Reference Manual

IND700 Shared Data Reference

Weighing Terminal

METTLER TOLEDO

METTLER TOLEDO Service

Congratulations on choosing the quality and precision of METTLER TOLEDO. Proper use of your new equipment according to this Manual and regular calibration and maintenance by our factory-trained service team ensures dependable and accurate operation, protecting your investment. Contact us about a service agreement tailored to your needs and budget. Further information is available at ▶ www.mt.com/service.

There are several important ways to ensure you maximize the performance of your investment:

1 **Register your product**: We invite you to register your product at www.mt.com/productregistration so we can contact you about enhancements, updates and important notifications concerning your product.

2 **Contact METTLER TOLEDO for service**: The value of a measurement is proportional to its accuracy – an out of specification scale can diminish quality, reduce profits and increase liability. Timely service from METTLER TOLEDO will ensure accuracy and optimize uptime and equipment life.

- ➡ **Installation, Configuration, Integration and Training**: Our service representatives are factory-trained weighing equipment experts. We make certain that your weighing equipment is ready for production in a cost effective and timely fashion and that personnel are trained for success.
- ➡ **Initial Calibration Documentation**: The installation environment and application requirements are unique for every industrial scale so performance must be tested and certified. Our calibration services and certificates document accuracy to ensure production quality and provide a quality system record of performance.
- ➡ **Periodic Calibration Maintenance**: A Calibration Service Agreement provides on-going confidence in your weighing process and documentation of compliance with requirements. We offer a variety of service plans that are scheduled to meet your needs and designed to fit your budget.

Table of Contents

1 Introduction and Overview

The Shared Data (SD) Server allows access to data in the IND700. It is also the primary interface for sending commands and exchanging data between remote applications and the Engine in the IND700. The Engine is the portion of the terminal firmware which controls the weighing functions.

Compliance information

National approval documents, e.g. the FCC Supplier Declaration of Conformity, are available online and/or included in the packaging.

▶ www.mt.com/ComplianceSearch

Manuals download

Please scan the QR code below and download from ▶ www.mt.com/IND700-downloads.

1.1 IND700 Shared Data Design

The Shared Data concept has been a very powerful and flexible tool. It provides mechanisms for both storing system data and for providing interfaces between remote applications, and IND700 Engine.

Shared Data Design Concepts

The following are some important Shared Data design concepts incorporated into the IND700 Shared Data:

Shared Data provides Local and Remote Applications and the Engine very fast access to the permanently stored data. Shared Data access time is less than 350 microseconds.

- Provides a consistent naming convention among all Shared Data fields. Local and Remote Applications access a Shared Data field using a six-character UNICODE name. Names provide consistency to Applications in accessing Shared Data fields in successive versions of Shared Data. The names for existing fields will remain the same even when new fields are added or when new physical storage locations are assigned to existing data. Shared Data uses a binary search of the names in the Data Dictionary to find a field definition quickly.
- Shared Data is organized into "object-oriented" blocks that make it is easier for Application programmers and users to understand how to use Shared Data.
- Data types are standardized and limited to a small, defined, consistent group.
- Provides data access control on an individual field basis. IND700 has four levels of access security: Operator, Supervisor, Service, and Administrator.
- The IND700 uses its SSD to store protected setup fields that change infrequently, and dynamic memory (flash and BRAM) to store data that changes often.
- Shared Data supports "callbacks" that alert a task when a Shared Data field changes. An Application or Engine can "Register a Callback Routine" for a particular Shared Data field. Then, when a task writes a new value to a Shared Data field that has a registered callback, Shared Data calls the registered callback routine.
- Shared Data supports both "native" and "string representation" access to data fields. However, Shared Data always stores the data fields in their native format. When an Application accesses a Shared Data field in its native data format, such as binary floating point or integer number representations, Shared Data simply copies the data between its storage and the Application interface. When Applications access the Shared Data using a string data format, Shared Data makes the data conversion between the native and the string data format.
- Shared Data provides access to an entire Shared Data block with a single read or write command. Applications can access the block of data in either native format or string format. When an application accesses the data in native format, Shared Data returns a "C-style structure" that matches the native format of the data. When an application accesses the data in string format, Shared Data converts each individual field to its string format, separating fields with a caret ('^').
- A Change Log file forms a service record that the customer or service technician can review to find or validate changes to the IND700 setup. Recording all process changes is becoming an important requirement for U.S. pharmaceutical applications.

- Validates changes to data fields. If Shared Data finds the new value is not legal, it does not update the field and returns an error status to the Application.

1.2 Shared Data Access

Access to Shared Data is controlled by the same user management as the terminal itself. Write access to all writable Shared Data requires an Admin login. Depending on the logged-in user, different SD variable classes will be available.

Communication can be carried out using either a serial (RS232/RS422) or Ethernet connection.

1.3 Shared Data Name Structure

Each Shared Data field has a six-character alphanumeric "name" that the Application uses to access the Shared Data field. The name contains the class, instance, and attribute of the Shared Data variable, each of which is two characters long. For example, Shared Data variable "sp0106" is the latched/unlatched target setting of the single (instance) scale the IND700 will support at one time. The name is constructed as follows:

sp	=	Class	Full Target Process Data (Note: Class can be written as all uppercase or all lower case)
01	=	Instance	Scale #1
06	=	Attribute	Target Latch Type

1.4 Shared Data Storage Types

There are two types of IND700 Shared Data: Dynamic data, and data stored on the terminal's SSD.

The following letters are used in this document to identify the data type of each block:

- D -- Dynamic (Dynamic RAM) Shared Data
- PP -- Protected Process and Setup (BRAM) Shared Data

1.5 Command Triggers

Shared Data can be used to trigger commands to the Engine. The Engine then uses other Shared Data status fields for reporting the activity and the results of its commands. Typically, command triggers reside in Dynamic Shared Data. There are many fields in Shared Data that enable applications to define command triggers.

Callbacks are a powerful mechanism for sending commands to the Engine or to Applications through writes to Shared Data. The destination task must first register a callback to Shared Data on its designated command field. Then, local or remote processes may initiate a write to the field to trigger a callback to the destination task. The IND700 designates the special Shared Data fields that can use callbacks as "real-time" fields. **In this document, "rt" designates real-time fields, while "na" designates non-real-time fields that do NOT support callbacks.**

Edge-Sensitive commands are also real-time fields, but the IND700 only makes a callback to process these commands when the field transitions from zero to a non-zero value. **In this document, "rc" designates edge-sensitive command fields.**

1.6 Application Commands to the Engine

Applications can issue commands to the IND700 Engine using the Shared Data Command Triggers. The Application sets the Command Trigger to 1 to issue the command. This generates a callback to the Engine, which detects that the Command Trigger is set and processes the command. When it has processed the command, the Engine sets the Command Trigger back to 0.

A Shared Data Command Status is associated with each Command Trigger. The Application can read the Command Status to determine the completion status of the command. 0 indicates that the command was successful, and 1 indicates the command is in progress. A status greater than 1 is a specific failure code. The Application can monitor the Command Trigger or Command Status to know when the command is complete.

For example, to issue a Tare Command to the scale, the Application sets Shared Data field wc0101 to 1. Then, the Application monitors for the Shared Data field wx0101 to be set to 1, which indicates the command is in progress. Then, it monitors for wx0101 to change again to get the completion status of the command. The Engine then sets wc0101 to 0 when it completes the command.

1.7 Data Format Types

IND700 Share Data supports the following data types:

IND700 Shared Data Format Types

Mnemonic	Data Type	Description
Bl	I1	Boolean fields are one-byte integers, but can only take avalue of 0 or 1
By	I1	One byte integer
US	UI2	Two byte unsigned integer
UL	UI4	Four byte unsigned integer
F	R4	Single precision floating point
D	R8	Double precision floating point
ABy nn[1] ABl nn[1]	Array I1 Array I1	Array of I1 Array of I1 Boolean
S mm[2]	Array U12	A Unicode String. NULL terminated. Array of UI2
AL nn[1]	Array U14	Array of UI4
Struct	Struct	Composite structure of the entire block

1. "nn" represents the length of the array
2. "mm" represents the maximum length of the Unicode String, including the null terminator

1.8 Shared Data Access Control

IND700 Shared Data provides data access control for individual fields, rather than on a block basis. The Shared Data Dictionary holds the "access privileges". The "access privilege" attributes for each Shared Data field determines how local and remote applications can access the fields. Generally, anyone can read any Shared Data element. The notable exceptions are password fields, which only the IND700 System modules may read. Hard-coding in Shared Data restricts read-access to the password fields. The Shared Data Dictionary defines the write-access privileges on an individual field basis, according to the class of the user.

There are three classes of user – Administrator, Supervisor, and Operator. The Administrator class always has the maximum possible write-access capability. However, not even an Administrator can write into "Read Only" fields. Typical Read Only fields are Real Time Data fields that contain the weight data for the scale.

There is no enforced class hierarchy below Administrator. Other classes have write-access to fewer Shared Data fields. By convention, the Operator class has the fewest rights, and the Supervisor class is a superset of the Operator class. Shared Data fields have factory-default access rights that meet most Application needs. In the default definition, each higher class has also write-access privileges to all data assigned to lower classes.

To satisfy legal metrology regulations or customers' security concerns, it is often necessary to limit terminal write-access after the customer has installed the terminal. For example, no user of any class may change setup parameters after a government inspector has certified and sealed the terminal.

If the IND700 is set to non-approved mode in the scale's metrological configuration, authorized users can write to Share Data fields according to their access rights. If the terminal is set to approved mode, scale configuration Shared Data fields are write-protected.

1.9 Validating Setup Data

IND700 Shared Data validates changes to Protected Setup and Calibration EEPROM fields. It compares the new value with the range of legal values stored in the Shared Data Dictionary. If Shared Data finds the new value is not legal, it does not update the field and returns an error status to the Application.

Shared Data does not validate all fields. It only validates those that it can validate using a table of values. It does not validate those fields that require special programming logic to validate.

Shared Data supports an Application command that returns the validation criteria for a particular field to the Application so the Application can display the list of legal values.

The Shared Data Dictionary has different validation criteria based on the type of validation required. Some of the validation types include:

- **Boolean validation** -- Only zero (0) or one (1) is a legal value

- **Range validation** -- Only values within a range are valid. The Data Dictionary contains the minimum and maximum legal values. For example, integer values from one to five are valid, or floating-point values from 0.0 to 9.9 are valid.
- **List validation** -- Only values in a list of values are valid.
- **No validation**

1.10 Shared Data Server Commands

After connecting to the Shared Data Server in the IND700, several commands are available for use. All commands can be given in either upper- or lower-case letters. The quotation marks shown are for clarity only and should not be transmitted. Valid commands are described in the following sections.

NOTICE

Response Format

"read", "write", and "callback" message responses have a formatted header. The first two characters indicate the status. "00" is the success status. "99" is a failure status. The next character is the type of message, "r", "w", or "c". The next three characters are a sequence number, which cycles from 001 to 999, and then starts over again.

"user" Command

A client must login to the SDSV using the "user" command before accessing Shared Data. The server validates the username and sends a response message back to the user. The SDSV responds with [Access OK] if no password is required or [Enter password] if a password is required.

A client can use only the "user", "pass", "help" and "quit" commands before successfully logging on.

Format: user username

Response 1: 12 Access OK

Response 2: 51 Enter Password

"pass" Command

The user enters a password using the "pass" command. If the password is valid, the server displays the [Access OK] message. If not valid, the server displays the [No access] message.

Format: pass password

Response: 12 Access OK

"help" Command

The "help" command returns the list of valid commands for IND700.

Format: help

Response: 02 USER PASS QUIT READ R WRITE W SYSTEM CALLBACK XCALLBACK GROUP RGROUP XGROUP CTIMER LOAD SAVE HELP NOOP CONTOUT XCOUNTOUT PRINTOUT XPRINTOUT

"quit" Command

The "quit" command terminates the TCP/IP connection.

Format: quit

Response: 52 Closing connection

"read" Command

The "read" command allows the client to read a list of one or more Shared Data fields. An individual field or an entire block can be read. If more than one field is requested, the fields should be separated by a space. If successful, the server responds with a separated list of values in ASCII format. The server separates individually requested fields with a "~"; and Shared Data separates items within a block with a "^". If an error is detected, the server responds with an error message. The maximum length of the reply message is 1,024 characters.

Format: read SDV#1 SDV#2

Example 1: read wt0101 wt0103

Response 1: 00R003~ 17.08~lb~

Example 2: read sp0100 (reads entire block)

Response 2: 00R012~XP/
0163M^1^^78^20.500000^0^0^0^1.200000^3.500000^0.150000^0.050000^0^0.000000^0.000000^0^0^0^0^0^0^1^0.000000^0.000000^0.000000^0.000000^0.000000^~

"write" Command

The ″write″ command allows the client to write a list of one or more Shared Data fields. A single field or an entire block can be written. The maximum length of the write message is 1,024 characters. Items within a list of writes must be separated with a ″~″. You must separate items within a block with a ″^″.

Format: write SDVblock#1=value1^value2^ value3 write SDV#1=value1~SDV#2=value2~SDV#3=value3

Example 1: write ak0100=abc^def^hij^lmn (writes fields into a block)

Response 1: 00W006~OK

Example 2: write aj0101=12.56~aj0150=987.653 (writes fields within a list)

Response 2: 00W007~OK

Note that the "write" command can be abbreviated to the letter "w" if desired.

"systat" Command

The ″systat″ command returns a description of the terminal′s resource utilization, such as CPU loada and memory use.

"noop" Command

The "noop" command performs no task; it checks communication and returns an [OK] response message.

Format: noop

Response: 00OK

"callback" Command

The ″callback″ command allows the client to define one or more fields for which the Shared Data Server sends a message to the client when the value of the callback field changes. Only certain SDV may be included in a callback command. These SDV are noted by an ″rc″ or ″rt″ status in the column after the structure column in the Shared Data document. Mainly, these are triggers that are used in the terminal.

SDV with a status of “na” are not real-time SDV and cannot be used in callbacks.

The callback message contains one or more changed field names and the new value for each field. A maximum of twelve callback fields can be specified. The ″ctimer″ command specifies the minimum time between repeated callback messages.

Format: callback SDV#1 SDV#2

Example: callback st0102 st0103 st0104

Response 1: 00B001~OK

Response 2: 00C005~st0102=0^st0103=1^st0104=1 (sent when all of the SDV change)

Response 3: 00C006~st0104=0 (sent when only st0104 changes)

"xcallback" Command

The ″xcallback″ command allows the client to remove one or more callback fields from the list of current SDV.

Format: xcallback SDV#1 SDV#2 or xcallback all (removes all callbacks)

Example: xcallback st0102 (removes st0102 SDV from callback)

Response: 00X008~OK

"contout" Command

The ″contout″ command allows the client to define the continuous output string as a callback field. The Console Print Server sends a message to the client at each continuous output. The continuous output message is either in the Standard METTLER TOLEDO Continuous Output format or in a continuous template format. The ″ctimer″ command specifies the minimum time between repeated callback messages. The ″xcontout″ command removes the registration from the terminal and the communication will stop.

Format: contout

Response: 00G008~number CONTOUT streams=1

When a continuous output occurs to the Ethernet port, the data will be sent to the client formatted as selected in setup:

Data: 00C004 4! 354 236

00C005 4! 354 236

"xcontout" Command

The "xcontout" command allows the client to remove the continuous output callback, thus ending the registration so no further continuous outputs will be available.

Format: xcontout

Response: 00X070~CONTOUT

2 Scale Data

When any scale configuration shared data is modified, it is necessary to trigger wc—08, apply setup shared data (page 2-3). This restarts the scale using the new configuration.

2.1 Dynamic Scale Weight (WT)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Access level is not customizable
Class Code	wt
Data Type	D
Instances	Instance 00 = Selected scale Instance 1- 4 = Scale platforms 1 – 4 Instance 5 = Sum scale.

wt--00	Composite wt block	Struct	na	Composite of entire block
wt--01	Displayed Gross Weight	S13	rt	
wt--02	Displayed Net Weight	S13	rt	When user has enabled MinWeigh, the first character contains an '*' when the MinWeigh conditions are not met.
wt--03	Weight Units	S4	rt	**lb** pounds, **kg** kilograms, **g**rams, **oz** ounces, **ozt**roy, **dwt** pennyweights, metric **t**ons, **ton**, or custom units name
wt—04	Displayed Aux Gross Weight	S13	rt	Shows the current displayed gross weight converted to Aux Units
wt—05	Displayed Aux Net Weight	S13	rt	Shows the current displayed net weight converted to Aux Units
wt—06	Aux Weight Units	S7	rt	**lb** pounds, **kg** kilograms, **g**rams, **oz** ounces, **ozt**roy, **dwt** pennyweights, metric **t**ons, **ton**, or custom units name
wt—08	Displayed Rate	S13	rt	
wt—10	Rounded Gross Weight	D	rt	Gross weight rounded to the selected increment size, but displayed in SD at smallest division value possible
wt—11	Rounded Net Weight	D	rt	Net weight rounded to the selected increment size, but displayed in SD at smallest division value possible
wt—16	Continuous Output Status A	By	rt	Status of Bit A of Standard METTLER TOLEDO Continuous
wt—17	Fine Gross Weight	D	Rt	Gross weight displayed to the smallest division value possible
wt--18	FineNet Weight	D	rt	Net weight displayed in the smallest division value possible.
wt--19	Weight Range	By	rt	0, 1, 2, or 3

Method

The Engine updates the Dynamic Weight Shared Data at every weight update, whenever the weight changes. Typically, this occurs up to 20 times per second, but can vary depending on the load cell type and the application-type setting in cs--21.

The Engine converts the weight from the raw filtered counts it receives from the scale boards to the Legal-For-Trade weight. The Engine signals the Weight Display and SmartTrac Visualization task or an Application Task indicating that new weight is ready. The Engine sets this signal whenever weight changes, up to a maximum rate of 10 times per second. If the weight does not change for an extended time, the Engine will set the trigger just to refresh the weight display.

The Engine periodically re-writes the Shared Data weight fields every few seconds even when there is no change to the weight data.

2.2 Scale Process Data (WS)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Read only. Access level is not customizable
Class Code	ws
Data Type	PP
Instances	5

ws--00	Current Scale Mode	By	na	G = Gross, N = Net
ws--02	Rounded Tare Weight	D	rt	Tare weight rounded to selected increment size, but displayed in SD at the smallest division value possible.
ws--09	Tare Source Stsring	S2	rt	PT = Keyboard tare, else T
ws--10	Displayed Tare Weight	S13	rt	
ws--11	Displayed Aux Tare Weight	S13	rt	

Method

The Engine maintains its scale process data in this block. This scale process data may change frequently but must be stored permanently. The Scale Tare Setup section describes how the Engine uses the tare process data in this block.

2.3 Scale Commands (WC)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Operator
Class Code	wc
Data Type	D
Instances	Instance 00 = Selected scale Instance 1- 4 = Scale platforms 1 – 4 Instance 5 = Sum scale.

wc--01	Pushbutton Tare Scale	Bl	rc	Appl. sets from 0 to 1 to trigger coommand
wc--02	Clear Scale	Bl	rc	
wc--03	Demand Print Scale	Bl	rc	
wc--04	Pushbutton Zero Scale	Bl	rc	
wc--05	Switch to Primary Units	Bl	rc	
wc--06	Switch to Secondary Units	Bl	rc	
wc--07	Toggle Primary/Secondary Units	Bl	rc	Set from 0 to 1 to toggle units
wc--14	Capture Raw Counts	Bl	rc	Toggle raw counts (py0121) capturing on/off. wx--48 contains toggle state

Methods

For example, to issue a Tare Command to Scale A, the application sets Shared Data field wc0101 = 1.

After receiving the callback, the Engine sets wx0101 = 1 to indicate the command is in progress. When the command is complete, the Engine sets wx0101 = 0 to indicate the command is successful or wx0101 = 2 to 255 for a specific error code. It sets wc0101 = 0 so the application can trigger the command again later. The application can register a callback on wx0101 to monitor when the command is complete and to get the completion status of the command.

2.4 Scale Statuses (WX)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Read only. Access level is not customizable
Class Code	wx
Data Type	D
Instances	Instance 00 = Selected scale Instance 1- 4 = Scale platforms 1 – 4 Instance 5 = Sum scale.

wx--01	Tare Scale Status	By	rt	General Command Completion Statuses: 0 = Success 1 = Command in Progress 2-255 = Specific error code 97 = Scale in invalid mode 98 = Invalid function parameter 99 = No SD access 0 = Tare completed successfully 1 = Tare in progress 2 = Scale in motion during tare = 2 3 = Pushbutton tare not enabled 4 = Programmable tare not enabled 5 = Chain tare not permitted 6 = Only incremental chain tare permitted 7 = Tare not in rounded increment value 8 = Tare value too small 9 = Taring when power up zero not captured 10 = taring over capacity 11 = taring under zero 12 = Tare value exceeds limit 13 = Must clear tare at gross zero 14 = Scale in wrong mode during tare 15 = IDNET scale error 16 = Clear Tare command while waiting for Tare command no motion
wx--02	Clear Tare Status	By	rt	Same as tare statuses

wx--03	Print Scale Status	By	rt	0 = Printing completed successfully 1 = Printing in progress 2 = Print connection not found 3 = Printing busy 4 = Printing error 5 = Printing not ready to print 6 = Printing scale in motion 7 = Printing scale overcapacity 8 = Printing scale under zero 11 = Printing not allowed 12 = Printing not enabled 13 = No demand print, but continuous print completed OK 14 = Scale below minimum print weight
wx--04	Zero Scale Status	By	rt	0 = Zero completed successfully 1 = Zero in progress 2 = Scale in motion during zero 3 = Illegal scale mode during zero 4 = Scale out of zeroing range 5 = IDNET zero command timeout 6 = Pushbutton zero disabled 7 = Command timeout error 8 = Scale communications disabled
wx--05	Switch to Primary Units Status	By	rt	
wx--06	Switch to Secondary Units Status	By	rt	
wx--07	Toggle Primary/Secondary Status	By	rt	
wx--14	Capture Raw Counts Status	Bl	rc	
wx--31	Motion	Bl	rc	0 = no, 1 = yes
wx--32	Center of Zero	Bl	rc	0 = no, 1 = yes
wx--33	Over Capacity	Bl	rc	0 = no, 1 = yes
wx--34	Under Zero	Bl	rc	0 = no, 1 = yes
wx--35	Net Mode	Bl	rc	0 = no, 1 = yes
wx--36	Printing in Progress	Bl	rc	
wx--44	Selected Scale	Bl	rc	
wx--48	Capture Raw Counts Status	Bl	rc	1 = scale capturing raw counts (py0102) at every weight update. wc—14 toggles state

Methods

The Engine sets the first set of statuses to reflect the status of commands to the scale. The second set of statuses to show the dynamic run-time status of the scale weight. An Application or PLC can get the multiple scale status bits with a single read of the Composite Status fields.

2.5 Scale Setup (CS)

cs—01	Scale Type	By	na	65 = Analog scale 78= none 82 = RBrick-MFR SICSpro balance 83 = SICS scale Lab Balance 84 = POWERCELL 85 = Sum Scale
cs—02	Scale Location	By	na	
cs--03	Scale ID	S21	na	
cs--04	Rate Weight Units	By	na	0 = None, 1 = pounds, 2 = kilograms, 3 = grams, 4 = metric tons, 5 = tons, 6 = lb-oz, 7 = troy ounces, 8 = penny weights, 0 = ounces, 10 = custom units
cs--07	Rate Time Units	S2	na	**N**o, **S**ec, **M**in, **H**our
cs--66	PDX Load Cell Filter Poles	By	na	
cs--68	Synchronous Continuous Output	By	na	
cs--70	PDX Load Cell Filter Frequency	D	na	PDX load cell filter corner frequency (Hz)
cs--71	Scale Location	S21	na	Second of the three text fields "model," "location" and "identification."

2.6 System Processing Data (XT)

xt0100	Currently selected measuring device	By	rt

2.7 Cell Calibration (CC)

cc--01	Calibrated Cell Counts 1-24	L	na
cc--02	Calibrated Span Counts 1-24	Aby 24	na

2.8 Scale Calibration (CE)

ce--01	Address of First Load Cell
ce--02	Number of Load Cells
ce--03	Primary Units
ce--04	Number of Ranges
ce--05	Increment Size
ce--06	Number of Ranges
ce--07	Number of Ranges
ce--08	Number of Ranges
ce--09	Number of Ranges
ce--10	Scale Capacity
ce--19	Calibration Units
ce--20	Zero Calibration Counts
ce--21	Span Calibration Counts
ce--22	Calibration Test Weight
ce--38	Base Serial Number

2.9 Cell Shift Adjust (CX)

cx--01	Shift Constants 1 – 24

2.10 PDX Dynamic Data (PY)

py0101	PDX Cell Scan Table	ABy 24	na	Ordered list of POWERCELL addresses used in polling the POWERCELLs
py0102	PDX Cell Counts	Struct	na	Composite of entire block
py0103	PDX Overload State	ABy 24	na	
py0104	PDX Zero Drift State	ABy 24	na	
py0105	PDX Error Status	ABy 24	na	
py0106	PDX Cell Temperature	Aby 24	na	There is one entry each for up to 24 POWERCELL cells, containing the temperature for the referenced cell expressed as a signed 32 bit floating point value, in degrees C

2.11 PDX Setup Table (PX)

px0101	CAN Bus PDX Cell SYNC rate. This update rate is configured automatically, depending on the number of PDX cells in the system. Up to 4 cells = 100 Hz 5 to 8 cells = 50 Hz 9 to 14 cells = 25 Hz 15 or more cells = 15 Hz Multi-scale systems run at 15 Hz regardless of the number of cells included.

2.12 PDX Diagnostic Monitor (MX)

mx--07	Generate alarm on enclosure break	Bl	na	1 = Generate operator alarm when enclosure break occurs
mx--15	Gas Concentration Check Interval	L	na	Time interval at which to dynamically check the gas concentration in the PDX POWERCELLs. Check only occurs when there is no motion on the scale. Time is expressed in one hour intervals

2.13 PDX Cell Identification (DX)

dx0101 – dx0124	IDENTITY Number Cell 1 to IDENTITY Number Cell 24	S20	na	Cell Identity number read from 1st to 24th cell, respectively. Characters 1 – 4 are the product code; characters 5 – 12 are digits 3 – 10 of the METTLER TOLEDO serial number; character 13 is Null
dx0130	Number of PDX load cells	By	na	Number of POWERCELL load cells

2.14 PDX Monitoring Process (PM)

pm0102	Current Zero Counts -- Cells 1-24	AL24	na

pm0117	Gas Concentration Level	AL25	na	There is one entry each for up to 24 POWERCELL cells. Each cell contains the Gas Concentration level measured at each cell as a %Seal value. %Seal is a percentage of helium in the load cell relative to a reference value set during manufacture. It is a twos-complement integer whole number containing the %Seal. The 25th entry is reserved Failures of the gas sensor result in extreme values well outside the normal expected range. A gas sensor membrane failure would result in % Seal of -365%. A disabled gas sensor would result in % Seal +600%. The terminal interprets %Seal as follows: 0 = -5%Seal < %Seal < 105%Seal: load cell gas detection is working. Breach detection is determined by rate of change of gas concentration 1 = %Seal > 105% -- gas sensor failure; gas level detection not working 2 = %Seal < -5% -- gas sensor failure; gas level detection not working
pm0120	Enclosure Break Detected	ABy24	na	State of Enclosure Break Detection; entries for up to 24 load cells 0 = Rate of change of gas leak is less than 10% per week. Enclosure break is not detected 1 = Rate of change of gas leak is greater than 10% per week. Enclosure break is detected
pm0142	PDX Digital LC Voltage	AL25	na	One entry for each of up to 24 PDX cells. Contains the measured power supply voltage for Digital LC electronics for each cell, in millivolts as twos-complement integer. The 25th entry is reserved.

3 Application Data

3.1 Application Message Table (AW)

aw--00	Composite aw block	Struct	na
aw— 01-99	String Setup Fields 1 -99	S101	

3.2 Classification Data

3.2.1 Classification Dynamic Data (CL)

Attributes

Access	All users
Class Code	cl
Data Type	PP
Instances	1

cl0100	Composite cl block	Struct	na	Composite of entire block
cl0101	Current class	D	rt	Class in which the current weight is unknown = -1 (Below Zero, Overload, Not Started, etc.)
cl0102	Class unit	By	na	Unit used for all classes.
cl0103	High Limit	D	na	High Limit of highest class
cl0104	Class 1 Limit	D	na	Lower Limit Class 1
cl0105	Class 2 Limit	D	na	Lower Limit Class 2
cl0106	Class 3 Limit	D	na	Lower Limit Class 3
cl0107	Class 4 Limit	D	na	Lower Limit Class 4
cl0108	Class 5 Limit	D	na	Lower Limit Class 5
cl0109	Class 6 Limit	D	na	Lower Limit Class 6
cl0110	Class 7 Limit	D	na	Lower Limit Class 7
cl0111	Class 8 Limit	D	na	Lower Limit Class 8

3.3 Counting Application Data

3.3.1 Counting Data (CD)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All users
Class Code	cd
Data Type	PP
Instances	1

cd0100	Composite cd block	Struct	na	Composite of entire block
cd0101	Count	D	rt	Current number of parts on the selected scale
cd0102	Uncertainty	D	na	Typical repeatability error of selected scale (weight)
cd0103	Accuracy	D	rt	Typical achievable accuracy (%) of counting with current reference build

cd0104	APW	D	rt	Average Piece Weight: Read: APW of current reference build Write: Overwrite current APW and start counting with this value
cd0105	MinAPW	D	na	Minimum APPW based on uncertainty of selected scale
cd0106	PTT	D	na	Process Tolerance Threshold, based on uncertainty of selected scale and process tolerance value (PTV)
cd0107	Build Reference Status	By	rt	Build reference result status 0 = successful 1 = in progress 2 = below MinRefWt 3 = below MinAPW 4 = below configured Process Tolerance 5 = cannot build Reference Weight 6 = Reference Weight is Zero 7 = Cancelled from below Process Tolerance 9 9 = Counting not active
cd0108	Counting Active	Bl	rt	Read: 0 = Inactive, 1 = Active Write: 0 = Clear all parameters and stop counting. 1 = Build reference with number of pieces stored in cf0103 (VarRef) and start counting
cd0109	Current Number of Reference Pieces	D	rt	Number of reference pieces used to build current APW. Read: -1 = Unknown (counting not started, material without number of reference pieces, etc.) Write: Overwrite number of reference pieces

Method

The Engine updates the Dynamic Weight Shared Data at every weight update, whenever the weight changes. Typically, this occurs up to 20 times per second, but can vary depending on the load cell type and the application-type setting in cs--21.

The Engine converts the weight from the raw filtered counts it receives from the scale boards to the Legal-For-Trade weight. The Engine signals the Weight Display and SmartTrac Visualization task or an Application Task indicating that new weight is ready. The Engine sets this signal whenever weight changes, up to a maximum rate of 10 times per second. If the weight does not change for an extended time, the Engine will set the trigger just to refresh the weight display.

The Engine periodically re-writes the Shared Data weight fields every few seconds even when there is no change to the weight data.

3.3.2 Counting Configuration Data (CF)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All useers
Class Code	cf
Data Type	PP
Instances	1

cf0100	Composite cf block	Struct	na	Composite of entire block

cf0102	Fixed # of Reference pieces	US	rt	Number of Reference Pieces (1-9999). Read: # of Ref. Pieces from configuration Write: Write into configuration field and RefFix key; this is also possible even if the ″lock # of ref pieces″ toggle is active.
cf0103	Variable # of Reference pieces	US	rt	Number of Reference Pieces (1-9999) Read: # of Reference Pieces used at last APW build Write: Build APW based on that number Initial value from fixed numbers
cf0104	Lock # of reference pieces	BI	na	Enable/disable changing RefFix key. Read: 0 = OFF, 1 = ON Write: 1 = OFF -- long touch of the RefFix key disabled; no change possible 0 = ON – long touch of RefFix key enabled. Context menu offers available values
cf0105	MinRefWt	BI	na	Monitoring of minimum reference weight based on the Process Tolerance Value. Read: 0 = OFF, 1 = ON Write: 0 = Default PTV to 20% 1 = Apply entered PTV and enforce using MinRefWt calculation based on PTV.
cf0106	Process Tolerance Value	D	na	PTV: 0.1 – 30.0%

4 Complex Target Data

4.1 Full Target Commands (SC)

Attributes

IMPORTANT: Currently, only instance 1 is suppported, a general target which applies to all scales.

Access	Access level is not customizable
Class Code	sc
Data Type	PP
Instances	Instance 1- 4 = Targets applied to scales 1 – 4 Instance 5 = Target applied to Sum Scale Instances 6-17 = Targets applied to flow meters (future implementation) Instances 18-22 = Reserved

sc--01	Restart Target	D	rt	Set from 0 to 1 to trigger a start
sc--03	Apply New Target	D	rt	
sc--06	Pause Target	D	na	Set from 0 to 1 to trigger command. Operates only if target is running. Command puts target in pause state and turns off feed status

4.2 Full Target Process Data (SP)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Supervisor
Class Code	sp
Data Type	PP
Instances	22 Instance 1- 4 = Targets applied to scales 1 – 4 Instance 5 = Target applied to Sum Scale Instances 6-17 = Targets applied to flow meters (future implementation) Instances 18-22 = Reserved

sp--05	Target Value	D	rt	For weight and jog Target targets, this field has a weight value. For rate Targets, this field contains the max value that can trigger a rate alarm. For Piece Count Targets, this field contains number of pieces. For LearnJog Targets, this field contains a time value. For a Dump to Empty Target, this field contains the dump-completion-trigger weight.
sp--09	Spill Weight Value	D	rt	
sp--10	Fill Weight Value	D	rt	
sp--11	Upper Tolerance Value	D	rt	The Target uses this field to determine if the actual cutoff weight falls within this specified upper tolerance. This is the last OK weight when transitioning from ″in tolerance″ to ″over tolerance″. Value is in absolute weight or deviation from target depending on sp--13.

sp--12	Lower Tolelrance Value	D	rt	The Target uses this field to determine if the actual cutoff weight falls within this specified lower tolerance. This is the Engine OK weight when transitioning from "under tolerance" to "in tolerance". Value is in absolute weight or deviation from target depending on sp--13.
sp--13	Tolerance Operation	By	na	0 = Weight deviation from target 1 = Absolute weight value 2 = % deviation from target
sp--14	Upper Tolerance Percent	D	na	If sp--13 = 2, the Target uses this field to calculate the upper tolerance value as a percent of the value.
sp--15	Lower Tolerance Percent	D	na	If sp--13 = 2, the Target uses this field to calculate the lower tolerance value as a percent of the value.
sp--20	Target Weight Units	By	na	0 = primary units 1 = secondary units
sp--64	5-zone Over/Under High Tolerance Weight	D	rt	High tolerance value for 5-zone over/under operation
sp--65	5-zone Over/Under Low Tolerance Weight	D	rt	Low tolerance value for 5-zone over/under operation
sp--66	5-zone Over/Under High Tolerancee Percentage	D	rt	High tolerance percent for 5-zone over/under operation
sp--67	5-zone Over/Under Low Tolelrance Percentage	D	rt	High tolerance percent for 5-zone over/under

Method

In its simplest form, a Target is a comparator having two numeric data inputs and one binary output. One of the two numeric data inputs is a Coincidence (or Target) Value, which an outside agency may update at any time. The other numeric data input is an available data stream from a device channel. The data stream choices include gross weight, net weight, piece count, and rate of flow. The Target also provides a direction specification of either Fill, Dose or Dump. A simple Target output truth table is as follows:

Inputs		**Output**
Enable = FALSE		FALSE
Enable = TRUE Direction - WEIGH IN	lDataStreaml > = lTarge**tl**	FALSE
	lDataStreaml > = **lT**argetl	TRUE

Enable = TRUE
Direction = WEIGH OUT
	lDataStreaml > = lTarge**tl**	FALSE

	lDataStreaml > = **lT**argetl	TRUE

You may associate the logical output of a Target with a physical Discrete Output or may use it as an internal status. Typically, you select this during IND700 configuration.

An application can set up and run feeds using a Target Instance and can monitor for its completion using the Target Commands and Statuses. The application must Engine setup a Target Instance to use it. At a minimum, it must setup the Assigned Device, the Target Data Stream Type, the Coincidence Value, and the Target Action within the Target Instance. When it is ready to begin feeding, the Engine turns on the Target in Progress status, st--12 = 1. When the feed is complete, the ENGINE turns off the Target in Progress bit. The Engine maintains the Target status in the ST block.

Note: The Weights and Measures seal does not protect the Target configuration data.

4.3 Full Target Statuses (ST)

Attributes

IMPORTANT: Currently, only instance 1 is suppported, a general target which applies to all scales.

Access	Read only
Class Code	St
Data Type	D
Instances	1

st0105	Below Low Tolerance Weight	Bl	rt	0 = Over Low Tolerance Weight 1 = Under Low Tolerance Weight
st0106	Above High Tolerance Weight	Bl	rt	0 = Under High Tolerance Weight 1 = Over High Tolerance Weight
st0107	In Tolerance	Bl	rt	0 = Out of Tolerance 1 = In Tolerance
st0122	Below Low Tolerance Weight	Bl	rt	0 = Above Low Tolerance Weight 1 = Below Low Tolerance Weight
st0123	Above High Tolerance Weight	Bl	rt	0 = Below High Tolerance Weight 1 = Above High Tolerance Weight

5 Simple Target Data

5.1 Simple Target Setup (DS)

Attributes

ds0111	Target Comparison Mode	By	na	0 = None, 1 = Material Transfer, 2 = Over/Under, 3 = Classification

5.2 Simple Target Process Data (SD)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	Supervisor
Class Code	sd
Data Type	
Instances	

sd--01	Name Descriptor	S21	na	Text name describing target
sd--02	Target is Active	By	na	Application sets: 1 = Target is active 0 = Target is disabled
sd--03	Shared Data source field	S7	na	Points to a shared data source field to be compared to the target
sd--04	Mode	By	na	
sd--05	Target Value	D	na	
sd--08	Target Comparison Operator	By	na	1: <, 2: <=, 3: =, 4: <>, 5: >, 6: >=
sd--09	Upper Weight Range Value	D	na	Used as a second target value, if target is a range
sd--10	Upper Weight Comparison Operator	By	na	1: <, 2: <=, 3: =, 4: <>, 5: >, 6: >=
sd--30	CP Source for Comparator	By	na	CP uses this field to determine SD field that is the source comparator: 0 = none, 1 = displayed, 2 = gross, 3 = rate, 4 = application

6 Discrete I/O Data

6.1 Discrete I/O Status (DI)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All users
Class Code	di
Data Type	D
Instances	7 Instance 0 = I/O on main board (2 I/2 O) Instance 00 = Main Board (IND700) Instance 1- 4 = I/O on scale interface in slots 1 to 4 (2 I/2 O for each) Instances 5 - 6 = I/O on optional DIO interface boards (4 I/6 O for each

di--00	Composite di block	Struct	na	Composite of entire block
di--01	Input Status 1	BI	rt	0 = Off, 1 = On
di--02	Input Status 2	BI	rt	0 = Off, 1 = On
di--03	Input Status 3	BI	rt	0 = Off, 1 = On
di--04	Input Status 4	BI	rt	0 = Off, 1 = On
di--05	Output Status 1	BI	rt	0 = Off, 1 = On
di--06	Output Status 2	BI	rt	0 = Off, 1 = On
di--07	Output Status 3	BI	rt	0 = Off, 1 = On
di--08	Output Status 4	BI	rt	0 = Off, 1 = On
di--09	Output Status 5	BI	rt	0 = Off, 1 = On
di--10	Output Status 6	BI	rt	0 = Off, 1 = On

6.2 Discrete Input Edges (DE)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All users
Class Code	De
Data Type	D
Instances	Instance 0 = I/O on main board (2 I/2 O) Instance 1- 4 = I/O on scale interface in slots 1 to 4 (2 I for each) Instances 5 - 6 = I/O on optional DIO interface boards (4 I for each)

de--00	Composite de block	Struct	na	Composite of entire block
de--01	Rising Input Edge 1	BI	rc	1 = Transition from 0 to 1 detected
de--02	Rising Input Edge 2	BI	rc	1 = Transition from 0 to 1 detected
de--03	Rising Input Edge 3	BI	rc	1 = Transition from 0 to 1 detected
de--04	Rising Input Edge 4	BI	rc	1 = Transition from 0 to 1 detected
de--05	Falling Input Edge 1	BI	rc	1 = Transition from 0 to 1 detected
de--06	Falling Input Edge 2	BI	rc	1 = Transition from 0 to 1 detected
de--07	Falling Input Edge 3	BI	rc	1 = Transition from 0 to 1 detected

de--08	Falling Input Edge 4	BI	rc	1 = Transition from 0 to 1 detected

6.3 Remote Discrete Input Edges (RE)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All users
Class Code	re
Data Type	D
Instances	8
	Instances 1-8 = I/O on remote I/O modules (e.g. ARM100 with 4 inputs for each module

re--00	Composite wt block	Struct	na	Composite of entire block
re--01-04	Rising Input Edge 1-4	BI	rc	1 = Transition from 0 to 1 detected
re--05-08	Falling Input Edge 1-4	BI	rc	1 = Transition from 0 to 1 detected

6.4 Remote Discrete I/O Status (RI)

Attributes

Note: The last two digits of each shared data variable is its attribute.

Access	All users
Class Code	ri
Data Type	D
Instances	8
	Instances 1-8 = I/O on remote I/O modules, e.g. ARM100 with 4 inputs for each module

ri--00	Composite ri block	Struct	na	Composite of entire block
ri--01-04	Input status 1-4	BI	rt	0 = Off, 1 = On
ri--05-10	Output status 1-6	BI	rt	0 = Off, 1 = On
ri--21	ARM100 Remote Unit Status	By	rt	

7 Other Data

7.1 Users and Security Data

7.1.1 Logged-In Users (XL)

xl--00	Composite xl block	Struct	na	
xl--01	Logged-on user name	S13	na	User name of logged-in user on SD server. 20 instances: 20 users can log in and communicate with terminal simultaneously across all available interfaces
xl--02	Access privilege of logged-on user	By	na	1 = Operator 2 = Supervisor 3 = Service 4 = Administrator

7.1.2 Keyboard Routing Commands (KC)

kc0137	Send Key Code to CP	By	rt	Send key code to control panel: 1 = Switch scale, 2 = Zero, 3 = Tare, 4 = Transfer, 8 = Clear

7.2 ID Sequence Data

7.2.1 Prompt Setup (PR)

pr--31-40	Prompt strings 1-10	S41	na	Title (prompt) of IDs 1-10 in the ID Form

7.2.2 Prompt Response (PA)

pa--0101-10	Response strings 1-10	S51	na	Data content for IDs 1-10 in the ID Form

7.3 Application Dynamic Variables

7.3.1 Data Connections Setup (DC)

Attributes

dc--01	Output Connection Type	By	na	Instance = position in the list of connections in setup. 0 = none, 1 = transfer and custom print, 2 = continuous output, 3 = multi-continuous output 1, 4 = multi-continuous output 2, 5 = reports, 6 = totals reports, 11 = extended continuous print
dc--04	Output Trigger	By	na	Instance = position in the list of connections in setup. 0 = none, 1 – 5 = Scales 1 -5, 6 – 25 = Custom print 1 – 20, 26 – 37 = Flow Meter 1 - 12

7.3.2 Serial Port Setup (RP)

Attributes

rp--01	Interface Type	By	na	0 = RS232, 1 = RS422, 2 = RS485
rp--09	Option Board Slot Number	By	na	0 = none, 1 – 6 = Slot Number

7.3.3 Custom Print Commands and Statuses (CP)

Attributes

cp0101 - 0110	Custom Print 1-10	Bl	na	From 0 to 1 = command to cp01nn: start custom print

cp0111 - 0120	Custom Print 1-10 Status	By	rt	Command completion statuses cp0120: 0 = Success, 1 – 255 = Specific error code
cp0121 - 0130	Custom Print 11-20	Bl	rc	From 0 to 1 = command to cp01nn: start custom print
cp0131 - 0140	Custom Print 11-20 Status	By	rt	Command completion statuses cp0120 0 = success, 1 – 255 = Specific error code

7.3.4 System State (XD)

Attributes

xd0103	Current Date	S12	na	Format defined in xs0110
xd0104	Time of Day	S12	na	Format defined in xs0111
xd0105	Week Day	S11	na	
xd0107	Second ticks	UL	rt	
xd0108	Number of scales	By	na	Engine initializes these four fields on power-up, based on the hardware configuration it detects
xd0109	Number of flow meters	By	na	
xd0110	Number of discrete inputs	By	na	
xd0111	Number of discrete outputs	By	na	
xd0119	Multi-Continuous Print Stream			
xd0120	Selected Standard Continuous Out			
xd0153	Current System Message Display			
xd0151	iButton EEPROM Read Image	ABy48	na	Required to verify licensing
xd0171	Logged in Client User	S13	na	
xd0172	Client User Access Lvel	By	rt	0 = None 1 = Operator 2 = Supervisor 3 = Service 4 = Administrator

7.3.5 System Setup (XS)

Attributes

xs0103	Software ID	S21	na	Textual description of the installed software
xs0104	Software Part Number	S14		Part numbers are 13 digits plus null terminator
xs0105	Terminal Serial #	S14		Serial numbers are 13 digits plus null terminator
xs0106	Terminal ID #1	S21		Terminal ID
xs0107	Terminal ID #2	S21		Project ID
xs0108	Terminal ID #3	S161		User textual description of the IND700

7.3.6 System Monitoring & Service Data (XP)

xp0101	Transaction Counter	UL	na	Transaction counter incremented according to the Transaction Counter Setup. FTP does not restore this field.
xp0103	Terminal Accumulation Total	D	na	
xp0104	Terminal Accumulation Sub-Total	D	na	

xp0105	Terminal Transaction Total	UL		na
xp0106	Terminal Transaction Sub-Total	UL		na
xp0110	Last Print Message	S1001		na
xp0111	Last Error message	S81		rt
xp0130	Last Alibi Record ID Number	UL		na

7.3.7 System Feature Triggers & Controls (QC)

qc0117	PDX Proprietary Services Lock		
qc0131-0135	Select Scale 1 – 5		
qc0161	Reboot terminal	Bl	rc
qc0178	Activate Remote Viewer		
qc0199	Activate PDX Load Cell Filters		

7.3.8 System Feature Triggers & Controls (XC)

xc0106	Disable keypad	Bl	rt	All keys on keypad and touchscreen off
xc0139	Reprint last demand print	Bl	rc	Applications use this trigger to issue a duplicate print request

7.3.9 Network Node Status (NS)

ns0124	PLC Online	By	rt	0 = Offline 1 = Online
ns0125	FTP/sFTP Currently Active	By	rt	0 = OFF 1 = ON

7.3.10 Board Identifications (BD)

Attributes

bd--05	Board Type

METTLER TOLEDO Service

To protect your product's future:

Congratulations on choosing the quality and precision of METTLER TOLEDO. Proper use according to these instructions and regular calibration and maintenance by our factory-trained service team ensure dependable and accurate operation, protecting your investment. Contact us about a service agreement tailored to your needs and budget.

We invite you to register your product at www.mt.com/productregistration so we can contact you about enhancements, updates and important notifications concerning your product.

www.mt.com/IND700

For more information

Mettler-ToledoGmbH
Im Langacher 44
8606 Greifensee, SWITZERLAND

30753890 Rev. B, 12/2025

30753890