# Application Note #1

## Truck I/O on the M2000

**The M2000 can support different types of truck in/truck out applications.**

- **Tare database supporting up to 150 trucks stored (non volatile FLASH memory)**
- **IN/OUT loop memory (battery backed SRAM) supporting up to 150 trucks with the truck deleted on the outbound loop.**
- **IN/OUT loop memory supporting up to 150 trucks with the truck <u>not deleted</u> on outbound loop.**
- **The tare database and the loop database can be combined to work together.**

**Three examples are shown here supporting truck tares stored in the database memory and tares stored in the loop memory. <u>Software Version 1.29</u> is required for the examples below.**

# Trucks stored in the tare database

- **The tare database is meant for storing tare weights along with a 4 digit ID number in permanent memory.**
- **A tare database is meant for tare weights that are recalled on a regular basis and do not change often.**
- **Up to 150 tare weights can be stored in the tare database**
- **The tare database memory can be preserved up to 10 years without power**

# Managing Truck tares stored in the tare database

This application is identical to the previous truck in truck out examples in the manual but the difference is that we are storing trucks to the FLASH memory database. The M2000 FLASH memory is not affected by power loss of any kind. Use **101 print select** to erase all tare IDs in the tare database if required.

This examples demonstrates the following:

- Storing a truck tare to the database
- Recalling a truck tare from the database
- Manually entering a keyboard tare to the truck database
- Deleting a truck ID from the database
- Printing all weights and IDs in the database

The tare database is a permanent memory storage system for storing tare weights that do not change often. These tare weights are maintained regardless if the internal battery of the M2000 is removed.

If you are changing tare weights on a daily basis then it is recommended that the loop database be used instead.

The M2000 can store up to 150 trucks into non-volatile memory, which can be called by a 4-digit id number.

This application uses 6 tickets to perform the operations above.

**Key assignments:**

The IN key is assigned to ticket 200 for recalling tares from the database.
The OUT key is assigned to ticket 201 for storing scale tares to the database
PRINT/SELECT should be assigned to 0

To add a keyboard tare to the database the user must enter 202 followed by the PRINT/SELECT key. Enter the weight followed by the TARE key, and then enter an ID followed by the PRINT/SELECT key.

To delete a truck ID the user must enter 204 followed by PRINT/SELECT key. Enter the ID you want to delete followed by the print select key.

To print a list of IDs and weights in the database then enter 205 followed by the print select key.

This ticket uses formatting codes for an Epson tape printer.

**Ticket 200 (truck out sequence) assigned to OUT key**
This ticket is assigned to the OUT key. The user is prompted for a truck ID and then the indicator retrieves the stored tare weight for that truck. Gross, Tare and Net is printed.

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date

**C73** call truck out function and ask operator for truck ID in database
note: if an invalid ID or if CLEAR is pressed the tickets aborts here
**P916** print string "Truck ID:" to printer
**C79** print truck ID number

**13** CR
**10** LF

**P901** send the string "Gross" to the printer
**C74** print trucks gross weight
**13** CR
**10** LF

**P902** send the string "Tare " to the printer
**C75** print trucks tare weight
**13** CR
**10** LF

**P903** send the string "Net " to the printer
**C76** print trucks net weight
**13** CR
**10** LF

**13** CR
**10** LF

```
Johns's Gravel Co
19:50:38 14/07/2000

Outbound Truck ID: 123
Gross 5550 kg
Tare  3580 kg
Net   1970 kg
```

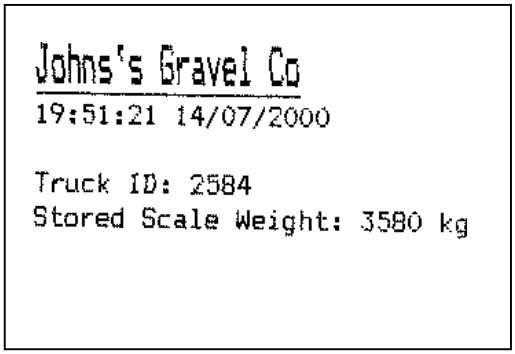
## Ticket 201 Store scale weight to database

This ticket can be assigned to the IN key. This ticket is called when the user wants to store a truck tare weight that is on the scale to the tare database.

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date

**C72** call add tare to database function and ask operator for truck ID
note: if an invalid ID or if CLEAR is pressed the tickets

**P906** print string "Truck ID:" to printer
**C79** print truck ID number
**13** CR
**10** LF
**P918** send the string "Stored Scale Weight:"
**C30** print the gross weight of the truck on the scale
**13** CR
**10** LF




## Ticket 202 Keypad tare weight to database

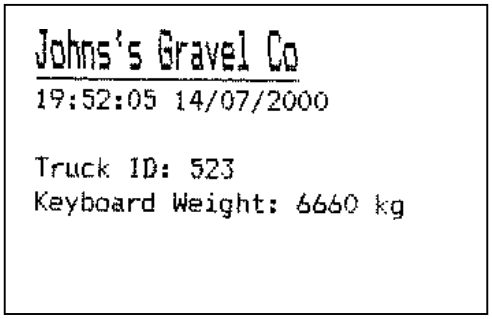
This function is the same as ticket 201 but instead of weighing a truck on the scale, tare weights can be entered from the keypad. Enter 202 followed by the print/select key. Enter the weight on the keypad followed by the tare key. Then enter the truck ID followed by the print/select key.

**C400** weight entry to ACC3 (enter weight followed by TARE or PRINT key)
**C88** store weight in ACC3 to the TARE database (ask for TARE ID)
note: if an invalid ID or if CLEAR is pressed the ticket aborts here

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date

**P906** print string "Truck ID:" to printer
**C79** print truck ID number
**13** CR
**10** LF

**P919** send the string "Keyboard Weight"
**C262** copy weight in ACC3 to ACC4
**C255** print weight in ACC4
**13** CR
**10** LF



## Ticket 203 Company name, time and date.

The ticket below prints the company name and time and date.

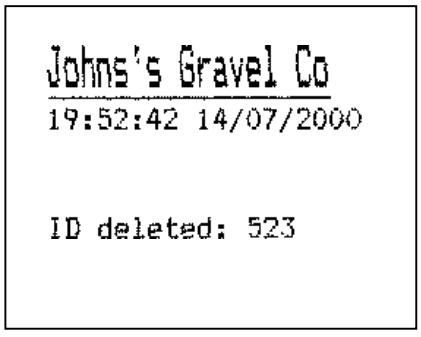
**13** CR send a carriage return and line feed to start with to the printer (just a habit)
**10** LF
**P100** set the printer to emphasized mode (darker letters)
**P101** turn on the double height for larger fonts
**P104** turn on underline mode
**74,111,104,110,115,96,115, 32 ,71,114,97,118,101,108, 32, 67,111** "John's Gravel Co."
**P108** reset the printer fonts
**13** CR
**10** LF
**C20** print the time
**32**
**C21** print the date
**13** CR
**10** LF
**10** LF

## Ticket 204 Delete Tare weight from database

This ticket is called to delete a truck from the database. A tare weight must be deleted from the database before the ID number can be reused or assigned a new tare weight. This function prints a ticket as a confirmation, but printing is optional.

**C80** ask for truck ID and delete truck from database
note: if an invalid ID or if CLEAR is pressed the ticket aborts here

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date
**10** LF
**P920** sent text "ID deleted:"
**C79** print truck ID number
**13** CR
**10** LF



## Ticket 205 Print a list of the truck database

This ticket is called when you want to print a list of the truck tares in the database

**C84** list all tickets in database (this is the only code allowed in this ticket)

```
TARE DATABASE memory
[ ID ] weight
-----------------
[ 123] 5550 kg
[5261] 5960 kg
[6237] 5960 kg
[4567] 5960 kg
[5679] 9130 kg
[3762] 9130 kg
[9641] 277770 kg
[1111] 6000 kg
[5673] 55550 kg
[5634] 3580 kg
[5555] 6170 kg
[2584] 3580 kg
[4444] 60 kg
[4448] 55550 kg
[ 567] 100 kg
[ 163] 630 kg
[6215] 780 kg
[5620] 2210 kg
[3563] 22220 kg
[ 486] 33360 kg
[5923] 123560 kg
[5623] 123560 kg
[7842] 123560 kg
[ 512] 5620 kg
[5298] 12630 kg
[9630] 12350 kg
[9318] 55620 kg
[ 564] 1230 kg
[ 524] 12670 kg
[5221] 11110 kg
[5539] 12340 kg
[ 541] 1230 kg
[3338] 810 kg
[2398] 5550 kg
[5418] 2220 kg
[ 159] 790 kg
[6632] 800 kg
[  12] 790 kg
[ 562] 120 kg
-----------------
END
```

# Truck Tares stored in the Loop Database

- **The loop database is meant for storing tare weights along with a 4 digit ID number in battery backed up memory.**
- **A loop database is meant for tare weights that change on a regular basis and usually are deleted right after use (truck in truck out applications).**
- **Up to 150 tare weights can be stored in the loop database**
- **The loop database memory uses an internal battery to preserve the memory when power is lost.**
- **There is an optional feature to <u>not delete</u> the truck ID and weight on the outbound loop.**

# Working with the loop database

Trucks stored in the tare database is designed for situations where you have a fleet of trucks that have tare weights that do not change often. In situations where trucks weigh in and weigh out on a regular basis the loop memory tare storage is the preferred choice.

The difference in the loop memory compared to the tare database memory is that the indicator uses static memory, which is faster and less wear then using flash memory. The loop memory is backed up by an internal battery in case of power loss. The loop memory can store up to 150 trucks and is completely independent to the tare database memory. Use **100 print select** to erase all tare IDs in the loop memory if required.

There are two scenarios for truck in/ truck out.

1. The first one is where the truck weighs in and then weighs out. On the weigh out the tare weight is deleted from the loop memory.

2. The second scenario is where the operator wants to weigh truck tare weights in the morning, or once a week, and does not want the truck tare weight deleted from the out loop memory. This improves productivity, as trucks only have to weigh out. Only a single C87 code "NO DELETE" is required which tells the indicator not to delete the truck ID number on the outbound loop. C87 should be inserted in the beginning of the inbound and outbound tickets.

The ticket example below is very similar to previous examples of truck in/truck out but demonstrates the following:

- The truck inbound function
- The truck outbound sequence with the "NO DELETE" feature
- Printing all weights and IDs in the loop memory
- How to add an incrementing serial number to the ticket
- Adding the no delete function

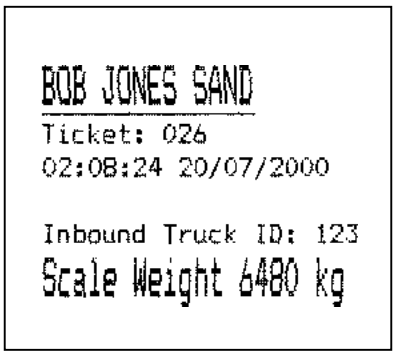
The example that follows is designed for an Epson tape printer. Printer codes in ticket 203 may need to be modified for other printers. This example shows the "no delete feature". To remove "no delete" simply delete C87 in ticket 200 and 201.

### Ticket 200 truck in loop:

This ticket should be assigned to the IN key.
Asks for an ID and stores the scale weight to the loop memory.

**C103** jump to another ticket (203) and prints it
**203** ticket 203 is called to print the ticket header, time and date
**C87** NO DELETE, tell the indicator that the no delete option is used
**C70** call truck in function and ask operator for truck ID
note: if an invalid ID or if CLEAR is pressed the tickets
**P915** print string "Inbound Truck ID:" to printer
**C79** print truck ID number
**13** CR
**10** LF

**P101** turn on the double height for larger fonts
**P909** send the string "Truck Weight"
**C30** print the gross weight of the truck on the scale
**P108** reset the printer fonts
**13** CR
**P114** form feed multiple lines
**9** number of lines for form feed is 9



### Ticket 201 truck out loop:

This ticket should be assigned to the OUT key.
Asks for a truck ID number and recalls the tare weight from the loop memory.
Because the "no delete" C87 is executed the indicator will not delete the truck ID from the loop memory.

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date
**C87** NO DELETE, tell the indicator that the no delete option is used
**C71** call truck out function and ask operator for truck ID
note: if an invalid ID or if CLEAR is pressed the tickets aborts here
**C916** print string "Truck ID:" to printer
**C79** print truck ID number

**13** CR
**10** LF
**P101** double height ON

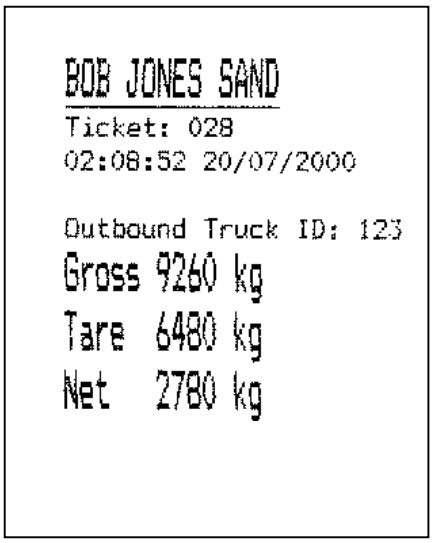
**P901** send the string "Gross " to the printer
**C74** print trucks gross weight
**13** CR
**10** LF

**P902** send the string "Tare " to the printer
**C75** print trucks tare weight
**13** CR
**10** LF

**P903** send the string "Net " to the printer
**C76** print trucks net weight
**13** CR
**10** LF

**P108** reset printer font
**13** CR
**10** LF

**P114** form feed 5 lines
**5** number of lines is 5



## Ticket 203 Company name, time and date and ticket number.

The ticket below prints the company name and time and date and serial number. It is called from tickets 200 and 201. The ticket serial number increments every time this ticket is called.

**13** CR send a carriage return and line feed to start with to the printer (just a habit)
**10** LF
**P100** set the printer to emphasized mode (darker letters)
**P101** turn on the double height for larger fonts
**P104** turn on underline mode
**66,79,66, 32, 74,79,78,69,83, 32, 83,65,78,68** "Bob Jones Sand"
**P108** reset the printer fonts
**13** CR
**10** LF
**P921** Send the string "Ticket:" to the printer
**C28** print the ticket number
**C27** increment the ticket number count
**C20** print the time
**32**
**C21** print the date
**13** CR
**10** LF
**10** LF

### Ticket 204

The ticket below prints all the ID numbers in the loop memory

**C78** print all trucks in the loop memory (no other codes should be in this ticket)

```
IN/OUT DATABASE memory
[ ID ] weight
------------------
[ 123] 6480 kg
[1111] 3550 kg
[ 555] 740 kg
[ 444] 740 kg
[9237] 740 kg
[5180] 9160 kg
------------------
END
```

# Combining the tare database with the loop database

- **The tare database and loop database are independent from each other.**
- **A tare database and loop database can be setup to work together.**
- **In this mode if the indicator does not find an ID in the tare database, it will proceed to search in the loop database**
- **Used in applications where customers have permanent truck tares for account customers/fleets but also want in/out weighing for infrequent customer visits.**

# Integrating the tare and loop databases

BJs rock & fill has a fleet of their own trucks, and some regular account customers. The truck weights for these trucks are stored in the tare database so trucks only have to weigh out. BJ also has cash customers who visit infrequently with different types of trucks. For the cash customers he wants to issue them a temporary ID and do a truck in/truck truck/out loop sequence.

This ticket is very similar to the two pervious examples but we are integrating them together. To integrate the two databases careful planning is required for the ID numbers. The ID number used in the tare database cannot be the same as in the loop database. If there are duplicate ID numbers the tare database will take precedence over the loop database.

In this example BJ will use 4 digits ID numbers for his account customers, and 3 digits ID numbers for his cash customers, that way he will never have duplicate IDs by mistake.

To integrate the two databases insert C89 in the beginning of the outbound ticket for the tare database. This tells the indicator that if the ID cannot be found in the tare database then search in the loop database.

We are simply going to implement this example simply by cutting and pasting from the previous two examples.

## Tickets used:

**Ticket 200**
The truck out ticket and is assigned to the OUT key.
This ticket will search the tare database first, if no match is found then it searches the loop database. Truck IDs are automatically deleted from the loop database.

**Ticket 201**
The inbound ticket for weighing trucks into the loop database and is assigned to the IN key.

**Ticket 202**
Is the inbound ticket for weighing trucks into the tare database. This ticket is called when Bob want to add another account customer to the database.

**Ticket 203**
Company name and time and date. This ticket is the prints header and is called from ticket 200,201.

**Ticket 204**
Delete a truck from the tare database. This ticket asks for an ID number to delete a truck from the database.

**Ticket 205**
Print all trucks in the tare database and the loop database.

**Ticket 200 (truck out sequence) assigned to OUT key**
This ticket is assigned to the OUT key. The user is prompted for a truck ID and then the indicator retrieves the stored tare weight for that truck. Gross, Tare and Net is printed. This ticket has the special C89 code added in the beginning of the ticket.

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date
**C89** tell the indicator to check the loop database after checking the tare database
**C73** call truck out function and ask operator for truck ID in database
note: if an invalid ID or if CLEAR is pressed the tickets aborts here
**P916** print string "Truck ID:" to printer
**C79** print truck ID number

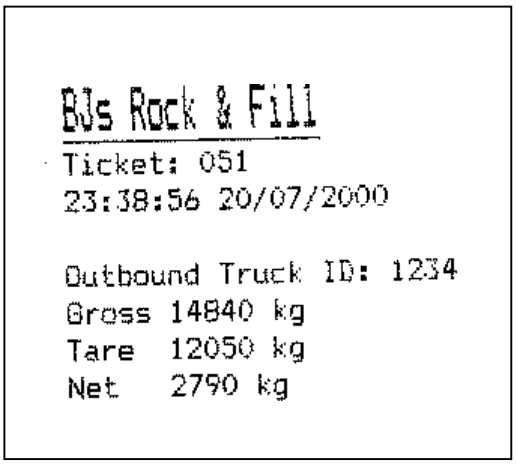
**13** CR
**10** LF

**P901** send the string "Gross" to the printer
**C74** print trucks gross weight
**13** CR
**10** LF

**P902** send the string "Tare " to the printer
**C75** print trucks tare weight
**13** CR
**10** LF

**P903** send the string "Net " to the printer
**C76** print trucks net weight
**13** CR
**10** LF

**13** CR
**10** LF



## Ticket 201 truck-in stored to the loop database

This ticket should be assigned to the IN key.
Asks for an ID and stores the scale weight to the loop memory.
This is ticket is called for truck in/truck out weighing.

**C103** jump to another ticket (203) and prints it
**203** ticket 203 is called to print the ticket header, time and date

**C70** call truck in function and ask operator for truck ID
note: if an invalid ID or if CLEAR is pressed the tickets
**P915** print string "Inbound Truck ID:" to printer
**C79** print truck ID number
**13** CR
**10** LF

**P101** turn on the double height for larger fonts
**P909** send the string "Truck Weight"
**C30** print the gross weight of the truck on the scale
**P108** reset the printer fonts
**13** CR
**P114** form feed multiple lines
**9** number of lines for form feed is 9

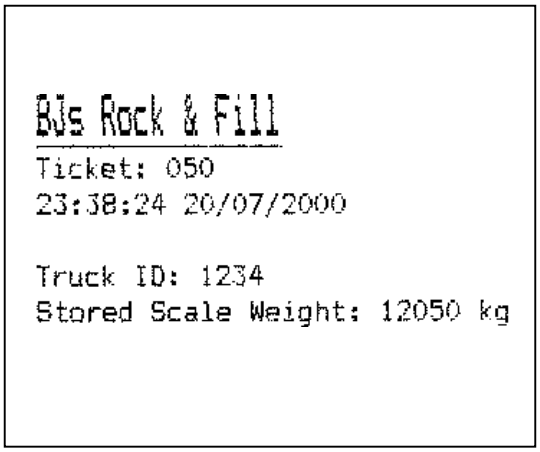
## Ticket 202 truck-in stored to the tare database:

This ticket is called when the user wants to store a truck tare weight that is on the scale to the tare database. The user enters 202 printselect to call this ticket.
This ticket is used to add account customers to the tare database.

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date

**C72** call add tare to database function and ask operator for truck ID
note: if an invalid ID or if CLEAR is pressed the tickets

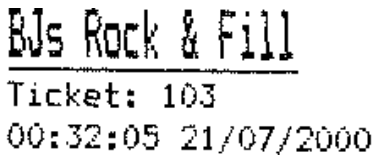
**P906** print string "Truck ID:" to printer
**C79** print truck ID number
**13** CR
**10** LF
**P918** send the string "Stored Scale Weight:"
**C30** print the gross weight of the truck on the scale
**13** CR
**10** LF



## Ticket 203 Company name, time and date and ticket number.

The ticket below prints the company name and time and date and serial number. It is called from tickets 200, 201 and 204. The ticket serial number increments every time this ticket is called.

**13** CR send a carriage return and line feed to start with to the printer (just a habit)
**10** LF
**P100** set the printer to emphasized mode (darker letters)
**P101** turn on the double height for larger fonts
**P104** turn on underline mode
**66,74,115, 32, 82,111,99,107 32, 38, 32 70,105,108,108** "BJs Rock & Fill"
**P108** reset the printer fonts
**13** CR
**10** LF
**P921** send the string "Ticket:" to the printer
**C28** print the ticket number
**C27** increment the ticket number count
**C20** print the time
**32**
**C21** print the date
**13** CR
**10** LF
**10** LF

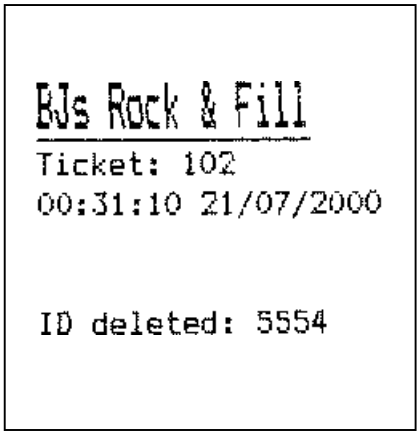


## Ticket 204 Delete Tare weight from database

This ticket is called to delete a truck from the tare database. A tare weight must be deleted from the database before the ID number can be reused or assigned a new tare weight. This function prints a ticket as a confirmation, but printing is optional.

**C80** ask for truck ID and delete truck from database
note: if an invalid ID or if CLEAR is pressed the ticket aborts here

**C103** jump to another ticket (203) and print it
**203** ticket 203 is called to print the ticket header, time and date
**10** LF
**P920** sent text "ID deleted:"
**C79** print truck ID number
**13** CR
**10** LF



## Ticket 205 Print a list of the truck database and loop database

This ticket is called when you want to print a list of the truck tares in the database and the loop database.

**C84** list all tickets in tare database (C84 and C78 are only allowed in this ticket)
**C78** list all tickets in loop database

```
TARE DATABASE memory
[ ID ] weight
----------------
[5261] 5960 kg
[5679] 9130 kg
[3762] 9130 kg
[9641] 277770 kg
[1111] 6000 kg
[5673] 55550 kg
[5634] 3580 kg
[5555] 6170 kg
[2584] 3580 kg
[4448] 55550 kg
[1234] 12050 kg
[5620] 2210 kg
[3563] 22220 kg
[5923] 123560 kg
[5623] 123560 kg
[5298] 12630 kg
[9630] 12350 kg
[9318] 55620 kg
[5221] 11110 kg
[5539] 12340 kg
[3338] 810 kg
[2398] 5550 kg
[5418] 2220 kg
----------------
END

IN/OUT DATABASE memory
[ ID ] weight
----------------
[ 567] 18060 kg
[ 552] 17450 kg
[ 555] 11880 kg
[ 569] 9100 kg
[ 123] 6310 kg
----------------
END
```