

# UNIVERSAL BATCH – DISPATCH COMMUNICATIONS LINK

Revision C: May 4, 2000 ©1999-2000 ALKON Corporation

ALKON Corporation, 5168 Blazer Parkway, Dublin, Ohio, 43017 Service Department: (614) 792-0677 • FAX: (614) 793-0608 Copyright © 1999-2000 Alkon Corporation. All rights reserved.

ALKON believes the statements contained herein are accurate as of the date of publication of this document. HOWEVER, ALKON HEREBY DISCLAIMS ALL WARRANTIES EITHER EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event will ALKON be liable to you for any damages, including any lost profits, lost savings, or other incidental or consequential damage arising out of the use or inability to use any information provided through this publication even if ALKON 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 to you.

This publication may contain technical inaccuracies or typographical errors.

This information is not intended to be an assertion of future action. ALKON expressly reserves the right to change or withdraw current products that may or may not have the same characteristics listed in this publication. Should ALKON modify its products in a way that may affect the information contained in this publication, ALKON assumes no obligation whatever to inform any user of the modification.

This publication is intended only for the direct benefit of users of ALKON products, and no right is granted for any other use.

Alkon Corporation 5168 Blazer Parkway Dublin, Ohio 43017 (614) 799-6650

www.alkon.com

# **TABLE OF CONTENTS**

List of Revisions	4
General Considerations	5
Hardware Specifications	6
Physical	6
Electrical	7
Baud Rate	7
Timing	7
Character Format	7
Parity	7
Software Specification	8
Dispatch Computer to BCC	
BCC to Dispatch Computer	9
BCC Status Codes	
Solution Non-Fatal Status Codes	
Sector Fatal Status Codes	
Transmission Error Checking	
Conventions Used in Message Transmission	
Message Formats & Sequence	
BCC Wake Up	
Sync Transmissions	15
Ticket Transmissions	16
How Ticket Data is Packaged for Transmission	17
Solution Format Rules for Ticket Transmissions	17
Series Response to Non-Recoverable Errors	
Scanceling a Ticket	
Solution Street Script	
Layout of Ticket Data Fields	20
Mix Transmissions	
b How Mix Design Data is Packaged	27
Solution Format Rules for Mix Transmissions	
Series Response to Non-Recoverable Errors	
Layout of Mix Transmission Fields	
Return Batch Results Query	
How Batch Results Request Data is Packaged	
Script	
Layout of Brief Batch Results Data Fields	
Solution Strain	
Return Material Inventory	
b How Return Material Inventory Data is Packaged	
System Script Sc	
Solution Content State S	
Purge Batch Results	
b How Purge Batch Results Data is Packaged	

Send	Next Delivery	40
$\mathcal{C}$	How Send Next Delivery Data is Packaged	40
$\mathcal{C}$	Typical Send Next Delivery Script	41
¢	Layout of Return Material Inventory Data Fields	41

Appendix A: TCP/IP Third Party Interface Addendum

# LIST OF REVISIONS

Date	Revision	Change(s)
Nov. 24, 1999	Rev. B	<ul> <li>Reformatted the document.</li> </ul>
May 4, 2000	Rev. C	<ul> <li>Changed Admix dosage fields from Num/6 (NNNN.N) to Num/7 (NNNNN.N) per CDI spec Ver. 1.04 dated July 9, 1999.</li> </ul>
		<ul> <li>Added "Send Next Delivery" section from the Cspec3.doc Ver. 1.03 dated January 3, 1996.</li> </ul>

# **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 the dispatch computer and the batch computer as long as the "rules" for a null modem are used.
- The interface will support point-to-point or multi-drop connections. In either case, the dispatching computer is considered the "MASTER" and the batch computer 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 communications channel.
- In this document, the Batch Control Computer may be referred to as BCC. The Dispatch System Computer may be referred to as the DSP.

# HARDWARE SPECIFICATIONS

# <u>Physical</u>

- 1. The connector presented to the batch computer is a 25-pin miniature "D" (DB-25S) connector with female contacts. The batch computer provides a mating (DB-25P) connector.
- 2. Typical equipment and configuration:



3. The communications equipment connected to the batch computer is configured as "DCE" (Data Communications Equipment). This is typically the data connector on a modem, null modem, multiplexer, or code-activated switch.

# <u>Electrical</u>

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
Circuit			· · · · · · · · · · · · · · · · · · ·
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.

# <u>BAUD RATE</u>

300/1200/2400/4800/9600 baud.

# <u>TIMING</u>

Asynchronous.

# <u>Character Format</u>

8 data bits -- 1 start bit, 1 stop bit.

<u>PARITY</u>

None.

## SOFTWARE SPECIFICATION

#### All transmissions must be embedded in the BISYNC protocol.

# DISPATCH COMPUTER TO BCC



# BCC TO DISPATCH COMPUTER



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

# <u>BCC Status Codes</u>

The BCC status code represents a meaningful response to the latest interaction with the DSP. It allows the DSP to quickly determine if there was an error, and what action to take in response.

Status codes are represented by a single character, and are grouped into two broad categories: (1) non-fatal, and (2) fatal. Non-fatal status codes are informative in nature. Fatal status codes indicate a failure of the BCC to carry out the last interaction.

Note the following convention. Status codes are sent in lower-case letters if there are batch results pending. This informs the dispatch computer that it should poll for results. Otherwise, status codes are sent in upper-case letters.

The tables shown below and on the following pages lists each status code, its cause, and suggested error recovery procedure.

# ✤ Non-Fatal Status Codes

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

## STATAL STATUS CODES

STATUS CODE	CAUSE	Recovery
В	Format/syntax error	The DSP should retry the 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 the transmission. If it fails on the second attempt, the DSP should alert the operator. Then poll for batch results.

Status Code	CAUSE	RECOVERY
С	Product name mismatch	The DSP should retry the transmission. If it fails on the second attempt, the DSP should alert the operator.
С	Product name mismatch; batch results pending	The DSP should retry the 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 the 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 the transmission. If it fails on the second attempt, alert the operator. Then poll for batch results.
F	Message too long	The DSP should retry the 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 the transmission. If it fails on the second attempt, alert the operator. Then poll for batch results.
Н	Ticket # already in queue	The DSP should retry the transmission. If it fails on the second attempt, the DSP should alert the operator.
h	Ticket # already in queue; batch results pending	The DSP should retry the transmission. If it fails on the second attempt, the DSP should alert the operator. Then poll for batch results.
Ι	Batching in progress	The DSP should wait a few moments, then retry the transmission.
i	Batching in progress; batch results pending	The DSP should wait a few moments, then retry the transmission. Then poll for batch results.
J	Ticket not available	The DSP should retry the 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 the transmission. If it fails on the second attempt, the DSP should alert the operator. Then poll for batch results.

Status Code	CAUSE	RECOVERY	
Μ	Mix design error	The DSP should retry the 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 the transmission. If it fails on the second attempt, the DSP should alert the operator. Then poll for batch results.	
Ν	Plant not ready	The DSP should wait a few minutes, then retry the transmission, or signal the operator to contact the plant.	
n	Plant not ready; batch results pending	The DSP should wait a few moments, then retry the transmission, or signal the operator to contact the plant. Then poll for batch results.	
Р	Block mismatch	The DSP should abort the present sequence and begin over. If it fails on the second attempt, then the DSP should alert the operator.	
р	Block mismatch; batch results pending	The DSP should abort the present sequence and start 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 the transmission. If it fails on 4 consecutive retries, the DSP should alert the operator.	
r	Receive error; batch results pending	The DSP should retry the transmission. If it fails on 4 consecutive retries, the DSP should alert the operator then poll for batch results.	
Т	Timeout	The DSP should abort the present sequence and start over. If it fails on the second attempt, the DSP should alert the operator.	
t	Timeout; batch results pending	The DSP should abort the present sequence and start over. If it fails on the second attempt, the DSP should alert the operator then poll for batch results.	

# TRANSMISSION ERROR CHECKING

There is a single error check which occurs on each message transmission. This check occurs over the length of the message block, utilizing CRC (Cyclic Redundancy Checking). Checking is based on the CCITT-CRC polynomial,  $X^{16} + X^{12} + X^5 + 1$ . The 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.

Some BCC's support an ASCII instead of a binary CRC. When ASCII CRC's are selected, these two bytes are expanded to four ASCII bytes. The value and ordering remains the same.

# CONVENTIONS USED IN MESSAGE TRANSMISSION

Message formats are designed to meet industry standards for dispatch/batch control interface specifications. Message formats are imbedded in a modified BISYNC protocol for purposes of transmission error detection. Below are some conventions used in the transmission of messages. Details of message syntax are covered later in this document.

- The field header is transmitted only at the beginning of a field segment. It is not transmitted between each field. Once you transmit a T007 field header, you will not need to do it again until another type of field header is sent.
- At the block level, the DSP should receive the BCC ACK/NAK response within 5 seconds. If it does not, the DSP must begin an error recovery procedure with the assumption that a NAK was received.
- To handle ticket transmission from the DSP, the BCC must be in the "awake" mode. The DSP sends the WAKE-UP/POLL message to the BCC and waits for an ACK response from the BCC. The BCC is now "awake" and will stay this way as long as there is transmission activity. If the BCC fails to receive any transmissions from the DSP after a period of 5 minutes, the BCC will automatically go back to "sleep." At this point, the DSP must send another WAKE-UP/POLL message to the BCC.
- Only *active* fields are transmitted from the DSP. If a field is not transmitted, it is assumed that it has spaces (if text) or zeroes (if numeric), unless otherwise noted.
- Field lengths are variable. Decimal places are sent for numeric fields. All alpha fields are left-justified. All numeric fields are right-justified.
- Only a carriage return (CR) is used as a field delimiter.
- The DSP must generate "keep-alive" traffic with each BCC. As mentioned earlier, each BCC must receive a transmission from the DSP within 5 minutes, or it will go back to "sleep." It will also alert the batch plant operator that the line is down.
- A transmission sequence, once begun, must be completely and successfully transmitted before any other type of transmission can be started. When transmitting ticket data, the DSP assumes the ticket failed to be inserted into the BCC ticket queue if the DSP does not receive a "Success Reply" message from the BCC. The DSP will then re-transmit the entire ticket.
- In general, if the DSP does not receive an anticipated ACK message, it assumes that a NAK message with a CRC error was sent. Similarly, the lack of an anticipated "Success Reply" message is construed as a "Failure Reply."

# MESSAGE FORMATS & SEQUENCE

# BCC WAKE UP

It is recommended that each morning the dispatch computer send a WAKEUP message to each BCC site. This ensures that each system is active and on-line. Additionally, synchronization information should be 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.

The ID #'s transmitted with the WAKE-UP/POLL/IDLE messages from the DSP allow a unique address to be assigned to each BCC. This selective addressing allows for a multidrop system. These numbers are equivalent to the BCC ID#'s in the sync transmissions (discussed next).

## SYNC TRANSMISSIONS

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.

			WA = With Acknowledge	ement
Fr DSP	om BCC	Syntax	Explanation	WA
Х		W001[Date Time]+	DD-MMM-YY HH:MM Ex: 01-Jan-99 13:10	Х
	Х	W017[Status Code]000000[ID info]+	Success Reply	
	Х	W021[Status Code]000000[ID info]+	Failure Reply	

# TICKET TRANSMISSIONS

Throughout the day, the DSP transmits instructions to the BCC to batch loads and print tickets. The BCC queues this information until batch time. (On the Series 2000 BCC, this queue is limited to a maximum of ten tickets.)

Ticket number assignment is the responsibility of the dispatch system. If the dispatch system maintains a single sequential ticket number, and several BCC systems are interfaced, 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 sequence of a typical ticket transmission is as follows:

- 1. The DSP has a ticket ready to send. It sends a WAKE-UP to the BCC, then waits for an ACK/NAK.
- 2. The BCC receives the WAKE-UP and responds with an ACK.
- 3. The DSP sends the first block, then waits for an ACK/NAK.
- 4. The BCC successfully receives the first block, then responds with an ACK.
- 5. The DSP sends the second block, then waits for an ACK/NAK.
- 6. The BCC receives the second block with a CRC error and responds with a NAK.
- 7. The DSP re-sends the second block, then waits for an ACK/NAK.
- 8. The BCC successfully receives the second block, then responds with an ACK.
- 9. The DSP sends the final block, then waits for an ACK/NAK.
- 10. The BCC successfully receives the final block, then responds with an ACK.
- 11. The DSP awaits a Success Reply from the BCC.
- 12. The BCC checks the entire ticket. If the check is O.K., it sends a Success Reply. Ticket transmission is now complete.

#### How TICKET DATA IS PACKAGED FOR TRANSMISSION

Actual data is transmitted via T007 fields. The T007 field identifier is sent once. Then the data fields follow. Once the T007 field identifier is sent, it remains in force until a different one is sent. A description of each "T" field is given below.

+ = carriage return		+ = carriage return	WA = With Acknowledgeme	
Fr DSP	om BCC	Syntax	Explanation	WA
X		T002[Ticket #]+	Begin Ticket	
Х		T007[Field #][Data]+	Ticket Data Field	
Х		T003[Ticket #]+	End Ticket	Х
Х		T006[Ticket #]+	Cancel Ticket	Х
	Х	T017[Status Code][Ticket #]+	Success Reply	
	Х	T021[Status Code][Ticket #]+	Failure Reply	

#### Section 2018 FORMAT RULES FOR TICKET TRANSMISSIONS

- Only one ticket transmission can be active at one time. Transmission of the next ticket cannot begin until after the previous transmission has been received and fully acknowledged by the BCC. A *full acknowledgement* consists of an ACK for each block and a Success Reply (T017). If the ticket format violates the syntax rules, a Failure Reply (T021) is sent.
  - **NOTE!** If an ACK is not received for a single block, only that block is retransmitted. If a Success Reply (T017) is not received, then the entire ticket should be re-transmitted.
- Each ticket transmission must start with a Begin (T002) field and end with an End (T003) field. Each Begin and End marker must contain a ticket number. For a given ticket, the Begin ticket number must match the End ticket number. If they do not, then the BCC will request re-transmission of the entire ticket.

#### Service Servic

If the entire ticket is sent without a non-recoverable error, the BCC sends a Success Reply (T017). However, if the ticket is sent with a non-recoverable error, the BCC sends a Failure Reply (T021). The lack of a Success Reply causes the DSP to re-transmit the entire ticket. This cycle can repeat itself over and over, so it is important that the DSP enforce error recovery.

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.

## ♦ CANCELING A TICKET

After a ticket has been completed and before it has been batched, the interface allows the dispatcher to cancel the ticket. This in the only change the dispatcher can make to a dispatched ticket. The Cancel message (T006) is sent to the BCC system for appropriate action.

# ✤ <u>Typical Ticket Script</u>

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

Header Field # Data	Comments
T00212345678+	Begin Ticket #12345678
T00700101+	Plant #01
00212345678+	Ticket #12345678
003345+	Truck #345
0048.00+	Load Size 8.00 vds
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
03/F1+	Extra Product Unit
0380.50+	Extra Product Price \$0.50
03925.00+	Extra Product Amt \$25.00
072457.00+	Subtotal \$457.00 $T_{1} = T_{1} + 1$ \$27.42
074484 42	$1 \text{ ax } 10 \text{ al } \frac{52}{.42}$
0/4484.42+	Uranu 10tal \$484.42 Legal Ticket Change allowed
0020+	Local Licket Unange allowed
100312343078+	End 11cket #12343678

# ✤ Layout of Ticket Data Fields

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 \* denotes values which directly affect a given batch.

Field #	Field Description	Type/Max	Format	Req
001	Plant ID#	Num/2		Х
002	Ticket #	Num/8		Х
003	Truck #	Num/4 *		Х
004	Load Size	Num/5 *	NN.NN	Х
005	Mix Code (Batchbook ID)	AN/8*		Х
006	Mix Description	AN/24		
007	Slump	Num/7	NNNN.NN	Х
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		Х
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	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/6	NNN.NN	
033	Mix Extended Amount	Num/8	NNNNN.NN	

# <u> Universal Batch – Dispatch Communications Link</u>

Field #	Field Description	Type/Max	Format	Rec
Extra Pro	ducts – Sent if on ticket			
034	Extra Product #1 Load Qty	Num/7	NNNN.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/6	NNN.NN	
039	Extra Product #1 Amount	Num/8	NNNNN.NN	
040	Extra Product #2 Load Qty	Num/7	NNNN.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/6	NNN.NN	
045	Extra Product #2 Amount	Num/8	NNNNN.NN	
046	Extra Product #3 Load Qty	Num/7	NNNN.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/6	NNN.NN	
051	Extra Product #3 Amount	Num/8	NNNNN.NN	
052	Extra Product #4 Load Qty	Num/7	NNNN.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/6	NNN.NN	
057	Extra Product #4 Amount	Num/8	NNNNN.NN	
058	Extra Product #5 Load Qty	Num/7	NNNN.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/6	NNN.NN	
063	Extra Product #5 Amount	Num/8	NNNNN.NN	
064	Extra Product #6 Load Qty	Num/7	NNNN.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/6	NNN.NN	
069	Extra Product #6 Amount	Num/8	NNNNN.NN	
Minimum	Load Charge Fields			
070	Minimum Load Charge Prt	AN/24		
071	Minimum Load Charge Amt	Num/8	NNNNN NN	

#### <u>Universal Batch – Dispatch Communications Link</u>

Field #	Field Description	Type/Max	Format	Req
Price Total	ls - COD only			
072	Ticket Subtotal	Num/8	NNNNN.NN	
073	Ticket Tax Total	Num/8	NNNNN.NN	
074	Order Cum Total	Num/8	NNNNN.NN	
General Pu	ırpose Fields			
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 *	Ν	Х
	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 @		

@ descriptive, but not affecting batch values

#### Admix/Water Trims

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

% Calcium	Num/4 *	N.NN
% Air Entrainment	Num/3 *	NNN
% Super Plasticizer	Num/3 *	NNN
% Hot Water	Num/3 *	NNN
Pounds of Ice	Num/4 *	NNNN
Loads Delivered	Num/4 *	NNNN
Admixture Code (if Batchbook ID)	AN/3	AAA
Customer Job Number	AN/16	
Print Weights (Y/N)	AN/1	'Y' or 'N'
	<ul> <li>% Calcium</li> <li>% Air Entrainment</li> <li>% Super Plasticizer</li> <li>% Hot Water</li> <li>Pounds of Ice</li> <li>Loads Delivered</li> <li>Admixture Code</li> <li>(if Batchbook ID)</li> <li>Customer Job Number</li> <li>Print Weights (Y/N)</li> </ul>	% CalciumNum/4 *% Air EntrainmentNum/3 *% Super PlasticizerNum/3 *% Hot WaterNum/3 *Pounds of IceNum/4 *Loads DeliveredNum/4 *Admixture CodeAN/3(if Batchbook ID)AN/16Print Weights (Y/N)AN/1

**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.

092	% Calcium	1.00
093	% Air Entrainment	100
094	% Super Plasticizer	100

Field #	Field Description	Type/Max	Format	Rea

#### **Adjusted Mix**

These are used when the mix design needed differs from the mix design held in the local file of the BCC. If a variation of the mix design is desired, then the entire mix design is sent. Note that the variation in admixes for batching purposes is sent in this format. The printout type data (price, um, etc.) is sent in the extra product fields.

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/L	Num/6	NNNN.N
117	Admix #1 Product Name	AN/8	
118	Admix #1 Ozs/ml/% Cement	Num/7	NNNNN.N
119	Admix #2 Product Name	AN/8	
120	Admix #2 Ozs/ml/% Cement	Num/7	NNNNN.N
121	Admix #3 Product Name	AN/8	
122	Admix #3 Ozs/ml/% Cement	Num/7	NNNNN.N
123	Admix #4 Product Name	AN/8	
124	Admix #4 Ozs/ml/% Cement	Num/7	NNNNN.N
125	Admix #5 Product Name	AN/8	
126	Admix #5 Ozs/ml/% Cement	Num/7	NNNNN.N
127	Admix #6 Product Name	AN/8	
128	Admix #6 Ozs	Num/7	NNNNN.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
135	Agg #5 Product Name	AN/8	
136	Agg #5 Weight	Num/4	NNNN
137	Percent Air	Num/5	NN.NN

Field #	Field Description	Type/Max	Format	Req
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*	Ν	
142	Metric (Y/N)	AN/1**	'Y' or 'N'	
143-148	RESERVED			
149	Zone Travel Mileage	Num/3	NNN	

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

\*\* 'Y' = Metric Ticket – all values in ticket are based on Metric units. This does NOT necessarily indicate an adjusted-mix ticket.

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

Agg	0 – 9999
Cem	0 – 9999
Water	0 – 9999.9
Admix	0 – 9999.9

**NOTE**: Earlier versions of some manufacturer's systems may have other limitations on product quantities.

#### **User Defined Fields**

These 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.

150	User Defined #1	AN/32	Left Jus.
151	User Defined #2	AN/32	Left Jus.
152	User Defined #3	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	User Defined #9	AN/32	Left Jus.
159	User Defined #10	AN/32	Left Jus.
160	User Defined #11	AN/32	Left Jus.
161	User Defined #12	AN/32	Left Jus.
162	User Defined #13	AN/32	Left Jus.
163	User Defined #14	AN/32	Left Jus.
164	User Defined #15	AN/32	Left Jus.
165	User Defined #16	AN/32	Left Jus.
166	User Defined #17	AN/32	Left Jus.
167	User Defined #18	AN/32	Left Jus.
168	User Defined #19	AN/32	Left Jus.

Field #	Field Description	Type/Max	Format	Req
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	User Defined #26	AN/32	Right Jus.	
176	User Defined #27	AN/32	Right Jus.	
177	User Defined #28	AN/32	Right Jus.	
178	User Defined #29	AN/32	Right Jus.	
179	User Defined #30	AN/32	Right Jus.	
180	User Defined #31	AN/32	Right Jus.	
181	User Defined #32	AN/32	Right Jus.	
182	User Defined #33	AN/32	Right Jus.	
183	User Defined #34	AN/32	Right Jus.	
184	User Defined #35	AN/32	Right Jus.	
185	User Defined #36	AN/32	Right Jus.	
186	User Defined #37	AN/32	Right Jus.	
187	User Defined #38	AN/32	Right Jus.	
188	User Defined #39	AN/32	Right Jus.	
189	User Defined #40	AN/32	Right Jus.	
190	User Defined #41	AN/32	Right Jus.	
191	User Defined #42	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	User Defined #49	AN/32	Right Jus.	
199	User Defined #50	AN/32	Right Jus.	
200	Extra Prod #1 Order Qty	Num/7	NNNN.NN	
201	Extra Prod #1 Cum Qty	Num/7	NNNN.NN	
202	Extra Prod #1 Long Desc.	AN/40		
203	Extra Prod #1 Price U/M	AN/8		
204	Extra Prod #2 Order Qty	Num/7	NNNN.NN	
205	Extra Prod #2 Cum Qty	Num/7	NNNN.NN	
206	Extra Prod #2 Long Desc.	AN/40		
207	Extra Prod #2 Price U/M	AN/8		
208	Extra Prod #3 Order Qty	Num/7	NNNN.NN	
209	Extra Prod #3 Cum Qty	Num/7	NNNN.NN	
210	Extra Prod #3 Long Desc.	AN/40		
211	Extra Prod #3 Price U/M	AN/8		
212	Extra Prod #4 Order Qty	Num/7	NNNN.NN	
213	Extra Prod #4 Cum Qty	Num/7	NNNN.NN	
214	Extra Prod #4 Long Desc.	AN/40		
215	Extra Prod #4 Price U/M	AN/8		

Field #	Field Description	Type/Max	Format	Req
216	Extra Prod #5 Order Qty	Num/7	NNNN.NN	
217	Extra Prod #5 Cum Qty	Num/7	NNNN.NN	
218	Extra Prod #5 Long Desc.	AN/40		
219	Extra Prod #5 Price U/M	AN/8		
220	Extra Prod #6 Order Qty	Num/7	NNNN.NN	
221	Extra Prod #6 Cum Qty	Num/7	NNNN.NN	
222	Extra Prod #6 Long Desc.	AN/40		
223	Extra Prod #6 Price U/M	AN/8		
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/8	NNNN.NN	
229	Ticket Grand Total	Num/8	NNNN.NN	
230	Prev. Order Cum. Total	Num/8	NNNN.NN	
231	Mix Max Water/Yard	Num/8	NNNN.NN	
232	Mix Max Water/Load	Num/8	NNNN.NN	
233	Sand Water/Yard	Num/8	NNNN.NN	
234	Sand Water/Load	Num/8	NNNN.NN	
235	Plant Water/Yard	Num/8	NNNN.NN	
236	Plant Water/Load	Num/8	NNNN.NN	
237	Allowed Water/Yard	Num/8	NNNN.NN	
238	Allowed Water/Load	Num/8	NNNN.NN	
239	Ticket Misc. 1	Num/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	Sale Type Short Desc.	AN/8		
249	Sales Type Long Desc.	AN/32		
250	Skip Ticket (Y/N)	AN/1		
251	Wet/Dry (0/1)	Num/1	Ν	
252	Previous Cum. Quantity	Num/8	NNNNN.NN	
253	Start Batch Immediately upon receipt	AN/1		Note 1
254	Mix Sequence Code	<b>AN</b> /10		Note 2

**NOTE 1:** "Start Batch Immediately upon receipt" is true <u>only</u> if the character 'Y' is present. All other characters are treated as false. (Spectrum V6 only)

**NOTE 2:** "Mix Sequence Code," left blank, uses the default code in the mix design. (Spectrum V6 only) An invalid sequence code should return an error.

# MIX TRANSMISSIONS

The DSP serves as a central storage and maintenance point for all mix designs. Mix designs are created or updated at the DSP then transmitted to each BCC. Each BCC has its own local storage file of current mix designs. Local mix designs can be updated either by the Batch Operator or by the DSP via a mix transmission. The DSP can also purge the entire local mix design file at a BCC.

<u>For Series 2000 BCC's</u>, mix transmissions can only be stored in the local mix design file when the BCC is not busy with batching. Therefore it is best to reserve mix transmission to Series 2000 systems when it is less busy (say, after normal working hours).

#### Solution How MIX DESIGN DATA IS PACKAGED

Actual data is transmitted via M007 fields. The M007 field identifier is sent once. Then the data fields follow. Once the M007 field identifier is sent, it remains in force until a different one is sent. A description of each "M" field is given below.

		+ = carriage return	WA = With Acknowleds	gement
Fr DSP	om BCC	Syntax	Explanation	WA
X		M001ALL+	Purge Current Mix File	Х
Х		M002[Mix Name]+	Begin Mix	
Х		M007[Field #][Data]+	Mix Data Field	
Х		M003[Mix Name]+	End Mix	Х
	Х	M017[Status Code][Mix Name/ALL]+	- Success Reply	
	Х	M021[Status Code][Mix Name/ALL]+	- Failure Reply	

#### Server State For Mix Transmissions

Storage of the mix design in the local mix design file will be inhibited if the mix components violate certain format rules as listed next:

- Only one mix transmission can be active at one time. Transmission of the next mix design cannot begin until after the previous transmission has been received and fully acknowledged by the BCC. A *full acknowledgement* consists of an ACK for each block and a Success Reply (M017). If the mix design format violates the syntax rules, a Failure Reply (M021) is sent.
  - **NOTE!** If an ACK is not received for a single block, only that block is retransmitted. If a Success Reply (T017) is not received, then the entire mix design should be re-transmitted.
- Each mix transmission must start with a Begin (M002) field and end with an End (M003) field. Each Begin and End marker must contain a mix name. The Begin mix name must match the End mix name. If they do not, then the BCC will request retransmission of the entire mix design.
- Material names transmitted for a given mix design must match those contained in the BCC files. If not, the mix design will not be accepted by the BCC.
- The material name must not be used twice on a given mix design.
- 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.
- The material quantities must be within the specified ranges.

#### ✤ Response to Non-Recoverable Errors

If the mix design is sent with a non-recoverable error, the BCC sends a Failure Reply (M021). The lack of a Success Reply (M017) causes the DSP to re-transmit the entire mix design. This cycle can repeat itself over and over, so it is important that the DSP enforce format rules.

Non-recoverable errors include:

- Failure to match Begin and End mix names.
- 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.

## ✤ LAYOUT OF MIX TRANSMISSION FIELDS

Below is a description of each of the allowed entries for data in the M007 type data fields. Individual product quantities are given for a 1-yard load.

Field #	Field Description	Type/Max	Format	Req
001	Mix Code (Batchbook ID)	AN/8		Х
002	Mix Description	AN/24		
003	Agg #1 Product Name	AN/8		
004	Agg #1 Weight	Num/4	NNNN	
005	Agg #2 Product Name	AN/8		
006	Agg #2 Weight	Num/4	NNNN	
007	Agg #3 Product Name	AN/8		
008	Agg #3 Weight	Num/4	NNNN	
009	Agg #4 Product Name	AN/8		
010	Agg #4 Weight	Num/4	NNNN	
011	Cement #1 Product Name	AN/8		
012	Cement #1 Weight	Num/4	NNNN	
013	Cement #2 Product Name	AN/8		
014	Cement #2 Weight	Num/4	NNNN	
015	Water #1 Product Name	AN/8		
016	Water #1 Weight/Gal/L	Num/6	NNNN.N	
017	Admix #1 Product Name	AN/8		

#### <u>Universal Batch – Dispatch Communications Link</u>

Field #	Field Description	Type/Max	Format	Req
018	Admix #1 Ozs/ml/% Cement	Num/7	NNNNN.N	
019	Admix #2 Product Name	AN/8		
020	Admix #2 Ozs/ml/% Cement	Num/7	NNNNN.N	
021	Admix #3 Product Name	AN/8		
022	Admix #3 Ozs/ml/% Cement	Num/7	NNNNN.N	
023	Admix #4 Product Name	AN/8		
024	Admix #4 Ozs/ml/% Cement	Num/7	NNNNN.N	
025	Admix #5 Product Name	AN/8		
026	Admix #5 Ozs/ml/% Cement	Num/7	NNNNN.N	
027	Admix #6 Product Name	AN/8		
028	Admix #6 Ozs	Num/7	NNNNN.N	
029	Cement #3 Product Name	AN/8		
030	Cement #3 Weight	Num/4	NNNN	
031	Water #2 Product Name	AN/8		
032	Water #2 Weight/Gal	Num/6	NNNN.N	
033	Mixer Time	Num/3	NNN	
034	Maximum Load Size	Num/5	NN.NN	Х
035	Agg #5 Product Name	AN/8		
036	Agg #5 Weight	Num/4	NNNN	
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*	Ν	
042	Metric (Y/N)	AN/1	'Y' or 'N'	

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

\*\* 'Y' = Metric Ticket – all values in ticket are based on Metric units. This does NOT necessarily indicate an adjusted-mix ticket.

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

 Agg
 0 – 9999

 Cem
 0 – 9999

 Water
 0 – 9999.9

 Admix
 0 – 9999.9

**NOTE**: Earlier versions of some manufacturer's systems may have other limitations on product quantities.

# **RETURN BATCH RESULTS QUERY**

The *Return Batch Results* query allows the DSP to receive the latest batch results from the BCC. For each ticket sent to the BCC, there is eventually a batch result or a cancellation. Spectrum Systems have no practical limit to the number of stored batch results, but the Series 2000 stores no more than ten batch results at a time. The DSP should request batch results before sending a ticket. By doing this it ensures that the build-up of batch results in the BCC is kept to a minimum.

When the DSP requests batch results, it receives the oldest batch results for one ticket. If there are no batch results to report, the first 'ACK' response from the BCC contains an upper case Status Code character (indicating no batch results are pending). The DSP will receive no other transmissions from the BCC at this time.

There are two kinds of batch results: (1) **Brief**, and (2) **Extended**. Both kinds embed results according to the standard Message Block format. **Brief Batch Results** are sent as a fixed-length block of 64 bytes. **Extended Batch Results** are variable in length, with a maximum of 801 bytes. In both cases, fields are fixed in length and delimited by a carriage return. Even blank fields are delimited by carriage returns.

If the DSP receives the entire batch result block without an error, it sends an ACK response. If the BCC receives the ACK response, it must send a Success Reply (T017). If it does not receive the ACK response, the BCC attempts to re-transmit the entire batch result. The BCC tries 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 DSP 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. The BCC sends the oldest pending Batch Result block.
- 4. If the DSP receives the entire Batch Result block without an error, it sends an ACK.
- 5. The BCC receives the ACK and replies a Success Reply (T017). If the BCC does not receive the ACK response, it re-transmits the entire Batch Result block. The BCC tries up to 3 times to send the Batch Result block before sending a Failure Reply (T021).
- \*\* The above sequence is for a Brief Batch Results request. To perform an Extended Batch Results request, the (T009) is changed to a (T013).

# How BATCH RESULTS REQUEST DATA IS PACKAGED

The following fields represent the 'header' type fields in a Return Batch Results transmission.

		+ = carriage return	WA = With Acknowledg	ement
Fr DSP	om BCC	Syntax	Explanation	WA
Х		T009+	Send Brief Batch Result	Х
	Х	T010+	Brief Batch Result	
Х		T013+	Send Extended Batch Result	Х
	Х	T014+	Extended Batch Result	
	Х	T017[Status Code][Ticket #]+	Success Reply	
	Х	T021[Status Code][Ticket #]+	Failure Reply	

#### Stypical Brief Batch Results Script

The following is a typical example of a Brief Batch Results transmission. The modified BISYNC protocol has been omitted for clarity. The '+' means carriage return. This data would be embedded 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.

#### Schule Content of Brief Batch Results Data Fields

Below is a description of the data in each field in the Batch Result block. Field numbers are supplied to help reference printed data. Each item is filled with default data to achieve its maximum length. Numeric items are right-justified with leading zeroes. Alphanumeric items are left-justified with trailing spaces.

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

#### Stript

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 % Moisture
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

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+	Non-metric Ticket*; Terminate with CR (594)

\*Applies to CDI Eagle Revision 8.10 and later.

## **RETURN MATERIAL INVENTORY**

The *Return Material Inventory* query allows the DSP 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. To obtain the complete inventory status of a single BCC, the DSP must initiate the query/response for every material at the BCC.

In response to a query for material inventory, the BCC sends an ACK and status code. The BCC will then begins sending a fixed-length material inventory block. This block contains the material name, on hand amount, daily usage, long-term usage, units, and daily received quantity.

If the DSP requests a non-existent material name, the message block from the BCC contains \*'s in its numeric fields and ?'s in its units field. Also the upcoming Success Reply will contain the status code 'C' or 'c' to reflect the material name mismatch.

A typical batch results interaction is as follows:

- 1. The DSP transmits a Return Material Inventory query (T011) to a BCC.
- 2. The BCC receives the Return Material Inventory request and responds with an ACK.
- 3. The BCC sends the requested Material Inventory block (T012).
- 4. If the DSP receives the entire Material Inventory block without an error, it sends an ACK response.
- 5. When the BCC receives the ACK response, it sends a Success Reply (T017). If the BCC fails to receive the ACK response, it attempts to re-transmit the entire Material Inventory block. The BCC tries up to 3 times to send the material inventory block before sending a Failure Reply (T021).

# How Return Material Inventory Data is Packaged

The following fields represent the 'header' type fields in a Return Material Inventory transmission.

r		+ = carriage return	WA = With Acknowledge	ement
Fr DSP	om BCC	Syntax	Explanation	WA
Х		T011[Material Name]+	Return Material Inventory	Х
	Х	T012[Material Name]+	Material Inventory	
	X X	T017[Status Code][Material Name]+ T021[Status Code][Material Name]+	Success Reply Failure Reply	

#### SCRIPT

An example of data transmitted for a Return Material Inventory query is shown next. The modified BISYNC protocol has been omitted for clarity. The '+' represents a carriage return. This data would be embedded 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

## ✤ LAYOUT OF RETURN MATERIAL INVENTORY DATA FIELDS

A description of each of the allowed entries for data in the Material Inventory block is given next. Each field is filled with default data to achieve its maximum length. Numeric items are right-justified with leading zeroes. Negative numeric values have the leading zero replaced by a minus sign (-). Alphanumeric fields are left-justified with trailing spaces. For invalid data, the numeric quantities become \*'s and units become ?'s.

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

\* Allowed units of measure:

tons = Short or metric tons G = U.S. gallons

G = 0.5. gam1 = liters

Also, any user-defined 4-character units (i.e., Mt3, Sacs)

# PURGE BATCH RESULTS

In some circumstances the DSP does not use the batch results. This can cause the batch results in the BCC to queue up. Spectrum Systems have no practical limit to the number of stored batch results, but the Series 2000 stores no more than 10 batch results at a time. The *Purge Batch Results* query allows the DSP to quickly clear out a BCC's batch results queue.

When the BCC receives a Purge Batch Results request, it responds with a normal ACK and status code. The BCC removes ALL pending batch results from the batch results queue, and then responds with either a Success Reply (T017), or Failure Reply (T021).

A typical Purge Batch Results interaction is as follows:

- 1. The DSP transmits a Purge Batch Results query (T015) to a BCC.
- 2. The BCC receives the query and responds with an ACK.
- 3. The BCC purges its batch results queue and transmits a Success Reply (T017).

#### Scheme Batch Results Data is Packaged

The following fields represent the 'header' type fields in a Purge Batch Results transmission.

		+ = carriage return	WA = With Acknowl	edgement
Fr DSP	om BCC	Syntax	Explanation	WA
Χ		T015+	Purge Batch Results	Х
	X X	T017[Status Code][Reserved]+ T021[Status Code][Reserved]+	Success Reply Failure Reply	

# SEND NEXT DELIVERY

Each delivery ticket entered at the BCC can be sent to the DSP. Only the oldest delivery ticket is sent, and only one delivery ticket is sent. If there are no delivery tickets to send, the BCC responds with a Next Delivery message (T020) with a value of "NONE" in the text block.

The sequence of a typical delivery ticket interaction is as follows:

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

#### Send Next Delivery Data is Packaged

The following fields represent the 'header' type fields in a Send Next Delivery transmission.

II <u>A</u> cknowledgement
n WA
elivery X
/
s Reply
Reply

#### ✤ TYPICAL SEND NEXT DELIVERY SCRIPT

An example of data transmitted for a Send Next Delivery query is shown next. The modified BISYNC protocol has been omitted for clarity. The '+' represents a carriage return. This data would be embedded 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
1:27PM 12Dec99 +	Date and Time Entered
ZQ4385 +	Delivery Truck Number
Wilson Delivery, Inc+	Hauler

If there are no deliveries to transmit, the script looks like this:

Data	Comments
T020 +	Next Delivery
NONE+	No deliveries available

## Schule Content of Send Next Delivery Data Fields

A description of each of the allowed entries for data in the Send Next Delivery block is given next. Each field is filled with default data to achieve its maximum length. Numeric items are right-justified with leading zeroes.

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

# TCP/IP Generic Third Party Interface Addendum

11/16/99 gtpa-100-1199



The contents of this manual are for informational purposes only and are subject to change without notice. Command Data Inc. assumes no responsibility for any errors that may appear in this manual. The contents of this manual may not be used for any purposes other than which provided.

© 1997, 1998, 1999 Command Data Inc.

Information disclosed herein was originated and is the property of Command Data Inc., and except for rights granted by written consent, such information shall not be disclosed, disseminated, or duplicated in whole or in part. Command Data Inc. reserves all patent, proprietary, design, manufacturing, use, and reproduction rights.

Command Data Incorporated

2204 Lakeshore Drive Suite 206 Birmingham, AL 35209-6719 (205) 879-3282 605 East Safari Pkwy., Suite C-4 Grand Prairie, TX 75050-2325 (972) 262-2692

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

DC, ARD, RKH, MJL, DEM, JPT

# **Table of Contents**

TABLE OF CONTENTS	I
INTRODUCTION	1
SOURCE NOTES	1
SOFTWARE SPECIFICATION	2
TRANSMISSION PROTOCOL FOR TCP/IP (SEE NOTE 3) Dispatch Computer To BCC BCC To Dispatch Computer	2 2 2
PLANT EDITOR	4
LOG EXAMPLES	5
TABLE OF NONPRINTABLE CHARACTERS	8
EXAMPLE C CODE	9

## **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**

```
    WAKE-UP/POLL station

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

    Return to IDLE (see Note 1)

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

    Message Block

            SYN | 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 Code	EOT	CR			
2.	Message I	Block						
	SYN	SYN	STX	Mess	sage Text	ETX	EOT	CR

Message text is "unlimited." A typical length might is 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.)

If the port is blank then the field IP address is made available. An IP address or a non-blank port is required on this screen. If a port is input the interface will use the COMM port routines (bisync protocol). If the IP address is set the interface will use the TCP/IP routines. If the IP address is set then the listen port must be a non-zero value. This value must not conflict with any TCP/IP tools (i.e. telnet is port 23) or any listen ports set up in COMMAND network.

📱 Plant Communicatio	on Setup (1 Plant 1)		×
Plant Copy Manager Number Port Name TCP/IP Address Selective Address Document Format	1	Listen Port	5001
Auto Dial Flag Modem Phone Number Modem Code			

# 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>
11:53:44.850 [0010r] <sy><sy><sx>T019<cr><ex><et>
11:53:45.020 [0010r] <sy><sy><sx>T020<cr>NONE<cr><ex><et><11:53:45.130 [0027r] <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><e> 1<e>
```

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>T020<cr>NONE<cr><ex><et><cr>
11:54:46.150 [0010r] <sy><sy><sx>T013<cr><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>SAND <cr>004848<cr>004840<cr>Lb
<cr>01.0<cr>
            12<cr>007347<cr>007360<cr>Lb
<pr><cr><cr><cr><cr><cr><cr>WATER <cr>000117<cr>000115<cr>Gal <cr><cr><cr><cr>>cr><cr>>dr>John Birdsong
<cr>0000<cr>Gal
<cr>N<cr><ex><et><cr>
11:54:46.260 [0027r] <sy><sx>W00101-Feb-1999 11:54<cr><et>
11:54:46.260 [0012s] <sy><sy>(sy>(sy)(17a)(cr)(eta)(cr))
11:54:46.310 [0007r] <sy><ec> 1<et>
```

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

11:56:48.250 [0007r] <sy><sc> 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><et><11:58:30.250 [0012s] <sy><sy><sx>W017A<cr><ex><et><cr>
11:58:30.360 [0007r] <sy><sy><e> 1<et>
```

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

```
11:59:30.990 [0007r] <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
            12<cr>007347<cr>007360<cr>Lb
<cr>01.0<cr>
<pr><cr><cr><cr><cr><cr><cr><cr><cr>WATER
<cr>000117<</p>
<cr>>Gal
<cr><cr><cr><cr>>dr><cr>>dr
<cr>0000<cr>Gal
<cr>N<cr><ex><et><cr>
11:59:31.160 [0027r] <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>csy>Cor>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>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>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>23500175.00<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.

TCPIP Generic Third Party Interface Addendum.doc gtpa-100-0299

```
<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>0075<cr</p>
>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>02800200
.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>0970002
<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>16
600875.00<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>237000
00.00<cr>23800000.00<cr>2453000<cr>2471<cr>248Commerc<cr>249Commercial<cr>25200005.00<cr>T00311055<cr
><ex><et>
12:03:43.320 [0012s] <sy><sy><sx>T017a<cr><ex><et><cr>
12:03:43.490 [0007r] <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>
12:04:35.120 [0005s] <sy><ak>A<et><cr>
12:04:35.170 [0027r] <sy><sy><sy>W00101-Feb-1999 12:04<cr><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>>015850<cr>>015850<cr>>0160035.0<cr>>017805<cr>>0180000.5<cr>>033030<cr>>042N<cr><et><12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><12:04:35.230 [0012s] <sy><sy><sx>M017A<cr><ex><et><12:04:35.280 [0007r] <sy><sy><ec> 1<et><12:04:35.280 [0007r] <sy><sy><ec> 1<et><12:04:35.280 [0007r] <sy><ex><et><1<</p>
```

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>
12:06:04.810 [0013r] <sy><sy><sy><sx>M001ALL<cr><ex><et>
12:06:04.970 [0013r] <sy><sy><sy><sx>M017A<cr><ex><et>
12:06:05.080 [0007r] <sy><sy><sy><ex> 1<et>
12:06:05.630 [0007r] <sy><sy><ex> 1<et>
12:06<05.630 [0007r] <sy><sy><ex> 1<et>
12:06<05.630 [0007r] <sy><sy><ex> 1<et>
12:06<05.630 [0012s] <sy><sx>W00101-Feb-1999 12:06<cr>><ex><et>
12:06:05.960 [0145r] <br/>
<sy><sy><sx>M00230001<cr>>001250 <cr>>015850<cr>>015850<cr>>0160035.0<cr>>017805<cr>>0180000.5<cr>>033030<cr>>03410.00<cr>>042N<cr>>M00330001<cr
><ex><et>
12:06:05.960 [0012s] <sy><sx>M017A<cr>><ex><et>
12:06:05.960 [0012s] <sy><sx>M017A<cr>><ex><et>
12:06:05.960 [0012s] <sy><sy><sx>M017A<cr>><ex><et>
12:06:05.960 [0012s] <sy><sy><sx><et>
12:06:05.960 [0012s] <sy><sy><sx><et>
12:06:05.960 [0012s] <sy><sy><sx><et>
12:06:05.960 [0012s] <sy><sy><sx><et>
12:06:05.960 [0012s] <sy><sy><ex><et>
12:06:05.960 [0012s] <sy><sy><ex><et>
12:06:05.960 [0007r] <sy><sy><ex><ex><et>
12:06:05.960 [0007r] <sy><sy><ex><ex><et>
12:06:05.960 [0007r] <sy><sy><ex><ex><et>
12:06:06.070 [0007r] <sy><sy><ex><ex><ex><ex><ex><</p>
```

|--|

Text	ASCII	Hex	Character Name	Symbol
<nu></nu>	0	0x00	null	
<sh></sh>	1	0x01	^A	Θ
<sx></sx>	2	0x02	^B	8
<ex></ex>	3	0x03	^C	¥
<et></et>	4	0x04	^D	٠
<eq></eq>	5	0x05	^E	2
<ak></ak>	6	0x06	^F	ŧ
<bl></bl>	7	0x07	^G	-
<bs></bs>	8	0x08	^H	-
<ht></ht>	9	0x09	^I	0
<lf></lf>	10	0x0a	٦^ J	Ξ
<vt></vt>	11	0x0b	^K	3
<ff></ff>	12	0x0c	^L	<b>Ŷ</b>
<cr></cr>	13	0x0d	^M	₽
<so></so>	14	0x0e	^N	Л
<si></si>	15	0x0f	^0	*
<dl></dl>	16	0x10	^P	
<d1></d1>	17	0x11	^Q	-
<d2></d2>	18	0x12	^R	<b>‡</b>
<d3></d3>	19	0x13	^S	
<d4></d4>	20	0x14	^T	<b>¶</b>
<nk></nk>	21	0x15	^U	§
<sy></sy>	22	0x16	<b>^</b> V	
<eb></eb>	23	0x17	^W	Ŧ
<cn></cn>	24	0x18	^X	1
<em></em>	25	0x19	^Y	₽
<ef></ef>	26	0x1a	^Z	<b>→</b>
<ec></ec>	27	0x1b	ESCAPE	÷
<fs></fs>	28	0x1c		
<gs></gs>	29	0x1d		++
<rs></rs>	30	0x1e		<b></b>
<us></us>	31	0x1f		Ŧ

# **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)
ł
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 crcl6(char *data_p,unsigned short start,unsigned short
                               TCPIP Generic Third Party
```

```
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++);</pre>
      do
      {
            for(i=0,data=(unsigned int)0xff & *data_p++;i<8;i++,data>>=1)
            ł
                   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
*/
#if defined(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);
}
2/5/99
                               TCPIP Generic Third Party
```