	^N	♂
<si>	15	0x0f	^O	✱
<dl>	16	0x10	^P	▶
<d1>	17	0x11	^Q	◀
<d2>	18	0x12	^R	↑
<d3>	19	0x13	^S	!!
<d4>	20	0x14	^T	¶
<nk>	21	0x15	^U	§
<sy>	22	0x16	^V	—
<eb>	23	0x17	^W	‡
<cn>	24	0x18	^X	↑
<em>	25	0x19	^Y	↓
<ef>	26	0x1a	^Z	→
<ec>	27	0x1b	ESCAPE	↵
<fs>	28	0x1c		␣
<gs>	29	0x1d		␣
<rs>	30	0x1e		▲
<us>	31	0x1f		▼

## Example C Code

The following C code applies only to serial port interfaces.

```
#define LSB(x)      ( x >> 8 )
#define MSB(x)      (( x << 8 ) >> 8)

void set_crc(char *msg,unsigned short crc,int ascii_crc_flag,int flip)
{
char crc1,crc2,*top;
int tmp;

    crc1=(char)LSB(crc);
    crc2=(char)MSB(crc);
    if( ascii_crc_flag==0 )
    {
        top=msg;
        if(flip==1)
        {
            *(msg++)=crc2;
            *(msg++)=crc1;
        }
        else
        {
            *(msg++)=crc1;
            *(msg++)=crc2;
        }
        *msg=0;
        BinToRead(top,2);
        strcpy(top,gd.txtbuf2);
        return;
    }
    else
    {
        tmp=(crc1>>4)&0xf;
        if( tmp>9 )
            *(msg++)=tmp+'7';
        else  *(msg++)=tmp+'0';
        tmp=crc1&0xf;
        if( tmp>9 )
            *(msg++)=tmp+'7';
        else  *(msg++)=tmp+'0';
        tmp=(crc2>>4)&0xf;
        if( tmp>9 )
            *(msg++)=tmp+'7';
        else  *(msg++)=tmp+'0';
        tmp=crc2&0xf;
        if( tmp>9 )
            *(msg++)=tmp+'7';
        else  *(msg++)=tmp+'0';
        *msg=0;
    }
}
```

```

unsigned short crc16(char *data_p,unsigned short start,unsigned short
end,unsigned int poly)
{
unsigned char i;
unsigned int data;
unsigned int crc;
unsigned int length;

    crc=0x0;
    if(start>end) return(crc);
    length=(end-start)+1;
    for(i=0;i<start;i++,*data_p++);

    do
    {
        for(i=0,data=(unsigned int)0xff & *data_p++;i<8;i++,data>>=1)
        {
            if((crc & 0x0001) ^ (data & 0x0001))
                crc = (crc >> 1) ^ poly;
            else
                crc >>= 1;
        }
    }
    while(--length);

    data = crc;
    crc=(crc<<8) | ((data >> 8) & 0xff);
    return(crc);
}

/*
 * OUT_PSICRC
 *
 * $1 = Text string to get checksum for
 * $2 = Ascii flag (0=binary 1=ascii)
 *
 * output:
 *
 * $1 = checksum
 */
#ifdef UNIFACE5
XLONG XEXPORT OUT_PSICRC()
#else
XEXPORT(long) OUT_PSICRC()
#endif
{
int ascii_crc_flag,len;
unsigned short crc;
char answer[20];

    UGETREGS(1,gd.txtbufc,1024);
    len=ReadToBin(gd.txtbufc);
    gd.txtbufc[len]=0;
    len--;
    ascii_crc_flag=(int) UGETREG(2);
    crc=crc16(gd.txtbufc,0,len,0x8408);
    set_crc(answer,crc,ascii_crc_flag,0);
    UPUTREGS(1,answer);
    return(SUCCESS);
}

```