QCOM Protocol

Version 1.6.7j – Draft for comment

Table of Contents - Brief

Table of Contents - Full

1 Introduction

This document defines a communications protocol specification for Electronic Gaming Machines (EGMs) known as ~~the QCOM Protocol (also referred to as simply~~ "QCOM"~~).or the QCOM Protocol~~.

Policy:

All EGMs installed in licensed venues in the state of Queensland must be connected to an Electronic EGM monitoring system. The OLGR requires all new EGMs to communicate using the same protocol with respect to the Basic Monitoring Service.

Please refer to the revision history for any policy in regards to the incept date of the requirements in this version of the document.

Purpose:

The purpose of this document is to provide a precisely defined standard of communication for an EGM, requiring low CPU processing requirements while maximising functionality.

Scope:

This document is applicable to EGM manufacturers, EGM testing laboratories, EGM monitoring system developers and developers of Value Added Services such as player loyalty and jackpot systems. This document does not completely specify the physical interface or the Site Controller QCOM operating procedures as they are a separate standard. Refer section 1.1 for more information.

Prerequisites:

This document assumes the reader has a thorough knowledge of EGMs and their operation as well as a Software IT/Computer systems engineering background with previous experience in communications protocols.

Typical scenario for QCOM shown below. (Note, QCOM is a LAN Protocol and the LAN type shown is an example only)
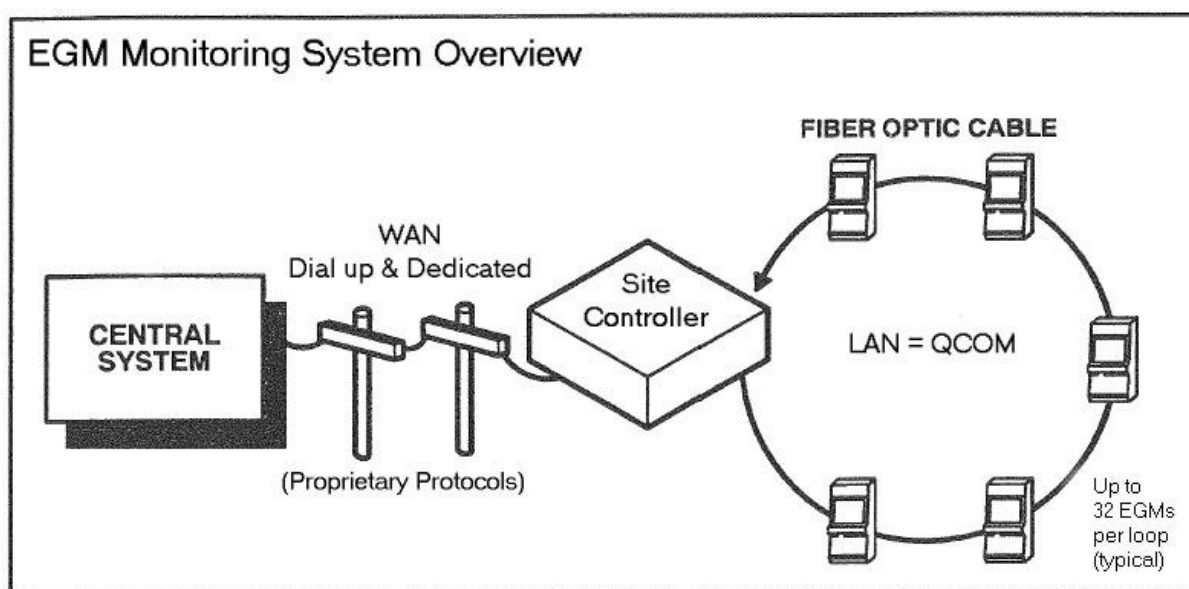
QCOM Protocol General Specifications:

- QCOM is a Serial based Protocol designed for low bandwidth LANs.
- Typically no more then 32 EGMs per LAN is recommended. Max ~ 250
- QCOM assumes a "Trusted" single (primary) master Site Controller. (EGMs will not enable without a SC.)
- Half Duplex - two way communication.
- Multi-point mode or point-to-point mode
- Asynchronous 19200 baud, 1 start bit 8 data bits, 1 wake-up bit & stop bit.
  - (0.5729 msec per character @ 19200 baud)
- Character Orientated Poll-Response Protocol.
- 1 second poll cycle time, typical. 250 msec minimum (*QPv1.6*).
- 9 continuous character times for the EGM Response Timeout period. (~5 milliseconds @19200 baud)
- 1 character time inter-character frame time-out in EGM responses.
- Byte Aligned.
- Little Endian. (i.e. Least Significant Bit/Byte first.)
- 1 cent base meter denomination.

QCOM Protocol Features:

- Remote auditing of EGMs.
- Remote EGM Monitoring and Control (an EGM cannot be enabled without being connected to a SC)
- Dynamic EGM Poll Address Configuration (~128 poll addresses available per LAN) (15.5.4)
- EGMs can be found on the LAN without knowing their serial number (refer 21) *(QPv1.6)*
- Cashless EGMs via Electronic Credit Transfer (Refer section 16)
- EGM Program Hash Verification (Refer section 11)
- Linked progressive Games (up to 65,533 groups, with up to 8 levels per group) (Refer Section 10)
- Promotional/Advisory Messages, two EGM specific (one with option of increased prominence - *QPv1.6)* (refer 15.4.20) and one global (refer 15.5.3)
- Multi-game support for up to 65535 games per EGM (but no more than 255 games may be offered for play at any time. (Refer Section 9).
- Multi-variation EGM support for up to 99 variations per game (no more than 16 may be hot switchable at a time *QPv1.6*). (Refer Section 9)
- Bet type meter monitoring (15.6.5) *(QPv1.6)*
- Player Choice meter monitoring (15.6.7) *(QPv1.6)*
- Note Acceptor version and status monitoring (15.6.2) *(QPv1.6)*
- Switchable on-screen player clock display *(QPv1.6)*
- PID & PID version control (refer 15.4.5) *(QPv1.6)*
- Each EGM can be configured to display externally triggered jackpot current amounts (refer 15.4.7) *(QPv1.6)*. *Display may be either in $.x or points.*
- Hot switching of game variations. (Refer Section 9)
- Stand Alone Progressives (up to 8 levels per game) (Refer to section 10)
- Stand Alone Progressive Level Contribution Local Area Backup (via SC) (Refer to section 10)
- Customisable Stand Alone Progressive support (Refer to section 15.4.6 CustomSAP field). *(QPv1.6)*
- Support for awarding system triggered jackpots via System Lockup Request Poll (15.4.9)
- EGM Tower Light control (15.4.21)
- Configurable Random Events (CRanE) (Refer to section 16.5.1.1). Up to 8 events with configurable probability per EGM.
- Cash Ticket In/Out support (TITO). (Refer Section 22) *(QPv1.6)*
- EGM Denomination Monitoring and Control (refer 3.1.3)
  - Tokenised EGMs
  - Fractional Denomination *(i.e. game denomination less than one cent) (QPv1.6)*
  - Denomination hot switching *(QPv1.6)*
- QCOM Command Prompt (Refer to section 20) *(QPv1.6)*

- EGM to EGM Communications (Refer to section 19) *(QPv1.6)*
- QCOM and RUGMs (Refer to section 18) *(QPv1.6)*

QCOM Protocol EGM Hardware Requirements:

- A Serial Communications port
    - A UART with parity address bit capability (wake-up parity bit) is ~~strongly~~ recommended to reduce processor interrupt load.
    - A UART with a FIFO buffer is also ~~strongly~~ recommended to reduce processor interrupt load.
    - A UART with specific and global address byte capability may also be considered.
    - A UART with automatic CRC generation may also be considered.
- CPU requirements depend on which UART is used. A UART with all the above features would allow the protocol to be implemented using even the slowest CPU.
- An NV RTC is recommended.
- A video display driver, firmware or hardware capable of overlaying transparent text boxes over game graphics with negligible CPU effort is recommended. Examples: refer Figure 9 as well as figures 1,3,4,5,6,9 & 10. The ability for the EGM to display overlayed transparent text boxes will become a mandatory requirement in the next major revision of the QCOM protocol.
- A minimum of a 4.5Kv ESD protected UTP / Ethernet / TCP/IP capable port is mandatory on all new EGM models submitted from 2007. [CP:1] However this port must not be enabled until further notice from the OLGR. Refer section 18.

QCOM EGM Software Requirements:

- GMNS compliance
- RCR feature implemented as per GMNS (for EGMs supporting hoppers) [CP:2]
- Floating point number support.
- NVRAM required for QCOM (estimate):
    - ~4k bytes (event queue + misc) + ~ 32..64 bytes per game variation.

A PC compatible based protocol simulator is available from OLGR subject to licensing arrangements. A QCOM Protocol Software Development Kit is also available.

1.1   Other Related Documents

The following documents are available for download from ~~http://www.OLGR.qld.gov.au/~~the Queensland Office of Liquor and Gaming Regulation industry portal website.

QCOM Protocol Site Controller Operating Procedures

> This is an important companion document to QCOM. This document lists all the SC requirements on how to communicate to an EGM via QCOM. This document is highly applicable to EGM monitoring system developers and developers of Value Added Services such as player loyalty and jackpot systems.

OLGR Local Area Network EGM Interface Specification

> This document defines the Local Area Network EGM Interface Specification that QCOM uses in QLD. EGM manufacturers and monitoring system operators and developers must obtain a copy of the above document

~~OLGR EGM Program~~ Hashing Algorithms

> EGM manufacturers must obtain a copy of the above document in order to implement program hashes. It contains a list of acceptable algorithms and also details what data must be included in the overall program hash result.

ANZ National Standards for Gaming Machines

Both EGM manufacturers and monitoring system providers must obtain a copy of these this documents.

Electronic Seal Minimum Requirements

The above is required by EGM manufacturers when wishing to implement remote software upgrade support.

Data Requirements for Monitored EGMs

This document specifies the electronic data transfer formats between the OLGR and QLD based LMOs. EGM monitoring system developers for the QLD market must obtain a copy of this document.

Gaming Venue Electronic Meter Access
OLGR Monitoring System Minimum Requirements Document

EGM Monitoring System developers for the QLD market must obtain a copy of these documents.

1.2   Optional Sections of QCOM.

Unless specified otherwise throughout this document, implementation of all sections of QCOM by an EGM manufacturer is mandatory. (Search with keywords: "optional" or "mandatory") An exception to this rule is that if an EGM does not have support for a particular piece of hardware, then the corresponding QCOM support for sections pertaining to that hardware are not mandatory. E.g. if the EGM does not have a note acceptor, then applicable sections of polls and responses pertaining to note acceptors do not have to be implemented.

A summary of mandatory and optional sections will be added to this section at a later stage (todo).

Also refer to Appendices for any specific jurisdictional requirements.

1.3   Notes on implementing QCOM for EGM Manufacturers

QCOM evaluations are extensive and anything that can be done to reduce the time spent testing is to the EGM manufacturer's advantage, for example; reusing existing code and utilising the QCOM SDK. Try to make the EGM QCOM implementation driver as generic as possible for future portability.

When developing QCOM from scratch, begin from the low level and work up. The recommended order of implementation of QCOM sections is the same as the recommended order of a QCOM evaluation. Refer to the QCOM checklist introduction for the order. As the highest priority the EGM must correctly identify its protocol version in the DLL (see 14.3.1). This is to ensure the protocol simulator knows what protocol version EGM it is communicating with.

It is recommended to start the implementation by firstly trying to decode messages from the protocol simulator within the EGM and displaying the results on the EGM. Once working, try implementing the General Status Response (15.6.1) with constant data in it and a hard-coded poll address. An issue to note here is that QCOM is not completely a typical poll-response protocol, as the EGM must delay all requested data for one response before sending. This is to lower CPU requirements by spreading the message processing load over a longer period. Refer section 14.1 on how this is to be handled.

Thoroughly test the implementation at this stage before continuing. E.g.:

- Extend the response to be much larger than it should be to ensure no problems later when it comes time to implement the larger responses.

- Extend polls and broadcast messages to their maximum length to ensure no side effects in processing large messages.
- Look for side effects when the poll cycle time is slowly decreased to as fast as it will go. Watch for corruptions or errors. Worst case, the occasional missed response is ok. In fact, during development it is recommended to keep the poll cycle running at least 200msecs the whole time, only slowing it down for debugging.

Utilising the QCOM SDK sample event code implementation is also recommended and can save much time in implementation of the QCOM event queues.

A special mention is made regarding QCOM's Response Timeout Period (14.1.7). Please refer to this section for more information.

1.4 ~~QCOM~~ Document Navigation

This document is designed to be read within MS Word™ or ~~a~~ compatible MS Word™ document reader. MS Word format versions of this document are available on request for QCOM Licensees. The MS Word version enables all in-document hyper-links for easier document navigation. For example:

- All references in the document are hot-links; i.e. Control-click on a reference to jump to that location.
- Endnotes (which define the EGM QCOM Protocol checklist) may be jumped to/from by double-clicking on the endnote number. Hovering the mouse pointer over an endnote number in the document body should reveal the endnote contents in a pop-up after a second or so.
- A number of Bookmarks have also been defined for quick navigation to major sections in the document.

2 Definitions

2.1 List of Abbreviations and Acronyms

^	"to the power of" (mathematical exponential operator)
ACIA	Asynchronous Communications Interface Adaptor
BCD	Binary Coded Decimal
CC	Cancel Credit
CDC	Communications Disabling Condition (Refer 6)
CEO	~~Chief Executive Officer -~~ refers to a delegated authority within the OLGR
CFF	Common Field Format. Refer to section 15.1
CRC	Cyclic Redundancy Check
CRanE	Configurable Random Event. Refer to section 16.5.1.1.
CMCS	Central Monitoring and Control System hardware and software
CMS	As above
CPU	Central Processing Unit
DS	Digital Signature
ECT	Electronic Credit Transfer
EGM	Electronic Gaming Machine
ESD	Electro-Static Discharge
GMNS	Gaming Machine National Standards Document for EGMs (1.1)
HS	Hot Switching
ID	Identification
IIS	Implement In Simulator (Internal Use Only)
LAN	Local Area Network
LP	Linked Progressive
LPCPA	Linked Progressive Current Prize Amount (See Glossary)
LS	Least Significant

LSB   Least Significant Byte
LSD   Least Significant Digit
NA   Not Applicable
NV   Non-Volatile
PID   Player Information Display (Refer GMNSNS for more information)
PHA   Program Hash Algorithm
PSD   Program Storage Device
QCOM   Queensland Local Area EGM Communications Protocol
OLGR   Office of Liquor and Gaming Regulation (Queensland)
QPvx.x QCOM Protocol Version X.X (Refer 14.3.1 & document Revision History)
QSIM   QCOM Protocol Simulator Software
RAM   Random Access Memory
RCR   Residual Credit Removal
RCRF   Residual Credit Removal Feature
RTP   Return To Player (see glossary for more information)
RSA   Rivest-Shamir-Adelman encryption
RTC   Real Time Clock.
RUGM   Remotely Upgradeable Gaming Machine
SAP   Stand Alone Progressive
SC   Site Controller (the master device talking QCOM). Refer QLD Gaming Machine Act 1991.
SCC   Serial Communications Controller IC
SDK   Software Development Kit
TBA   To Be Announced.
TITO   Cash Ticket In/Out
UART   Universal Asynchronous Receiver Transmitter
VDU   Visual Display Unit of an EGM
XOR   Exclusive OR

If an abbreviation used is not defined above, then refer section on Common Field Formats (15.1)

2.2   Glossary

Audit Mode

> Refer to the latest version of the Gaming Machine National Standards for EGMs (1.1).

"C"

> Refers to the C Programming Language

Cancel Credit

> Means a manual payment or 'hand pay'.

Cashless Mode

> Refer 16.2.

Critical Section

> Power or other disruptions can occur at any time on an EGM. In this document, a critical section refers to any area of EGM program code that if interrupted while being executed, wcould leave one or more areas of the EGM's NV memory corrupted and upon restart, the corruption be both undetected and uncorrectedor cause a function or calculation to return an incorrect result because the state has changed. Examples: updating any NV variable that takes more one CPU instruction to update; updating an event queue; running self audit formula, or processing poll data (14.1.22).

Door Open

Refers to the open state condition of either the main, console (cash box), processor, belly or note acceptor doors on an EGM. Also in this protocol the stacker removed condition is to be treated as if it was a door open condition (i.e. it is not a fault condition and a return of the stacker will clear the condition). Refer 7.10.3.12 for more information

Events

In this document, the term "event" & "event queue" and all similar forms, unless specified otherwise in terms of context, refer only to QCOM defined events and event queues. Refer to section 7 for more information.

Fault Condition

A fault condition means an event has occurred on the EGM where either a hardware component(s) of the machine can no longer perform the function(s) for which it was designed, or some action has taken place which the EGM considers to be illegal behaviour.

NB: Door opens and stacker removed conditions are not to be considered fault conditions in this document. CP:3

The EGM must remain in a fault condition lockup until it is manually keyed off by an attendant or via the General Reset Poll.

External Jackpot/s

This refers to any type of linked jackpot that can be won on the EGM that the EGM does not trigger as a part of any of its games. I.e. The jackpot is triggered by a device 'external' to the EGM. (If the EGM has not received the External Jackpot Information poll (15.4.7) then it would have no way of telling if it was on an external jackpot.)

Game

1. To play; refer "Play" definition below. 2. Refer GMNS

Group Meter

Refer to section 12.1

Idle Mode

Refer to section 3.3.1

Message Data

'Message Data' when stated and where its context is not specified, refers to the area of a broadcast, poll or response message, starting from the function code of the message, up to but not including the message CRC. (i.e. it excludes the Data Link Layer protocol items.)

Message Format and Order:

Any string following "Message Format and Order:" indicates the fields and order of the fields as they would appear in actual poll/response message packets. Each separate field is enclosed in < > brackets. [ ] brackets enclosing fields indicates optional fields that only appear depending on certain conditions and configurations. New lines in a message format string mean nothing.

Log /

Logging

In this document, the terms "log" and "logged" and all similar forms, when used in the context of "events" refer to the process of an EGM creating a new QCOM defined event in memory and pushing it onto the applicable QCOM event queue for subsequent transmission via an Event Response (15.6.9). Refer to section 7 for more information.

Linked Progressive Turnover Meter or 'LP Turnover Meter'

In this document the 'LP Turnover Meter' refers to the applicable turnover towards a jackpot (i.e. before the percentage contribution is taken out). An EGM must have one unique LP Turnover Meter per game with a LP component. In a single variation game this meter is equal to a games multi-game/variation turnover meter. In a multi-variation multi-game EGM, each LP Turnover Meter is equal to the sum of each game's multi-game/variation turnover meters. In a single game EGM, it is also equal to the EGM's total turnover meter except it excludes RCR turnover[*] and other games total turnover in a multi-game EGM. [CP:4]

The LP Turnover Meter is currently sent by the EGM in two response types; the Progressive Meters Response (15.6.3) and the Meter Group/Contribution Response (15.6.8) PAMT fields. Refer to section 10.6 for more information on linked progressives.

[*] RCRF Turnover is excluded from EGM triggered jackpots because this type of jackpot cannot be won from playing the RCRF. However for external jackpots, typically all EGM turnover including RCRF turnover, contributes to the jackpot and can trigger the jackpot.

Linked Progressive Jackpot Current Amount

This means the current value of a jackpot prize of a linked progressive jackpot level. This is the amount which is usually displayed to participating players. If the LP jackpot was won this is the amount that would be paid to the winning player.

Lockup Condition

Lockup conditions are similar to fault conditions in that they suspend play and credit acceptance/redemption until manual intervention and verification by an authorised device or person eg. an attendant. But lockup conditions do not indicate a problem has occurred like a fault condition does and they occur as a normal part of the EGMs operations. They indicate an authorisation is required, usually by an attendant. They do not interrupt other states of the EGM like a fault condition (but the System Lockup Request Poll (15.4.9) now has this option). Lockup conditions are; cancel credits, RCRF cancel credit, cashless transfers (from EGM), System Lockup (15.4.9), progressive awards and large wins. Lockup conditions are not fault conditions and different requirements apply. Only one lockup condition can be active at any instant on the EGM. A lockup condition may be interrupted by a fault condition, door open or audit/test modes.

Mask Out

If any bit fields are labelled 'mask out', then this means that when reading any other field from the hosting byte, the field/s labelled 'mask out' must not affect or be a part of the data extracted. For example if a byte field is defined as:

ByteAAA:
      Bit 0 FlgA
      Bit 1…7 reserved, mask out

To access the value FlgA, an operation equivalent to (ByteAAA & 0x01) must be performed (in "C" notation)

Meter

A fault tolerant, Non-Volatile protected and error-checked EGM software variable, which stores EGM accounting/auditing information. The term "meter" is a carry-over from the days of pulsed mechanical meters. With the exception of a meter roll-over at its maximum value, a meter generally only ever increments.

Play

A play refers to the sequence of events on an EGM initiated by a player through an irreversible wagering of credit and ends when the play show is over, the results of the play revealed to the player and either the wager has been lost or winnings become payable (i.e. winnings are able to be transferred to the EGMs total win meter). Free games, free spins, bonus features and double up/gamble, are considered to be a part of a single play. A play is considered finished upon entry into return to idle mode procedures (refer section 3.3.2).

Poll cycle

Refer section 14.1

Residual Credit Removal Feature (RCRF)

This feature activated at the end of a hopper collect, gives the player an option to gamble any remaining credit less than the value of one hopper coin/token for a chance to win a whole coin/token at the regulatory minimum RTP set for the jurisdiction. Refer to the latest version of the Gaming Machine National Standards for EGMs (1.1).

Response Timeout Period

Refer Section 14.1.7

Return To Player

Unless specified otherwise, RTP is represented as a percentage in this protocol (0%...100%). RTP refers to the theoretical return to player of a game or component of a game. I.e. an 85% game would mean on average for every dollar bet in the game the average prize (including prizes of zero) would be 85 cents.

Site

Means a gaming venue or licensed gaming premises.

System

The CMCS, including the SC.

Session

A "session" in the context of communications, refers to the period of communication between an EGM and SC, starting when the EGM resumes responding to a SC after poll address designation and ending upon the next EGM power down or EGM detected communications time-out (6.1.2) or Communications Fault(QPv1.6)(6.1.3). *(NB: in QPV1.5 EGMs, a Communications Fault did not end a session for an EGM).*

Test Mode

Refer to the latest version of the Gaming Machine National Standards for EGMs (1.1).

Turnover

Means a total or partial total of bets made on an EGM.

Variation

Refers to a specific percentage return to player of a game. Usually represented by a 2 digit ID number. Each game in an EGM has one or more variations. Each variation in a game usually has a different RTP. Specific variations are denoted by their variation ID number (VAR)

2.3 Document Conventions

2.3.1   Date format used in this document is DD/MM/YYYY.

2.3.2   All bytes in this protocol are least significant bit transmitted first (denoted bit 0).

2.3.3   All multi-byte fields are LSB first.

2.3.4   BCD fields in the protocol document are packed BCD, LSB and LSD first (with respect to bit 0) within the byte. eg. the BCD number 1234 would be transmitted as 0x34 first, then 0x12. When required to display BCD numbers they are simply displayed as decimal numbers.

2.3.5   '0x' preceding a number (or 'h' following), denotes a hexadecimal number (base 16), otherwise the number is decimal.

2.3.6   All value fields are unsigned numbers unless specifically stated as signed.

2.3.7   32 bit Floating point numbers are represented in the protocol by IEEE 754 floating point numbers. (The IEEE-754 floating-point format is used in ANSI C & C++ compilers) Eg. The number 987654321000.0 (9.87654e[11]) typically would be represented in hex as 0x5365F4C4.

2.3.8   In QCOM, ASCII printable characters are defined as characters in the ASCII range 0x20 (' ') to 0x7A ('z') inclusive and must be displayable by the EGM. [CP:5]

Specifically： !"#$%&'()*+,-./0…9:;<>?@A…Z[\]^_`a…z

(NB: does not include '{', '|', '}' and '~' characters (ASCII 0x7B…0x7E) but there is no issue if these characters are supported by the EGM)

The '^' character and Cash Ticket Printers – special case:

Many brands of ticket printers consider the '^' character (ASCII 0x5E) to be a control code and will not print this character (and possibly the whole string it is embedded in). Accordingly the EGM for any QCOM text field destined for a cash ticket must either (pick one): [CP:6]

- treat the '^' character as a non-printable character and behave as per the respective TITO poll mandates for non-printable characters, or
- before the EGM applies any is-printable sanity check, substitute any '^' characters with a space characters (ASCII 0x20), or
- before the EGM applies any is-printable sanity check, delete any '^' characters from the string

The above only applies to cash-ticket related text fields. Specifically these are the Site Details (15.5.5): STEXT & LTEXT fields and the Cash Ticket Out Request Acknowledgement Poll (15.4.10):CTEXT fields. The EGM must still support the '^' character (i.e. treat it as printable) for all other text string fields used in QCOM. [CP:7]

2.3.9   [] brackets indicate an optional item or field usually depending on other variables.

2.3.10   "/" when used in a formula denotes a division operation.

2.3.11   "x" or "*" when used in a formula denotes a multiplication operation.

2.3.12   Unless specifically stated otherwise or by inference, all requirements in this document primarily apply to EGM manufacturers.

3   General

3.1   Electronic Gaming Machine Configuration

3.1.1   EGM Configuration Sequence from RAM Clear

A typical EGM configuration after a RAM clear must follow the following sequence:

1. After a RAM clear the EGM must not respond to or process any polls or broadcasts. Also refer to section 8.1 for EGM Defaults upon an EGM RAM clear.

2. An EGM serial number must be manually entered in a function provided in either test or audit mode on the EGM (See 3.1.2 for requirements).

3. Any other required, one-time EGM setup parameters not covered by the protocol (e.g. button panel selection) must be requested and entered before the EGM proceeds any further. [CP:8]

4. Once configured with a serial number and other manually entered items, the EGM must no longer allow a Demonstration Mode (if applicable) [CP:9].

5. The EGM may now commence responding to the SC once it has been assigned an initial poll address. In order to obtain a poll address to respond on, the EGM must start processing Broadcast Messages for the Poll Address Configuration Broadcast Message (15.5.4).

6. Once responding, all remaining items required for configuration will then be sent from the SC to the EGM via the EGM Configuration Poll (15.4.2), EGM Game Configuration Poll/s (15.4.3) and Progressive Configuration Poll/s (15.4.6)which assigns the EGM with denomination, game variation, progressive information (if applicable) and other various parameters.

This completes EGM initialisation on the QCOM protocol. Also refer to the section on EGM defaults at RAM clear (section 8.1). Upon a power failure at any stage of the configuration process, normal state recovery rules apply (i.e. you should not have to start over and the EGM should return to the state (above) it was in prior the power fail). [CP:10]

3.1.2   EGM Serial Number Setup Function

To configure an EGM to operate using this protocol, the EGM's serial number (15.1.8) must be manually entered (or confirmed if the default value is already correct) [CP:11] in a function provided in either test or audit mode on the EGM. [CP:12]

The setup function must accept only valid serial numbers which are 6 digit (packed BCD) numbers. [CP:13]

Any value entered in the setup function must be able to be verified by the user before being permanently accepted and write protected by the EGM. The act of permanently saving the serial number must be unique and difficult to perform by accident [CP:14]

Once saved, the serial number must not be able to be changed unless the EGM is RAM cleared. [CP:15]

The default serial number initially offered in the setup function is zero, alternatively it is permissible to restore a previously saved serial number from EEPROM (or equivalent) as the initial default for confirmation or possible change.

(*Acceptance of the overall legibility of the serial number setup feature on the EGM is at the discretion of the CEO [CP:16].)

3.1.3 Denomination - (Configuration, Control & Hot Switching)

An EGM's coin/token denomination and credit denomination are set initially at EGM configuration via the EGM Configuration Poll (refer 15.4.2).

3.1.3.1 Denomination Hot Switching (QPv1.6)

(Denomination hot-switching was added in QPv1.6) EGMs may support Denomination Hot Switching at the discretion of the EGM manufacturer. If supported, the EGM will allow its credit denomination to be changed via the EGM Configuration Poll (15.4.2). Capability for Denomination Hot-Switching is indicated by the EGM via the EGM Configuration Response (15.6.12).

The following requirements apply to EGMs that support Denomination Hot Switching: [CP:17]

- The EGM must have an on-screen display of its current credit denomination. E.g. "1 credit = 1 cent". The display priority must be equivalent to the player clock display (refer Appendix A).
- If the EGM has progressive games, then the progressive game component parameters must not change on a denomination switch otherwise the EGM must not offer Denomination Hot Switching. Progressive parameters (such as SUP, CEILING, or HRATE) typically have to change for each denomination to maintain the RTP and will therefore progressive EGMs will generally not be able to have their denomination changed.
- The current reported list of variations for a game (15.6.11) must not change over a denomination hot-switch. i.e the reported list of variations must still all be valid for the new denomination otherwise the EGM must reject the new denomination.
- The credit denomination of each play must be clearly displayed in last play recall.
- The EGM must not offer hot-switching of denomination if any static artwork on the EGM needs to be manually updated.

If denomination hot-switching is supported by the EGM, the EGM may allow its credit denomination to be changed via the EGM Configuration Poll (15.4.2) when the following conditions are satisfied: [CP:18]

- Configuration via the EGM Configuration Poll (15.4.2) has been previously successfully completed on the EGM;
- The EGM is in idle mode; [CP:19]
- The EGM has zero credit;
- The EGM MEF = 0,
- An EGM Configuration Poll is received that contains a valid manufacturer ID, valid serial number and a new valid credit denomination value in the DEN field.

If the above conditions hold, then the EGM will accept the new credit denomination, log the 'EGM Denomination Enabled Event' (7.10.3.31) and update the EGM's display of all current meters in units of credit to allow correct betting to take place for the next play. [CP:20] Due to possible side effects that can occur from changing an EGM's denomination (e.g. number of playlines may change), then it is also acceptable for the EGM to set the current game display to its RAM clear default state (i.e. display a non-winning combination and zero the bet & win meters etc).

On EGMs that support denomination hot-switching, once configuration has been completed via the EGM Configuration Poll (15.4.2), the EGM must not process any other fields in the EGM Configuration Poll apart from the validating the MID & SER fields and processing the DEN field for a possible credit denomination change. [CP:21] i.e. it must not be possible (without an EGM RAM clear) to change any other field set exclusively by the EGM Configuration Poll.

3.1.3.2 Fractional Denomination Support

If this feature is allowed by the regulatory authority, manufacturers may supply games that support a credit denomination of less than 1 cent. E.g. 1 credit = 0.1 cent. If supported by the EGM for a game, a credit meter multiplier must be hard-coded into the game. From the protocol's point of view the EGM will still appear to the SC at whatever credit denomination was set by it, but the EGM will be displaying credit to the player multiplied by the hard-coded amount. All bets wagered and all non-progressive wins

in the game must still be a multiple of the EGM's credit denomination as set by the EGM Configuration Poll (15.4.2). Games with this feature must be identified as such by being allocated a unique GVN.

(If this feature proves popular there may be extended support this feature in future versions of QCOM.)

3.1.3.3 Player Selectable Denomination

'Player Selectable Denomination' refers to games where the player may choose to play the game at a denomination of their choosing. EGM manufacturers wishing to provide player selectable denomination games may do so with the approval of the regulatory authority. The EGM manufacturer **must** contact the regulatory authority before implementing player selectable denomination games for any special requirements.

There are two possible ways QCOM can support player selectable denomination games in QCOM. Which particular method to be used is at the discretion of the regulatory authority.

1. Each available denomination version of the game appears via QCOM as a separate game (i.e. GVN) in the EGM (the reason for doing this is this allows for individual game denomination meters to be tracked via the Multi-game/Variation Meters response (15.6.6)). All denominations offered must be evenly divisible by the EGM's credit denomination (15.4.2) which the EGM must require to be set at 1 cent, thus the EGM will appear via QCOM to be still operating at the base credit denomination even though their actual denomination may be a multiple of this value. All QCOM requirements pertaining to a multi-game EGM would apply.
2. The game appears via QCOM as a single game operating at a one cent credit denomination.

The maximum denomination that may be offered in a player selectable denomination game is limited by MAXDEN (15.4.2).

3.2 EGM RAM Errors

3.2.1   EGMs operating under this protocol specification must state the date, time (adjusted by TZADJ) and source of a RAM error condition on the main display when a RAM error occurs in the EGM. Eg. "Self Audit Failure" [CP:22], "Meter Corruption" [CP:23], "Event Queue Corruption" [CP:24], "RNG State Corrupted" [CP:25], etc [CP:26]. Refer GMNS for other types of RAM errors.

3.2.2   Meters in audit mode must be able to be displayed on the EGM in a RAM error condition. [CP:27]

3.2.3   Upon the occurrence of any RAM corruption, program mismatch or fault condition requiring a RAM clear to reset, the EGM must immediately stop responding to polls from the SC [CP:28]. This is to prevent the EGM from transmitting any corrupt information while it resides in any of these error conditions.

3.3 Idle Mode

3.3.1 Definition of Idle Mode

In this document, Idle Mode is considered active on an EGM when a play is not in progress, all doors are closed, note stackers are returned, no form of credit redemption is in progress, no fault conditions or lockup conditions (as defined by QCOM) are active, audit mode or test mode are not active and the EGM has finished all foreground power up procedures.

In addition, the following EGM functions:

- bet adjustment,

- player instigated machine reserve feature, or
- game rules/paytable display (if present),
- the PID function (if implemented),
- full screen attract mode,
- power-save mode,
- game selection in a multi-game EGM via a game select screen,
- the display of any active Communications Disabling Conditions (section 6),

must all be considered to be a part of idle mode under these requirements. This is important for the implementation of all 'return to idle' mode operations (refer section 3.3.2 below) and other requirements that apply 'during idle mode'. Refer Appendix A.

Idle mode may be considered "paused" if interrupted by a higher priority concurrent state such as audit/test modes, door open, or fault condition. If idle mode is paused then 'during idle mode' requirements do not apply.

Please note that GMNS has a different definition of idle mode. QCOM's def. of idle mode does not extend beyond the scope of this document.

3.3.2   Return To Idle Mode

Please note the Idle Mode definition above.

If queued, QCOM requires an EGM to perform a number of tasks upon return to idle mode. The EGM may return to idle mode from any of the following situations; a power up, a completed play, a fault condition, a door open condition, a lockup condition or test/audit modes.

Under QCOM, any combination of the following events may be queued in the EGM to occur upon return to idle mode:

1. A large win lockup with respect to the LWIN threshold. Refer 15.4.5. (Test once per play only.)
2. A win payout with respect to the NPWINP/SAPWINP thresholds. Once per play only. Refer 15.4.5. *(QPv1.6)*
3. A System Lockup request as a result of the System Lockup Request Poll (15.4.9).
4. An ECT out as a result of the 'ECT from EGM' Lockup Request poll (16.3.2).
5. Any accumulated ECT-to-EGM as a result of the ECT to EGM poll (16.1.1) including Cashless mode bit updates.
6. A cash-out as a result of the Cash-Out Request Poll (0).
7. Any changes as a result of a queued EGM Game Configuration Change Poll (15.4.4)

The order in which the EGM performs the above items if queued is important for constistent behaviour between different manufacturer's EGMs. Accordingly, the above tasks must be performed by the EGM in the order listed above. [CP:29] *(QPv1.5 preferred order of priority was: ECT-Out, ECT-In, SL, CC, & Game changes. Order was not mandatory except for ECT out/in)*

The EGM must check for any of the above events upon every return to idle mode. If the EGM is returning to idle mode and has to perform one of the above events, then when finished performing an event, the EGM must restart from the top of the above list again. [CP:30] *(QPv1.5 this requirement was not mandatory)*

EGMs must 'return to idle mode' between every play to allow any of the above items to be executed if queued. [CP:31]

If the EGM is already in idle mode then any of the above conditions are initiated immediately as they occur. It is important to note that idle mode has a number of sub-modes as listed in the previous section (3.3.1). None of the idle mode sub-modes listed in 3.3.1 are allowed to delay any queued return to idle mode operations listed above, because the sub-modes are all defined as a part of idle mode and must occur immediately when the EGM is in idle mode. [CP:32]

3.4 Hoppers

An EGM may or may not have a hopper present under QCOM. Presence of a hopper is indicated by the EGM via the EGM Configuration Response (15.6.12).

QPv1.6 QCOM EGMs must not automatically resume a hopper pay-out after an interruption during the pay-out, e.g. door access, fault, power etc. The hopper pay must terminate upon return from the interrupting condition. [CP:33] A hopper payout may be restarted by pressing collect after the condition is cleared. This is to allow a patron to be easily hand-paid by a cancel credit (via the Cash Out Request Poll 0 or via Cancel Credit function in audit mode (if supported)) in the event of a recurring hopper fault.

3.4.1 Hopper Level Meter

EGMs using this protocol with hoppers must perform software based hopper level tracking for the purpose of reporting the Hopper Empty and Hopper Jam Events (refer 7.10.1.6 & 7.10.1.7) . A current hopper level meter must be maintained by the EGM by monitoring the physical movement of coins/tokens in/out of the hopper.

QPv1.6 EGMs with hopper weight sensors must cross check the hopper level ascertained from the weight sensor with the current EGM hopper level meter. Mismatches are reported via the Hopper Level Mismatch Event (7.10.1.5).

The EGM hopper level meter must never be allowed to go negative (either as a result of an adjustment made in test/audit mode [CP:34] or a collect [CP:35]); a negative operation on the hopper level meter must produce a zero value for the hopper level meter.

ECT in/out must not affect the hopper level meter. [CP:36]

3.4.2 Hopper Refills

It is recommended the refill procedure should be in units of $ and not in units of coins/tokens, as this allows the EGM to perform refills before denomination is setup, otherwise, the refill function must be disabled until after denomination is setup. [CP:37]

All refills which an EGM is commanded to record, will be of a variable amount which is specified by the attendant at the time of the refill [CP:38], with the default refill amount as set via the Hopper Maintenance Poll. [CP:39]

A refill option should still be possible as an option during hopper empty and jam fault conditions and at any time test or audit modes are available, as per GMNS. [CP:40]

EGMs must also provide the following functions in either test or audit mode of the EGM:

- The ability to record a refill of any amount with the default value as set by the hopper maintenance poll (15.4.17). [CP:41]
- Display and adjustment of the current hopper level meter (3.4.1) [CP:42].

All meters displayed must be labelled with units. [CP:43] The above functions must be intuitive (no user manual should be required), logical and foolproof. * This cannot be stressed enough. All values must be confirmed before accepting. [CP:44]

(*Acceptance of the overall legibility of the hopper maintenance feature on the EGM is at the discretion of the CEO [CP:45].)

3.4.3 Divertor Operations

Diverter operations must be primarily dependent on the hopper full probe and not the current hopper level meter. [CP:46] However the hopper level meter (3.4.1), is used for determining whether the EGM

should log the Hopper Empty of Hopper Jammed events. Refer to sections 7.10.1.6 & 7.10.1.7 for more information.

3.5 Credit Redemption

3.5.1   Credit redemption process under QCOM.

Upon a collect button press in idle mode with credit, or upon a win payout (*QPv1.6* refer section 15.4.5), as the highest priority, if the EGM was in cashless mode, then an 'ECT-from-EGM' lockup (16.3.1) results so long as the amount to be transfered is less than MAXECT (Section 16). Failing that, or if the EGM was not in cashless mode, then a hopper payout, Ticket Out, or Cancel Credit will result, depending on the EGM's current hardware configuration and the thresholds set in the Hopper/Ticket Printer Maintenance Poll (15.4.17). [CP:47]

Summary of the 'collect' / 'win payout' process:

If in **Cashless mode** then perform ECT-out unless credit is >= **MAXECT,** then act as if not in cashless mode (below).

If not in **Cashless mode** then the following rule applies with respect to the value of the credit meter:

Perform Hopper Payout < **COLLIM** (15.4.17) <= Perform *Ticket Payout*
< **TICKET** (15.4.17) <= Perform *Cancel Credit*

I.e in code form:

```
if (Cashless_mode && (credit_meter < MAXECT)) then ECT_from_EGM()
else
    if (credit_meter < COLLIM) then Hopper_Payout()
    else
        if (credit_meter < TICKET) then Cash_Ticket_Out()
        else Cancel_Credit()
```

Related: sections 3.5.3 (below) and 16.3.

3.5.2   All hopper [CP:48]/ticket printer collects [CP:49], cancel credits [CP:50], residual credit removal [CP:51] and electronic credit transfers [CP:52], must be displayed on the EGM in $.¢.

3.5.3   RCRF in EGMs with ticket printers.

Typically Cancel Credit is an available option for the player as a part of the RCRF. However if the EGM has a ticket printer, then the RCRF Cancel Credit option must be replaced with a "Ticket Out" option instead. [CP:53]

3.5.4   Denomination Display

QCOM EGMs must display its configured denomination to the player via its VDU and not via physical static artwork [CP:54]

3.6 Other

3.6.1   QCOM Parameter Access Rights.

Any parameter or variable that QCOM can controls must not be set or overridden by any other means on the EGM (eg. via a function in EGM audit mode), unless an overriding feature is a requirement specifically stated by this document.

3.6.2   QCOM State Variable Volatility.

Unless a QCOM variable or parameter already has a power up default listed in sections 8.2 or 8.3, any variable used in the implementation QCOM by the EGM must be maintained in the EGM's non-volatile memory.

Where a QCOM variable or parameter requires special handling, such as being stored in EGM critical memory (refer GMNS), then this requirement or other requirements will be specifically stated in this document where necessary.

Where use of critical memory (refer GMNS) is not mandated, consideration must still be given to minimising or eliminating critical sections (refer 2.2) of EGM program code.

3.6.3   Display of Text Data – General Requirements.

There are numerous requirements throughout QCOM where the EGM is required to display textual information to the user. For example; CDCs (6.2), SPAM (15.4.20), System Lockup (15.4.9), & GPMs (15.5.3), clock display (15.5.1) to name a few. There will always be a statement; "acceptability is at the discretion of the CEO, OLGR" or similar associated with any requirements to display QCOM related information. This is because as that the aesthetics involved in evaluating evaluation the display of any type of data displayed on the EGM's VDU means acceptability can be subjective.

However, as a general minimum requirement for the display of any QCOM related text data on the EGM, unless stated otherwise, any text display of QCOM related text data, must use a **10 point** or higher font and be at least **5mm** high[1] (unless stated otherwise) when measured directly off a properly adjusted EGM VDU. In addition, every pixel in the original font definition, when displayed, must map to at least one corresponding VDU pixel. For example, a 12 point font would be unacceptable if it was displayed in VDU area which was only 8 VDU pixels high in resolution.

For the purposes of evaluating the above requirements, for VDUs with separated Red, Green & Blue pixels, each set of RGB pixels is to be considered one VDU pixel.

Prominence, clarity (especially when scrolling), contrast and colour blending are also factors in determining acceptability of any text display and acceptability is at the discretion of the CEO, OLGR.

In regard to the display of faults and "Lockup Conditions" on the EGM, (Refer section 15.6.1 STATE field description for a list of possible 'lockup conditions' on the EGM), ideally any fault/lockup condition on the EGM should appear as a box overwriting the game result window (e.g. the reel display) with the required information displayed clearly inside. If the display box can be transparent so the the underlying graphics are still distinguishable then this would be even better. All meters on the game display must remain visible during the lockup. Acceptance of the overall legibility of the display of any fault/lockup condition on the EGM's display is at the discretion of the CEO [CP:55].

In general, if a hardcoded message is displayed on the EGM, then it should not use abbreviations or acronyms, or little understood terms if the target audience may be a player or attendant

---

[1] Font height is to be measured off any single capital letter of the font used.

4 Audit Mode Requirements

This section details the requirements for EGM audit mode regarding QCOM.

4.1 Audit Mode General Items

4.1.1   It must be possible to exit audit mode at any time on the EGM. [CP:56] (in order to be able to view the current state of the EGM, view all messages, CDC's, etc)

4.1.2   Display of Units.

All monetary meters must be displayed in $.¢. Hex numbers with no default units (eg. GVNs) must be displayed with leading "0x". BCD values must be displayed as decimal, not hex (i.e no "0x"). All other values must be displayed in decimal unless stated otherwise. Units should always be clearly indicated. [CP:57]

The following items must be included in the EGM's audit mode displays:

Note if the values to be displayed in EGM audit mode under section 4.1 are not displayed/refreshed in real time, then the EGM must also display the following message 'Values at: <ts>' (where <ts> equals the timestamp at which the data was sampled for display) on each applicable audit mode screen. [CP:58]

4.1.3   The version number of this protocol document under which the EGM is programmed to operate. This number is hard-coded into the EGM program. eg. "1.6.6" (the trailing letter which denotes the checklist version is not mandatory). Labelled PVERSION. [CP:59]

4.1.4   The Manufacturer ID (2 digits decimal) and EGM Serial Number (display in decimal, 6 digits) [CP:60]

4.1.5   The last received Site Details Broadcast message data [CP:61].

4.1.6   All Meters referred to in this document [CP:62]. Refer Meter Group/Contribution [CP:63], Multi-game/variation meters (in addition, also display per Variation, the current total game metered RTP as well as theoretical RTP [CP:64]), Progressive Meter, Bet Meter and Player Choice Meter responses [CP:65].

It is noted some meters have different definitions between QCOM and GMNS (e.g. Cancel Credit, Progressive and Money Out meters). ~~As the first priority, t~~The EGM must display GMNS's meters in audit mode as required/defined in GMNS as the default EGM meters. [CP:66] QCOM meters must be displayed as defined on a separate screen from GMNS meters in audit mode as a complete list. [CP:67] Meters transmitted via QCOM must be as per QCOM's definition.

4.1.7   Current total metered EGM RTP. In % to 2 decimal places [CP:68]. Refer to 12.1.1 for the formula.

4.1.8   Indicate whether the EGM has been successfully configured via the EGM Configuration Poll (15.4.2) [CP:69]

4.1.9   QPv1.6 For each game in the EGM, indicate if the game has been configured successfully via the EGM Game Configuration Poll (15.4.3). [CP:70]

4.1.10   QPv1.6 For each progressive game in the EGM, in addition to PRET (15.6.11), the EGM must display each game's current total ~~theorectical~~theoretical RTP (as PRET does not include the progressive component). [CP:71] (NB: A formula for a standard progressive level's RTP [based on parameters] is available in the OLGR Jackpot Systems Min. Requirements document. For LPs the EGM must assume 0% RTP contribution from the

LP levels until the Progressive Configuration Poll (15.4.6) has been received and accepted.)

4.1.11   QPv1.6 In general, all information defined as 'message data' which appear in all broadcast messages, polls and responses must be displayed in audit mode unless specifically excluded. [CP:72] This may exclude the following; reserved fields, unimplemented or non-applicable feature data fields, Data Link Layer fields, fields and flags which are used to request data, indicate a field size, or indicate the number of repeated entries. If unsure, err on the side of displaying the field. Each data field must be displayed in the units in which it is defined and be clearly and suitably labelled*. Units must always be indicated. Bit fields must be displayed as either YES/NO, or ENABLED/DISABLED, or ON/OFF, whichever is most applicable.

(*Acceptance of the overall legibility of the displayed information on the EGM is at the discretion of the CEO [CP:73].)

4.2   Real Time Data Display

The following information must be displayed in <u>real time</u> on the EGM's audit mode display preferably on the same display or where appropriate:

Note: If an item value is being displayed in audit mode in real time, then this requires no further clarification, however, it must be clear throughout EGM audit mode as to which items are not being displayed in real time and which are. E.g. separate the two types via pages or clearly marked sections. [CP:74]

4.2.1   The current poll address (display in decimal, 1…250, 0 may be used to indicate the EGM needs a PA) [CP:75]. (QPv1.5 PA display was not real time)

4.2.2   The last received date and time received from the SC. [CP:76] [CP:77] Labelled SYSTEM TIME. [CP:78]

4.2.3   The EGM's current RTC value, if present. TZADJ adjusted. Labelled RTC (LOCAL) TIME. [CP:79] [CP:80]

4.2.4   The total number of successful polls (i.e. with a valid CRC) to the EGM's ~~assigned~~ <u>designated</u> poll address <u>(15.5.4)</u> ~~(since last RAM clear)~~. Labelled TOTAL POLLS. [CP:81] [CP:82]

4.2.5   The total number of completed poll messages with CRC errors* received by the EGM on its <u>designated</u> ~~assigned~~ poll address <u>(15.5.4)</u> ~~(since last RAM clear)~~. (A QPv1.5 EGM also counted incomplete poll messages here, a QPv1.6 EGM tallies incomplete poll messages separately, see below) Labelled CRC ERRORS [CP:83]. [CP:84]

4.2.6   The total number of CRC errors* (since last RAM clear) received on Broadcast Messages with the broadcast poll address 0xFF. Labelled BCRC ERRORS. [CP:85]

(*A CRC error must be only counted when a CRC check is actually performed by the EGM on a received message and the result fails. i.e. do not count CRC errors for messages with invalid length fields or incomplete polls)

4.2.7   (QPv1.6) The total number of received incomplete messages to the EGM's address or on broadcast messages. Labelled INCOMPLETE. [CP:86]

4.2.8   The total number of UART overrun errors. Labelled OVERRUN [CP:87]. This should always be zero [CP:88] and is mainly for testing purposes. Also refer section 13.2.4.

4.2.9   The total number of negative acknowledged EGM responses (since last RAM clear). i.e. the total number of NAKs received (14.2.1). Labelled NON ACKNOWLEDGE [CP:89]. [CP:90]

4.2.10   QPv1.6 Last received QCOM protocol EGM program hash seed and completed* calculated hash. In hex. Labelled SEED and HASH respectively. [CP:91] Display a QPv1.5 seed and hash as MSB first hex numbers, for QPV1.6, display the values as LSB first hex numbers with a space (or highlight) every 4th character.

*The EGM must not display a hash result if a program hash calculation is in progress.

4.2.11   Primary event queue % full. Labelled EVENT QUEUE % [CP:92] [CP:93]

4.2.12   The current number of events yet to be sent and acknowledged by the SC. Labelled QUEUED EVENTS [CP:94]. [CP:95]

4.2.13   Primary and secondary[(QPv1.6)] QCOM event queue displays [CP:96] showing for each event:

- date and time (format: dd/mm/yy hh:mm:ss & not TZADJ adjusted) [CP:97]
- Event Sequence Number (2 digits hex)
- Event Status i.e. queued, sent, or purged (either colour or letter codes are fine)
- Event Code (4 digits hex)
- Short description (EGM ROM space permitting) [CP:98]
- Extra event data (clearly formatted) if any [CP:99].

Events must be displayed in the order in which they were logged. Purged events must still be displayed until overwritten. [CP:100] Refer section 7.3.

4.3   Last Play Recall

Last play recall must also display the following items in addition for each play result:

4.3.1   The game's name, variation and theoretical total percentage return*. Labelled: "name VAR XX XX.XX%" [CP:101]. [CP:102] (* special coding will be required here to incorporate the RTP from customSAP [CP:103] & LP [CP:104] games. NB: The EGM must assume LP PRTP = 0 until the Progressive Configuration Poll has been received and accepted {refer 15.4.6})

4.3.2   The local date and time (i.e. a TZADJ adjusted RTC timestamp) the play commenced. [CP:105]

4.3.3   Total ECT in/out, refer section 16. [CP:106]

4.3.4   Jackpots won. Refer section 10.4. [CP:107]

4.3.5   The game denomination of the play (only mandatory if the game supports Denomination Hot-Switching 3.1.3.1). [CP:108]

5   This section is intentionally left empty.

Ed: Keep this section to maintain section equivalence between QPv1.5 & QPv1.6 documents.

6 Communication Disabling Conditions (CDCs)

Communications Disabling Conditions (CDCs) in this document refer to the QCOM Protocol specific conditions defined in this section that can disable play and credit insertion on the EGM (except ECT). They are not to be confused or treated with as fault conditions or lockup conditions; they are a special separate set of disabling conditions displayed as a part of idle mode on the EGM.

All Communication Disabling Conditions in this protocol are defined in this section. Communications Disabling Conditions disable only the following functions on the EGM:

- new game play (including all bet and play buttons) [CP:109]
- new coin/token [CP:110]/bill/ticket acceptance [CP:111] (i.e. physical credit acceptance)

With one or more active CDCs, any play (including free games) in progress or state/fault/lockup in progress must first complete until the EGM returns to idle mode [CP:112], however the disabling of physical credit acceptance must occur immediately [CP:113] *(QPv1.6.4)*.

During a CDC, current credit must still be able to be collected by all the usual methods (3.5.1) [CP:114], all EGM test functions [CP:115] and audit mode must be accessible [CP:116], System Lockup must still function [CP:117], game configuration changes must still take place (as a result of 15.4.4) [CP:118] and ECT to/from the EGM must still function (for those CDCs where communication with the EGM is still possible, or if queued before the CDC). [CP:119]

CDC's must not prevent the EGM from returning to idle mode from any other mode [CP:120]. CDCs must not be considered to be a stand alone EGM STATE (15.6.1) or fault condition. CDCs may be active concurrently across all EGM modes and states, where active CDCs are simply displayed during EGM idle mode.

(If the SC wants to implement a disable where credit input is disabled but game play is not, then it may use the credit meter limit field CRLIMIT sent in the EGM Parameter Poll.)

An EGM will only enable new game play and credit acceptance, provided no communication disabling conditions are active and no other fault/lock up conditions exist.

During active Communications Disabling Conditions in idle mode, the EGM must remove enticements to insert credits [CP:121] and play (eg. 'play now' and button lights) [CP:122] from its display. If there is credit on the EGM in idle mode, any enticements to collect must remain during Communications Disabling Conditions. [CP:123]

The EGM may emit a sound denoting CDC entry if desired; however the EGM must not emit a sound denoting CDC entry if the EGM was already in one or more CDCs. [CP:124]

It must be possible to access EGM test mode and audit mode, during communication disabling conditions. [CP:125]

6.1   CDC Definitions

6.1.1   EGM Configuration Required

Denotes the EGM requires setup of a serial number, denomination or any other EGM configuration data (via the protocol or manual entry) that the EGM needs before being able to enable initially after an EGM RAM clear. This condition must end when the EGM accepts a valid EGM Configuration Poll (15.4.2). (This condition does not apply to individual game setup within an EGM. Use the 'No Game Enabled' disabling condition below. Individual games must be enabled as they are configured via the protocol). [CP:126]

6.1.2   Communication Time-Out

If an EGM does not receive a poll with a valid CRC to its designated poll address& for more than 10 consecutive seconds, the machine shall automatically disable in a "Communications Time-Out" CDC.[CP:127] The Communications Timeout Event may need to be logged (refer 7.10.3.6see 7.10.3).

The EGM must also perform the procedures in section 8.3 (EGM Communication Defaults) upon entry into a Communication Time-Out.

&Refer section 8.3.2 regarding when a poll address becomes 'undesignated' and section 15.5.4 regarding poll address designation. If the EGM does not have a designated poll address, then a Communications Timeout CDC cannot occur. (This prevents EGM's popping in/out of CT when being continually NAK'd)

QPv1.6 The EGM shall clear this CDC on the next successful poll address configuration for the EGM received via the poll address broadcast (15.5.4), and re-enable again provided no other disabling conditions exist. [CP:128]

(QPv1.5 FYI EGMs re-enabled on the next successful poll to its address (after poll address configuration has been received), provided no other disabling conditions exist. [CP:129]
NB: QPv1.5 - An issue was that a QPv1.5 EGM will would not accept a new poll address once assigned until either the EGM is powered down or until at least one successful poll is received on that address [CP:130])

6.1.3   Communication Fault

If a machine receives three (3) consecutive negative acknowledgements on its designated poll address from the SC (14.2.1), the EGM must automatically disable in a "Communication Fault" CDC. [CP:131]

The EGM must clear this CDC on the next positive acknowledgement from the SC (refer 14.2.1) to the EGM's designated poll address and enable provided no other disabling conditions exist. [CP:132] *(Despite the name this CDC (or any other CDC for that matter) is not to be treated as a fault condition by the EGM).*

(QPv1.6) The EGM must also perform the procedures in the EGM Communication Defaults (section 8.3) upon entry into a Communications Fault. [CP:133]

6.1.4   Event Queue Full

A machine must automatically disable upon a primary event queue full condition. [CP:134] It shall clear this CDC once enough events in the queue have been purged by the SC and the queue is no longer full, and re-enable provided no other disabling conditions exist. [CP:135]

6.1.5   Site Disable

An EGM will disable if the Site Enable Flag (SEF) (15.1.1) sent in the broadcast messages is not set. [CP:136] The EGM must clear this CDC when a SEF = 1 is received and re-enable provided no other disabling conditions exist.

6.1.6   EGM Disable

An EGM will disable if the Machine Enable Flag (MEF) (15.1.2) sent in any poll is not set. [CP:137] The EGM must clear this CDC when a MEF = 1 is received and re-enable provided no other disabling conditions exist.

6.1.7   No Game Enabled

An EGM will disable if all games in the EGM are disabled via their corresponding GEF (refer section 15.1.3). (An unconfigured EGM must be considered disabled). The EGM will clear this CDC if at

least one game is enabled via its GEF and re-enable if no other Communications Disabling Conditions exist.

6.1.8 LP Broadcast Timeout

New for QPv1.6. Applicable to LP games only. An EGM will disable if any currently enabled LP game has not received a LP broadcast/s (15.5.2) for **all** its LP levels for 90 consecutive seconds. [CP:138] *(Note the timeout must be implemented on a per level basis, e.g. the EGM must disable if 2 of 3 levels received an update in the timeout period, but a 3rd level did not. The EGM must not make any assumptions about the order of data in LP broadcasts! See 15.5.2 for more information)* [CP:139] This CDC will clear once the required LP Broadcasts have been received containing the outstanding levels and the EGM will also re-enable if no other Communications Disabling Conditions exist. [CP:140] This CDC is not applicable to external jackpots sent to the EGM (via 15.4.7).

6.1.9 Game Variation RTP Out of Range

New for QPv1.6.1. As a failsafe, an EGM will disable if the game currently selected by the user has a total RTP outside the allowable range as set via the EGM Configuration Poll (MINRTP & MAXRTP), refer 15.4.2. The EGM will clear this CDC when the problem is addressed[*] or another game or variation is selected whose total RTP is acceptable with respect to MINRTP & MAXRTP.

This CDC is **optional** but is provided because some EGMs may find it easier to apply VAR/total game RTP validation in idle mode of the EGM rather than during configuration of the game. [CP:141] E.g. For progressive games, there is a period of time from when the game is configured (15.4.3) until the Progressive Configuration Poll (15.4.6) is followed in the game's RTP may be temporarily out of range. (Another option with respect to progressive games, would be for the EGM to consider the game to be disabled until the required Progressive Configuration Poll is received)

[*] *In some cases, this CDC may not be addressable without an EGM RAM clear unless the EGM is a multi-game EGM, or the game is a progressive, or the game supports variation hot switching (9.2).*

6.2 Display of Communication Disabling Conditions

All currently active Communication Disabling Conditions on the EGM must be clearly indicated[*] by the EGM during idle mode in real time [CP:142], along with a generic "PLAY DISABLED" text message. [CP:143] The display of Communication Disabling Conditions must be a static display, i.e. CDC messages cannot be alternated with other information on the display [CP:144]. It must still be possible to determine the last play's final outcome while any number of CDC's are being displayed in idle mode [CP:145].

The display of Communications Disabling Conditions during idle mode must be considered in this protocol to be a part of idle mode. [CP:146] (Refer glossary section 2.2 definition of "Idle mode").

([*]Acceptance of the overall legibility of the display of CDC's on the EGM is at the discretion of the CEO (refer 3.6.3) [CP:147].)

There are two acceptable methods for the display of the individual CDC messages, display the full text or abbreviated letter codes.

The corresponding text used to display communication disabling conditions in full (as described above) must be as follows:

> **"PLAY DISABLED"** (displayed for all CDCs)
> **"Configuration Required"** [CP:148]
> **"Communication Time-Out"** [CP:149]
> **"Communication Fault"** [CP:150]
> **"Event Queue Full"** [CP:151]
> **"Site Disabled"** [CP:152]
> **"Machine~~EGM~~ Disable"** [CP:153]

"No Game Enabled" CP:154
"LP Broadcast Time-Out" CP:155 *(QPv1.6)*
"RTP Out of Range" CP:156 *(QPv1.6.1)*

Alternatively, or when the EGM cannot display all of the above text at once, the EGM must display "PLAY DISABLED XXXXX", where "XXXXX" represents a string where each character denotes an active communications disable condition. CP:157

'C'	if "Configuration Required" active
'T'	if "Communication Time-Out" active
'N'	if "Communication Fault" active (i.e. EGM is being NAK'd)
'F'	if "Event Queue Full" active
'S'	if "Site Disabled" active
'D'	if "EGM Deisabled" active
'G'	if "No Game Enabled" active
'L'	if "LP Broadcast Time-out" active *(QPv1.6)*
'R'	if "RTP Out of Range" active *(QPV1.6.1)*

For Example: "PLAY DISABLED ND", indicates Communications Fault and EGM Disabled are current.
"PLAY DISABLED TFS", indicates Communications Time-Out, Event Queue Full and Site Disabled are current.

The ideal CDC display would be a transparent box overlaid on the last game result display area.

Figure 1 CDC Display Example.

Screen dump courtesy of IGT (Australia) PTY LTD.

QPV1.6.7: The CDC display while visible, must incorporate within its display window, a poll cycle heartbeat display indicator. The indicator must 'tick' upon each received date and time broadcast message (15.5.1) with a good CRC processed by the EGM. This indicator needs to be uniform across all makes of machines, so make it a simple line roughly the size of a character (if not a use an actual character) located on the RHS of the CDC display, that rotates 45 degrees clockwise each 'tick'. (QSIM for example cycles a text character through: /-\| to achieve this). E.g. "Play disabled: TD /", "Play disabled: TD -", "Play disabled: TD \", "Play disabled: TD |". (The indicator will allow easier debugging of QCOM LANs without having to access EGM audit mode.)

7　EGM QCOM Event Queues

The purpose of the EGM QCOM event queues is to allow for the redundant or duplicated storage of events while they are in transit to the host system's fault-tolerant database. The event queues also ensure no new event data is lost if communications to the SC is broken for a period of time.

Please refer to the glossary (section 2.2) for a definition of the terms "log" and "logged" and similar forms when used this document in the context of "events".

Any queries regarding implementations of the EGM QCOM event queues can be readily answered by also having a look at the demo implementation program (qceq.exe) and source code contained in the QCOM SDK.

7.1　General

7.1.1　The EGM must have two QCOM event queues, a large primary event queue and a smaller secondary event queue. The main functional difference between the two queues is that the EGM automatically purges (deletes) events from the secondary event queue as the SC acknowledges them after being transmitted. However, the primary event queue is only purged by the EGM upon command from the SC. Note, a SC will only send the purge poll for events that have been confirmed to been stored in the host monitoring system computer in a fault tolerant manner.

7.1.2　The event queues must be implemented as a **circular buffer**. This allows for events 'purged' via the Purge Events Poll (15.4.15) to still be displayed in audit mode (4.2.13) for the longest possible period of time until they are overwritten by new events (also refer 7.3). Refer 7.11 for an illustration.

7.1.3　The EGM must store events in either the primary event queue or the secondary event queue as specified in the sections following. ~~and~~ The EGM must ~~report~~ ensure that events ~~to~~ received by the SC (via Event Responses (see 15.6.9) are received in the order in which they were created (from each event queue respectively) CP:158. This rule applies under all conditions; e.g. during normal operation, after interruptions and after negative acknowledgements (14.2.1). Related 7.5.4.

7.1.4　New events are always pushed onto the primary event queue unless it is full, or the event is defined as an "unnumbered" event in which case the event is pushed onto the secondary event queue.

7.1.5　All events must have a date and time stamp (not TZADJ adjusted) and must be logged upon the occurrence of the event as per its definition (Refer section 7.10) CP:159.

7.1.6　Upon the occurrence of an event as defined, the EGM must transmit the event, no later than the following two responses (provided no higher priority responses or events were pending). I.e. either of the next two responses must contain the event. CP:160

7.1.7　Event time-stamping.

Unless otherwise specified for a particular event (e.g. 7.10.3.10 & 7.10.3.11), events when triggered must be created & time-stamped with the current EGM Non-Volatile Real Time Clock (NV-RTC) date and time (refer to section 7.8 for important notes regarding RTC use). For EGMs which are not using a NV-RTC, the EGM must use the last received date and time from the broadcast information messages for the time-stamping of events CP:161.

Event time-stamps must not change for the life of the event.

EGMs may also use a software clock or volatile RTC for event time-stamping so long as it is used under the same conditions as a NV-RTC (section 7.8) with the noted exception that the time will be undefined upon EGM power up until receipt of the first broadcast message. In this case the last received broadcast time is used for event time stamping until a broadcast message is received to seed the software clock.

7.1.8   Assuming the SC does not change the EGM's time backwards; events must always be logged in chronological order.

7.1.9   Events must be stored in Non-Volatile RAM on the EGM. [CP:162]

7.1.10   Events are variable in length as some events have additional data associated with them. For example, a cancel credit amount or jackpot information (Refer Section 7.7).

7.2   The Primary Event Queue

7.2.1   The primary event queue must be capable of storing at least 200 events[*] [CP:163] (QPv1.5 was 100) but never more than 255 events (assuming events of the smallest size) [CP:164]. (The upper limit is due to the 8 bit event sequence numbers and also to do with the way systems are required to detect EGM event runaway). On storing the second last event before becoming full[*], an Event Queue Full event is stored in the last event position and the EGM declares itself in an event queue full condition. [CP:165]

7.2.2   The EGM remains disabled in an event queue full condition until a Purge Events Poll is received which results in enough free space of at least two events[*] in the primary event queue. [CP:166]

          [*] as the last event is always the Event Queue Full event (7.10.3.5).

7.3   The meaning of "Purged" Events in QCOM Event Queues

The only way to purge events from the primary event queue in an EGM, is via the 'Purge events Poll' (15.4.15). Events in the secondary event queue (see below) are purged as they are sent.

In this document, the meaning of the word "purge" when applied to QCOM event queue events means to flag the event/s as "purged", however the events are not actually destroyed by the EGM at that time. An event flagged as "purged" is an event that is not resent to the SC upon a "Request All Logged Events Poll" (15.4.14). So from the SCs point of view, purged events appear destroyed, but events flagged as "purged" are not actually destroyed by the EGM until overwritten by new events. The reason for all of this is to allow for "purged" events to be still displayed in audit mode of the EGM for as long as possible. (4.2.13)

Purged events in a given buffer (main or temp), must be overwritten in the order ~~of oldest first~~in which they were created. A purged event must not be overwritten until all other available event queue space has been first used to store new events.

7.4   Event sequence numbers

All events have an event sequence number which is used for purging events. Event sequence numbers are one byte hex numbers following the sequence:

          Event N sequence No = ((Event N-1 sequence No) modulus 255) + 1

          i.e. 1...255 [CP:167]

The next expected event sequence number to be used by the EGM must be stored in NV-RAM. [CP:168]

7.5   Secondary Event Queue

7.5.1   The secondary event queue must be at least 10 events in size[*] [CP:169].

7.5.2   The secondary event queue is used in a primary event queue full condition and for events defined as unnumbered events (section 7.6). [CP:170]

7.5.3   Events stored in this queue are automatically purged by the EGM as they are acknowledged by the SC. [CP:171]

7.5.4   If both the primary and secondary event queues have events to send, the oldest event must be sent first (as indicated by the event's date and timestamp). [CP:172] (Related: 7.1.3)

7.5.5   If both event queues should become full, the EGM must discard new events until space becomes available. [CP:173]

The EGM must consider the secondary event queue to be full until there is enough free space of at least two events in the secondary event queue.

* as the last event is always the Secondary Event Queue Full Event (7.10.4.3).

7.5.6   Events which are logged and stored in the secondary event queue must have an event sequence number of zero [CP:174] and must not affect the next event sequence number being utilised for the primary event queue. [CP:175]

7.6   Unnumbered Events

Some events are considered non-critical (for general information only) and are specifically logged to the secondary event queue, even if the primary event queue is not full. These events are called unnumbered events (Section 7.10.4) and they are only ever logged to the secondary event queue. Examples are the hopper pay out and ~~residual~~ RCRF cancel credit events.

7.6.1   If the secondary event queue is full, then new unnumbered events are discarded until space in the secondary event queue becomes available.

7.7   Extended Events

Some selected events (e.g. cancel credit, progressive award events) have additional event data associated with them. This is to indicate specific event information, such as the amount of a cancel credit and so on. An extended event is indicated by a non-zero size (ESIZ) field in the Event response (Refer section 15.6.9).

7.8   RTC Use

EGMs are encouraged to use their own NV Real Time Clock (if present) for time stamping of events. This is useful for more accurate [CP:176] time keeping when the SC is off-line. However, please ensure the following:

7.8.1   The EGM RTC must be immediately updated by the SC broadcast message date and time if the broadcast date and time is less than the current EGM date and time [CP:177], or if the EGM date and time is less then the broadcast date and time by more than a second. [CP:178] (Refer 15.5.1)

7.8.2   The EGM RTC is never allowed to get ahead of the SC time when logging events, as this will cause the SC to generate invalid date and time events for the EGM. However, setting the EGM RTC via the system time received in the broadcast message (15.5.1) should result in the EGM RTC being slightly behind system time, due to protocol ~~transmission and processing delays~~latencies. To also help prevent this, when the EGM

sets its the RTC off the broadcast ~~messge~~message, it must ensure to also reset the current RTC fraction of a second to zero. CP:179

7.8.3   The EGM RTC if used must be year 2000 compliant (note however that an EGM no longer has to deal with any date less than the year 2000). CP:180

7.9   Event Queue Integrity Requirements

7.9.1   Events must be stored in EGM NV memory via a methodology that supports ~~Three-way memory (as per GMNS) is not mandatory for the QCOM EGM event queues; however~~error detection. Error correction is optional. ~~t~~The EGM must maintain at a minimum, a 16 bit CRC (or better) over the event queue contents CP:181.

Two examples of methods are: 1. simply maintain a CRC per event. 2. To save memory, it is possible to maintain one overall event queue CRC that is updated as events are added the queue without having to recalculate the overall event queue CRC each time an event is added. To do this, the overall event queue CRC is calculated by XOR'ing the CRC for each event in the queue together. Then as events are added (or overwritten) to the queue, to get the new overall CRC result for the event queue, simply XOR each added/overwritten event's CRC into the overall CRC result to add / remove it from the overall event queue CRC result. By this method, adding or overwriting events from the event queue does not require a re-calculation of the entire event queue CRC for each operation in order to maintain one overall event queue CRC. CP:182 Refer to the QCOM SDK for an example of this methodology.

7.9.2   Updates to event queues and event queue pointers/indexers/CRCs must be treated as critical sections (refer 2.2) of program code.

7.9.3   The event queue integrity (i.e. CRC's) must be validated at least once upon every EGM power up. CP:183 An uncorrectable corruption of the event queue must result in a RAM error on the EGM (Refer Section 3.2). CP:184

Refer to the QCOM SDK for an example implementation of the QCOM event queues.

7.10   EGM Event Definitions

Events are divided up into four major categories:

- Fault Conditions.   (Section 7.10.1)
- Lockup Conditions.   (Section 7.10.2)
- Advisory Events.   (Section 7.10.3)
- Unnumbered Advisory Events.   (Section 7.10.4)

Events are reported via the EGM Event Response (Section 15.6.9)

All EGM fault and lockup conditions (which require an attendant) which exist in an EGM, must have a corresponding event code CP:185.   Undefined event codes are reserved for future expansion.   Event codes greater than 0x7FFF are reserved for system use.

Event codes are ordered into groups (fault conditions, advisory events etc.), the event type is denoted by the most significant 4 bits of the event code.

The implementation of all events listed in this section is mandatory unless stated otherwise, or if the EGM simply does not have the overall hardware device (eg. No note acceptor present). Events must be implemented as described.

Note to Monitoring Systems developers. For event reports produced by monitoring systems for the OLGR, the exact text to use for the event name is the section heading name when an event name

descriptor section is not already provided within the actual section for the event. Where the event has extended data, then this data must appear as specified in the "Format String" section for the event. Actual extended event data must be substituted for fields surrounded with < >.

For example, the EGM NV-RAM Cleared event (section 7.10.1.28) would appear in event reports as:

<SCID> <midserial> <dattime> <seq> <systemseqno> <ecode> **EGM NV-RAM Cleared   Serial No:** ***001654321***

Monitoring System developers should refer to the latest "OLGR Monitoring system Minimum Requirements Document" for more information on event reports for the OLGR.

7.10.1   Fault Conditions
CP:186

7.10.1.1   EGM Coin-In Fault

Event Code: 0x0004
Description: General coin in fault condition, eg. coin-in jam [CP:187].

7.10.1.2   EGM Excessive Coin Rejects Fault

Event Code: 0x0005
Description: Five or more rejected coins/tokens in any 10 second period the EGM was enabled to accept coins. Only applicable if hardware is capable. [CP:188]

7.10.1.3   EGM Coin-In Yo-Yo

Event Code: 0x0006
Description: A coin/token travelling in the reverse direction though the coin-in optics was detected. [CP:189]

7.10.1.4   EGM Cash Box Optic/Diverter Fault

Event Code: 0x0007
Description: A coin/token was detected going inadvertently to the cash box [CP:190] or vice versa [CP:191], or the coin-in diverter was detected to be in the wrong position for a time-out period [CP:192], or a blocked cash box optic [CP:193].

7.10.1.5   EGM Hopper Level Mismatch

Event Code: 0x0009 *QPv1.6*
Description: Only applicable to hoppers with built-in weight sensors, this event may be logged by the EGM whenever it detects an unacceptable* difference between the hopper level meter (refer 3.4.1) and the current physical level reported by the hopper weight sensor. [CP:194] EGMs with hopper sensors should check for this event on a periodic basis, such as every few seconds.
(*The tolerance threshold for the logging of this event may be set by the EGM manufacturer.)

Extended Data:

AMT   *4 bytes* **signed** *hex, equals current hopper sensor level minus hopper level meter, in cents. LSB first, display as signed decimal currency of up to 11 characters.*

Format String: "SL-HLM = $<*AMT*>"

7.10.1.6   EGM Hopper Empty

Event Code: 0x000A
Description: The EGM detected no coins out in the time-out period during a hopper collect and there were less than or equal to ten coins/tokens in the hopper as indicated by the hopper level meter (3.4.1), refer Hopper Maintenance Section (the ten coins may need to be

adjusted depending on the hopper type. Eg. A EGM slant top hopper with an extension arm typically has 15+ residual coins in it at all times). [CP:195]

7.10.1.7   EGM Hopper Jammed

Event Code: 0x000B
Description: The EGM detected for a time-out period, either a blocked hopper coin out sensor, the hopper not spinning (if detectable), or no coins out and the hopper level meter (3.4.1) is greater than ten coins [CP:196].

Exception - If the hopper can actually detect a true internal jam (eg. rotor not turning) and it is also of the type that consistently always pays out to the last coin, then the hopper level meter (3.4.1) does not need to be used in the generation of the hopper jam or hopper empty events.

7.10.1.8   EGM Hopper Runaway/Overpay

Event Code: 0x000C
Description: Unexpected coin paid from hopper at any time [CP:197]. Report one event only for each fault condition and not for each coin [CP:198]. EGMs using hoppers with weight sensors may also use this event to indicate an authorized negative reduction in hopper weight. Also see 7.10.3.39.

7.10.1.9   EGM Hopper Disconnected

Event Code: 0x000D
Description: If detectable by the EGM and a hopper is expected by the EGM (as indicated in the EGM Configuration Response (15.6.12)). This event is logged upon any detected disconnection of the hopper [CP:199].

7.10.1.10   EGM Mechanical Meters Disconnected

Event Code: 0x000E [CP:200]
Description: *QPv1.6* The EGM detected a disconnection of one or more pulsed mechanical meters. Only applicable if the EGM was expecting the mechanical meters to be present. It is recommended the EGM auto-detect their presence at RAM clear, if found, then the EGM should always expect to see them else this fault condition will be generated.

7.10.1.11   EGM Progressive Contr. Fault

Event Code: 0x0010
Description: Any fault/disconnection of an EGM progressive controller or progressive display controller [CP:201]

7.10.1.12   EGM I/O Controller Fault

Event Code: 0x0011
Description: Any fault/disconnection of an EGM I/O controller not already covered by another event [CP:202]

7.10.1.13   EGM Touch Screen Fault

Event Code: 0x0015
Description: Any fault/disconnection of a touch screen controller [CP:203].

7.10.1.14   EGM Manufacturer Specific Fault A

Event Code: 0x0017

Description: This event has been replaced by 7.10.1.16. The following Manufacturer Specific Fault events cover any fault conditions on an EGM which do not fall under any of the existing event code categories. This is in support of the requirement that all fault conditions (which require an attendant) must have an event code (refer 7.10). Manufacturers must advise and obtain approval (QPv1.6) if it is intended to use these event codes and for what fault conditions [CP:204].

7.10.1.15   EGM Manufacturer Specific Fault B

Event Code: 0x0018
Description: This event has been replaced by 7.10.1.16.

7.10.1.16   EGM Manufacturer Specific Event

Event Code: 0x0019
Description: See *7.10.1.14* above *(New for QPv1.6)*. As this event provides more information (with respect to the addition of the RTEXT field) compared with the previous two event types, QPv1.6 EGMs must use this event instead of either of the above two events where a Manufacturer Specific Fault event is required. The only time the A/B Man. Specific Fault events above should be used over this event, is when the event may be logged in high volumes.

Extended Data:

RTEXT   *16 bytes printable ASCII (refer section 2.3.8) including NULL (0x00) characters.*
            Right padded with 0x00's as required~~Zero padded~~. This field must always contain at least one terminating NULL character (0x00).
            RTEXT is a manufacturer assigned description of the fault (15 chars max).
            This event is not intended for high volume use, use events 0x0017, or 0x0018 instead if high volume logging of this event may be possible. Accordingly, all uses of this event by the EGM must be approved by the OLGR before implementation [CP:205].

            As result of increasing difficulty in diagnosing EGM configuration problems, EGM manufacturers ~~will be~~are required[*] to use this event as an advisory event for the purpose of providing more information in cases where the EGM rejects any of the configuration poll type messages (refer to the EGM Configuration Poll (15.4.2), EGM Game Configuration Poll (15.4.3), EGM Game Configuration Change Poll (15.4.4), or the EGM Progressive Configuration Poll (15.4.6)). E.g. If a Game Configuration Poll was received with an invalid PGID, after logging the 0x2036 "Invalid Game Configuration" event (7.10.3.22), the EGM would also log this event with RTEXT = "PGID Invalid". Refer to "RTEXT" ~~suggestions~~ text logging requirements located throughout the document. [CP:206] This will provide the monitoring system with more information when trying to diagnose configuration issues. There must be no situations where a QCOM host is not given enough information (via events) as to what to try next. [CP:207]

            *Be Advised; the above paragraph will become a mandatory QCOM requirement in all EGM software submissions by 2007.

Format String: "Reason: *<RTEXT>*"

7.10.1.17   EGM Stepper Reel Fault

Event Code: 0x001A
Description: Any fault/disconnection of a stepper reel mechanism [CP:208]

7.10.1.18   EGM Bonus Device Fault

Event Code: 0x001B
Description: *QPv1.6* Any fault with a Bonus Device such as spinning wheels etc. [CP:209]

7.10.1.19 EGM Note Stacker Full Fault

Event Code: 0x0020
Description: The Note Stacker is full [CP:210]

       EGM has disabled in a Note Stacker Full fault condition.
Note this condition may not lockup the EGM as GMNS allows this condition to be an advisory event. For this case there is an alternative advisory event provided; refer 7.10.3.18 (QPv1.6). Use this event instead.

7.10.1.20 EGM Note Acceptor Jammed

Event Code: 0x0021
Description: Covers any jam of the note acceptor or stacker unit [CP:211]

7.10.1.21 EGM Note Acceptor Disconnected

Event Code: 0x0022
Description: If a NA is expected then this event is logged upon any detected disconnection of a the Note Acceptor [CP:212].

7.10.1.22 EGM Note Acceptor Fault

Event Code: 0x0023
Description: Covers any general fault with the Note Acceptor or Stacker [CP:213]

7.10.1.23 EGM Excessive Note/Ticket Acceptor Rejects Fault
(QPv1.6 changed name/defn re Ticket-In)

Event Code: 0x0024
Description: Five or more notes/cash tickets rejected in any 40 second period and the EGM was enabled to accept notes/cash tickets (the GMNS definition is also acceptable) [CP:214]. (Cash tickets that could be read but were subsequently rejected via the Cash Ticket In Request Acknowledgement Poll (15.4.11) must not be counted for the purposes of detecting this event. [CP:215])

7.10.1.24 EGM Note Acceptor Yo-Yo

Event Code: 0x0025 *(new for QPv1.6)*
Description: A note/ticket travelling in the reverse direction though the note acceptor was detected [CP:216]. Only applicable if hardware is capable.

7.10.1.25 EGM Low NV-RAM Battery

Event Code: 0x0027
Description: Low NV RAM Backup Battery [CP:217]
Not applicable if the EGM uses Zero-Power[tm] NVRAM or equivalent.
The EGM should detect for this fault condition once at every power up.

7.10.1.26 EGM Low PF Door Detection Batt.

Event Code: 0x0028
Description: Low Power Fail Door Open Detection Battery [CP:218]. The EGM should detect for this fault condition once at every power up.

7.10.1.27 EGM EEPROM Error/Fault

Event Code: 0x0029

Description: Any detected general fault or corruption of an EEPROM device or equivalent if present. The EGM should detect for this fault condition at least once at every power up, or more frequently as required by GMNS if it stores critical data. ^{CP:219}

7.10.1.28   EGM NV-RAM Cleared

Event Code: 0x002A
Description: This event must be the first event logged by the EGM after an EGM NV-RAM clear. It is logged upon the first initialisation of the EGM with a poll address (15.5.4) ^{CP:220}. This event is not actually a fault condition but an advisory event.

Extended Data:

SER   *EGM Serial number (15.1.8), 3 bytes BCD, LSB first, display as 6 characters of unsigned decimal.*
MID   *EGM Manufacturer I.D. (refer section 15.1.7), 1 byte BCD, display as 2 characters of unsigned decimal.*

Format String: "Serial No: *<MID><SER>*"

7.10.1.29   EGM Ticket Printer Paper Out

Event Code: 0x002C ^{CP:221}
Description:

Logged by the EGM when it detects a paper out condition on the ticket printer.
(This event is only expected to be seen immediately following a successful ticket print as EGM ticket printers all appear have paper loaded detection, meaning that paper out detection will always occur well before any print is attempted.)

7.10.1.30   EGM Ticket Printer Paper Jam

Event Code: 0x002D ^{CP:222}
Description:

Logged as per event descriptor implies. Also refer section 22.1.1 for additional fault condition display requirements in the event this fault occurs while printing a ticket.

7.10.1.31   EGM Ticket Printer General Fault

Event Code: 0x002E ^{CP:223}
Description:

General purpose ticket printer fault event code. Log this event if no more specific event can be found. Also refer section 22.1.1 for additional fault condition display requirements in the event this fault occurs while printing a ticket.

7.10.1.32   EGM Cash Ticket Out Print Failure

Event Code: 0x002F
Description: *New for QPv1.6.*
                Applicable to EGMs with ticket out printers where the system is assigning authentication codes. Logged by the EGM after a Cash Ticket Out Request has received acknowledgement (15.4.10) and a Cash Ticket Out print was interrupted or has failed. ^{CP:224}

Extended Data:

TSER   *Ticket Serial number. 2 bytes hex, LSB first, display as 5 digits of unsigned decimal*
             The Ticket Serial Number is from the last generated EGM Cash Ticket Out Request Event (7.10.4.11).
TAMT   *4 bytes hex in cents, amount of ticket out from the last EGM Cash Ticket Out Request Event, LSB first, display as decimal currency of up to 11 characters.*

Format String: "Ticket Serial: <TSER> Amount $<TAMT>"

7.10.1.33 EGM Low Memory

Event Code: 0x0030
Description: *QPv1.6.* Refer GMNS. [CP:225]

7.10.1.34 EGM Power Down Incomplete

Event Code: 0x0031
Description: *QPv1.6.* Logged by the EGM upon power up if it detected that power down procedures were not completed prior the last power down. [CP:226] This event is applicable to EGMs with power fail procedures, a non-maskable power down interrupt or watchdog timers. This event is useful for detecting ESDs, software hangs or problems with an EGM's power supply.

7.10.1.35 EGM Auxiliary Display Device Failure

Event Code: 0x0032
Description: *QPv1.6.* Logged by the EGM upon any detected failure of an auxiliary display device or system, if applicable. [CP:227]

7.10.1.36 EGM Primary Display Device Failure

Event Code: 0x0033
Description: *QPv1.6.* Logged by the EGM upon any detected failure of the EGM's primary display device or system, if applicable. [CP:228]

7.10.1.37 EGM Tertiary Display Device Failure

Event Code: 0x0034
Description: *QPv1.6.* Logged by the EGM upon any detected failure of the EGM's tertiary display device or system, if applicable. [CP:229]

The above display device failure events are not mandatory if the hardware is not inherintely capable.

7.10.1.38 EGM License Key Missing/Failure

Event Code: 0x0035 *(QPv1.6.3)*
Description: Logged by the EGM whenever it detects a required license key in the EGM is missing or not responding [CP:230]. Also see 7.10.3.40

Extended Data:

KEYID   *8 bytes hex, Last inserted License Key ID, zero if none. Display as a 16 character hex zero padded LSB first unsigned number.*

Format String: "KeyID: *<KEYID>*"

7.10.2 Lockup Events

7.10.2.1 EGM Linked Progressive Award

Event Code: 0x1000
Description: This event is generated the instant (as a part of a linked progressive jackpot feature) a specific level won is revealed to the winning player (either directly or by inference, refer 10.6.1). [CP:231]
After this event is logged to the EGM's event queue, the EGM must indicate a LP lockup no later than the very next built General Status Response (15.6.1). [CP:232]

Extended Data:

GVN	*Game Version Number (15.1.5) of the award, 2 bytes hex, display as 4 upper case characters of unsigned hex.*
VAR	*Game Variation Number (15.1.6) of the award, 1 byte BCD, display as 2 characters of decimal*
PGID	*Progressive Group I.D. (15.1.10) of the award, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
PLVL	*1 byte hex,* *Bits 0…2 denote the progressive level of the award 0...7, display as 1 character of decimal* *Bits 3…7 reserved = 0. (mask out these bits when reading any data from this byte) QPv1.6.*
LJAMT	*The last received jackpot current amount* (15.5.2) *for the level won at the time the jackpot hit is revealed to the player* [CP:233] *, 4 bytes hex in cents, LSB first, display as unsigned decimal currency of up to 11 characters.*

Format String: "Gme: 0x*<GVN>* Var: *<VAR>* Group: 0x*<PGID>* Lev: *<PLVL>* $*<LJAMT>*"

7.10.2.2 EGM SA Progressive Award

Event Code: *0x1001*
Description: *(QPv1.6)* **This event is no longer used. Superseded by 7.10.4.8**
In fact, this event was never used even in QPv1.5 EGMs as QPv1.5 SAPs were all implemented silently. This event is generated, the instant a Stand Alone progressive level is won on an EGM [CP:234].

Extended Data:

GVN	*Game Version Number (CFF) of the award, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR	*Game Variation Number (CFF) of the award, 1 byte BCD, display as 2 characters of decimal*
PLVL	*1 byte hex, 0...7 progressive level of award, display as 1 character of decimal*
JPAMT	*Amount of award, 4 bytes hex in cents, LSB first, display as unsigned decimal currency of up to 11 characters.*

Format String: "Gme: 0x<GVN> Var: <VAR> Lev: <PLVL> $<JPAMT>"

7.10.2.3 EGM Large Win

Event Code: 0x1002
Description:

A win greater than or equal to the Large Win Lockup parameter value (refer 15.4.5) has occurred causing the machine to lock up for manual verification of the win and the win amount. [CP:235] This event is generated upon entry into the Large Win Lockup condition. *In QPv1.5 EGMs, if the large win or credit meter amount was to be cancelled (this was an option), then the EGM did not log this event, it logged the cancel credit event instead.*

Extended Data:

GVN	Game Version Number (15.1.5) of the large win, 2 bytes hex, display as 4 characters of unsigned hex, upper case.
VAR	Game Variation Number (15.1.6) of the large win, 1 byte BCD, display as 2 characters of decimal
WAMT	4 bytes hex in cents, amount of large win amount, LSB first, display as decimal currency of up to 11 characters.

Format String: "Gme: 0x*<GVN>* Var: *<VAR>* $*<WAMT>*"

7.10.2.4   EGM Cancel Credit

Event Code: 0x1003

Description: This event is generated upon entry into the lockup condition [CP:236]. Note, this event is not logged for cancel credits as a result of the residual credit removal feature, this is a separate event (see 7.10.4.10) [CP:237].

Extended Data:

CC	4 bytes hex, amount of cancel credit. LSB first, display unsigned decimal currency of up to 11 characters.

Format String: "$*<CC>*"

7.10.3   Advisory Events

7.10.3.1   EGM All Faults Cleared

Event Code: 0x2000
Description: Logged after only all current EGM faults have been reset, cleared from the display and no others are pending [CP:238] (not including Communications Disabling Conditions, they are not fault conditions [CP:239]).

7.10.3.2   EGM Lockup Cleared

Event Code: 0x2001
Description: Logged after the current EGM lockup condition has been cleared. Specifically, this event is logged after the Cancel Credit [CP:240], Large Win [CP:241] and LP Award [CP:242] lockup conditions.

It is not logged not after the ~~Residual~~ RCRF Cancel Credit Lockup [CP:243], System Lockup [CP:244], ECT lockup [CP:245], SAP Awards [CP:246], Ticket Out Lockup [CP:247], CRanE Lockup, or after any Communications Disabling Conditions [CP:248].

7.10.3.3   EGM Cancel Credit Cancelled

Event Code: 0x2002
Description: The cancel credit lockup was cancelled by the attendant (ie. the cancel credit was not performed. E.g. in the case where the patron inadvertently had pressed collect). The event 0x2001 would still be logged. This event is not logged when a RCR CC is cancelled. [CP:249]

EGM Manufacturers may provide for this feature at their discretion. The feature must only be accessible from test or audit mode [CP:250].

*QCOM does not encourage **player** cancellable CCs at this time. This is because if the feature did exist, it would be possible for a player to ~~SPAM~~ spam CC events and disable an EGM with an event queue full.*

7.10.3.4   Reserved for the QSIM Protocol Simulator

Event Code: 0x2003

7.10.3.5   EGM Event Queue Full

Event Code: 0x2006
Description: Logged upon the EGM primary event queue being declared full. This event must be logged as the last event in the primary event queue upon the queue becoming full [CP:251]. Refer Section on Events.

7.10.3.6   EGM Communications Time-Out

Event Code: 0x2007
Description: Logged upon the EGM detecting a 10 second Communications Time-out (CT) as defined in Section 6.1.2 (provided none of the exceptions below apply). [CP:252]

This event must only be logged ~~only~~ once per Communications Time-out (6.1.2) timeout ~~timeout~~ period. [CP:253]
This event must not be logged on an EGM power up (as the EGM is already in CT on power up by default, see 8.2). [CP:254]
QPv1.6 This event must not be logged if the EGM never received a valid poll (i.e. CRC) to its designated poll address (via 15.5.4) since ~~on~~ its last ~~assigned (15.5.4)~~ poll address designation (via 15.5.4) occurred. [CP:255]

This event must not be logged if the EGM does not currently have a designated poll address [CP:256]. (Section 15.5.4 designates a poll address & 8.3.2 un-designates a poll address.) For example, if the EGM is being constantly NAK'd; after three NAK's the EGM must assume communications defaults (i.e. reset its poll address) and after 10 seconds you should **not** see a communication timeout or event as the EGM does not have a poll address to time-out on.

7.10.3.7 EGM Game Variation Enabled

Event Code: 0x2008
Description: Logged when the EGM hot-switches a game's variation in the EGM to a new value (as a result of the EGM Game Configuration Change Poll (15.4.4)) [CP:257] and when a variation is configured initially after a RAM clear via the EGM Game Configuration Poll (15.4.3) [CP:258].

Extended Data:

GVN   *Enabled Game Version Number (15.1.5), 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Enabled Game Variation Number (15.1.6), 1 byte BCD, display as 2 characters of decimal*

Format String: "Gme: 0x*<GVN>* Var: *<VAR>*"

7.10.3.8 EGM RTC Refreshed

Event Code: 0x2009
Description: Logged upon the EGM updating its RTC time from the system time (refer 15.5.1) and the difference was greater than 5 seconds. This event must be logged with the new time [CP:259]. Only applicable if the EGM has a NV-RTC and is using it.

7.10.3.9 EGM Hopper Refill Recorded

Event Code: 0x200A
Description: This event is logged by the EGM whenever it is told to record a hopper refill (eg. by either via an attendant in audit mode, or via a QCOM poll request (15.4.17)). [CP:260] EGMs that use hoppers with weight sensors may automatically detect this event when the main door is open, but should wait until the EGM door has been closed before logging this event (3.4.1).

Extended Data:

AMT   *4 bytes unsigned hex, amount of refill in cents. LSB first, display unsigned decimal currency of up to 11 characters.*

Format String: "$*<AMT>*"

7.10.3.10 EGM Power Up

Event Code: 0x200B
Description: If the EGM is using a NV-RTC for event time-stamping, this event is logged once per EGM power up immediately after the EGM power down event below with the current NV RTC time [CP:261].
                If the EGM is not using a NV-RTC for event time-stamping, this event is logged once per EGM power up upon receipt of the first date and time broadcast message. The timestamp on the event must be the timestamp that was received in the first broadcast message. [CP:262]

7.10.3.11 EGM Power Down

Event Code: 0x200C
Description: If the EGM is using a NV-RTC for event time-stamping, this event is logged once per EGM power up with the last known date and time just prior the last power fail. To ensure the current power down time is recorded, an EGM using the NV-RTC must maintain a last known date and time NV variable refreshed from the RTC at least once every second for use with this event. [CP:263]

If the EGM is not using a NV-RTC for event time-stamping, this event is logged once upon EGM power up with the last known date and time received via the broadcast message immediately prior the power fail [CP:264].

Except after an EGM RAM clear, this event must be the first event the EGM logs to its event queue upon a power up. [CP:265]

7.10.3.12   EGM Door Open/Close and Stacker Removed/Returned Events

The following group of door open/close and stacker removed/returned events are logged;

- upon the occurrence of the event
- upon EGM RAM clear with their initial state
- upon EGM power up if the state of the door/stacker has changed since the power off.

In this protocol the stacker removed condition is to be treated as if it was a door open condition (i.e. it is not a fault condition and a return of the stacker will clear the condition).

QCOM EGMs must not lockup or disable after any door closure or door power off access. The monitoring system will decide if the EGM is to be disabled in each case based on jurisdictional requirements.

0x2010 **"EGM Main Door Opened"** [CP:266]
0x2011 **"EGM Main Door Closed"** [CP:267]
0x2012 **"EGM Cash Box Door Opened"** [CP:268]
0x2013 **"EGM Cash Box Door Closed"** [CP:269]
0x2014 **"EGM Processor Door Opened"** [CP:270]
0x2015 **"EGM Processor Door Closed"** [CP:271]
0x2016 **"EGM Belly Panel Door Opened"** [CP:272]
0x2017 **"EGM Belly Panel Door Closed"** [CP:273]
0x2020 **"EGM Note Acceptor Door Opened"** [CP:274]
0x2021 **"EGM Note Acceptor Door Closed"** [CP:275]
0x2022 **"EGM Note Acceptor Stacker Removed"** [CP:276]
0x2023 **"EGM Note Acceptor Stacker Returned"** [CP:277]
0x204A **"EGM Mechanical Meter Door Opened"** [CP:278] *(QPv1.6)*
0x204B **"EGM Mechanical Meter Door Closed"** [CP:279] *(QPv1.6)*
0x204C **"EGM Top Box/Aux Door Opened"** [CP:280] *(QPv1.6)*
0x204D **"EGM Top Box/Aux Door Closed"** [CP:281] *(QPv1.6)*

EGM Implementation notes re mapping actual monitored doors to QCOM events:

Some model gaming machines have many monitored doors (in some cases monitored doors in locations never envisaged by QCOM). Questions are sometimes raised as to the best way to map these doors onto currently defined QCOM events and door flags (refer General Status Response) in EGM software. Accordingly, following are some notes that should be taken into consideration when a EGM manufacturer is initially planning how their inherently monitored doors with be reflected in QCOM.

QCOM machines should be programmed to log a minimal number of QCOM events during operation up to the point that EGM monitoring ~~system~~ security or integrity ~~is~~may be ~~not reduced~~adversely affected. ~~This means that~~In other words, if there is no added value (regarding EGM integrity or security) in reporting two door access event (as opposed to one event) to access a

given piece of hardware in the EGM, then the EGM manufacturer is encouraged to propose logically OR'ing in software door sensors together[1] with respect to what is reported via QCOM.

Acceptance of the way the EGM maps its monitored doors to QCOM event / flags is at the discretion of the CEO [CP:282].

When two or more doors are logically OR'ed together in the QCOM software implementation, the EGM must still clearly identify which specific door/s are open on its primary display and in audit mode at all times. [CP:283] Without this, working with doors (especially diagnostics) will be difficult and confusing for the user. For this reason QCOM does not encourage chaining doors together at the hardware level.

Examples of desired implementations relating to specific EGM hardware:

Hardware: Logic Board
Desirable number of events logged to access: No limit.

As the machine logic board is rarely accessed and requires the highest security, there is no desirable figure on the number of events that are logged in order to access this device

Hardware: Coin Hopper, Banknote acceptor, coin acceptors, QCOM interface / connection
Desirable number of events logged to access: 1 (e.g. Main door)

A minimum of one event to access is desirable for these frequently accessed types of hardware.

Hardware: Note Stacker
Desirable number of events logged to access: 2 (e.g. main door and stacker door.)

Door event logging across machine resets and power fails

Below is a state table that must be used for events logged for a door over an EGM power fail or reset:

Door state before power fail	During power fail	Door events logged on power up
Opened	No change	None
Opened	Door is closed	Log door closed event
Closed	Door is opened and closed	None
Closed	Door is opened	Log door opened event

7.10.3.13   EGM Cash Box Cleared

Event Code: 0x202A *(QPv1.6.2)*
Description: An attendant must be presented with an option of logging a cash box cleared event upon detection of <u>either</u> a cash box door open or close. [CP:284] This event is logged if the attendant indicates in response that the cash box was actually cleared [CP:285]. (See also 7.10.3.37)

To expedite the process, drilling down through a number of attendant menus to a clearance option should be avoided.

Example implementation: Upon a cash box door open, in addition to the door open message, the user is presented with a message which says "To record a cash box clearance, turn the attendant key" If the user takes no action and closes the cashbox, then the EGM assumes a clearance is not occurring and simply returns from the cash box door open condition. If the attendant key is turned; the user is then presented with

---

[1] Where open = 1

a clearance confirmation via another button press. (NB: This example involved the use of a key to record a clearance to ensure that only an attendant can record a clearance for example in the event that a cash box sensor is faulty.)

7.10.3.14   EGM Note Stacker Cleared

Event Code: 0x2024
Description: An attendant must be presented *(QPv1.6)* with an option of logging a note stacker cleared upon detection of a Note Acceptor Stacker return/insertion. [CP:286] This event is logged if the attendant indicates in response that the stacker was actually cleared [CP:287]. (See also 7.10.3.37)

               Example implementation: Refer to EGM Cash Box Cleared event in section 7.10.3.13 above

7.10.3.15   Power Off Door Access Events

Power off door access events are logged upon power up using the current NV RTC time for a specific door if the EGM detected the door opening while the mains power to the EGM was either off or disconnected. If the EGM loses its ability to monitor a door during the power off state (i.e. backup battery drains), then the EGM must assume the door has been opened.

Note, Power off door accessed events must not be logged for a door if the EGM was powered down with that door already open [CP:288].

State table for events logged for a power off monitored door over a power fail:

Door state before power fail	If during the power fail:	Door events logged on power up
Opened	No change	None
Opened	Door is closed	Log door closed event.
Closed	Door is opened and closed	Log pwr off door accessed event
Closed	Door is opened	Log both pwr off door accessed & door opened events

Descriptor: "EGM Pwr Off Processor Door Access"
Event Code: 0x2025
Description: See above. Implementation of this event is mandatory. [CP:289]

Descriptor: "EGM Pwr Off Cash Door Access"
Event Code: 0x2026
Description: See above. Implementation is optional in an EGM. [CP:290]

Descriptor: "EGM Pwr Off Main Door Access"
Event Code: 0x2027
Description: See above. Implementation is optional in an EGM. [CP:291]

Descriptor: "EGM Pwr Off Note Acceptor Door Access"
Event Code: 0x2028
Description: (or power off note stacker removal) See above. [CP:292]
               Implementation is optional in an EGM.

Descriptor: "EGM Pwr Off Mechanical Meter Door Access"
Event Code: 0x2029
Description: See above. Implementation is optional in an EGM. [CP:293]

7.10.3.16   EGM Cash Ticket Printed

Event Code: 0x2030
Description: Only used by EGMs with ticket printers were the EGM is generating the ticket's serial
number and authentication code. [CP:294]

(**This event has been superseded in QPv1.6 (namely 7.10.4.12) and should not be implemented**)
(*NB this event code is now now being utilised by the XCOM-KCMS system solution – do not re-use*)

Extended Data:

TSER	*4 bytes hex, Ticket Serial Number. Display as 10 characters of unsigned decimal*
TAC	*4 bytes hex, Ticket Authentication Code, display as 10 characters of unsigned decimal*
AMT	*4 bytes hex, value of ticket in cents. LSB first, display unsigned decimal currency of up to 11 characters.*

Format String: "Ser: *<TSER>* TAC: *<TAC>* $*<AMT>*"

7.10.3.17   EGM Ticket Printer Ink Low

Event Code: 0x2031
Description: (if applicable) [CP:295]

7.10.3.18   EGM Note Stacker Full – Note Acceptance Disabled

Event Code: 0x2032
Description: (*QPv1.6*) Logged by the EGM when the Note Stacker becomes full and the EGM is not
disabling, but only the note acceptor is (as allowed under GMNS). [CP:296] If the EGM
does disable on a Note Stacker Full condition, then a fault condition event is provided
(see 7.10.1.19).

7.10.3.19   EGM Note Stacker High Level Warning

Event Code: 0x2033 [CP:297] (*QPv1.6*)

7.10.3.20   EGM Ticket Printer Paper Low

Event Code: 0x2034 [CP:298] (*QPv1.6)*

7.10.3.21   EGM Invalid EGM Configuration

Event Code: 0x2035
Description: Logged by the EGM whenever it receives invalid data in the EGM Configuration Poll
(15.4.2) [CP:299], or no games would be able to be offered as a result of the parameters
in the poll *(QPv1.6).* This event must be immediately followed by the logging of at least
the one "EGM Manufacturer Specific Fault – Extended" event (refer 7.10.1.16)
explaining the exact cause for the logging of this event. Where there are multiple
issues, it is acceptable for the RTEXT message in the event (7.10.1.16) must at least
provide a way forward e.g. by clearly stating one specific issue concerning a received
parameter.

7.10.3.22   EGM Invalid Game Configuration

Event Code: 0x2036
Description: Logged by the EGM whenever it receives invalid data in either the EGM Game
Configuration Poll (15.4.3) or EGM Game Configuration Change Poll (15.4.4). Refer
EGM Game Configuration Poll & EGM Game Configuration Change Poll for more
information [CP:300]. This event must be immediately followed by the logging of the "EGM
Manufacturer Specific Fault – Extended" event (refer 7.10.1.16) explaining the exact
cause for the logging of this event.

7.10.3.23 EGM Invalid Progressive Configuration

Event Code: 0x2037
Description: *(QPv1.6)* Logged by the EGM whenever it receives invalid data in the Progressive Configuration Poll (15.4.6). [CP:301]. Clarification: If the GVN in the Progressive Configuration Poll (15.4.6) was invalid, then this event should be logged with the GVN of the invalid GVN and a VAR = 0. [CP:302] This event must be immediately followed by the logging of the "EGM Manufacturer Specific Fault – Extended" event (refer 7.10.1.16) explaining the exact cause for the logging of this event.

Extended Data:

GVN   *Applicable Game Version Number (15.1.5), 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Current Game Variation Number (15.1.6), 1 byte BCD, display as 2 characters of decimal*

Format String: "Gme: 0x<*GVN*> Var: <*VAR*>"

7.10.3.24 EGM Processor Over-Temperature

Event Code: 0x2040 [CP:303]
Description: *(QPv1.6)* Only applicable for EGMs with actively cooled CPUs and processor boards with temperature detection. This event indicates the EGM's processor temperature has exceeded an acceptable operating threshold level and indicates the processor may shortly overheat if the temperature continues to rise. Suggest the over-temperature level be set as low as possible, however consideration should be taken into account with respect to the wide ambient temperature range of gaming venues. If used, this event should be logged when the critical temperature is exceeded in the positive direction. Some hysteresis may be necessary in the implementation of this event to avoid repeatedly logging the event in borderline situations.

EGMs monitoring a processor/s temperature should also implement a critical over temperature threshold. After this threshold is exceeded, the EGM must do everything possible to protect its processor. This includes automatically shutting down, or throttling down the CPU, or entering a low/power-save mode to protect its CPU from damage. If an EGM is unable to keep up with communications with its CPU throttled, then it must cease all QCOM communications to avoid sending garbage onto the LAN.

It is acknowledged that in some exceptional circumstances (e.g. failure of a heat sink) that there simply may not be enough time to log this event. However, it is expected in the case of a sudden complete failure of any cooling fan, that the EGM will be able to shutdown before incurring any permanent damage.

It is recommended that the thresholds be reasonably configurable in audit mode on the EGM.

It is also permissible to operate this event as a fault condition for better prominence; allowing multiple occurrences per power cycle (and saves implementing hysteresis).

7.10.3.25 EGM Cooling Fan Low RPM

Event Code: 0x2042 [CP:304]
Description: *(QPv1.6)* Only applicable for EGMs which can detect a failure or low RPM of one or more cooling fans in the device. This event should be logged upon detection of the event and every subsequent power up until the problem is fixed. Some hysteresis may

be necessary in the implementation of this event to avoid repeatedly logging the event in borderline situations.

It is also permissible to operate this event as a fault condition for better prominence; allowing multiple occurrences per power cycle (and saves implementing hysteresis).

7.10.3.26 EGM Call Service Technician

Event Code: 0x2043
Description: *(QPv1.6)* This event is to be used in instances when the EGM detects it really needs technical assistance from a qualified service technician. [CP:305] This event is not for RAM errors, as an EGM cannot communicate during a RAM error. This event may be implemented at the EGM manufacturer's discretion but only with prior approval from the OLGR.

7.10.3.27 Reserved Events for QCOM FTP

Event Code: 0x2045…0x2049   Reserved for possible use in QCOM FTP *(QPv1.6)*

7.10.3.28 Reserved for Mech Meter and Aux Doors

Event Code: 0x204A…0x204D
Description: Refer 7.10.3.12

7.10.3.29 EGM Recoverable RAM Corruption

Event Code: 0x204E
Description: *(QPv1.6)* Logged by the EGM upon detection of any recoverable RAM corruption.
[CP:306]

There is the possibility that this event can cause an event runaway in an EGM. To prevent an event runaway from occurring, after detecting and logging this event as defined the first time since the EGM was last powered up, the EGM must only log this event again once every 10 minutes if another recoverable RAM corruption has occurred in that time and has not been powered down. (I.e. Once per power up, then every 10 mins)

EGM manufacturers are encouraged to log an RTEXT event (7.10.1.16) immediately after this event to indicate the area that was corrupted or containing any information for diagnostic purposes. [CP:307]

7.10.3.30 EGM New PID Session Started

Event Code: 0x204F
Description: (QPv1.6) Logged after a player elects to start a new session (i.e. reset PID statistics) via the PID function. (Note, not all PID displays will have a new session button.) The EGM must not log this event if there was no play in between two successive PID sessions or if a PID is not implemented. [CP:308]

7.10.3.31 EGM Denomination Enabled/Changed

Event Code: 0x2050 *(QPv1.6)*
Description: Logged whenever the EGM instigates a denomination hot-switch to a new denomination value [CP:309] and when the EGM is successfully configured initially after a RAM clear via the EGM Configuration Poll (15.4.2) [CP:310]. Refer section 3.1.3.1on Denomination Hot-Switching.

Extended Data:

DEN   *(New) Game Denomination. 4 bytes hex in cents. Display unsigned decimal currency of up to 11 characters.*

Format String: "Denom $<*DEN*>"

7.10.3.32   EGM Invalid Denomination

Event Code: 0x2051 (*QPv1.6*)
Description: Logged by the EGM whenever it receives what it ~~consideres~~considers to be an invalid or unreasonable denomination in the EGM Configuration Poll (15.4.2) either upon EGM initialisation [CP:311], or denomination hot-switch [CP:312].   Refer section 3.1.3.1on Denomination Hot-Switching.

7.10.3.33   EGM Hopper Calibrated

Event Code: 0x2052 (*QPv1.6*)
Description: Only applicable to hoppers with built-in weight sensors, this event must be logged by the EGM whenever it is told (currently only via a command in audit/test mode) to perform a hopper weight sensor calibration. [CP:313]
Extended Data:

AMT   *4 bytes unsigned hex in cents, amount of money in the hopper after the calibration (in cents). LSB first, display unsigned decimal currency of up to 11 characters.*

Format String: "Level $<*AMT*>"

7.10.3.34   EGM Progressive Configuration Changed

Event Code: 0x2053
Description: Logged by the EGM whenever it receives and accepts a change in progressive configuration via the Progressive Configuration Change Poll (15.4.6) [CP:314] *QPv1.6*.
This event must *a*lso be logged immediately after logging the "EGM Game Variation Enabled" event (7.10.3.7) when a progressive game is configured initially after a RAM clear via the EGM Game Configuration Poll (15.4.3) [CP:315] QPv1.6.7.
This event must also be logged whenever the reported value of HRATE for the game changes. (Refer section 15.6.3 for specifics. Note here that the event must not be logged as a result of a player action.) [CP:316] QPv1.6.7

Extended Data:

GVN   *Game Version Number (15.1.5) of updated game, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Current Game Variation Number (15.1.6) of updated game, 1 byte BCD, display as 2 characters of decimal.*
PRTP   *Total RTP of updated progressive game component x 10000. 4 bytes hex.*
PRTP is the total RTP of the startup prize, plus percentage increment, plus auxiliary RTP of all progressive levels for the game denoted by GVN. Display as a percentage. E.g. "1.2345%" equates to PRTP = 12345. Any significant digits dropped due to precision limits must truncated (i.e. rounded down) [QPV165]

$$PRTP = (RTP_{level\ 0} + \ldots + RTP_{level\ n}) * 10000$$

*Where (typically) RTP$_{level\ n}$ = HRATE * SUP * 100.0 + (PINC / 10000) + (AUXRTP / 10000)*

(To verify an implementation of the above RTP formula is working when using the typical formula above, set HRATE = 0.5, SUP = 1c, PINC & AUXRTP = 250000 (i.e 25%), then RTP$_{level\ n}$ should equal 100)

If PRTP varies (e.g. for different bet values) the EGM must report the PRTP for the minimum bet. (Note that systems will add PRTP from this event to the game's PRET

field value typically for the purpose of ensuring the EGM's total RTP is within MINRTP and MAXRTP).

Format String: "Gme: 0x*<GVN>* Var: *<VAR>* RTP (*<PRTP>*/10000)%"

7.10.3.35   EGM Top NP Prize Hit

Event Code: 0x2054 *(new for QPv1.6)*
Description: Logged by the EGM whenever a player wins the highest paying tabulated non-progressive prize on the EGM's payscale in any game for a given bet, irrespective of any subsequent gamble result. Not Applicable to Roulette or games where the top prize is a regular occurance (i.e. this event should be able to be enable or disabled on a game by game basis).

Not to be implemented in EGMs from QPv1.6.3 onwards – Alternatively the LWIN parameter can be used by systems in order to log similar information (15.4.5***)*** CP:317

Extended Data:

GVN   *Game Version Number (15.1.5) of winning game, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Game Variation Number (15.1.6) of winning game, 1 byte BCD, display as 2 characters of decimal*

Format String: "Gme: 0x*<GVN>* Var: *<VAR>*"

7.10.3.36   EGM Invalid Ticket Out Acknowledgement

Event Code: 0x2055 *(new for QPv1.6)* CP:318
Description: Logged as directed elsewhere in this protocol. Refer section 15.4.10 (*basically an incorrectly echoed TSER or TAMT*). This event is only applicable to EGMs with TO support. Refer 22 for more information on TITO.

7.10.3.37   EGM Period Meters Reset

Event Code: 0x2056 *(new for QPv1.6.1)*
Description: Logged by the EGM whenever any period meters are reset by an attendant, or with respect to any other local jurisdictional requirements. CP:319 (See also 7.10.3.14)

7.10.3.38   EGM EXTJIP Icon Display Enabled

Event Code: 0x2057 *(New for QPv1.6.1)*.
Description: Refer to section 15.4.7 (EXTJIP).

Extended Data:

ICONDESCR   *16 bytes printable ASCII (refer section 2.3.8) including NULL (0x00) characters.* Right padded with 0x00's as required~~Zero padded~~. *This field must always contain at least one terminating NULL character (0x00).*
               Refer EXTJIP section 15.4.7.

7.10.3.39   EGM Hopper Overpay Amount

Event Code: 0x2058 (*QPv1.6.2)*
Description: Only applicable to EGMs with hoppers, this event is logged upon the clearing of a Hopper Runaway/Overpay fault condition (7.10.1.8). CP:320
Extended Data:

AMT   *4 bytes unsigned hex in cents, the amount of money associated with the last* Hopper
                      Runaway/Overpay fault condition *(in cents). LSB first, display unsigned decimal
                      currency of up to 11 characters.*

Format String: "$*<AMT>*"

7.10.3.40   EGM License Key Detected

Event Code: 0x2059 *(QPv1.6.3)*
Description: This event is intended for EGMs that utilise some form of software licensing system
                 and its implementation in an EGM is optional. Logged by the EGM whenever it detects
                 an newly inserted, or changed or **production** license key in the EGM [CP:321]. Also
                 logged at EGM RAM clear to record the initial key ID (not every power up) [CP:322]. Also
                 see 7.10.1.38 and below.

Extended Data:

KEYID   *8 bytes hex, License Key ID (zero if not known). Display as a 16 character hex zero
                  padded LSB first unsigned number.*

Format String: "KeyID: *<KEYID>*"

7.10.3.41   EGM Non-Production License Key Detected

Event Code: 0x205A *(QPv1.6.3)*
Description: This event is intended for EGM's that utilise some form of software licensing system
                 and its implementation in an EGM is optional. Logged by the EGM whenever it detects
                 an newly inserted, or changed or license key in the EGM that is intended for
                 development, testing or other non-production use [CP:323]. Also logged at EGM RAM
                 clear to record the initial key ID (not every power up) [CP:324]. Also see 7.10.1.38 and
                 above.

Extended Data:

KEYID   *8 bytes hex, License Key ID (zero if not known). Display as a 16 character hex zero
                  padded LSB first unsigned number.*

Format String: "KeyID: *<KEYID>*"

7.10.3.42   EGM Ticket-In Timeout

Event Code: 0x205B *(QPv1.6.5)*
Description: This event is logged as defined in section 22.2.1 [CP:325]
Extended Data:

AUTHNO   *Ticket Authorisation Number, 16 bytes hex, LSB first.
                  This number is read from the tickets barcode (refer 22.3).
                  E.g. if the barcode value was 57-6460-7523-0342-3488 (decimal) the AUTHNO would
                  be set equal the same value (which in MSB first hex is:
                  0x0000000000000000**8**00000000000000)*
                  Display AUTHNO values in the EGM's event queue display (re section 4.2.13) as per
                  the numeric format on the printed ticket, e.g. "57-6460-7523-0342-3488" [CP:326]

Format String: "Auth No: ~~0x~~<AUTHNO>"

7.10.3.43   EGM Ticket-In Aborted

Event Code: 0x205C *(QPv1.6.5)*
Description:

This event must be logged by the EGM if ever after logging a Cash Ticket In Request event (7.10.4.13) it subsequently aborts that Ticket-in process for any reason (excluding "fail" acknowledgements re 15.4.11 and timeouts re 22.2.1). [CP:327] For more information regarding interruptions to the TI process, refer section 22.2.2

Extended Data:

AUTHNO   *Ticket Authorisation Number, 16 bytes hex, LSB first.*
              This number is read from the tickets barcode (refer 22.3).
              E.g. if the barcode value was 57-6460-7523-0342-3488 (decimal) the AUTHNO would
              be set equal the same value (which in MSB first hex is:
              0x0000000000000000800000000000000)
              Display AUTHNO values in the EGM's event queue display (re section 4.2.13) as per the numeric format on the printed ticket, e.g. "57-6460-7523-0342-3488" [CP:328]

Format String: "Auth No: 0x<AUTHNO>"

7.10.3.44   EGM Transaction Denied - Credit Limit Reached

Event Code: 0x205D *(QPv1.6.6)*
Description:

This event must logged by the EGM as a result of any ECT-to-EGM (16.1.1) [CP:329] or Cash Ticket In (15.4.11) [CP:330] transaction which the EGM denied because the transaction would cause the EGM's credit meter display capabilities to be exceeded (this must include any transaction that would cause the credit meter to rollover), or any other credit meter limit implemented by the EGM to be exceeded. [CP:331]

This event replaces the former logging of the Manufacturer Specific Fault event (7.10.1.16) with RTEXT = "ECTIN DispLimit" previously required as a result of an ECT-to-EGM denial (16.1.1).

7.10.4 Unnumbered Advisory Events

There events are logged only to the secondary event queue. Refer 7.5 for more information.

7.10.4.1 EGM Hopper Payout

Event Code: 0x3000
Description: Logged upon completion of every hopper payout [CP:332].

(QPv1.6) EGMs also terminate a hopper pay if a fault occurs (refer section 3.4). If a hopper pay is aborted before any coins have been paid, the EGM doesn't have to log the event in this case.

Extended Data:

AMT   *4 bytes hex, amount of hopper pay in cents. LSB first, display unsigned decimal currency of up to 11 characters.*

Format String: "$<*AMT*>"

7.10.4.2 EGM RCRF ~~Residual~~ Cancel Credit Lock Up

Event Code: 0x3001
Description: Used by QPv1.5 EGMs only. **(QPv1.6 This event has been superseded, see below)** This event is generated upon entry into the residual credit removal cancel credit lockup condition [CP:333].

Extended Data:

CC   *2 bytes hex, amount of cancel credit in cents. LSB first, display up to 5 characters of unsigned cents*

Format String: "<*CC*>¢"

7.10.4.3 EGM Secondary Event Queue Full

Event Code: 0x3002
Description: Logged upon the EGM secondary event queue becoming full. This event must be logged as the last event in the secondary event queue upon the queue becoming full [CP:334]. Refer section 7.5.

7.10.4.4 EGM New Game Selected

Event Code: 0x3003
Description: Applicable to multi-game EGMs only. This event logged whenever a new selected game is actually played which was a different game from the last played game. This is with respect to GVN and not applicable to variation changes [CP:335]. This event must always be logged on the first game played after a RAM clear on a multi-game EGM [CP:336].

Extended Data:

GVN   *Game Version Number (15.1.5) of the new game, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Game Variation Number (15.1.6) of the new game, 1 byte BCD, display as 2 characters of decimal*

Format String: "Gme: 0x<*GVN*> Var: <*VAR*>"

7.10.4.5 EGM ECT from EGM

Event Code: 0x3004
Description: (QPv1.5 EGMs only) Refer 16.3.3

7.10.4.6   EGM CRanE Hit

Event Code: 0x3005
Description: Refer section 17.4 on CRanE

7.10.4.7   EGM System Lockup User Response

Event Code: 0x3006 [CP:337]
Description: QPv1.6 Refer to the System Lockup Request Poll (15.4.9)
Extended Data:

YESNO   *1 byte hex.*
    Bit 0 Yes/No
        A value of one denotes a yes response
        A value of zero denotes a no response
        Display as either "Yes" or "no" respectively.
    Bits 1…7 Reserved = 0

Format String: "Ans: *<YESNO>*"

7.10.4.8   EGM SAP Award

Event Code: 0x3008
Description: This "unnumbered event" *(added in QPv1.6)* is generated the instant a SAP level is won on an EGM [CP:338]. (This event supersedes the older SAP Award Event used by QPv1.5 EGMs. 7.10.2.2)
Extended Data:

GVN   *Game Version Number (15.1.5) of the award, 2 bytes hex, display as 4 characters of unsigned hex, upper case.*
VAR   *Game Variation Number (15.1.6) of the award, 1 byte BCD, display as 2 characters of decimal*
PLVL   *1 byte hex*
    Bits 0…2   0...7 progressive level of award, display as 1 character of decimal
    Bits 3…7   Reserved = 0, (mask out these bits when reading any data from this byte) QPv1.6.
JPAMT   *Amount of award, 4 bytes hex in cents, LSB first, display as unsigned decimal currency of up to 11 characters.*

Format String: "Gme: 0x*<GVN>* Var: *<VAR>* Lev: *<PLVL>* $*<JPAMT>*"

7.10.4.9   EGM ECT From EGM

Event Code: 0x3009
Description: (QPv1.6 EGMs) Refer 16.3.3

7.10.4.10   EGM RCRF ~~Residual~~ Cancel Credit Lock Up

Event Code: 0x3010
Description: *New for QPv1.6 EGMs and replaces 0x3001.* This event is generated upon entry into the residual credit removal cancel credit lockup condition [CP:339].
Extended Data:

CC   *4 bytes hex, amount of cancel credit in cents.*
    LSB first, display up to 10 characters of unsigned cents

Format String: "*<CC>*¢"

7.10.4.11 EGM Cash Ticket Out Request

Event Code: 0x3011
Description: *New for QPv1.6.*
Applicable to EGMs with ticket out printers where the system is assigning authentication codes. Logged by the EGM upon entry into a Cash Ticket Out lockup condition [CP:340] this event instigates a Ticket Out authorisation or otherwise via the Ticket Out Request Acknowledgement Poll (15.4.10). Refer section 22 for more information on TITO.

Extended Data:

TSER *Ticket Serial number. 2 bytes hex, LSB first, display as 5 digits of unsigned decimal.*

TSER is assigned by the EGM, **starting from one at the last EGM RAM clear**. [CP:341] TSER must be a global NV value in the EGM. ~~This~~ TSER ~~number is~~ must be incremented by one for each Ticket Out Lockup entered by the EGM and must wrap around to 1 when its maximum value is reached. [CP:342]

TAMT *4 bytes hex in cents, amount of ticket out, LSB first, display as decimal currency of up to 11 characters.*

Format String: "Ticket Serial: <TSER> Amount $<TAMT>"

7.10.4.12 EGM Cash Ticket Out Print Successful

Event Code: 0x3012
Description: *New for QPv1.6.*
Applicable to EGMs with ticket out printers where the system is assigning authentication codes. Logged by the EGM after a Cash Ticket Out Request has received acknowledgement (15.4.10) and a Cash Ticket Out has been "successfully" printed by the EGM' ticket printer device. [CP:343] Refer section 22.1.3 for the definition of a successful cash ticket print~~Refer section 22 for more information on TITO.~~

Extended Data:

TSER *Ticket Serial number. 2 bytes hex, LSB first, display as 5 digits of unsigned decimal*
The Ticket Serial Number is from the last generated EGM Cash Ticket Out Request Event above.
TAMT *4 bytes hex in cents, amount of ticket out from the last EGM Cash Ticket Out Request above, LSB first, display as decimal currency of up to 11 characters.*

Format String: "Ticket Serial: <TSER> Amount $<TAMT>"

7.10.4.13 EGM Cash Ticket In Request

Event Code: 0x3013
Description: *New for QPv1.6.*
Applicable to EGMs with a Ticket-In device where the system is assigning/verifying authentication codes. Logged by the EGM upon insertion of a ticket in with a readable barcode while the EGM was enabled to accept credit. [CP:344] This event instigates a ticket in verification via the Cash Ticket In Request Acknowledgement Poll (15.4.11). Refer section 22 for more information on TITO.

Extended Data:

AUTHNO *Ticket Authorisation Number, 16 bytes hex, LSB first.*
This number ~~is read from the~~must denote the ticket's barcode (refer sections 22.2 & 22.3).
E.g. if the barcode value was 57-6460-7523-0342-3488 (decimal) the AUTHNO would be set equal the same value (which in MSB first hex is:
0x00000000000000008000000000000000)

Display AUTHNO values in the EGM's event queue display (re section 4.2.13) as per the decimal format on the printed ticket, e.g. "57-6460-7523-0342-3488" [CP:345]

Format String: "Auth No: <AUTHNO>"

7.10.4.14   Reserved – NZ (EGM New PID Session Started)

Event Code: 0x3014 *(QPv1.6.3)*

7.10.4.15   Reserved – NZ (EGM PID Session Stopped)

Event Code: 0x3015 *(QPv1.6.3)*

7.11 QCOM Event Queue Summary

In QCOM, "Event queues" are circular buffers (the term 'event queues' is just a legacy term). This section summarises their operation.
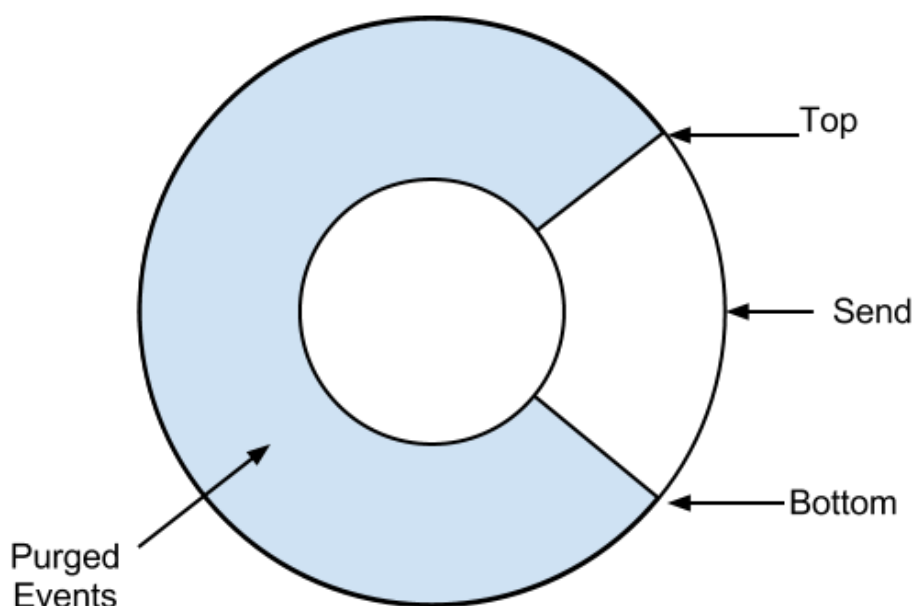
Figure 2

In the above diagram, all pointers (top, send, bottom) rotate in a clockwise direction.

"Top" buffer pointer:

New events go here, overwriting the next purged event (Top moves anti-clockwise in above diagram). **Top** must not pass **bottom**. If **top** reaches **bottom** the queue is deemed full.

"Send" buffer pointer:

Denotes the next event to send via the Event Response (15.6.9). **Send** moves by one event for each Event Response acknowlegement. **Send** chases top. On the primary event buffer, a Request all Logged Events Poll (15.4.14) resets **Send** to equal **Bottom**. On the secondary event buffer Send always equals Bottom.

"Bottom: buffer pointer:

Bottom chases **Top** but must not pass **Send**. Only a Purge Poll (15.4.15) for the primary event buffer, or a Event Response acknowledgement (14.2.1) for the secondary event buffer will move **Bottom**.

"Purged events" denotes events that have been "purged" and thus will be overwritten by new events over time.

8   EGM Defaults

8.1   EGM Defaults upon an EGM RAM clear

8.1.1   The EGM must not process or respond to polls from the SC.

8.1.2   The EGM must check the integrity of the event queue, meters [CP:346] and critical memory (refer GMNS) storage areas.

8.1.3   The EGM must require a serial number to be confirmed (3.1.2) and any other required manually entered parameters must be setup as well (3.1.1).

8.1.4   The EGM must not accept credit via any method (eg. Coins/notes, cashless in or other) until at least EGM configuration has been completed via the EGM Configuration Poll (15.4.2) [CP:347]

8.1.5   The EGM must be able to receive and process broadcast messages at all times, even before EGM configuration has been completed (as this is how an EGM receives its poll address once it is assigned a serial number).

8.1.6   The EGM must assume a default date and time of 01/01/2000 (QPv1.6. Was 1991 in QPv1.5) and 00:00:00, until an update is received by the SC. [CP:348]

8.1.7   The EGM must not log any QCOM events onto its QCOM event queues (refer section 7) until after the EGM's poll address configuration (via the broadcast Message 15.5.4) has been completed for the first time* [CP:349], at which time the NV RAM clear event must be logged [CP:350] and then the current state of all EGM doors and note stacker must be logged via events [CP:351].

    *I.e, the default date and time must never appear on events sent by the EGM and the RAM Clear Event will be the first event logged with an Event Sequence Number of 0x01. [CP:352]

8.1.8   The EGM must assume an initial event sequence number of 0x01. [CP:353]

8.1.9   The EGM must disable all games via their Game Enable Flag (15.1.3). This also applies to single game EGMs.

8.1.10   The EGM must default to disabled via the Machine Enable Flag (MEF), Site Enabled Flag (SEF) and all Game Enable Flags (GEF). Refer section 15.1.

8.1.11   The EGM must zero all counters, EGM group meters [CP:354], multi-game/variation meters [CP:355] and LP Turnover Meters. [CP:356]

8.1.12   The EGM must assume default denominations fields (DEN & TOK) of zero after a RAM clear until configured by the EGM Configuration Poll (15.4.2). [CP:357] Refer EGM Configuration Response (15.6.12).

8.1.13   Other EGM Configuration Poll/Response (15.6.12) field defaults are as follows (note, these figures are always overwritten by the SC upon configuration, the defaults are just for initial reporting purposes):

MAXDEN	0
MINRTP	0.00%
MAXRTP	99.99%
MAXSD	65535
MAXLINES	65535
MAXBET	4294967295
MAXNPWIN	4294967295

MAXPWIN	4294967295
MAXECT	$10,000

8.1.14 A LP or SAP game must assume a default Progressive Group ID number (PGID, Refer 15.1) of zero until configured. [CP:358] The default PGID number a non-progressive game will assume after a RAM clear is 0xFFFF. [CP:359]

8.1.15 The EGM must assume default Poll Sequence Numbers (PSN, Refer Section 15.1.9) of zero. [CP:360]. Ie. the first expected PSNs will be 0x01

8.1.16 The EGM must default to accept all Australian plastic banknote types until otherwise configured via the Note Acceptor Maintenance poll (0). [CP:361]

8.1.17 The EGM must assume EGM Parameter Poll (15.4.5) defaults of: [CP:362]

Parameter	Default Value
Reserve feature	1
Autoplay	0
CMLIMIT mode	0
Operator ID	0x00
Large Win Lockup Amount	$10,000
Credit Acceptance Lockout Limit	$100
Maximum Number of Double-Ups/Gambles	5
Double-Up/Gamble limit	$10,000
Time Zone Adjust	0
Power-save Time-out Value	900 (secs)
Player Information Display ID	0x00
End of Day time	180
NP Win Payout Threshold	$10,000
SAP Win Payout Threshold	$10,000

8.1.18 Hopper/Ticket Printer Maintenance Poll (15.4.17) defaults:

Parameter	Default Value
Hopper Collect Limit (COLLIM)	$50 for $1 token EGMs, otherwise the equivalent of 300 coins/tokens [CP:363]
Default Hopper Refill Amount [CP:364]	$160 for $1 token EGMs, otherwise the equivalent of 400 coins/tokens
Hopper level meter (refer 3.4.1) [CP:365]	$0 *(QPv1.6, was as above in QPv1.5)*
Ticket Out Limit (TICKET) [CP:366]	$0

8.1.19 A General Status Response (15.6.1) default state of 0x01 (Idle). [CP:367]

8.1.20 External Jackpot Information Poll (15.4.7) details of a RTP of zero and with zero jackpot levels. [CP:368] QPv1.6

8.2 EGM Defaults upon an EGM power up

The EGM must perform these items upon power up in the order listed below *(QPv1.6)*. [CP:369]

8.2.1 The EGM must check the integrity of the event queue, meters [CP:370] and critical memory (refer GMNS) storage areas.

8.2.2 Log the EGM Power Down event (7.10.3.11).

8.2.3   The EGM must detect and log any power off door accessed events and low battery events.

8.2.4   The EGM must check the state of all doors/note stacker at power up and log the applicable events where a change in state of the door/stacker has occurred since power down. CP:371

8.2.5   Log the EGM Power Up event (7.10.3.10).

8.2.6   The EGM must not resume responding to the SC upon a power up, until after all power up integrity checks have been completed. CP:372

8.2.7   (QPv1.6) A LP EGM must default to a LP Broadcast Timeout (6.1.8). CP:373

8.2.8   The EGM must default to a Ccommunications tTimeout condition CDC (6.1.2) and must also assume the Communication Defaults as listed in the section below.

Also be aware of section 3.6.2 regarding QCOM State Variable Volatility.

8.3   EGM Communication Defaults

The EGM must assume the following defaults upon power up and upon entry into a Communication Time-Out (6.1.2) or Communications Fault(QPv1.6) (6.1.3).

8.3.1   The EGM must default to disabled via the Machine Enable Flag (MEF) Refer 15.1.2. CP:374

8.3.2   The EGM must stop responding to polls to its current poll address . and consider its poll address as undesignated. The EGM must not resume responding to the SC until the EGM has been designated a new poll address again via the EGM Poll Address Configuration broadcast message (15.5.4). CP:375

8.3.3   As the EGM may be assigned a new poll address, any pre-built/pre-queued response/s must be discarded. Any pending group meters (either for transmission or awaiting acknowledgement) are set back to idle. CP:376 However, events have special handling in this regard, see below.

8.3.4   QPv1.6: If the EGM's last response in the session was an event response (15.6.9), then the EGM must assume a NAK for this event and it must ensure that the same event response is automatically queued up for transmission again in the next session and that all events that were pending transmission prior in the last session are still sent in the next session. (This will create a duplicate event in the SC in this case (but this is of no concern because consecutive duplicate events are currently automatically ignored by SCs). Overall, this is a benefit because it reduces any possibility of an event being lost between sessions.) CP:377

        QPv1.5: If the EGMs last response prior the power down or communications time-out was an event response (15.6.9), then to avoid sending the same event twice, the EGM must ensure it processes the ACK or NAK on the first received poll (refer 14.2.1) in the next session and apply it to the event response sent previously.

8.3.5   Any program hash calculation in progress must be aborted. CP:378

8.3.6   Reset the consecutive NAK counter to zero. (tThis is the internal EGM variable that the EGM uses to detect 3 consecutive NAKs - refer section 6.1.3. This should not affect the EGM's CDC display of communications fault (refer section 6.1.3); this CDC must be reset as per section 6.1.3. *Note, this implies that the EGM's NAK counter and "flag" variable denoting the display of the communications fault CDC need to be two separate variables*). CP:379

9 Multi-Game/Multi-Variation Support

A QCOM EGM must indicate as having at least one game available at all times. If an EGM is a multi-game EGM, then this means it has more than one game available and it also means there will be some sort of a game select screen used by a player to select a game for play. However, QCOM treats single game and multi-game EGMs exactly the same way in that the same messages are used for both types.

The current number of available games in the EGM and the maximum number of games that a player may select from (via the game selected screen) is indicated in the EGM Configuration Response (15.6.12).

New for QPv1.6: **Extended Multi-game Support**. A QPv1.6 EGM may theoretically have up to 65535 resident games but may only offer a maximum of 255 games at a time* based on EGM configuration settings (refer EGM Configuration Poll (15.4.2)). Initially after EGM RAM clear, the EGM will report a total number of games via the EGM Configuration Response (15.6.12). The EGM will not report Game Configuration Responses at this stage. After EGM configuration has been completed via the EGM Configuration Poll (15.4.2), the number of games reported via the EGM Configuration Response (15.6.12) may decrease as a result of the parameters in the EGM Configuration Poll making one or more games ineligibleillegal. However, the EGM will now provide details of each remaining available game via Game Configuration Response/s (15.6.11). Games made ineligibleillegal by EGM Configuration Poll settings must never be reported by the EGM via any response types [CP:380]. The EGM will also not accept an EGM Configuration that will result in all games being made illegalineligible; refer EGM Configuration Poll (15.4.2 second paragraph) for what events to log in this case. After EGM configuration is completed via the EGM Configuration Poll (15.4.2), the reported list and number of available games reported by the EGM, must not change unless the EGM is subsequently RAM cleared [CP:381]. **Any EGM manufacturer wishing to utilise this feature must first advise the OLGR to ensure that all concerned systems support this feature before proceeding.**

(A QPv1.5 EGM could only offer a static set of up to 255 games. All the games present would be static and valid for the jurisdiction).

(*EGM manufacturers must check with the monitoring system provider or regulator before proposing EGMs with more than 16 games that can be enabled at any time, as most monitoring systems have a limit of the maximum number of enabled games per EGM that they can support.)

Once a multi-game EGM is completely setup, typically a player is presented with some sort of game selection display as a part of idle mode, to allow them to choose which game they want to play next.

In a multi-game EGM, configured games are enabled / disabled by their corresponding Game Enable Flag (15.1.3). Disabled games (GEF=0) may still be shown on the game select screen, but if shown, they must be indicated as disabled (e.g. grey-out the game icon), however it may be better (reduces risk of confusion) if the games are hidden from the game select screen when the game's GEF = 0.

An un-configured game (re 15.4.3) must not be displayed on the game select screen in a multi-game EGM. [CP:382]

An example of how to handle a configured game with GEF=0 in a multi-game EGM, is for the EGM to simply "grey-out" the game on the game select screen, however the user can still select and enter the disabled game (to view last play, rules etc), but are simply not allowed to start a new play. The words "Game Disabled" would also be displayed inside the game.

9.1 Game Variation Support

The EGM must offer at least one variation for each game up to a maximum of 16 variations per game *(QPv1.6).*

Extended Variation Support *(QPv1.6)*: An EGM may have up to 99 resident variations for a game, but it may only offer for selection at any time a maximum of 16[(QPv1.6)*] variations per game. The EGM may decides which variations are offered for selection based on the current EGM configuration settings (refer EGM Configuration Poll (15.4.2)). [CP:383] I.e. the EGM will only offer variations applicable to the current EGM settings, such as max bet, denomination etc. The offered list of variations must not change once reported via the Game Configuration Response (15.6.11) unless the EGM is subsequently RAM cleared. **Any EGM manufacturer wishing to utilise this feature must first advise the OLGR to ensure that all concerned systems support this feature before proceeding.**

*QPv1.5 EGMs only offered a maximum of 8 variations per game.

Only one variation per game may be enabled for play at a time.

9.2 On-The-Fly (or Hot) Switching of Game Variations

The ability to switch from the currently enabled variation of a game to another variation of the same game without RAM clearing the EGM is supported by the QCOM protocol. Only one variation per game can ever be enabled for play at a time. Switching variations must be possible only via the EGM Game Configuration Change Poll (15.4.4). [CP:384] Refer to this polls description for more information.

Hot switching support is **mandatory** in all multi-variation games except progressive games. *(QPv1.6)*

A prerequisite before an EGM may offer a game with hot switchable variations, or as a game with multiple variations, is that the **static artwork** on the EGM must not have to the changed between variations.

If an EGM has on-the-fly switching of game variations, the EGM must maintain a set of multi-game/variation meters for each game variation. [CP:385] In all other cases there is only one set of multi-game/variation meters required per game in the EGM.

If the game has a progressive component, the progressive component (i.e. denoted by the percentage increment, start-up, ceiling, overflows, trigger probability and jackpot current amounts) must not change between game variations, otherwise the EGM must not offer that game to be hot-switched.

Reel strip and combination test modes, if offered, must be available for at least the currently selected variation, in addition the currently selected variation must be displayed within these test functions. E.g. "VAR: 99" [CP:386]

10   Progressive Game Support

This section applies to the EGM triggered SAP & LP EGM games. This section has nothing to do with system jackpots (or jackpots that are triggered externally by a device that is not an EGM).   System jackpots (or external jackpots) are facilitated in QCOM via the System Lockup Request Poll (15.4.9).

10.1 Progressives General

Progressive support is optional for an EGM manufacturer.

A game may or may not have a progressive component at the discretion of the EGM manufacturer.

A multi-game EGM may have games with and without progressive components.

QCOM progressive level percentage increment amounts in the EGM (i.e. PINC & AUXRTP as a %), must utilise no more than of 4 decimal places. In addition, there must be no rounding of this figure when transmitted, displayed or utilised (i.e. these values must be exact and no more than 4 decimal places). CP:387

Upon a jackpot win, any contribution amount not of a whole cent must be carried over to the next jackpot. This ensures that no partial contribution is lost over the course of a jackpot win.

Implementations of SAP & LP progressives in EGM or SC software must have no cumulative rounding errors with respect to their specific ~~theorectical~~theoretical auditing formulas.   E.g. the ~~theorectical~~theoretical progressive balancing formula in section 10.4 for a basic progressive implementation must always balance over time. CP:388 (Generally speaking there shouldn't be an issue if the progressive implementation in software avoids any form of 'division' operation and that precision is consistent in all calculations)

A SAP jackpot refers to a progressive jackpot level prize that is controlled and triggered by a single EGM.

A LP jackpot in this document refers to a jackpot level that is triggered by the EGM but where the prize amount (encompassing the start-up amount, increment and ceiling) is controlled by a SC. There is usually more than one EGM contributing and competing for the prize amount.

Each progressive level may be offered by the EGM as either SAP only, LP only *(QPv1.6)*, or either (refer 15.6.11). The EGM may also allow a game to be configured with some levels as SAP and others as LP. CP:389 *QPv1.5 EGMs if LP had to also be able to be setup as a SAP, but exemptions were granted on occasion.*

Once per EGM RAM clear, initial jackpot level contributions (including overflows if SAP) are setup by the SC before the EGM is enabled for play. This is required in case of an EGM RAM clear to restore last known current amounts or when required to carry over a jackpot amount from somewhere else, for example in the event of a decommissioned jackpot.

Turnover from the Residual Credit Removal Feature (RCRF) must not contribute to an EGM triggered progressive level amount or chance of winning. Typically the only time RCRF turnover would contribute to a jackpot is when the jackpot is an 'external jackpot' (i.e. not a part of the EGM's game).

10.2 Level Numbering

QCOM progressive level numbering convention. In QCOM, progressive level numbers must always be assigned in the following manner: level numbering must start at level 0 and it must correspond to the

progressive jackpot prize with the highest start-up amount (typically this is the least probable prize), level 1 as the next highest start-up and so on, for every defined progressive level in the game.[1] [2] CP:390

Level numbers are only used internally and must not be displayed to players, use level names instead.

QCOM currently supports a maximum of 8 progressive levels per game.

Also refer to Section 15.1.10 for an explanation of Progressive Group ID numbers (PGID).

10.3 Jackpot Displays

For games which contribute to a QCOM progressive, the EGM must automatically display all current progressive level amounts to the player. However, games designed with hidden progressive levels, or games designed for use with an external progressive display are exempt from having to have the EGM display the amounts. CP:391 For more general requirements on jackpot displays refer to the OLGR Jackpot System Minimum Requirements Document and the Jackpot Display System Requirements documents.

Displayed jackpot current amounts must be rounded down to the smallest display unit (typically cents, however 5c may also be acceptable in some cases).

For progressive games that are intended for use with large external jackpot display signage, it is recommended for redundancy purposes, that those games also have a small, discrete built in display of jackpot current amounts (display space permitting). This may mean that the EGMs can remain in play in the event of a failure of the primary external jackpot display.

---

[1] If this still isn't clear for a particular game (e.g. the game has levels with equal SUPs), then a sub-order which takes into account to the physical order in which the levels are displayed to a player is an acceptable alternative. In this case make level 0 is the highest (LHS) level as displayed (i.e. in altitude and horizontal alignment) and the remainder follow in order from left to right, top to bottom.
[2] FYO. Ordering by start-up amounts is assumed by the integrity check in section 15.4.6

10.4 Progressive Game Audit Mode Display Requirements

10.4.1   Progressive Meters per Jackpot Level per Game

The EGM must have the following additional meters and information in audit mode for each progressive level in a game: (Labelled as shown in bold below and units must be indicated. NB: GMNS also mandates ordering starting from 'Current Value' and this is reflected in the table below) CP:392

Each page of progressive meters must be titled with the applicable game name and GVN *(QPv1.6.4)* CP:393. This is because the Hits, Wins and Turnover progressive meters below must always be metered and displayed on per game basis; even in a shared progressive EGM (refer 10.9).

Label	Description / notes
Level Number	0…7 (remember 0 = Top/Highest prize)
Level Name	As used on artwork, eg. MAXI, GRAND, MINI, etc.
Type	Indicate either **LP** or **SAP** (or "**---**" or similar if not configured yet). Also append the word "**Shared**" if the level is a shared progressive (10.9). CP:394
Current Value ($.¢)	I.e. the amount as displayed on the jackpot display to participating players. For LP levels, this will be the last received amount for the level via the LP broadcast message. Refer 15.5.2.
Overflow ($.¢)	(SAPs only. Display "NA" for LP levels) Display overflow as the amount of contribution exceeding the ceiling amount eg. with no ceiling the current amount is $123.45 and with a ceiling of $100 the overflow amount to display is $23.45. The EGM must include hidden increment amounts in this value.
Hits (count)	Number of times the game has won this jackpot level. CP:395
Wins ($.¢)	Total amount of wins on this level for the game. CP:396
Start-Up Amount ($.¢)	Default start-up amount (SUP) of the jackpot level. Do not include any overflow or background increment here. (If the level is a LP, then before the Progressive Configuration Poll (15.4.6) poll has been received for the game, the EGM must display "???" for this field).
Ceiling Amount ($.¢)	This is the amount the jackpot displays actually freeze at when reached by the current amount. Any contributions in excess of the ceiling must go into the overflow meter. (If the level is a LP, then before the Progressive Configuration Poll (15.4.6) poll has been received for the game, the EGM must display "???" for this field).
Percentage Increment (%)	Percentage of turnover that contributes to the jackpot current amount. E.g. 1.5000% (four decimal places must be displayed) (If the level is a LP, then before the Progressive Configuration Poll (15.4.6) poll has been received for the game, the EGM must display "???" for this field).
Hidden Increment (%)	Refer GMNS. Equals AUXRTP. *NB most systems do not currently support LP totalisation including a hidden increment. EGM manufacturers should check before implementing a LP game with a hidden increment.*
Initial Contribution ($.¢)	Initial contribution amount received via the EGM Game Configuration Poll (15.4.3) (+/-$.¢) Note the value is in units of contribution, not turnover, i.e. the amount taken out by the percentage increment. Display "NA" if level is LP ***If the level is a ~~custom~~SAP level, ~~this value~~'Initial Contribution' must be updated by the EGM upon every accepted SAP level parameter change (this ~~ensures~~ the new SAP reconciliation formula still balances over a SAP configuration change~~is not broken~~) CP:397~~. See the notes below for more information.~~ Note also that while the contribution received via the EGM Game Configuration Poll (15.4.3) is an unsigned value, ~~'Initial Contribution'~~

	may essentially become **negative** as a result of a customSAP level parameter change. Therefore to prevent confusion, 'Initial Contribution' must be now be **displayed as a signed integer** in audit mode. [CP:398] (This is related to note 2 below)
Turnover ($.¢)	Total applicable game turnover towards the jackpot since last EGM RAM clear. Note the value is in units of turnover, not contribution. I.e. This must not include initial contribution above, or RCRF turnover.
~~**Hit Rate (p)** *(QPv1.6)*~~	~~Refer 15.6.3 HRATE field. Display in scientific notation (e.g. 4.0245e-10) with all significant digits. Units must be labelled "/ cent" or "/ ¢").~~
Expected RTP *(QPv1.6)*	Current Theoretical Percentage RTP of level (i.e. SUP $PRTP_{level}$ + % INC_{level} + AUXRTP). Units "%". (If the level is a LP, then before the Progressive Configuration Poll (15.4.6) has been received for the game, the EGM must display "???" for this field). If the expected RTP is in a range, then state two figures: minRTP%-maxRTP%.
…	*For non standard progressive levels, insert all additional information required in order to fully audit the progressive level here.*
Win Log *(QPv1.6)*	Date, time (TZADJ adjusted RTC timestamps) and the amount of at least the last five hits of this level for the game. [CP:399] The EGM manufacturer may store a longer history if desired. The EGM must include all the data in the LP/SAP award events in each entry of this log (refer sections 7.10.2.1 & 7.10.4.8). [CP:400]

Any other meters or information relevant to reconciliation of the current jackpot amounts must also be suitably displayed. (E.g. a next Start-Up amount percentage increment and contribution) [CP:401]

The level name and amount won in a progressive jackpot must also be displayed in the EGM's last play recall information. [CP:402]

FYI. To reconcile or self audit a standard or vanilla progressive jackpot level:

> Jackpot Current Displayed Amount + Overflow =
> (Total turnover to jackpot * Percent Increment)$^{Rounded\ Down}$ + ((Hits+1) * Reset Amount) + Initial Contribution - Total Wins

Issues with implementing the above formula in a QCOM EGM:

1. The formula may have to be adjusted by the EGM manufacturer to handle proprietary SAP features/calculations for a particular game.

2. A SAP parameter change (specifically any change in either Percent Increment or Reset Amount parameters, refer 15.4.6: customSAP) will cause the above audit formula to fail unless at least one term in the formula is adjusted in compensation of the change. This must be done as follows:[CP:403]

    To avoid a **self audit error** occurring after accepting a new set of SAP parameters in CustomSAP game, upon accepting the new SAP level parameters, the EGM must solve its SAP reconciliation audit formula for 'Initial Contribution' and save the result. Subsequent SAP self audits for the level should then balance with the new level parameters. [CP:404]

3. The formula will also fail upon any participating meter rollover e.g. the turnover meter. (As this is unlikely to occur in the life of a game operating in a live gaming venue, this concern does not need to be addressed at this time)

    However this issue may occur during fast-play game testing in which case there are several options that may be considered. For example: disable SAP reconciliation in test game software, or use 64 bit meters, or have the EGM automatically disable SAP reconciliation the first time the turnover meter crosses an upper threshold.

4. To minimise the risk of inadvertent self audit errors, reconciliation formulas like the above must be implemented in the EGM as a critical section. (Related 12.4)

5. The above formula is only for a vanilla progressive, for example it does not take into account a hidden increment (i.e. AUXRTP) or any other special features and the audit formula utilised may need to be adjusted accordingly. *(For example, to accommodate an AUXRTP implemented as a hidden increment which is added to the new reset amount after a jackpot hit, simply add/bracket in AUXRTP to the Percentage Increment term in the above formula)*

10.5 QCOM Stand Alone Progressive Jackpots

A SAP jackpot refers to a progressive jackpot level prize that is controlled by a single EGM and is winnable only on that EGM. The main reason SAPs are supported is to handle progressive features in a game which may not be able to be supported by the basic LP support that a Site Controller provides. SAPs can also be much more rapid than LPs (depending on the number of EGMs on the LP).

In QCOM, the requirements for handling SAPs are slightly different than for a LP level. This section details the requirements specific to SAPs.

When a SAP level is won, the EGM must log the SAP Award event (7.10.4.8) and prominently and constantly *(QPv1.6)* display the level name, the amount to be awarded (in $.¢) to the player for a sufficient period. [CP:405] E.g. "Mini Progressive Jackpot $12.34". Acceptance of the overall legibility of the award text message on the EGM display is at the discretion of the CEO [CP:406].

SAP wins must be automatically paid directly to the EGM's Credit Meter by the EGM and after a short delay, the SAP win must also be automatically cleared by the EGM with no manual intervention required (except if the total win amount (including the SAP win amount) at the end of the play exceeded the Large Win Lockup threshold (15.4.5), in which case a Large Win Lockup must result). [CP:407]

SAP win shows are acceptable and recommended. Note that there is no STATE in the General Status Response (15.6.1) to indicate a SAP win. The EGM must stay in either the 'In-play' state or the 'In-Play Feature' state for the duration of the SAP win unless it was a Large Win.

SAP wins must be added to group meters 0x02 and 0x0C$^{QPv1.6}$ by the EGM. [CP:408] Refer 12.1.1.

In EGM audit mode for the GMNS master meter display, SAP wins must be added to the total wins meter of the EGM and not to the total progressive wins meter on this display (if one exists – the QLD Appendix to GMNS that mandated this meter was abolished in 2013). [CP:409]

The EGM must not log a Lockup Cleared event for SAP wins. [CP:410]

In QPv1.5 EGMs, SAP wins were 'silent' and the following special requirements applied: [CP:411]

> *SAP wins were only added to group meter 0x02 (EGM total wins meter).*
> *SAP wins were not added to group meters 0x07, or 0x0C. However the EGM still sent group meter 0x07 (SAP wins) when group zero meters are requested, but its value was always zero (this overrode the requirement in section 12.1, that the EGM must never to send a meter that would always be zero).*
> *SAP wins were added only to the WINS meter in Multi-Game/Variation Meter response (15.6.6). (Same in QPv1.6)*
> *No state set in the General Status Response (15.6.1) (same in QPv1.6)*
> *EGMs did not log the SAP Award event (7.10.4.8).*
> *EGMs did not log the 'Lockup Cleared' event after the SAP win. (Same in QPv1.6)*

10.6 QCOM Linked Progressive Jackpots

LP jackpots in this document refer to EGM triggered, system controlled linked progressive jackpots. That is, the EGM triggers only the award (via a combination win, or mystery method) and the system (SC) maintains the jackpot current amounts by collecting the applicable turnover to the progressive, taking the percentage contribution towards the jackpot, applying the ceiling limit and broadcasting the current progressive amounts back to the EGMs. The LP EGMs have no knowledge of or concern for the jackpot parameters of startup, ceiling and percentage increment, as only the current jackpot amounts sent to the EGM via the LP broadcast message.

QCOM can support up to 65534 possible linked progressive groups and up to 8 possible progressive levels per group. A multi-game EGM may contribute to a different LP group for each game, or the same group for all games. The EGM must maintain a unique LP Turnover meter (refer 2.2) for each LP game regardless. It is acceptable for a QCOM LP game to trigger one or more LP jackpots per 'play' (def: 2.2), e.g. during a free game series. A QCOM LP game must not log a new LP jackpot award event until any current LP jackpot has been first acknowledged (via the LP Ack Poll 15.4.18) and the lockup reset (see 10.6.1 below). Games which could trigger two or more LP jackpots on a single game feature are allowed, however this requires special handling on the part of the EGM. For more information regarding this, refer to section 10.8 below.

In QCOM, the requirements for handling SAPs are slightly different than for a LP level. This section details the requirements specific to LPs.

EGM manufacturers should give some consideration to the hit rate before deciding to make a LP game as some jurisdictions place restrictions on how many EGMs can be on a LP in order to reduce the number of ~~simulataneous~~simultaneous wins. Also, some jackpot display systems that perform win shows can have difficulty in keeping up if the hit rate for a LP is too high. To avoid these problems, consider making the game a mixed LP/SAP game with the lower levels available as SAP.

10.6.1   LP Lockups

When a specific LP jackpot feature has been won, at the moment the jackpot level which has been won is revealed to the winning player (either directly or by inference) the EGM must immediately:

- log the LP Award Event* (7.10.2.1) [CP:412],
- prominently and constantly[QPv1.6] display the message "VERIFYING JACKPOT AMOUNT" [CP:413], and
- display the amount to be awarded (this must be displayed in units of $.¢), and
- indicate to the SC that it has entered the LP Award lockup condition by setting the LP Lockup state in the General Status Response [CP:414] (15.6.1).

Acceptance of the overall legibility of the text message on the EGM display is at the discretion of the CEO [CP:415].

It is stressed that once the winning player is aware of which level they have won (even if only by association, such as implied by a specific winning combination), the EGM must log the LP Award Event at that time. [CP:416] This is to reduce the potential of a simultaneous win claim and security reasons

The "VERIFYING JACKPOT AMOUNT" message must be displayed until receipt of the next Linked Progressive Award Acknowledged Poll (15.4.18) at which time this message must be removed [CP:417].

Linked progressive jackpots must not be able to be cleared or reset by any means (e.g. the EGM General Reset Poll (15.4.19) or manual key-switch) until they are first acknowledged via the Linked Progressive Award Acknowledged Poll (15.4.18). [CP:418]

The EGM must not display a 'you have won' message [CP:419], or the equivalent (**this includes a win show** [CP:420]) until after the Linked Progressive Award Acknowledged poll (15.4.18) has been received.

Once the LP award has been acknowledged and any win shows have completed, the EGM must prominently and constantly [QPv1.6] display the winning level's name, the amount to be awarded (in $.¢), until the lockup is reset. [CP:421] E.g. "Mini Progressive Jackpot $12.34". Do not display "call attendant" or the equivalent. Acceptance of the overall legibility of the award text message on the EGM display throughout the LP lockup process is at the discretion of the CEO [CP:422].

(QPv1.6) It is highly recommended that for an EGM performing a LP award win show, the EGM does not allow the LP Award lockup to be reset via the General Reset Poll (15.4.19) until the win show has been completed (as permitted under section 15.4.19). This is because for a LP level being auto-paid/reset by a SC, the SC typically spams ~~out~~ the General Reset Poll until the EGM reacts to it, potentially cutting short any LP award win show. In addition, the EGM should reserve a few more seconds post win show to allow the level ID and amount won to be adequately displayed (Note, faults that occur during a LP win show must still be able to be reset [CP:423]). Note however that the LP Ack Poll must never be ignored by the EGM that is in a LP lockup (refer 15.4.18) [CP:424].

The amount of the LP award that the EGM must state in the LP Award lockup and use in the LP Award Event, must be the last received amount for the level sent via the LP broadcast messages (15.5.2) immediately prior the logging of the LP Award Event. [CP:425]

Very shortly after an EGM logs the 'LP Award Event', the EGM will receive in addition to the "LP Ack. Poll', a LP broadcast for the winning level containing the new SUP current amount. While it is mandatory for the EGM to clearly display to the player the amount won for the duration of the LP lockup, internally, the EGM must not under any circumstances, ignore an applicable LP Broadcast at any time for subsequent display. While, it is acceptable for the duration of the LP lockup for the EGM to freeze the winning level's current amount on its LP displays, in background the EGM must still be processing applicable LP Broadcasts so that when the EGM is finally reset from the LP lockup, the EGM has the correct current amount ready to display. [CP:426]

In the Meter Group/Contribution Response (15.6.8), LP wins are added only to group meter: 0x08 (Total EGM LP Wins) upon exit from the lockup condition (refer 12.2).

In the Multi-game/Variation Response (15.6.6) LP wins are added only to the PWIN meter.

10.7 QCOM SAP/LP Requirements Comparison Table

In QCOM, the requirements for handling a SAP level are slightly different than for a LP level. This section summarises the differences in the requirements. Please read the previous sections for more information.

Requirement:	Type: SAP Protocol Version: QPv1.6	*QPv1.5*	LP All
Auto-pay & auto-clear progressive awards	Yes	*Yes*	No
Audit Mode – GMNS master meter display; QLD App: PROGRESSIVE meter. *(NB The QLD Appendix to GMNS was abolished in 2013 so this meter is not longer mandatory in QLD.)*	NA	*NA*	Add wins
Audit Mode – GMNS master meter display; WINS meter	Add wins	*Add wins*	NA
QCOM Group Meter 0x02 (EGM Total Wins Meter)	Add wins	*Add wins*	NA
QCOM Group Meter 0x07 (formerly the HPSAP Wins Meter)	NA	*NA*	NA
QCOM Group Meter 0x08 (LP Wins Meter)	NA	*NA*	Add wins
QCOM Group Meter 0x0C (SAP Wins Meter) *(New for QPv1.6.1)*	Add wins	*NA*	NA
Multi-Game Meters – WINS meter	Add wins	*Add wins*	NA
Multi-Game Meters – PROGR meter	NA	*NA*	Add wins

Lockup Condition & General Status Response – set STATE	NA	*NA*	Yes
Log Award Event	Yes	*No*	Yes
Log Lockup Cleared Event	No	*No*	Yes
EGM Controls/increments current jackpot values	Yes	*Yes*	No

10.8 Multiple LP Award Handling
(QPv1.6.4)

This section deals with games that may trigger multiple LP jackpots on a single game feature outcome. E.g. multiple jackpot hits as a result of a single spin of game reels.

As mentioned in section 10.6, a constraint exists in existing QCOM monitoring systems which requires QCOM LP games to instigate a separate LP lockup for each hit of a LP jackpot level, with the corresponding LP Award event (7.10.2.1) not to be logged until just prior its corresponding LP award lockup. I.e. the EGM must not log a new LP jackpot award event until any current LP jackpot has been first acknowledged (via the LP Ack Poll 15.4.18) and the LP lockup reset/cleared (refer 10.6.1 above). Also, each subsequent LP Award event must have a newer timestamp on the event than the last LP award event (else the system will ignore all but the first received LP Award Event if the time-stamps are all equal).

Accordingly, there are two options for games wishing to trigger multiple LP awards per play in order to satisfy existing QCOM system constraints:

Option 1:

The EGM snapshots all the won jackpot progressive meters at the instant they are indicated as won for subsequent reporting via LP Award events. The EGM then transmits the LP events as per the system constraint (see above) from lowest prize level up. (This order helps minimise the risk of simultaneous LP wins with respect to other EGMs on the link).

This method is best suited for jackpot levels which are auto-paid/cleared by the SC.

Pros/Cons of option 1:

+ the jackpot amounts at the time they are actually won is exactly what the player/s are going to be subsequently awarded in all cases.
- a slightly elevated risk of a simultaneous win overpay with respect to higher LP jackpot levels.
- It may appear to the SC that some wins are coming in very late with respect to the current jackpot amount, however this should be fine under current QCOM SCP LP requirements.

Option 2:

The EGM awards the hit jackpot levels from the highest prize level down as per the system constraint (see above) and it does not take a snap-shot of the jackpot current amount for the LP Award event until it actually comes time to log the corresponding LP award event.

Pros/Cons of option 2:

+ No increased risk of a simultaneous win overpay.
- the jackpot amount which is on display at the time game result indicated a win, may not be what the player is going to be subsequently awarded (i.e. in the case when another player wins the level in the interim).

If option 2 is chosen, then the behaviour of the game in the event of multiple LP award level hits must be stated in the game rules. E.g. "If multiple jackpots are won on a single <feature>, then they are awarded one at a time in sequence from the highest to lowest jackpot level. The amount won will be snapshot at the time the game locks up for that award level"

Games which can log multiple simultaneous LP awards for the same jackpot level must always implement it via option 2 in order to avoid jackpot overpayment. i.e. The first prize amount will be the metered amount and then subsequent amounts will all be the startup amount (assuming no overflow to be carried over).

10.9 Shared Progressives
(QPv1.6)

Scope: Applicable only to progressive multi-game EGMs.

Identification: Whether of not an EGM is a shared progressive is reported by the EGM via the EGM Configuration Response (15.6.12) FLGS field.

If an EGM is NOT a shared progressive, then this denotes that each progressive game's component in a multi-game machine is mutually exclusive. I.e. all progressive parameters and meters are ~~mutally~~mutually exclusive on a per game basis [CP:427] This means that playing games will only increment that game's progressive jackpots and not the progressive values of other games in the EGM. *(This has been the default behaviour in the previous version of QCOM (QPv1.5.x), however in practise it is decidedly more common that the progressive component will be shared across progressive games in a multi-game EGM. Accordingly this flag was created to allow the EGM to clarify the issue to the SC and behave accordingly)*

If an EGM is a shared progressive, then this denotes that the progressive component of any progressive game in a multi-game EGM, is identical across all other progressive games in the EGM. I.e. the jackpot amount for a given level is the same value for all progressive games in the EGM at all times.   This means that playing games will also increment all other resident progressive game progressive jackpots in the EGM.

The specifics of shared progressive implementation is as follows:

A shared progressive EGM must maintain and report the following progressive data fields with identical values for all other progressive games in the EGM at all times (i.e. one copy for all games):

- CAMT and HRATE fields (refer sections Progressive Meters Response 15.6.3), [CP:428]
- LP/SAP flag, SUP, PINC, CEIL, AUXRTP fields, refer Progressive Configuration Response 15.6.4 [CP:429]

Exceptions:

- the LP Turnover meter (PAMT, refer section 2.2) must always be maintained and sent, on a per game basis by the EGM [CP:430] and
- progressive level HITS, WINS & PWINS meters (refer sections 15.6.3 & 15.6.6), must always be maintained and sent, on a per-game-per-level basis by the EGM. [CP:431]

A shared progressive EGM must also automatically apply any progressive configuration or changes thereof to all other progressive games in the EGM (even if game configuration (15.4.3) has not yet been completed on those games). [CP:432] This means that when the SC configures or changes progressive configuration in one progressive game via any of the following polls:

- EGM Game Configuration Poll (15.4.3 PGID, PFLG & CAMT fields) [CP:433]
- EGM Game Configuration Change Poll (15.4.4 PGID field) [CP:434]
- Progressive Configuration Poll (15.4.6 SUP, PINC, CEIL, AUXRTP fields) [CP:435]

~~then~~ the SC does not have to also apply the configuration or change to any other progressive games in the EGM because the EGM will.

11   EGM Program Hashes

(This chapter was formerly called "EGM Program Signatures". The name was changed for correctness and to avoid confusion with cryptocrapically secure ~~the future use of~~ Digital Signatures)

In general cryptography terms, a QCOM Program Hash is the equivalent of a 'digital fingerprint' or a 'key dependent one way hash function'. They should not be confused with digital signatures used in cryptography.

11.1 General

11.1.1   Each EGM has one overall program hash value reported on requested via the Program Hash Response (15.6.13).

11.1.2   The EGM must initiate a Program Hash verification using a regulatory approved algorithm when requested to do so via the Program Hash Request Poll (15.4.8).

11.1.3   The EGM will extract and use the seed from the message truncating it as required for its designated hash algorithm.

11.1.4   When an EGM receives a Program Hash Request Poll, it must always restart a new hash calculation, even if it was already in the process of calculating a program hash [CP:436].

11.1.5   The EGM must abort a hash calculation if it is powered down during the process. [CP:437]

11.1.6   Program Hash Time-out.

An EGM must complete the Program Hash Calculation and be ready to transmit the Program Hash Response (15.6.13) as soon as possible and within a specified time-out period after receiving the request for the hash calculation. [CP:438] (The time-out period is specified by the OLGR and as of 10/2001 the period was set at 90 seconds but is subject to change. Contact the OLGR for the current value.)

Manufacturers should endeavour to make their program hash calculation on their EGM as quick as possible, because in most cases an EGM will be disabled by a SC until it returns a valid hash. A hash response time of no more than 30 seconds is preferable. Typically program hashes are usually requested when an EGM resumes' responding on the network, but it is also possible an EGM could be in play, in a lockup, a fault condition or any other state when it receives a program hash request.

11.1.7   The Program Hash Calculation should be invisible to a patron and the EGM must continue to respond to the SC at all times and still perform all functions. Visual degradation in EGM game performance is acceptable, but all other operations and security functions of the EGM must be in no way affected [CP:439].

11.1.8   In some circumstances, if an EGM is still having difficulty making a hash within the program hash time-out, it is acceptable to temporarily suspend non-critical processes in the EGM (~~eg.~~e.g. idle mode, play, hopper pay out, etc) for a short period, provided an appropriate message is displayed [CP:440]. E.g. "PLEASE WAIT – Calculating Hash". Consideration could also be given to a progress indicator in this case.

11.1.9   Data to be included in the program hash calculation.

This is now a separate OLGR technical standard "Hashing Algorithms" (refer 1.1). [CP:441]

11.1.10   To expedite a hash result, the EGM may calculate program hash in any desired ROM/byte/bit order and over separate CPU controlled peripheral or sub-systems in parallel. Multiple hash results over sub-systems must be combined to give a single final

result via modulo 2 addition (i.e. XOR) [CP:442] . E.g. BIOS ^ OS ^ EGM_Base_App ^ Game A ^ Game B … etc = final hash (where '^' = XOR operator)

11.2 Program Hash Algorithms

The QCOM protocol has support for multiple Program Hash Algorithms in that the protocol has no knowledge of the actual hash algorithm being used as it only deals with seeds and hash results.

QCOM *(QPv1.6)* can now support all hash algorithms types up to 160 bits maximum. *(QPv1.5 was limited to a maximum of 64 bits hashes)* The specific algorithm the EGM must use is defined in a separate standard (Refer section 1.1). [CP:443]

A list of acceptable Program Hash Algorithms are available from OLGR as a separate document (refer 1.1).

12 Meters

12.1 EGM Group Meters

An EGM has one overall set of EGM meters referred to as "group meters". They are called this simply because they are arranged into arbitrary groups for reporting purposes via QCOM. Each group meter is assigned a unique 2 digit group ID number in hex. The most significant digit of the group ID number denotes what group the meter is in. Group meters are reported by the EGM via the Meter group/contribution response (15.6.8). See this response type for more information.

All meters are in units of cents unless defined otherwise.

12.1.1   Group 0 Meters

Group Meter ID 0x	Label & Description	Applicability (QPv & H/W)	Updated. (Refer 12.2.2)
00	**Total EGM Stroke (S)** [CP:444] [CP:445] *A count of total games played, not including free spins or free games. RCRF gamble attempts must not increment this meter* [CP:446]*.*	All	1
01	**Total EGM Turnover (T)** *A total of all bets made on the EGM* [CP:447]*, includes RCRF turnover.* [CP:448]	All	1,8
02	**Total EGM Wins (W)** [CP:449] *Total wins paid to the EGM's credit meter. This meter includes all SAP & RCRF wins, but excludes all LP wins.*	All	2 & 8 (on RCRF win)
03	**Total EGM Cancel Credit (CC)** [CP:450] *Excludes cash ticket out.* [CP:451] *Note the GMNS Cancel Credit meter includes Cash Ticket out* [CP:452]*, refer Appendix B.*	All	3
04	**Total EGM Cash Ticket Out* (CTO)** [CP:453] *Refer Section 22.1*	TO	3
05	**Total EGM Cents In (CI)** [CP:454] *Includes Coins/Tokens, Notes and Cashless In.* *Excludes Cash Ticket In* [CP:455]*.*	All	4,7
06	**Total EGM Cents Out (CO)** [CP:456] *Coins/Tokens and Cashless Out* *Not including Total EGM Cancelled or Cash Ticket out* [CP:457] *and extra coins paid re hopper runaways/overpays.* *Note the GMNS Money Out meter includes Cash Ticket out* [CP:458] *and Cancel Credit; refer Appendix B.*	All	6,3
07	**Reserved*** [CP:459] *Formerly the* **Total EGM HP SAP Wins** *Meter. Intended to be a total of all hand paid SAP wins on the EGM but* ***was never used (i.e. incremented by SAP wins). If sent by any EGM this meter was always reported as zero.*** *This meter was made obsolete in QPv1.6.1. (i.e. must not be sent) and replaced by 0x0C below for reasons of backward compatibility in existing systems regarding how gaming tax is calculated (NB in QPv1.6 EGMs, all SAP Wins are also added to group meter 0x02).* *(In QPv1.5 EGMs, this meter was never used either, as SAP wins were hidden to allow SAP wins to be auto-paid/cleared by the EGM; i.e. SAP wins were paid directly to the credit meter and only added only to the total wins meter above. This meter if sent in a QPv1.5 EGM would always be zero).*		
08	**Total EGM Linked Progressive Wins* (LP)** [CP:460]	LP	3

Group Meter ID 0x	Label & Description	Applicability (QPv & H/W)	Updated. (Refer 12.2.2)
	Total of all LP wins on the EGM.		
09	**Total EGM Cash Ticket In* (CTI)** [CP:461] *Refer Section 22.2*	*QPv1.6,* TI	7
0A	**Total EGM PID Accessed*** [CP:462] *Total times the PID has been accessed on the EGM. NB: To meter this value correctly, the methodology in order to access the PID must be a unique action on the EGM and for example, it must not be necessary to pass through the PID in order to access any other information or function on the EGM, such as game rules display for example (otherwise an inflated PID Accessed Meter would result).*	*QPv1.6*	Upon PID entry
0B	**Total EGM Games Won (GWON)** [CP:463] *Total number of games won, incremented at the end of each play if the play (including all features and gambles) results in a win. (This would be incremented at the same time as the 0x02 meter)* *Excludes RCRF wins[CP:464] and winning games whose wins were completely lost from subsequent double-up/gamble.*	*QPv1.6*	2
0C	**Total EGM SAP Wins* (SAP)** [CP:465] *Total of all SAP wins on the EGM. New for QPv1.6.*	*QPv1.6,* SAP	3
0D	**Reserved *** *(do not send) (NB: NZ QCOM EGMs add credit received from a 3rd party jackpot protocol to this meter ID as well as meter ID 0x14)*		
0E…0F	**Reserved*** *(do not send)*		

Note the following properties regarding group 0 meters: [CP:466]

- The EGM current credit meter amount = W – T – CC + CI – CO – CTO + CTI
- The total EGM percentage return (RTP) = (W + LP) / T
- The RTP of the non-progressive component = (W – SAP) / T *(QPv1.6 only)*
- The percentage return of the LP component (LPRTP) = LP / T
- The percentage return of the SAP component (SAPRTP) = SAP / T
- The base game percentage return (i.e. excludes progressive component) BRTP = (W – SAP) / T
- Total wins (includes SAP wins) $W = sum^{ii}(WIN_{game\ 1}…WIN_{game\ n}) + RCRW$
- Total EGM Turnover $T = sum^{ii}(TURN_{game\ 1}…TURN_{game\ n}) + RCRT$
- Total Linked Progressive Wins $LP = sum^{ii}(PWIN_{game\ 1}…PWIN_{game\ n})$
- Taxable Metered Win (TMW), FYI only, this formula is QLD specific :

  TMW = T – W – LP + (Wide Area LP Wins[i]) – (Wide Area Jackpot Contributions[i])

([i] Not defined in this document. Contact the OLGR for more information on TMW)
([ii] From Multi-Game Meter Responses (15.6.6))

12.1.2 Group 1 Meters

Group Meter ID 0x	Label & Description	Applicability (QPv & H/W)	Updated. (Refer 12.2.2)
10	**Total EGM Hopper Refills* (R)** [CP:467] *Total amount of Hopper Refills recorded by an attendant (3.4.2)*	Hopper	On the event
11	**Total EGM Coins/Tokens in* (COI)** *(excl. refills)* [CP:468]	CV	4
12	**Total EGM Coins/Tokens out* (COO)** [CP:469] *I.e. hopper collects, excl. extra coins paid re hopper runaways/overpays.*	Hopper	6
13	**Total EGM Coins/Tokens to Cash box* (CB)** [CP:470] *I.e. in cents, excluding notes in.*	CV	5
14	**Total EGM Cashless Credit In (CSI)** [CP:471]	All	7
15	**Total EGM Cashless Credit Out (CSO)** [CP:472] *Not including cashless out failed/cancelled*	All	3
16	**Total EGM Note Acceptor Cents In (NI)*** [CP:473] *Not including Cash Ticket In (this is metered separately)* [CP:474]	Note Acc.	4
17	**Total EGM Notes In Count (NIC)*** [CP:475] *Not including Cash Ticket In count* [CP:476]	Note Acc.	4
18	**Total Residual Credit Removal Turnover* (RCRT)** [CP:477]	Hopper	8
19	**Total Residual Credit Removal Wins* (RCRW)** [CP:478]	Hopper	8
1A	**Total Rejected Enabled Notes*** [CP:479] *'Rejected' means that the NA tried to validate a note against its current note set and no acceptable match was found. Counted only while acceptor was enabled.* *Do not count notes that authenticate but are rejected because they are disabled. This meter is for tracking note acceptor performance.* *Include Cash Ticket In Rejects here (as sometimes the acceptor wont know if it was a banknote or a TI anyway). However do not include TI negative Acknowledgements (re 15.4.11) or TI timeouts* [CP:480]	*QPv1.6,* Note Acc.	Upon the event
1B	**Total EGM Gamble Turnover (GT)** [CP:481] *Incremented by the amount bet from every gamble attempt. (I.e. this meter may be incremented multiple times per play)*	*QPv1.6,* All	8
1C	**Total EGM Gamble Wins (GW)** [CP:482] *Incremented by each win (if any) from every gamble attempt. (I.e. this meter may be incremented multiple times per play)*	*QPv1.6,* All	8
1D	**Total EGM Coins/Tokens Cleared.*** *Total Value of all coins cleared from the EGM's Cash Box since the last RAM Clear of the EGM. This field is a cumulative field updated by the amount cleared each time a Cash Box clearance is recorded by an attendant.*	*QPv 1.6.2* CV	Upon the event
1E	**Total EGM Notes Cleared.*** *Total Value of all notes cleared from the EGM's Notes Stacker since the last RAM Clear of the EGM. This field is a cumulative field updated by the amount cleared each time a Note Stacker clearance is recorded by an attendant.* *Do not include Cash Ticket In amounts on this meter* [CP:483]	*QPv 1.6.2* Note Acc.	Upon the event
1F	**Reserved*** *(do not send)*		

Note the following properties of group 1 meters:

- Total cents in   CI   = COI + CSI + NI
- Total cents out CO   = COO + CSO
- RCRW / RCRT approaches MINRTP / 10000 (15.4.2)
- GW / GT approaches 100% = Gamble / double up theoretical RTP

12.1.3 Group 2 Meters

Note acceptor meters:

Group Meter ID 0x	Label & Description	Applicability (QPv & H/W)	Updated. (Refer 12.2.2)
20	**Total EGM $5 Notes In** * CP:484 *(count)*	Note Acc.	4
21	**Total EGM $10 Notes In** * CP:485 *(count)*	Note Acc.	4
22	**Total EGM $20 Notes In** * CP:486 *(count)*	Note Acc.	4
23	**Total EGM $50 Notes In** * CP:487 *(count)*	Note Acc.	4
24	**Total EGM $100 Notes In** * CP:488 *(count)*	Note Acc.	4
25…2F	Reserved (do not send)*		

Meter groups 3-15 are reserved for future expansion.

* "Reserved", or meters which are 'not applicable' (see below), must not be transmitted by the EGM.

A meter is 'not applicable' if its associated peripheral device is not installed (e.g. Group 2 meters and other meters concerning note acceptance are 'not applicable' if the EGM does not have a note acceptor installed), or if a related software feature is not present (E.g. the "Total EGM Linked Progressive Wins meter is 'not applicable' if the EGM has no LP games). The requirement is that a meter must not be sent by the EGM if it would remain at zero regardless of any possible operation performed on the EGM (excluding RAM clears) CP:489. SAP & LP wins meters must not be sent until at least one game is setup a SAP or LP respectively CP:490. The PID Accessed meter must start to be included in applicable group meters responses no later than on the first PID access, but not before an implemented PID display has first been enabled on the EGM (via 15.4.5). (Basically the EGM should send the PID Accessed meter if it contains a non-zero value, or if it has ever been sent previously since last EGM RAM clear with respect to the previous sentence) CP:491

All meters listed above without an * next to them must be transmitted under all EGM configurations.

Once a group meter has been transmitted by the EGM in any Meter Group response (15.6.8), it must be always subsequently retransmitted when its group is requested via the EGM General Maintenance Poll (15.4.13) or when its value is changed by the EGM CP:492. This occurs until the next EGM RAM clear where the hardware options may change which may make some meters no longer applicable.

12.2 Meter Updating

12.2.1   As a general rule, all meters in this protocol must be updated by addition upon the occurrence of the event and never incremented (e.g. during a win show). [CP:493] However meters reflecting things like a hopper pay out, or coin/token input for example, are exceptions to this rule.

12.2.2   List of QCOM meter update times (refer 12.1) i.e. when a group meter response would be queued unsolicited:

1. Start of a new play.
2. Confirmed winnings (i.e. after gamble feature or double up/gamble or winning RCRF.) *If a large win lockup occurs the meters are updated and flagged for transmission upon lockup entry.*
3. Upon exit from the associated Lockup Condition.
4. Upon detection of a valid coin/note insertion.
5. Upon detected validated coin to cash-box.
6. Coin paid out from hopper.
7. Upon successful valid ticket insertion or cashless transfer. (Refer applicable chapter for more details)
8. Upon RCRF gamble, or play gamble being irreversibly instigated by a player.

Examples:

The following EGM group meters are updated and flagged for transmission* (via the Meter Group/Contribution response (15.6.8)) at the following times:

- Stroke, Turnover and LP Turnover meters upon play commencement [CP:494].
- Win meters and games-won meters upon confirmed winnings (i.e. after gamble feature or double up/gamble [CP:495]. If a large win lockup occurs the win meter is updated and flagged upon lockup entry [CP:496]).
- Cashless out, Cancel credit and Progressive win meters upon exit from the lockup condition [CP:497].
- Turnover [CP:498], RCRT meters upon a RCR which results in a loss. [CP:499]
- Turnover [CP:500], Wins, RCRT and RCRW meters upon a RCR which results in a win. [CP:501]
- GT upon each and every gamble attempt by a player resulting in a loss. [CP:502]
- GT and GW upon each and every gamble attempt by a player resulting in a win. [CP:503]
- CI, COI and CB meters upon a single coin/token in that also diverts to the cashbox. [CP:504]
- CI and CSI meters upon a successful cashless in. [CP:505]
- CO and CSO meters upon a successful exit from the ECT-from-EGM lockup. [CP:506]
- CI, NI, NIC and one of 0x20…0x24 meters upon a single note accepted. [CP:507]

All other meters must be updated and flagged for transmission precisely upon the occurrence of the event [CP:508].

Note; slow poll cycle times or communications issues will cause multiple meter updates to accumulate and combine into a subsequent single group meter response [CP:509] (space permitting).

Some additional examples of when to flag for transmission some of the other group meters:

- The total cents out and total coins out meters upon **every** coin paid. E.g. For a hopper collect of 100 coins taking ~10 seconds, a group meter response would be seen every second response containing the cents out and coin out meters with ~10 coins worth of increment in each response for the duration of the hopper collect.
- The total cents in and total coin in and cash box (if diverted to cash box) meters upon every coin in.

- Total RCR & EGM turnover meters and RCR & EGM win meters (if the RCRF was a win) upon a RCR gamble (i.e. all together), then followed with the coin out meters as the hopper pays the coin.

* Where a number of meters are updated at the same point in the EGM software (e.g. stroke and turnover are updated at the commencement of a play), then it is required that the meters are all updated first and only then flagged/queued for transmission as a critical section of code. [CP:510] This critical section must ensure that only one meter response will result in all cases provided the number of meters in the response doesn't exceed 8; in this case a subsequent response would also be sent. This is as opposed to updating a meter, flagging it for transmission, then moving onto the next meter. This later method (depending on poll timings and the EGM's QCOM implementation, e.g. when there is a separate communications thread with respect to the thread updating meters) can occasionally result in separate meter responses which is not acceptable. E.g. if the EGM ever sent a meter group response containing the stroke meter but with no turnover meter, then this does not meet the requirements here.

12.3 Meter Rollover

12.3.1   Meters in section 12.1 of this protocol must rollover after their highest possible value has been reached. Specifically, 2 to the power of 32 minus 1. (4,294,967,295) [CP:511]. This mandates 32 bit unsigned binary (integer) meters or multiples thereof (e.g. 64 bit meters) must be used by the EGM.

EGMs that use 64 bit meters (or higher) are acceptable provided all QCOM meters transmitted in all QCOM packets still rollover as per this specification and that all self audit and other formulas still function correctly as intended. Master meters in audit mode may appear as their native 64 bit meters, however all meters and number type parameters appearing on QCOM specific pages in audit mode must reflect what is being transmitted and received via QCOM. All QCOM 32 bit number fields must still function exactly as per this specification. Double all conversions for possible side effects. (For example in a QCOM machine there may exist a few places in code that compare a QCOM related value to -1; areas like this may need special consideration).

12.3.2   Except for a meter roll-over, under no circumstances, must the EGM ever transmit a cumulative meter in any response which is less than any previous transmitted value of that meter [CP:512].

12.3.3   Credit Meter Rollover [CP:513]

An EGM must never permit (within reason*) its credit meter to rollover via any combination events that increment the EGM's credit meter (e.g. ECT, TITO, coin insertion, banknote insertion, *including game wins).

Since credit meter rollover would never occur in normal operation, the concern here is that any ability to roll over a credit meter may create opportunities for exploit in an EGM or connected system.

Accordingly, all EGMs must implement a ~~soft~~ credit meter limit (if it doesn't already do so ~~regarding~~ e.g. credit meter display capability limiting) which is sufficiently lower than the credit meter rollover threshold with respect to any resident game's likely# large win scenarios. In applying the limit, EGM must disable physical credit insertion devices and in the case of ECT and TITO utilise the transaction denial option.

#Please ensure the limit is not set too low in that it will cause issues or need review in the expected operating life / environment of the EGM (consider worse case games and environments e.g. high-roller gaming environments).

12.4 Self Audit

EGMs must perform the following self audits of QCOM meters at the following times: [CP:514]

- the start and end of every play,
- at least once every 10 seconds while in idle mode *(QPv1.6)* and
- upon every power up:

$$CI \quad == (COI + NI + CSI)$$
$$CO \quad == (COO + CSO)$$
$$\text{Credit Meter} \quad == (CI + W - T - CC- CO + CTI - CTO) \text{ and}$$

Refer section 12.1 above for legend.

If an EGM has SAP levels then they must also be included in the self audit. The formula for self auditing a standard progressive is located in section 10.4.1 [CP:515] *QPv1.6.4*

The above formulas will automatically handle any meter roll-over if unsigned integer binary arithmetic is used.

Care must be taken to ensure the meters are in a reconcilable condition when the audit is performed. Additional meters may be required in the audit formulas to ensure reconciliation at all times (e.g. EGMs which allow credit input during play or when the EGM has a cashless transfer to credit meter queued [CP:516]).

To minimise the risk of inadvertent self audit errors, reconciliation formulas like the above must be implemented in the EGM as a critical section (see glossary).

Failure of the above formula to reconcile, if not correctable via meter backups, ~~will~~ must immediately result in a RAM error on the EGM (Refer Section 3.2). [CP:517]

EGM self imposed sanity checks on credit meter movements are encouraged. However please avoid credit meter threshold based sanity checks such as "if the credit meter is greater than x then RAM Error", as this potentially creates an arbitrary way of forcing the EGM into a RAM error by running up the credit meter (for example via QCOM ECT). [CP:518]

13 Physical Layer

13.1 Physical Interface

This is a separate technical requirements document. Please contact OLGR for a copy, or see the OLGR web pages. Note however, that the QCOM protocol is not dependant on a specific physical interface.
CP:519

13.2 UART Usage

13.2.1 Flow Control

There is no hardware or software flow control such as RTS/CTS or XON/XOFF. Therefore in order to achieve the required message and character frame timing, the EGM UART service interrupt or process may need to be set at a higher priority.

13.2.2 Message Synchronisation.

In QCOM, SC message/poll synchronisation/detection is achieved via a technique utilising the parity bit. In SC messages, the parity bit of each byte no longer denotes the byte's parity, but is used to indicate the start of new messages. In this mode, the parity bit is often referred to as the "wake-up bit" or "multi-point mode". Many UARTs have this programmable parity capability built-in. Protocols like QCOM which use this technique are often referred to a 9 bit protocols.

For SC messages/ (or "polls"), the parity bit is used in the following manner: when the parity bit is set, this denotes the start (i.e. the first byte) of a new message from the SC. In QCOM, the parity bit (or wake-up bit), is only ever set by the SC on the first byte (i.e. the address byte) of a message. The remainder of the poll message data, and as well as all EGM responses, must have parity forced to zero.

EGMs should utilise the multi-point mode of their UART, SCC or ACIA to efficiently detect the start of a new message on the LAN. If the EGM's UART does not have hardware support for multi-point mode, then this may still be done in software.

For EGM's with UARTs without wake-up bit capability, but with forced parity capability, the EGM can use the parity error indicator bit of the UART to also efficiently detect an address byte.

For EGM's with UARTs without even forced parity capability (i.e. they only support odd/even parity) must calculate the expected parity on the fly order to implement message synchronisation. This is harder, especially when transmitting (as parity has to be changed during transmission) and requires more CPU time and skill to implement. QCOM EGMs should avoid using UARTs with only odd/even parity capability.

Third party QCOM sniffing applications trying to find EGM response messages must first synchronise on SC messages as the SC it the only device setting the wake up bit which distinguishes between a poll and a response. Once a poll is identified Tthen using 14.1.5 and other requirements in section 14.1 the application can find a particular EGM's reaponse as it knows that the if next message has the same poll address, then it is a response to that poll. (Note, as of QPv1.6.6 there is now lamapoll / lamaresp flags (refer sections 14.2 & 14.3) which makes message synchronisation possible without having to use the parity / wake-up bit).

For QCOM devices using Windows OS and trying to detect SC messages/polls, the Windows API function of interest is *DeviceIoControl* using the *IOCTL_SERIAL_LSRMST_INSERT* parameter option. This mode of operation inserts parity errors directly into the received serial stream allowing for easy detection of wake-up bits associated with specific bytes.

EGM Requirements:

The ability to detect wake-up bits on all received bytes (indicating the start of a new message from a SC).

When an EGM transmits a response, it must always transmit with the parity bit forced zero [CP:520].

If during receipt of a message, an EGM receives a byte with the parity set (i.e. indicating a new message), the EGM must abort receipt of the current message, and process the byte with the parity set as the first byte of a new message [CP:521].

13.2.3 EGM CPU Loading Considerations

In QCOM, it is ~~strongly~~ recommended that a UART is used which has "wake-up bit" capability built in, as this reduces the processor load ~~(down to less than ~70 characters/second, even on a congested channel)~~.

UARTs with only forced parity capability are also perfectly acceptable, their only draw back ~~being~~ is that every received byte on the communication channel has to be interrogated by the CPU ~~(up to ~1500 characters/second in a congested multi-point EGM configuration)~~ which requires significantly more processor time compared with a UART with wake-up bit capability.

UARTs without either wake-up bit or forced parity capability can also work using a parity look up table. However to transmit, parity must be changed during transmission, which can be difficult, as with some UARTs, parity changes take effect immediately and the processing load is higher.

Another way to significantly reduce communication processing load, is to use the UART in FIFO mode, if present. This can further reduce the total amount of interrupts the EGM CPU has to process by up to 16 times in some UARTS. A UART with a variable FIFO trigger level is ~~strongly~~ recommended.

The ideal UART to use is one which has wake-up bit and FIFO buffer capability. ~~UARTs with automatic CRC generation are also an additional advantage.~~

13.2.4 Low Level Communication Errors

Framing errors on received bytes must simply be ignored or turned off by the EGM software, as the message CRC is the validating methodology [CP:522].

UART overrun errors indicate the EGM is not processing received UART characters quickly enough. Manufacturers of EGMs experiencing overrun errors could try making the UART interrupt or process a higher priority [CP:523].

13.2.5 Baud Rate Tolerance

The EGM's UART must communicate at 19200 baud with a 1% tolerance (unless borderline, an un-calibrated measurement is acceptable) [CP:524]. Ideally, the UART's clock should be sourced from a crystal oscillator (or the equivalent). Synthesising the UART clock for example from the EGM's CPU, is not recommended.

13.3 8 Bit Mode

QPV1.6.7 : 8-bit mode is being introduced in order to allow QCOM v1 EGMs to be more easily integrated into other networks types which do not natively support 9-bit protocols (such as QCOM). 8-bit mode will allow supporting QCOM v1 EGMs to be used with off-the-shelf RS-232 to Ethernet converters. (These devices usually support parity but not 9-bit protocols where the parity is programmed.)

Minimal changes to existing EGM QCOM implementations is required in order to implement QCOM 8 bit mode.

13.3.1 Implementation

Implementing 8-bit mode in an existing QCOM EGM should only require a few lines of code to be changed/added. (New developers implementing QCOM in EGMs should implement QCOM in its native 9-bit mode then come back here after.)

When the EGM is put into 8-bit mode:

- the EGM must ignore parity on received bytes.
- Once the EGM finished receipt on any current incoming poll message (if any in progress), then instead of then waiting for a byte with a wake-up-bit set (13.2.2), the EGM must assume that the very next byte received is a new poll message and receive it and following bytes as a poll message. Once the message is received the EGM must immediately assume the next received byte is a new poll message again.

Done. EGM manufacturers should not change anything else in the EGM's QCOM module. I.e. The EGM still transmits with parity forced zero as before and expect a parity bit on all rx bytes as it did before (it just ignores parity now where as before it checked it for a possible new poll message marker).

In more technical detail; the primary line of code to be changed in the EGM is the line/s of code equivalent to:

`If a rx byte's parity equals one; then a new poll message is incoming.`

Update these lines of code to a new conditional statement along the lines:

```
If 8 bit mode is off, then the original conditional statement executes.
If 8-bit mode is on, then the EGM assumes the byte received is the start of a new
poll message (provided the receive state machine was not already receiving a
message).
```

As an example implementation, in QSIM EGM MODE source code all that was required were the following changes:

The original lines of code equivalent to:

>   `if (byte.parity EQUALS 1) then …new poll message`

were replaced with:

>   `if ((8bitMode AND dll.idle) OR (!8bitMode AND (byte.parity EQUALS 1))) then`
>   `…new poll message`

The above variable: dll.idle denotes that the QCOM low level rx state engine was idle (i.e. not currently in the process of receiving any part of a poll message. To facilitate this variable in QSIM*, the following line of code was added immediately after where any poll message CRC was checked:

>   `If (8bitmode) then dll.idle = true;`

(*Many QCOM EGMs may already have the equivalent variable and not require any change to create the equivalent of the dll.idle variable. QSIM source code did because it used to let the wake-up-bit reset the rx state machine back to default)

It shouldn't really be any more complicated than that to implement QCOM 8-bit mode in an EGM's low level QCOM message handler.

13.3.2 Poll Message Synchronisation

QCOM 8-bit mode in an EGM must have no supporting poll message synchronisation code. EGMs must not care if for example some rx garbage bytes cause it to get out-sync and start processing bytes all over the place as a new poll message. The QCOM message CRC will prevent any garbage messages from being processed as a poll.

If a message synchronisation issue does occur, it has been found that the EGM will tend to get back in-sync after only a few seconds worth of poll cycles with no other help. Also, Site Controllers can manually force a re-sync on demand by sending out the following byte string: 255, 254, 253,…,0. It is recommended any SC utilising 8-bit mode, send this re-sync message on any poll cycle it sees any non-QCOM related bytes (i.e. garbage) or it in itself injects any non-QCOM related data onto to the QCOM LAN (i.e. some SC talk other protocol in between QCOM poll cycles.)

13.3.3   Turning on/off

The EGM must provide a new checkbox on the serial number setup function screen (3.1.2) labelled "8-bit Mode". 8-bit mode must default to off (unchecked) in a RAM cleared EGM.

Elsewhere in EGM audit mode on a suitable QCOM related page, the same checkbox must appear allowing the user to change modes again if desired.

13.3.4   Indication

If 8-bit mode is enabled on an EGM, then the EGM must display the append the figure 8 on the EGM's CDC display "Play Disabled" text string anytime a CDC is being displayed (Refer 6.2). E.g. "Play disabled: S8".

QCOM EGMs which support 8-bit mode must set CNTL:bit 1 in all responses. Refer 14.3.1.

14 Data Link Layer

14.1 Polling Cycle and Responses

QCOM may be categorised as a poll - response protocol. However, in order to minimise network response times and processor load, QCOM mandates a special poll-response methodology system with strict timing. This means that the most obvious implementation of a poll-response protocol in an EGM will **not** meet QCOM requirements. Accordingly, please pay particular attention to the requirements in this section.

In QCOM, messages sent by the SC are called "polls" and messages sent by the EGM are called "responses".

In QCOM, a **"Poll Cycle"** comprises of a single message or 'poll' sent to each EGM in turn by the SC, after each poll to an EGM the SC will wait for a response from the EGM for a short period before moving on to the next EGM. (An EGM will always try to respond to valid polls to its designated assigned poll address (15.5.4).) At the end of a poll cycle the SC will send out any broadcast messages. One broadcast message that is always sent every poll cycle is the Date and Time Broadcast (15.5.1). (EGMs do not respond to broadcast messages but must process them.) The Date and Time Broadcast denotes the end of a poll cycle. Poll cycles are instigated by the SC and occur periodically (see 14.1.3).
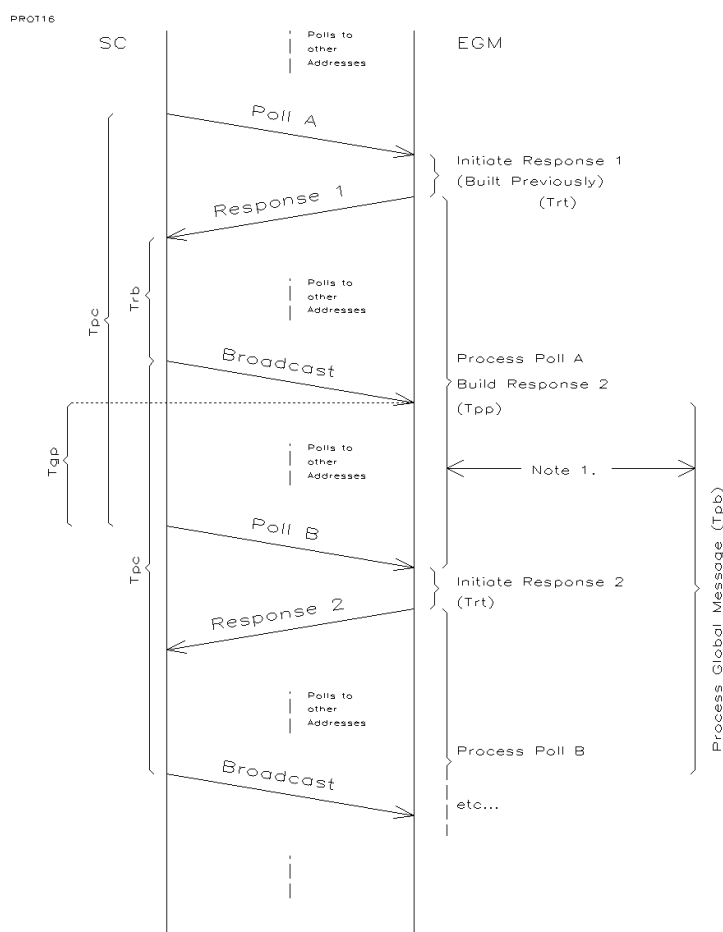
Figure 3 EGM Message Processing

(Refer following two sections)

Timings (refer diagram on left):

Tpc = One poll cycle time, Minimum (typical).

Tgp = 5 msec minimum.

Trb = 0 msec minimum.

Trt EGM Response Timeout Period.

5 msec maximum.

Notes :

Note the potential for the overlap of poll processing and broadcast receipt & processing. (Susceptibility for this depends on the EGM's CPU speed.) Refer 14.1.23

14.1.1   Poll / Response Processing Methodology

EGM processing of a poll message data must not be carried out until after the response to that poll has been initiated. [CP:525] Refer "Figure 3 EGM Message Processing" above.

This means that the EGM's response to the next poll is built by the EGM before the poll has been received. Refer "Figure 3 EGM Message Processing" above. This has the advantage of allowing the EGM to distribute the protocol processor load of processing the poll data and building a response over a longer period. Thus the only thing an EGM has to do within the Response Timeout Period is validate the poll message's CRC and if ok, initiate the last and latest pre-built response. The EGM must not try to build responses within the Response Timeout period, this is not allowed (since QCOM v1.5.5). [CP:526]

In other words, when polled for a specific response type, the EGM will not respond with the requested response type on the response to that poll, but the requested response type will be sent in response to the next poll (provided no other higher priority responses were pending).

E.g. → SC sends a Meter Poll Request
← EGM sends a response (built previously over last poll cycle)

→ SC sends the General Status Poll
← EGM now sends the Meter Response

The typical method in a poll-response protocol of processing the poll, building a response then sending the response to that poll off all within the response timeout period is not allowed. [CP:527] This is to standardise EGM response behaviour and it is also extremely CPU intensive as there can be under some circumstances too much to do within the response time-out period.

Thus, the only time the EGM may a build a response (for subsequent transmission) is after a valid poll (CRC ok) to the EGM's address has been processed. [CP:528]

In regards to unsolicited responses of applicable types e.g., group meter increments, events and program hash responses. When data pertaining to these types of responses is required to be transmitted, the software should flag the applicable response type as "need to send" such that the current data will be captured at the next response build at its given priority. This avoids building a response with aged data. This method has the advantage of only requiring the EGM to have **two** alternating response buffers; one is for the current transmission & acknowledgement processing, and the other to hold the next response (as built or being built) for pending transmission.

14.1.2 Broadcast Processing

The EGM must be able to process at least one Date and Time Broadcast each poll cycle and this must not interfere or be able to affect the processing of polls and building of responses to the EGM's poll address.

Refer to Figure 3 EGM Message Processing above and (14.1.23) for more information.

14.1.3 Poll Cycle Timing (for definition see 14.1 above).

Typically a SC will commence a new poll cycle no more than once per second. However QPv1.6 EGMs must now be able to sustain poll cycle periods as fast as 250 msecs without missing any responses. At a minimum the EGM must be able to process at least two messages per poll cycle; a poll to its address and one broadcast message. NB: the time between a poll-response from an EGM and a broadcast message may be negligible and vice-versa. [CP:529] NB: polls to other addresses can also occur at any time and must be ignored if not applicable to the EGM.

14.1.4 Maximum time to process message data.

To sustain a continuous stream of responses an EGM must take no more than the 250msec after receipt of a poll to its address to process the poll data and build the next response, or to process a Date and Time Broadcast. [CP:530]

However for some EGMs the processing of some polls under certain conditions may take a while to process. If the EGM can't process the poll in time it must behave as required in section 14.1.23. However this should not be considered allowable typical behaviour and a poll overrun should only occur under special circumstances (e.g. the EGM CPU was also heavily loaded at the time). *At this time one of the hardest polls to process in time is a Purge Poll when the EGM event queue was full. This is because the event queue CRC has to be completely recalculated in this case.*

Some message types may take a very long time for the EGM to process. On example is the Program Hash Request Poll (15.4.8). In this case the EGM will be granted a stated time-out period in which the poll message request must be processed, but in the interim the EGM must still process and respond to all received messages in a timely manner. These types of CPU intensive poll requests are typically processed by another task in background by the EGM.

14.1.5   The EGM must only respond to poll messages received on its designated poll address [CP:531] (Refer 15.5.4) that have a valid CRC. [CP:532]

14.1.6   Broadcasts.

An EGM must not respond to a message with the broadcast address (255) and the broadcast's control field must be ignored by the EGM.

14.1.7   Response Timeout period.

An EGM's response to a poll to its address must commence within 9 continuous character times (~5 milliseconds @ 19200 baud) upon receipt of the last byte of the poll [CP:533] provided the poll message had a valid poll address and CRC. [CP:534] This value is called the **Response Timeout period**.

Meeting this requirement can be challenge for EGM's using protected mode operating systems (i.e. Windows & Linux). In both cases meeting the required response time can require a change to the Operating Systems' Serial Port device driver.

14.1.8   Continuous Responses.

Once an EGM has resumed responding to a SC on its designated poll address (Refer 8.3), the EGM must respond to all subsequent received polls with a valid poll address and CRC, until the next power down of the EGM or communications timeout. After an EGM power up or communications timeout, the EGM must not resume responding until it has been assigned a poll address via the poll address broadcast message (Refer 15.5.4).

14.1.9   Response Priorities.

Upon building a response, the EGM must build the highest priority response that is pending but with the following exception; [CP:535] excluding the lowest priority response, responses of the same type (i.e. function code), must only be sent by the EGM every 2nd poll response. This allows for a lower priority response between consecutive responses of the same type. [CP:536] This behaviour should occur by default upon implementation the required QCOM response methodology (refer Figure 3 & 14.1.1).

14.1.10   Responses must reflect the latest EGM data.

All responses must be built no earlier than the one poll cycle prior their transmission to ensure they contain the most recent data. I.e. if a response type is requested but an opportunity for its transmission was delayed by higher priority responses, then when the response is finally sent it must reflect the latest EGM data no older than one poll cycle. [CP:537]

14.1.11   Acknowledging polls from the SC.

A complete EGM's response denotes an acknowledgement to the SC of its last poll to the EGM. (This is provided the response also has a valid poll address and CRC.)

14.1.12   Acknowledgement of responses from the EGM

A SC acknowledges each EGMs response via the CTRL field of the next poll to the EGM's address. Refer section 14.2.1.

14.1.13   An EGM must only process poll messages which have a valid CRC. [CP:538]

14.1.14   Queuing multiple responses of the same type. [CP:539]

If the SC sent two consecutive polls in a row requesting the same response type, then the EGM may respond with either one or two responses of that type requested, whichever is easier (usually one is easier). Both poll's message data must be processed. Similarly, if the SC sent a poll requesting a specific response and the response to that poll was the required response type, then the EGM is not required to send that response again. However, it may also send the whole response again if that is easier.

14.1.15   Maximum time between successive characters in a response.

Consecutive characters in a message response must not be separated by more than one character time (i.e. 0.5729 milliseconds @ 19200 baud). This time refers to the gap between two consecutive characters in an EGM's response. [CP:540]

14.1.16   Interrupted Responses.

If an EGM detects it receives any character while initiating or transmitting a response, it must immediately abort the response and process the received byte. Some EGMs only do this if the parity bit is enabled on the received byte. This is acceptable as it can be difficult to implement this requirement in some UARTs. [CP:541]

14.1.17   To distribute processor load in the processing of received polls and to achieve the fast response time, it is recommended poll message CRCs be calculated as the bytes are received.

14.1.18   While building a poll response, the message data must be protected from access by other processes, or change under interrupt. E.g. Metering data.

14.1.19   Unknown Function Codes.

If an EGM receives an undefined function code in a message poll, it must still respond [CP:542] and process the poll for an acknowledgement [CP:543], but ignore the unknown message data [CP:544]. For backward compatibility, an undefined function code must not be considered as a non-acknowledge by the EGM [CP:545]. It is the SC responsibility to ensure that it only sends known poll function types to an EGM (as the EGM reports its protocol version to the SC in the EGM Configuration Response (15.6.12)).

14.1.20   Extended Messages.

EGMs must be able to process poll messages normally that have additional message data inserted just before the CRC (i.e. a poll message must not be considered invalid because it has a length byte greater than what the protocol indicates for the given message type). [CP:546] This allows the SC to add additional information to poll message data for future expansion, but while maintaining backward compatibility.

14.1.21   There must be no sequence of invalid or corrupt poll data which, when received by an EGM, could adversely affect the EGMs integrity such as corrupting RAM. For example:

- Malformed message lengths; i.e. too short or too long. A purposely or otherwise malformed poll message length byte must not be able to cause a receive buffer overrun or corrupt EGM memory.

  There are four possibilities to consider:

  - Short malformed messages[*]. I.e. Valid FC & CRC with correct LEN field for the CRC position but one or more fields are missing. [CP:547]
  - Short messages that are cropped. I.e. the message is missing a CRC (see 'incomplete messages' below).
  - Long messages: additional bytes before the CRC (This is allowable behaviour, refer 14.1.20)
  - Long messages: additional bytes after the CRC. [CP:548]

  (Ideally an EGM as a validity check before processing a received message data properly, should cross check the message's LEN field against the theoretical expected length of the message as indicated by the message's function code and any applicable message data fields. This is because with increasing versions of QCOM, existing messages may expand in length, thus the probability that a SC may send an older QCOM version message to a newer QCOM version EGM is significant. However this validity check is not mandatory at this time, the numerous validity checks in section 15.4 should prevent any serious side effects if this was to occur.)*

- Rapid polling (much greater then one poll cycle per second). [CP:549]
- Invalid function codes (14.1.19).
- Incomplete messages. (causing communications state machine/process hang) [CP:550]
- Invalid message data within a valid message (all validity checks listed in section 15.4 must be performed by the EGM).

Anything transmitted to the EGM that can cause it to corrupt its memory, crash, or hang will be considered an EGM software bug that must be addressed.

14.1.22   Received Poll Message Integrity.

It must also not be possible to corrupt, overwrite, lose or abort the processing of a valid poll message in the EGM (wrt its CRC) once it has been acknowledged (refer 14.1.11) without resulting in an EGM RAM error. Specifically check an EGM power fail* and poll overrun condition (where new poll message is received before the current poll message is processed, refer 14.1.23).

* On slower EGMs there may be a critical section where EGM does not finish processing a poll before the poll has been acknowledged by the EGM (14.1.11), then there is a critical section of time following the last sent byte of the EGM's response (which denotes an ACK), until the EGM has finished processing the response, where a power interruption may result in a poll that has been ACK'd but is never actually fully processed by the EGM. The EGM must try to minimise or eliminate this critical section by either processing polls in a timely fashion or via state recovery. [CP: 551]

14.1.23   Poll Overrun.

Three cases (Poll-Poll, Poll-Broadcast & Broadcast-Poll):

If a new poll message (that is not a broadcast) is received before the previous poll message has been processed (including the building of a response if required), then the EGM must discard the new poll and ensure the current poll processing is completed. The EGM will miss a response as a result. This should not be possible in normal operation given the current poll cycle time but the EGM software must give consideration to this event regardless (this will be tested). [CP:552]

If a broadcast message is received before the previous poll message has been processed (including the building of a response if required), then the EGM must finish processing the current poll message then process the broadcast message. No data or response must be lost in this case. [CP:553]

If a new poll message (that is not a broadcast) is received before the previous broadcast message has been processed, then the EGM must finish processing the current broadcast message then process the new poll message. No data or response must be lost in this case. [CP:554]

14.1.24   Communications Buffer/s Integrity.

As per a poll overrun condition, at the low level, before an EGM can commence writing a new poll message to a receive buffer, it must ensure the buffer is not still in use (i.e. being processed). (This may not be applicable in all cases depending on each individual EGM manufacturer's implementation of QCOM). If the buffer is still in use, the EGM must simply abort receipt of the new poll message and ensure processing of the data in the current message buffer/s is completed. Similarly for the EGM transmit buffer/s. The EGM must ensure a transmit buffer is not still in use before commencing to build a new response. If the buffer is still in use, the EGM must wait until it is available before building any further responses. [CP:555] This may not be applicable in all cases depending on how QCOM is implemented by the EGM.

14.1.25   Please refer to the QCOM SDK for more information on the recommended EGM Message processing order.

14.2 Poll / Broadcast Message Format.

Message Format and Order:^(refer glossary)   **\<DLL\>[...]\<CRC\>**

DLL   = **\<ADRS\>\<LEN\>\<CNTL\>** where:

ADRS   Poll address of the EGM, 1 byte hex (1-254) ^{CP:556}.

LEN   Length of ensuing message, from the CNTL to CRC fields inclusive. 1 byte hex, LSB first.
For future expansion and backward compatibility the largest Message all EGMs must be able to process (from address byte to CRC inclusive) is 257 bytes. (i.e. maximum LEN= 255 or 0xFF) (*QPv1.6*)
(QPv1.5 EGMs typically can handle a max LEN = 0x9E (158 bytes) i.e. 160 bytes total)
The EGM must ensure LEN >= 4 bytes and <= Maximum EGM buffer size (i.e. 0xFF), if not the EGM must abort receipt of the poll (do not increment the CRC error counter in this case as this can occur frequently when there is more than one protocol on the LAN). ^{CP:557}

CNTL   Control Field, 1 byte hex. See 14.2.1 below.

[...]   Application Layer Message Data. Only one message type may be sent per poll. Refer sections 15.4 & 0.

CRC   16 bit CCITT CRC of the message, zero seeded, LSB LS bit first, utilising the standard polynomial, $X^{16} + X^{12} + X^5 + 1$. Arranged so a CRC result of zero is always returned for a valid message (address byte to CRC inclusive). The CCITT CRC Algorithm is available in "C" source code format with the protocol SDK.

14.2.1   Poll Message Control Field - CNTL

bit 0   Iamapoll flag. Must be set to 1 by all SCs.
This value is for future expansion and must be ignored by EGMs

bits 1-6 Reserved for future expansion = 0 (mask out these bits when reading any data from this byte) ^{CP:558}.

bit 7   ACK / NAK. Unnumbered Response Acknowledgement Bit.

A change of state of this bit with respect to the last received value indicates an acknowledgement to the EGM's last response. Valid states are 0 & 1. If the value of the bit does not change with respect to the last received value, this indicates a negative acknowledgement to the EGM's last response.

If the last response from the EGM is not acknowledged by the SC, (i.e. no change in this value), the EGM must re-queue the response to be rebuilt and sent again later ^{CP:559} (with respect to general response message priority ^{CP:560}). This is the only time a response is ever re-queued ^{CP:561}. When a response is NAK'd it must be rebuilt to ensure it reflects the latest EGM data at all times (e.g. the Meter Group/Contribution Response 15.6.8). ^{CP:562} ^{CP:563} NAKs in relation to Event Responses, see 7.1.3.

(QPv1.6) For any poll which contains a negative acknowledgement the EGM must not process the application message data ^{CP:564}. This prevents poll messages from being processed more than once by an EGM that is being NAK'ed. *(QPv1.5 EGMs will process the poll data again.)*

 (*QPv1.6*) This bit is not applicable on the first poll in a new session as there is no response yet to apply the acknowledgement to. ^{CP:565} However this bit must be~~it is used~~saved by the EGM to set the current Ack. state so acknowledgements can be detected on all subsequent polls in the current session. ^{CP:566}

In QPv1.5 EGMs, the last received value of this bit must be stored in the EGM's NV-RAM for the purposes of satisfying section 8.3.4. (This is no longer required for QPv1.6 EGMs)

This bit is not used in a broadcast message with the broadcast message poll address ^{CP:567}.

(QPv1.5 EGMs only) The control field from the last received poll to the EGM's address must be stored in non-volatile RAM on the EGM in order to correctly determine acknowledgements after a power down. ^{CP:568}

14.3 Response Message Format.

Message Format and Order: ^(refer glossary) **<DLL>[...]<CRC>**

DLL   = **<ADRS><LEN><CNTL>** where:

ADRS Poll address of the EGM, 1 byte hex (1-254).
LEN   Length of ensuing message, from the CNTL to CRC fields inclusive. 1 byte hex, LSB first.
(SCs must be able to receive response up to LEN = 0xFF, i.e. 257 bytes total)
CNTL Response Control field, 1 byte hex. Refer Section 14.3.1 below.
[...]   Application Layer Message Data. Only one message type may be sent per poll. Refer section
15.6.
CRC   16 bit CCITT CRC of the message, zero seeded, LSB LS bit first, utilising the standard
polynomial, $X^{16} + X^{12} + X^5 + 1$. Arranged so a CRC result of zero is always returned for a valid
message (address byte to CRC inclusive). The CCITT CRC Algorithm is available in "C" source
code format with the protocol SDK.

14.3.1   Response Control Field (CNTL)

Bit 0   Iamaresp flag. Must be set to 0 by the EGM (*all currently do*)
Bit 1   8BitMode flag. Must equal 1 if the EGM supports 8-bit mode (13.3). Zero otherwise.
bits 12-4   Reserved = 0.
Bits 6…5   QPV (QCOM Protocol Version) *(QPv1.6)*
This 2 bit field must always be set to a value of 01 by the EGM. This field denotes what
version protocol the EGM implements.
QPV = 00 denotes a QCOM Protocol v1.5.5 and below EGM
QPV = 01 denotes a QCOM Protocol v1.6 or later EGM
This is to help a SC easily recognise which version protocol the EGM communicates
on.

Also see the QPV field in the EGM Configuration Response (15.6.12)

bit 7   Unnumbered Broadcast Message Negative Acknowledgement

This allows the SC (if desired) to receive unnumbered acknowledgements to broadcast
messages sent to the broadcast poll address 0xFF.

An EGM must expect to receive at least one valid broadcast message (a valid broadcast
includes correct poll address & CRC, but exclude function code) between successive polls to
the EGMs address. If the EGM detects a missed or corrupt broadcast message, the EGM
shall set this bit in all subsequently built future responses until the next time a response with
the bit set is acknowledged by the SC in the normal fashion. ^{CP:569} The bit may then be
cleared from future responses.

Because of the pre-building of responses requirement (14.1.1), when a broadcast requires a
negative acknowledgement, it is not expected or required that this bit will be set in the very
next response from the EGM, but the NAK bit must be set no later than in the following
response from the EGM. Eg. refer Figure 3 EGM Message Processing. At the beginning of
this figure, if the broadcast immediately prior to Poll A was corrupt (this broadcast is not
actually shown on the diagram), then it is expected the global NAK be sent no sooner or later
than response 2. Attempting to send the global NAK any sooner can be very difficult to
implement as this would require rebuilding an already built response.

At all other times the bit is not set ^{CP:570}.

EGMs must ensure they do not inadvertently set this flag if it was to receive two or more
consecutive broadcast messages with no poll to the EGM's address in between ^{CP:571}.

15 Application layer, Message Types and Data

15.1 Common Field Formats

15.1.1 Site Enable Flag (SEF)

An EGM has one SEF. This flag is sent in all broadcast messages (0). A value of one means enable, zero means disable. Refer to section 6 for more information on what a SEF disable does with regards to being a CDC.

15.1.2 Machine Enable Flag (MEF)

An EGM has one MEF. Where this flag appears throughout the protocol it refers to this flag. A value of one means enable the EGM for play and allow physical credit input; zero means disable the EGM for play and disable physical credit input. Refer 6 for more information on the MEF and CDC's.

15.1.3 Game Enable Flag (GEF)

Each game in an EGM has one GEF. Where this flag appears in polls and responses it must be used in association with the GVN in the message to determine which game the GEF applies to. A value of one means the game is enabled, a zero means the game is disabled.

Specifically, the EGM will enable and disable individually configured games (even in a single game EGM) as indicated by their Game Enable Flag sent in the EGM General Maintenance Poll (15.4.13) and the EGM Game Configuration Poll (15.4.3). [CP:572]

Some multi-game EGMs allow only a limited number of games to be enabled for play at one time (often this limit is usually imposed by the game select screen), but actually have a larger number of games resident in the EGM to choose from. The EGM reports this information in the EGM Configuration Response, Section 15.6.12.

If the EGM is told to enable a game that causes the maximum allowable number of enabled games to be exceeded, the EGM must ignore the request to enable that game. [CP:573] However this does not stop a game from being configured.

Also refer to section 9 for how GEF affects the game select screen in a multi-game EGM.

15.1.4 Base Software Version Number (BSVN)

Also abbreviated to BVN in other documents, this field is a 2 bytes hex hash value, assigned by the EGM manufacturer, which must uniquely identify the EGM's generic base software and version for the manufacturer. [CP:574] This field is reported by the EGM only in the EGM Configuration Response (15.6.12).

EGM manufacturer's must ensure BSVNs must be unique with respect to all previously used BSVN & GVN numbers for the manufacturer. However, if a single game EGM does not have a generic base software set, then the BSVN may be set at the same value as for the GVN number for the game [CP:575]. Systems should refer to the latest version of the QCOM Site Controller Procedures document for what do with BSVN fields.

15.1.5 Game Version Number (GVN)

Manufacturer assigned game version I.D number. The GVN field is a 2 byte hex number, LSB first.

Every version of a game produced by an EGM manufacturer must have one GVN number which must uniquely identify the game and software version of the game for that manufacturer. In this protocol the GVN is used as the method of referring to a specific game. So long as GVN numbers are unique to the game and software version, the EGM manufacturer may use whatever GVN numbering scheme

they like (e.g. a 16 bit CRC over the game version string, a simple serial number, or some sort of hash algorithm over the game version string or the game itself).

An approved game denoted by a GVN, for a single game EGM, can have the same GVN if it is incorporated into a multi-game EGM on the condition that the game is identical in all respects; i.e. same game name, number of variations, PRTPs, reel strips, rules, etc.

GVNs for games ported to a new EGM base/shell version can have the same GVN on the condition that the game is identical in all respects; i.e. same game name, number of variations, PRTPs, reel strips, rules, etc.

A GVN of zero is reserved and must not be used as a game I.D. number.
When combined with the VAR field, this uniquely identifies a game and variation in an EGM of a manufacturer.

15.1.6   Game Variation Number (VAR)

The VAR field is always attached to a GVN and is the manufacturer assigned game program variation that denotes a game's percentage return (the game as denoted by the GVN).
The VAR field is a 1 byte BCD number. 00...99. A VAR of zero is reserved (it denotes the game requires configuration) and must not be used as a variation I.D. number.
When combined with the GVN field, this uniquely identifies a game and variation in an EGM.

15.1.7   Manufacturer I.D. (MID)

The MID field is a 1 byte BCD number. Gaming machine manufacturer I.D. numbers referred to in this document (MID fields) are assigned by the OLGR.

Refer to the appendices for the currently assigned MIDs [CP:576]

MID numbers must be hard-coded into the EGM's program [CP:577].

When combined with the serial number the two fields uniquely identify a QCOM Protocol EGM.

NB In QCOM, MID fields are Binary Coded Decimal (BCD) hexadecimal values. E.g. a MID of "14" is encoded as 0x14 within a QCOM message packet and QSIM would display "14" on its display if the MID is implemented correctly by the machine. (NB because of past confusion, QCOM no longer uses BCD for new fields, unfortunately for legacy fields like this one, its units cannot be easily changed due to the number of systems now built around it)

15.1.8   EGM Serial Number (SER)

EGM serial numbers are 6 digit EGM manufacturer assigned serial numbers. When combined with the manufacturer ID number (MID) above, the two fields uniquely identify a QCOM Protocol EGM. The MID & SER are always used together and the MID is the most significant number. The MID & SER is the primary identifier for communication with an EGM on the QCOM protocol. Refer to the Poll Address Broadcast Message for more information (15.5.4). Refer to section 3.1.2 on how a serial number is configured into an EGM.

15.1.9   Poll Sequence Numbers (PSNs)

PSN fields are 1 byte hex numbers and appear in selected poll message types only. PSNs are used to ensure that an EGM doesn't process the same poll message data that the PSN pertains to, twice.

Presently they are used by two poll message types, the Purge Events poll (Section 15.4.15) and 'ECT to EGM' poll (Section 16.1.1).

The EGM must maintain a separate non-volatile PSN for each Poll message type which uses a PSN field [CP:578]. I.e. Two. One for Purge Polls and another for ECT Polls.

After validating the poll message address, function code and CRC and a PSN number appears in a poll message, the EGM will only process the remaining poll message data after the PSN field, if the message PSN is equal to the last received valid PSN number plus one modulus 256 [CP:579].

i.e. $PSN_n = (PSN_{n-1} + 1) \% 256$ (in C notation).   => 0...255

If a PSN passes the above test on the processing of an applicable poll, then the EGM must update the next expected PSN as per the above formula for the given poll type at that time. If the PSN fails then its value must remain unchanged. [CP:580]

The default values for Poll Sequence Numbers the EGM must assume after a RAM clear is zero (i.e. the first expected PSNs will be 0x01).

PSNs may be reset on command by the SC in the EGM Configuration Request Poll (15.4.1) [CP:581].

15.1.10   Progressive Group I.D. (PGID)

The PGID field is a 2 byte hex number.

PGID numbers are used to denote groups of levels of the basic EGM triggered, linked progressive jackpots. They are arbitrarily assigned by the monitoring system as required by linked progressive games. If a game has a linked progressive component (i.e. if one or more progressive levels have been configured as LP) then its LP levels will be assigned a PGID number in the range 0x0001 to 0xFFFE (inclusive). A single game can have only one PGID number. Different games within a single EGM can have different PGID numbers (and usually will have).

The PGID's primary function is to enable the EGM to identify applicable current progressive amounts when broadcast out by the SC (15.5.2). Via the PGID, the EGM knows which current level amounts are applicable to its games.

A PGID number of 0x0000 in the EGM Game Configuration response (15.6.11) indicates the game has a progressive component with levels configurable as either LP or SAP and requires configuration via the EGM Game Configuration Poll (15.4.3).

PGID in the range 0x0001-0xFFFE indicates the game has been configured with one or more Linked Progressive levels and this is the LP group I.D. number contributed to by the game.

A PGID number of 0xFFFF indicates the game is either a non-progressive game or was configured as SAP only.

15.2 Poll Message List and Function Codes

Function Code	Message Description
0x39	\
…	Reserved (*FTP*)
0x36	/
0x33	EGM Configuration Request Poll (15.4.1)
0x31	EGM Configuration Poll (15.4.2)
0x2F	EGM Game Configuration Poll (15.4.3)
0x2E	EGM Game Configuration Change Poll (15.4.4)
0x2D	EGM Parameters Poll (15.4.5)
0x2C	Progressive Configuration Poll (15.4.6) *QPv1.6*
0x2A	External Jackpot Information Poll (15.4.7) *QPv1.6*
0x29	Program Hash Request Poll (15.4.8)
0x27	System Lockup Request Poll (15.4.9)
0x24	Reserved for ECT Operation - ECT Lockup Reset Poll (16.4)
0x23	Reserved for ECT Operation - 'ECT from EGM' Lockup Request Poll (16.3.2)
0x22	Reserved for ECT Operation - ECT to EGM Poll (16.1.1)
0x21	Cash Ticket Out Request Acknowledgement Poll (15.4.10) *QPv1.6*
0x20	Cash Ticket In Request Acknowledgement Poll (15.4.11) *QPv1.6*
0x1E	Cash-Out Request Poll (0)
0x1B	EGM General Maintenance Poll (15.4.13)
0x17	Request All Logged Events Poll (15.4.14)
0x16	Purge Events Poll (15.4.15)
0x12	Reserved for Configurable Random Events (CRanE. Section 17.5)
0x11	Reserved for Configurable Random Events (17.3)
0x10	Note Acceptor Maintenance Poll (0)
0x0F	Hopper/Ticket Printer Maintenance Poll (15.4.17)
0x0E	Linked Progressive Award Acknowledged Poll (15.4.18)
0x0C	EGM General Reset Poll (15.4.19)
0x08	Specific Promotional/Advisory Message A Poll (SPAM A 15.4.20)
0x07	Specific Promotional/Advisory Message B Poll (SPAM B 15.4.20)
0x05	EGM Tower Light Maintenance Poll (15.4.21)
0x02	Reserved for QCOM Command Prompt Poll (20.1)
0x01	General Status Poll (15.4.22)

Broadcast Message Summary

Function Code	Message Description
0xFF	Current Date and Time and Broadcast Information (15.5.1)*
0xFE	Reserved
0xFD	Reserved for QCOM EGM To EGM Communications (Section 19)
0xFC	Reserved for Finding EGMs (Section 21)

* Extended Current Date and Time Broadcast Messages (Refer Section 15.5)

0x02	Linked Progressive Jackpot Current Amounts (15.5.2)
0x03	General Promotional Message (GPM 15.5.3)
0x04	EGM Poll Address Configuration (15.5.4)
0xFE	Site Details (15.5.5)

15.3 Response Messages, Priority and Function Codes

Func. Code	Priority	Solicited / Unsolicited*	Message Description (section)
0x20	NA	Yes	Reserved for the Seek EGM feature. Refer section 21.3
0x19	1	Yes	Program Hash Response (15.6.13)
0x16	2	Yes	EGM Configuration Response (15.6.12)
0x13	3	Yes...	EGM Game Configuration Response (15.6.11)
0x10	4	Both...	Event Response (15.6.9)
0x0F	5	Yes	Purge Event Poll Acknowledgement Response (15.6.10)
0x0E	6	Yes	ECT-to-EGM Poll Acknowledgement Response (16.1.2) (*QPv1.6*)
0x0C	7	Both...	Meter Group/Contribution Response (15.6.8)
0x0B	8	Yes	Player Choice Meters Response (15.6.7) (*QPv1.6*)
0x0A	9	Yes	Multi-Game/Variation Meters Response (15.6.6)
0x09	10	Yes	Bet Meters Response (15.6.5) (*QPv1.6*)
0x08	11	Yes	Progressive Configuration Response (15.6.4) (*QPv1.6*)
0x07	12	Yes	Progressive Meters Response (15.6.3)
0x06	13	Yes	Reserved for Configurable Random Events (CRanE) (17.6)
0x05	14	Yes	Reserved for EGM to EGM Communications, refer 19.1 (*QPv1.6*)
0x04	15	Yes	Note Acceptor Status Response (15.6.2) (*QPv1.6*)
0x02	16	Yes	Reserved for QCOM Command Prompt Response (20.2) (*QPv1.6*)
0x01	17	No	General Status Response (15.6.1)

1 denotes highest priority response in the priority column above. Refer 14.1.9.

* In QCOM all EGM responses are solicited by the SC. However responses of a particular type have three categories; solicited, unsolicited or both. Unsolicited means that the EGM will send the response type in response to a poll when it needs to (wrt general response priority), solicited means that the SC will only get the responses type when it specifically requests it.

"**...**" above, denotes multiple responses of the same type may have to be queued simultaneously by the EGM as the result of one or more polls (more information is provided on this matter within the sections on each response marked with ...). For all other responses the EGM will only have to be able to queue one response of a type at a time.

15.4 Poll Message Data

Refer to the glossary (section 2.2) for an explanation of the "Message Format and Order:" string and meaning of a reference to "Message data".

Note.   All reserved message bits and bytes must always be masked out by the EGM during poll message processing to avoid any problems when the protocol is expanded.

15.4.1   EGM Configuration Request Poll

Upon receipt of this poll message type, the EGM will queue the EGM Configuration Response (15.6.12) [CP:582] and perform other functions as indicated by the FLG field.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x33, 1 byte hex
FLG   1 byte hex.
         Bits 0...4   reserved = 0 (mask out these bits when reading any data from this byte) [CP:583]
         Bit 5   Machine Enable Flag (MEF refer section 15.1.2) [CP:584]
                   Bit 5 = 1 EGM enable. Bit 5 = 0 EGM disable
         Bit 6   EGM Game Configuration Request Flag.
                   If set, commands the EGM to queue the EGM Game Configuration Response (15.6.11) for all available* games in the EGM [CP:585] (* Refer Section 9 and 15.6.12->NUMG field)
                   (NB: QPv1.6 EGMs will not report game configuration responses until EGM configuration has been completed. Refer Section 9)
                   If not set, no action [CP:586]
         Bit 7   Reset PSN Flag.
                   If set, command the EGM to reset all its current Poll Sequence Numbers (15.1.9) to zero (i.e. the next expected PSNs will be 0x01). This does not affect event sequence numbers. Refer to the Section Common Field Formats for an explanation of PSNs [CP:587].
                   If not set, no action [CP:588]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.4.2   EGM Configuration Poll

This poll message type is used by the SC to configure an EGM after a RAM clear. (Once an EGM is configured via this poll, games are then configured via the EGM Game Configuration poll, refer to the next section.)

Before accepting any of the data in this poll (which the EGM must only do if at least one game can be enabled based on the data in the poll), the EGM must confirm all the poll message data is valid, correct and applicable to the EGM. Any invalid or incorrect information detected in this poll (e.g. with respect to; MID [CP:589] and serial number [CP:590] (also DEN & TOK if applicable)) by the EGM, will result in the poll's 'message data' area being ignored and the "Invalid EGM Configuration" event (7.10.3.21) must be logged. *(QPv1.6)* The EGM must also reject the poll and log the event, if accepting the configuration would mean no games would be subsequently available as a result of the parameters (*RTEXT (7.10.1.16) = "No Games Avail")*. [CP:591]

Once the EGM has accepted the configuration in an EGM Configuration Poll, it must not process this poll type again until after its next RAM clear* [CP:592]. (* there is special casean exception here for EGMs supporting denomination switching (refer 3.1.3.1) ; for EGMs supporting denomination switching Tthis poll may be subsequently used to change an EGM's credit denomination).

The current settings for the data set via this poll may be verified by a SC by requesting the EGM Configuration Response (refer 15.6.12)

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x31, 1 byte hex
SER   EGM Serial number (15.1.8), 3 bytes BCD, LSB first
       Validity check: SER must be BCD and correct [CP:593]. (*RTEXT (7.10.1.16) = "SER not BCD" or "SER Incorrect")*.
MID   EGM Manufacturer I.D., 1 byte BCD. Refer section 15.1.7
       Validity check: MID must be BCD and agree with hard-coded value [CP:594]. (*RTEXT (7.10.1.16) = "MID not BCD" or "MID Incorrect")*.
       SER & MID uniquely identify the EGM being configured

Configuration item fields:

JUR   Jurisdiction, 1 byte hex.
               This field informs the EGM of which jurisdiction it is operating in. This allows the EGM to correctly execute code specific to a jurisdiction. *(In QPv1.5 this parameter set the RTP of the RCRF, the max bet and whether power-save and the player instigated machine reserve features were available.)* In QPv1.6, this field is no longer used except as a last resort and will be assigned by the OLGR as needed. Multi-jurisdiction support is generally achieved through a wider range of new parameters.

               For any gaming machine manufacturer still utilising JUR for any purpose other then recording/reporting its current value, please advise the OLGR the product name and version and for what purpose it is being used. [CP:595]

               Defined jurisdictions:

AU, QLD CLUBS & HOTELS	0x00
AU, QLD CASINOS	0x01
New Zealand	0x02
AU, Victoria	0x03
AU, South Australia	0x04
AU, Tasmania	0x05

FLG   Machine flags, 1 byte hex, reserved = 0.
ODEN   *2 bytes BCD. Credit Denomination used by QPv1.5 EGMs only*

OTOK   *2 bytes BCD. Token Denomination used by QPv1.5 EGMs only*
              Validity check (QPv1.5 only): must be BCD [CP:596]

New fields for QPv1.6:

DEN   Credit Denomination, 4 bytes hex in cents, LSB first.
              This field typically (refer 3.1.3) sets the denomination of all credit on the EGM. [CP:597]
              E.g. if DEN = 10, then 1 credit = 10 cents.
              All bets wagered and all non-progressive wins on the EGM must be evenly divisible by this amount. [CP:598]
              Validity check: DEN must be less than or equal to MAXBET & MAXDEN [CP:599]. (*RTEXT (7.10.1.16) = "DEN > MAXBET" or "DEN > MAXDEN"*).
              An EGM may automatically reject at its discretion, any denomination it considers unreasonable, or that it doesn't support. Also refer to the invalid denomination event (7.10.3.32).
              The EGM's denomination may be changed at a later time via this poll if the EGM supports this feature. Refer section 3.1.3.1 on Denomination Hot-Switching. Capability for Denomination Hot-Switching is indicated by the EGM via the EGM Configuration Response (15.6.12).
TOK   Coin/Token Denomination, 4 bytes hex in cents, LSB first
              Applicable only to EGMs with Coin Validators, this amount sets the denomination value of a single coin/token. [CP:600]
              TOK also denotes the win amount to be awarded for a successful the RCRF. [CP:601]
              The poll must be considered invalid if either DEN or TOK is equal to zero [CP:602]
              If the EGM does not have a Coin Validator, then this field is still accepted but is not used by the EGM except for display purposes in audit mode and is echoed back in the EGM Configuration Response.
              It is recommended the EGM place a ~~reasonableness~~ sanity test on DEN & TOK [CP:603] (However be careful not to restrict the EGM from being used in other jurisdictions)
              Support for DEN > TOK is at the EGM manufacturer's discretion. If not supported, the EGM must consider the poll invalid) [CP:604]
              An EGM may automatically reject at its discretion any coin/token denomination it considers unreasonable, or it doesn't support. Also refer to the invalid denomination event (7.10.3.32).
MAXDEN   EGM Credit Meter Max Denomination, 4 bytes hex in cents, LSB first (Refer 3.1.3)
              This field is the regulatory maximum allowable denomination of the EGM's credit meter. Refer Section 9 'Extended Multi-game Support' for intended use.
              Validity check: MAXDEN <= MAXBET [CP:605]. (*RTEXT (7.10.1.16) = "MAXDEN > MAXBET"*).
RES   Reserved 4 bytes hex.
MINRTP   Minimum RTP, 2 bytes hex, LSB first, in units of % x 100 (e.g. 8502 = 85.02% RTP).
              Validity checks: MINRTP must be <= MAXRTP, >= 5000 (50%) and <= 10000 (100%). [CP:606]. (*RTEXT (7.10.1.16) = "MINRTP Invalid"*).
              This field is the regulatory minimum percentage RTP for the jurisdiction for the EGM's information. It includes RTP from jackpots triggered by the EGM.
              MINRTP must set the RTP of the RCR feature if implemented. [CP:607]
              MINRTP & MAXRTP (below) do not control the RTP of any game, ~~but~~ and the EGM may use these fields at its discretion (e.g. for validity checks for the game variation selection set). Refer Section 9 'Extended Multi-game' Support for one intended use. Sanity checking Progressive Configuration Poll data (15.4.6) is another possible use.
MAXRTP   Maximum RTP, 2 bytes hex, LSB first, in units of % x 100 (e.g. 9250 = 92.50% RTP).
              Validity checks: MAXRTP must be <= 10000 (i.e. 100%). [CP:608]. (*RTEXT (7.10.1.16) = "MAXRTP>100%%"*).
              This field is the regulatory maximum percentage RTP for the EGM for the jurisdiction.
              MAXRTP upon application includes any RTP from jackpots triggered by the EGM.
              Refer Section 9 'Extended Multi-game Support' for intended use.
MAXSD   Regulatory Maximum (nominal) EGM Game Theoretical Standard Deviation. 2 bytes hex.
              MAXSD does not apply to the progressive component of the game, if any.
              The maximum value of 65535 denotes there is no limit. [CP:609]

	Refer Section 9 'Extended Multi-game Support' for intended use.
MAXLINES	2 bytes hex.
	The regulatory maximum number of playlines in any game in the EGM.
	Applicable only to reel based EGM games.
	The maximum value of 65535 denotes there is no limit. [CP:610]
	Refer Section 9 'Extended Multi-game Support' for intended use.
MAXBET	Maximum Bet per play. 4 bytes hex in cents.
	The regulatory maximum amount that may be bet per play on the EGM. No play's bet on an EGM may be allowed to exceed this value. [QPV:1.6.2] [CP:611] Since this value cannot be changed without an EGM RAM clear, it can be implemented at game configuration (i.e. the EGM will not allow a game to be enabled that would exceed MAXBET). Alternatively, the OLGR will also accept an implementation in which the EGM forces applies the limit during player betting. This allows games to be accessible at a lower MAXBET.
	Also refer Section 9 'Extended Multi-game Support' for another use.
MAXNPWIN	Regulatory maximum non-progressive EGM win permitted in any game element (refer to GMNS for definition of this field). 4 bytes hex in cents. [CP:612]
	Refer Section 9 'Extended Multi-game Support' for intended use.
MAXPWIN	Regulatory maximum SAP progressive EGM win permitted per SAP hit. 4 bytes hex in cents.
	The EGM must only apply this field to SAP level ceilings. [CP:613] Refer to the Progressive Configuration Poll (15.4.6) where MAXPWIN is used to sanity check ceiling values for *customSAP* games.
	~~(As a result of the 95% CI requirement on ceilings, this will also indirectly limit other SAP parameters such as a SAP level's maximum Average Trigger Value.)~~
	Also refer Section 9 'Extended Multi-game Support' for another use.
MAXECT	Regulatory maximum allowable ECT to/from the EGM. 4 bytes hex in cents. [CP:614]
	The EGM will limit ECT and Cash Ticket In (15.4.11) operations to amounts less than MAXECT. Refer Section 16.
	MAXECT also caps TITO transactions. Refer section 22.
	Operators in Australia ~~using QCOM~~ must be aware of the "Financial Transactions Reports Act " that contains requirements regarding 'significant cash transactions' and 'suspect transactions'. ~~As of the date of this document the threshold was $10,000au.~~

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

The EGM must not apply sanity checks to EGM Configuration Poll 'regulatory' max/min type parameters (see 'regulatory' above) other than the sanity checks specifically stated (see individual field descriptions above). An EGM Configuration Poll with all the 'regulatory' max/min type parameters set to their most liberal values must allow any EGM configuration to be accepted (provided the EGM also supports the specific values sent for JUR, DEN, TOK) and allow the EGM to expose all games in the EGM. EGMs should not apply 'equal to' constraints to MAXBET, MAXLINES and MAXSD fields etc.; as these parameters and similar may change over time during the operational life of a game or may be set to max value.

Additional EGM manufacturer implemented sanity checks or pre-requisites concerning JUR, DEN, TOK fields are acceptable provided the EGM provides adequate feedback on any issues via the logging of "Invalid EGM Configuration" events (7.10.3.21) followed with one or more suitable explanatory RTEXT (7.10.1.16) events. Feedback represented by the event data must always provide a clear way forward in resolving issues with EGM Configuration Poll data. [CP:615]

15.4.3 EGM Game Configuration Poll

This poll initialises each game in an EGM after a RAM clear. This poll type is sent by the SC after the EGM Configuration Poll (15.4.2) has been acknowledged by the EGM. The EGM must not enable a game until this poll has been received for that game ^{CP:616} (as identified by its GVN & VAR). Current game configuration may be verified via the EGM Game Configuration Response (15.6.11).

There is no specific response queued in response to this poll.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x2F, 1 byte hex
GVN   Game Version Number (*15.1.5*), 2 bytes hex. Identifies the game being configured.
    Validity Check: GVN must exist in the EGM. (*RTEXT (7.10.1.16) = "GVN Unknown"*).
VAR   Game Variation Number (*15.1.6*), 1 byte BCD. Indicates to the EGM what variation to use.
    Validity Check: VAR must be BCD ^{CP:617} & exist for the game (denoted by GVN) and be legal ('legal' w.r.t. the current EGM Configuration Poll settings (15.4.2) however the EGM has the option of applying the legality check in idle mode, refer 6.1.9). (*RTEXT (7.10.1.16) = "VAR Unknown", or "VAR not BCD", "VAR Illegal"*).
GFLG   1 byte hex.
    Bits 0..5 reserved = 0, mask out these bits when reading any data from this byte ^{CP:618}.
    Bit 6   Variation Lock. *QPv1.6*
        If set, the EGM must lock the above variation and not allow the variation to be changed. This effectively disables variation hot switching until the EGM is next RAM cleared. The status of the variation lock is reported via the EGM Game Configuration Response (15.6.11). ^{CP:619}
    Bit 7,   Game Enable Flag (GEF, refer 15.1.3) ^{CP:620}
        1 = enable game denoted by GVN, 0 = disable game denoted by GVN

If the EGM is a multigame with shared progressive component (refer 10.9), then processing from this point onwards for a progressive game must only be performed once per RAM clear by the EGM and be automatically applied to all applicable games involved in the shared progressive arrangement by the EGM. I.e. any differences in subsequent progressive configs would simply be ignored. ^{CP:621} *(QPv1.6.4)* This ensures any initial contribution is only accepted once.

PGID   Linked Progressive Group I.D. (15.1.10). The LP group contributed to by the game identified by GVN & VAR (0x0001-0xFFFE), ^{CP:622} 2 bytes hex
    0xFFFF indicates either a non-progressive game or indicates game will run as a SAP and has no LP levels ^{CP:623}.
    Validity Check: The message must be considered invalid if PGID is zero (see below) ^{CP:624}. *RTEXT (7.10.1.16) = "PGID = 0"*.
PNUM   1 byte hex, number of progressive levels in the game. 0...8
    Validity Check: The message must be considered invalid if PNUM is > 8 ^{CP:625}. (*RTEXT (7.10.1.16) = "PNUM >8"*).
    Validity Check: If PNUM is ~~zero or~~ incorrect with respect the game's actual number of progressive levels, the EGM will consider the poll invalid and log the "Invalid Game Config." Event (7.10.3.22) ^{CP:626} *(QPv1.6)*. (*RTEXT (7.10.1.16) = "PNUM Incorrect"*).
    NB QPv1.5 EGMs would default to SAP ~~in this case~~if PNUM was invalid.
SIZ   1 byte hex. Size of each repeated entry = 5 ^{CP:627}

The following message poll data is repeated PNUM times for each progressive level in the game, in order starting from level 0 (the highest expected jackpot). There may be up to eight repeated progressive entries in one poll message depending on PNUM.

PFLG   1 byte hex.
    bits 0...6   Reserved ^{CP:628}
    bit 7   Level type.
        if set, denotes the level is to be set as a LP ^{CP:629}.
        if not set, denotes the level is to be set as a SAP ^{CP:630}.

CAMT   4 bytes hex in cents, LSB first.
Initial jackpot contribution/current amount for progressive level.
Validity Check: CAMT < 2^31 [CP:631] (*RTEXT (7.10.1.16) = "GCP:CAMT Bad"*) [(QPv1.6.4)]
If SAP, CAMT is the initial contribution towards the progressive level.
i.e. The initial SAP jackpot current amount (uncapped) = SUP + CAMT [CP:632]
This contribution is usually a carry over as the result of a RAM clear on the EGM. The EGM must be able to restore the same jackpot current amount for the level given this initial contribution [CP:633].
Note, any overflow caused by CAMT must go to the level's overflow meter [CP:634]
If the SAP also has a hidden/auxiliary increment, then a~~n appropriate~~ AUXRTP proportion of CAMT ~~must~~ may be diverted to ~~this~~ the hidden component (if suitable/desired*). *(*However i~~l~~t might be best not to divert any unless a ceiling is breached, as it may confuse installers / operators if the whole amount does not appear as received.). All of the CAMT value must be accrued in any case.*
Regarding SAP level parameters, upon accepting this poll, the EGM must assume reasonable (wrt MINRTP & MAXRTP) default hard-coded SAP parameter values for all its SAP levels (E.g. SUP, PINC & CEIL, refer 15.6.4) [CP:635]. The EGM may allow the SAP parameters to be subsequently changed via Progressive Configuration Poll (15.4.6) if the game has CustomSAP support *QPv1.6*. Capability for this feature is indicated by the EGM via the Progressive Configuration Response (Refer 15.6.4).
If LP, CAMT is the initial jackpot current amount for the LP level and may be treated by the EGM as the same as a LP Broadcast to the EGM's PGID [CP:636].
Regarding LP level parameters, the EGM controls only the hit rate which must be hard-coded to the game's variation, all other LP level parameters are controlled by the monitoring system, however, the EGM is kept up to date with these parameters by the SC via the Progressive Configuration Poll (15.4.6) *QPv1.6*.

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

The EGM must enable games (provided their Game Enable Flag is also set) as they are configured by this poll [CP:637]. Also refer to the "EGM Game Variation Enabled" event (7.10.3.7).

Validity checks: The EGM will consider the poll invalid and not accept the poll's 'message data' if either:

- EGM Configuration has not been successfully completed via the EGM Configuration Poll (15.4.2) *QPv1.6.* [CP:638]. (*RTEXT (7.10.1.16) = "EGM not Cfg"*).
- The VAR [CP:639], GVN [CP:640] or PGID field, is illegal, incorrect or not applicable to the game and EGM. (*RTEXT (7.10.1.16) = "xxxx Unkown", or "xxxx Illegal" where "xxx" is the VAR, GVN or PGID respectively*).
- For a progressive game, if PGID indicates LP but no progressive levels were indicated as LP [CP:641], or, if PGID indicates SAP but one or more progressive levels were indicated as LP [CP:642]. (*RTEXT (7.10.1.16) = "PGID Inconsist"*).
- If a LP only or SAP only progressive level is attempted to be set as a SAP or LP respectively. (*RTEXT (7.10.1.16) = "Lev x is LP/SAP Only" where 'x' is substituted with the level ID number*)
- It fails any other validity check in this section.

If the message is considered invalid for any of the above reasons then the EGM must log the "Invalid Game Configuration" event (7.10.3.22) [CP:643].

The EGM must only process a valid poll message of this type once per game (GVN) per RAM clear [CP:644].

15.4.4 EGM Game Configuration Change Poll

(In QPv1.5 this poll type used to be called the "EGM Variation Change Poll")

This poll may be used to change aspects of a game's current configuration. Such as:

- GEF
- Variation [CP:645] (if hot variation switching is supported by the particular EGM (refer 15.6.11)),
- Linked Progressive Group I.D (PGID) [CP:646],

This poll's 'message data' area must be ignored by the EGM if it does not have zero credit on the credit meter upon the initial processing of this poll message [CP:647], or a play is in progress or incomplete [QPv1.6] [CP:648], or if the game has not yet been configured via the EGM Game Configuration Poll (15.4.3) [CP:649].

If the poll is accepted, changes must take effect immediately if the EGM is in idle mode [CP:650], otherwise if not in idle mode, the changes must be queued in NV-RAM [CP:651] and take effect upon the next return to idle mode (3.3.2). [CP:652]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x2E, 1 byte hex
GVN   Game Version number (*15.1.5*), 2 bytes hex. Identifies the game being re-configured.
       Validity Check: GVN must exist (see below *). (*RTEXT (7.10.1.16) = "GVN Unknown"*).
VAR   Game Variation number (*15.1.6*), 1 byte BCD.
       Indicates to the EGM what new variation to use for the game denoted by GVN. Refer 9.2.
       Validity Check: VAR must be BCD [CP:653] & exist for the game (denoted by GVN) and be legal ('legal' w.r.t. the current EGM Configuration Poll settings (15.4.2), however the EGM has the option of applying the legality check in idle mode, refer 6.1.9). (see below *). (*RTEXT (7.10.1.16) = "VAR Unknown", or "VAR !BCD", "VAR Illegal"*).
       If the game with GVN has had its variation locked via the EGM Game Configuration Poll (15.4.3), then the EGM will not allow its variation to be changed [CP:654]. *QPv1.6*
       The EGM must log the "Game Variation Enabled" event (7.10.3.7) when an change of a game's variation actually takes place in idle mode or upon return to idle mode [CP:655] and update the game's PID (if applicable) [CP:656] and possibly the payscale and reel strips [CP:657].
       A variation change (15.4.4) must not affect the variation of a game in progress [CP:658].
       Refer GMNS for requirements pertaining to the game display upon a variation change.
       Previously played games stored and displayed in last play recall in audit mode, must not be affected by variation changes and still display correctly (e.g. correct reels for played variation) [CP:659].
GFLG   1 byte hex.
       Bits 0..6 reserved = 0, mask out these bits when reading any data from this byte.
       Bit 7,   Game Enable Flag (GEF, refer section 15.1.3) [CP:660]
            1 = enable game denoted by GVN, 0 = disable game denoted by GVN
PGID   New Linked Progressive Group I.D. (Refer *15.1.10*) to contribute to by the game identified by GVN & VAR (0x0001-0xFFFE), 2 bytes hex [CP:661]. If the game denoted by GVN has no LP levels (i.e. a PGID of 0xFFFF) then this field must be ignored and the PGID cannot be changed [CP:662].
       If the game denoted by GVN has one or more LP levels then:
            The message must be considered invalid if PGID is zero (see below) [CP:663].
            The message must be considered invalid if the GVN's PGID was a LP group but the new PGID is 0xFFFF (SAP) [CP:664].
       Note, you can only change from one LP group to another LP group. To change from a LP to SAP and vice versa, an EGM RAM clear must be performed.
       If the EGM is a shared progressive (refer 10.9) then a change to a PGID here to one game must be automatically applied by the EGM to all other progressive games. [CP:665]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

*The EGM will ignore this poll's entire 'message data' area if either the VAR, GVN or PGID field is incorrect or not applicable to the game or EGM and log the "Invalid Game Configuration" event (7.10.3.22) ^{CP:666}.

15.4.5　EGM Parameters Poll

This poll changes various parameters on the EGM and may be sent at any time by the SC. The EGM must not allow any of these parameters to be changed except via this poll. Any received changes to these parameters must take effect straight away if the EGM is in idle mode [CP:667] otherwise the EGM may either delay the changes until next return to idle mode or process them immediately, whichever is easier (or safer). For defaults values of fields in this poll, refer to section 8.1.17 on EGM Default Data.

Message Format and Order:

DLL	Data Link Layer, Refer Section 14.2
FC	Function Code = 0x2D, 1 byte hex
FLG	1 byte hex.

Bits 0..2 Reserved = 0, mask out these bits when reading any data from this byte.

Bit 3　　Reserved for NZ. *Ref 27/11/2009 (QPv1.6.4)*

Bit 4　　Reserved for Tattersalls. *QPv1.6.3*

Bit 5　　Reserve Feature Enabled Flag (RFEF) [CP:668] *QPv1.6*

If set, the EGM may offer a player instigated machine reserve feature at its discretion.

If not set, the EGM may not offer a player instigated machine reserve feature.

Bit 6　　Autoplay Flag. *QPv1.6*

If set, then holding down a play button on the EGM, must allow games to be played on the EGM without having to release the button between plays in order to start a new play; for as long as the button is held down [CP:669]. In addition, if a play button is being held down, then it must also not be a requirement to press take win upon a win, or to start a feature, or perform a win show. [CP:670] All features requiring player input must also auto-play (using an optimal strategy if applicable, otherwise a random selection must be taken) after a short timeout period (e.g. 0..5 secs). [CP:671] However, except for LP lockups[1], the autoplay does not have to function across lockups, CDC's or fault conditions.

If not set, then all play buttons must be released between plays.

Bit 7　　CRLIMIT Mode Flag. First see CRLIMIT below. *QPv1.6*

If set the EGM will also reject the insertion of notes that would cause the credit meter to exceed CRLIMIT [CP:672] (e.g. If the limit was $100 and the credit meter was $80 then a $50 note would be rejected but a $20 note would be accepted).

If not set CRLIMIT must operate only as defined in CRLIMIT below.

EGMs must support both modes of operation of CRLIMIT for backward compatibility with QPv1.5 EGMs.

OPR	1 byte hex (*QPv1.6. This field was BCD in QPv1.5*). Monitoring System Operator ID. This allows the EGM to execute any operator specific code, if any.

0x00　　Default/Unused

Licensed Monitoring Operator ID/licence number, as allocated by OLGR

0x10　　Denotes Tattersalls (VIC)

LWIN	Large Win Lockup Threshold. 4 bytes hex, in cents.

At the end of a play upon return to idle mode (refer section 3.3.2), if a single play's total win (including double up/gamble, free games, spins, features, SAP wins etc.) is greater than or equal to this amount, the EGM will enter a large win lockup to allow manual verification of the win amount [CP:673]. In QPv1.6, the EGM must not automatically instigate a payout of the win amount as this is now controlled by other parameters below. LWIN does not apply to Linked Progressive wins, but does apply to SAP wins. [CP:674]

(In QPv1.5 EGMs, a manual payment of a large win was optional. If a manual payment did result and the EGM was in cashless mode then an ECT from EGM would result instead of a Cancel Credit.)

---

[1] I.e. after a LP lockup has been cleared, play must continue without having to release the play button at any stage.

The Large Win Lockup Condition.

Upon entry into the Large Win lockup condition, the EGM Large Win Lockup Event must be logged (7.10.2.3) [CP:675] and the win amount must be added to the EGM's total wins meter (12.2) and multi-game/variation win meters (15.6.6). [CP:676] Once in the lockup, the amount of the win and the game result must be displayed clearly to the player and the attendant. [CP:677] Acceptance of the overall legibility of the Large Win Lockup display is at the discretion of the CEO [CP:678]. Upon lockup exit, the EGM may either return to play or instigate a payment of the win amount depending on the NPWINP & SAPWINP parameters below.

CRLIMIT	Credit-in Lockout Value. 4 bytes hex, in cents. When the credit meter is greater than or equal to this limit, credit entry to the machine (via coins/tokens, ~~or~~ and banknotes) must be disabled [CP:679]. Cash tickets must still be accepted for possible TITO system approval.

While disabled, the EGM must display an appropriate explanatory message in idle mode (refer GMNS). [CP:680]
CRLIMIT does not apply to ECTs [CP:681] or Ticket-In (re TITO) [CP:682].
This parameter may be used as a general hard currency disable as used in some jurisdictions near the end of operating hours.

DUMAX 1 byte hex.
  bits 0...3   0...15
  Maximum allowable number of consecutive Double-Ups/Gambles (or the equivalent) per play, if applicable [CP:683]. If the EGM receives a value for DUMAX greater than it can support in software then it must assume its maximum possible value [CP:684]. A value of zero simply disables double-up/gamble [CP:685]. The EGM may allow this to be set lower by an attendant if desired.
  bits 4...7   Reserved [CP:686]

DULIMIT Double-Up/Gamble Limit. 4 bytes hex, in cents. Double-up/Gamble (or the equivalent) must not be offered if the total amount won for the play would exceed this limit if the gamble was won [CP:687].

It is recommended for multi-jurisdiction support that DUMAX & DULIMIT are displayed dynamically to the player in either idle mode or on the game rules display or within the in double up/gamble feature.

TZADJ Time Zone Adjust. 2 bytes signed hex. Units of minutes. *QPv1.6*
  TZADJ added to the current system time (15.5.1) to get the local time for display purposes.
  All date and time stamps logged and displayed on the EGM must be adjusted by this amount except where stated otherwise. This includes, but is not limited to: [CP:688]
  • The player clock display time (15.5.1).
  • The local time as printed on a Cash-Out Ticket (22.3).
  • Last play recall time stamping.
  • Audit mode RTC display.
  • All displayable event queues except for the QCOM event queues (see below).
  Date and time stamps recorded with one value of TZADJ must not change if TZADJ is subsequently changed. [CP:689]

  The only exception is the logging of QCOM events and the display of the QCOM event queues in EGM audit mode ((4.2.13). QCOM events (refer 7.10) must not be adjusted by this field. [CP:690]
  In audit mode, the EGM must display this field in the format "+/- hh:mm". [CP:691]
  Validity Check: TZADJ must be within the range +/- 1440 else the field must be ignored. [CP:692]

PWRTIME Power-save Time-out Value. 4 bytes hex in seconds. *QPv1.6*
  The time in seconds after which and EGM will enter power-save with respect to the following conditions. If PWRTIME is not equal to zero, the EGM must enter power-

save mode if it is in idle mode with zero credit and a site disable is current (i.e. SEF = 0; refer 6.1.5) and there are no faults and all doors are closed and PWRTIME period has expired. [CP:693]

A PWRTIME value of zero is a special case and must cause power-save to occur instantly the next time the EGM is in idle mode with zero credit, no faults, all doors are closed. This is **regardless** *QPv1.6.2* of the current value of the SEF. [CP:694] This allows the option to put EGMs into power-save that are on the same network as EGMs that are in play.

The purpose of the above special case is to allow a gaming venue operator to put individual EGMs into power--save if e.g. they wanted to close off a section of the gaming floor and have those EGMs go into power-save. An operator couldn't do this in former versions of QCOM running on a multi-drop LAN as the operator they had to disable the entire EGM LAN (SEF=0) in order to put a single EGM into power-save.

A PWRTIME value of 2^32-1 must disable power-save. [CP:695]

Exiting power-save:

- If PWRTIME is not equal to zero, any activity on the EGM (button push, fault, audit mode access via key-switch, ECT-in, or door open), or a change in state of the SEF from 0 to 1 *QPv1.6.2* [CP:696], must reset the power-save timer and bring the EGM out of power-save.
- If PWRTIME=0 there must be no instantaneous exit/re-entries to power-save (e.g. on button press or SEF 0 to 1 transition) [CP:697]
- If PWRTIME = 0 then only leaving idle mode (e.g. a change in GSR:STATE from Idle (e.g. System Lockup) or a change in state of the EGM to a fault condition, door open, audit mode access or reboot) will temporarily bring the EGM out of power-save until the next return to idle mode. [CP:698]
- Any change in value of PWRTIME to a non-zero value must reset the current power-save timer and therefore also cause the EGM to exit power-save if active *QPv1.6.2*. [CP:699]

As the SEF may now also equal one (site enable) during power-save (see special case above), the EGM must ensure that physical credit input devices are specifically disabled for credit input during power-save. *QPv1.6.2* [CP:700]

PID   Player Information Display ID. [CP:701] 1 byte hex. *QPv1.6*
Denotes which PID version the EGM must use.
      0x00 disables the PID
      0x01 denotes QLD version PID
      0x02 denotes VIC version PID
      0x03 denotes NZ version PID
      0x04 denotes TAS version PID
Other version ID's will be added as required.
Undefined values (e.g. 0xFF) must disable the PID
Changes must take effect on the EGM no later than the next return to idle mode.

EODT   End of Day Time. 2 bytes hex. *QPv1.6*
This field informs the EGM of the time of the each day when the system is signs off the day for accounting purposes. This field may be used by the EGM at its discretion. [CP:702]

In units of minutes since midnight. Valid range: 0…1439 minutes.
If out of range the EGM must ignore the EOD field. [CP:703]
(NB: When displaying this field, always display it as "hh:mm" 24 hour format.)

NPWINP   Non- Progressive Win Payout Threshold. 4 bytes hex, in cents. *QPv1.6*
Once per play, if a single play's total win at the end of a play (including double up/gamble, free games, spins, features, etc. Refer section 3.3.2) is greater than or equal to this amount, the EGM must automatically instigate a payment of the win amount [CP:704]. Payment of the win amount must be either an ECT, hopper pay, cancel

credit, or a ticket out, depending on the EGM's current hardware configuration, cashless mode (16.2) and other payout thresholds (refer 3.5 & 15.4.17) [CP:705]. I.e. the EGM will behave as if its collect button has been pressed except it will pay out only the win amount and not the whole credit meter amount. More requirements below.

SAPWINP   Stand Alone Progressive Win Payout Threshold. 4 bytes hex, in cents. *QPv1.6* Applicable to EGMs with SAP games only.

Once per play, if a single play's total SAP win at the end of a play is greater than or equal to this amount (refer section 3.3.2), the EGM must automatically instigate a payment of the SAP win amount [CP:706]. Payment of the SAP win amount must be either an ECT, hopper pay, cancel credit, or a ticket out, depending on the EGM's current hardware configuration, cashless mode (16.2) and other payout thresholds (refer 3.5 & 15.4.17) [CP:707]. I.e. the EGM will behave as if its collect button has been pressed except it will pay out only the SAP win amount and not the whole credit meter amount. If both the LWIN and SAPWINP thresholds are exceeded, then the SAP win payout must not occur until after the large win lockup has been cleared. [CP:708]

If both NPWINP & SAPWINP thresholds are exceeded for a single play then a payout must occur for the combined win amount. [CP:709]

Any win payout that occurs must be correctly accounted for in last play recall, as per GMNS. [CP:710]

Upon entry into any win payout as a result of a win exceeding either NPWINP or SAPWINP, the EGM must pause and display the following message for 5-10 seconds:

"Total win exceeds threshold – paying out win". [CP:711]

While this message is being displayed the EGM's state (refer 15.6.1) should still be on of the 'Play in Progress' states.

As a result of the above two new thresholds, in order to implement payouts of win amounts, to avoid confusion, all payout routines via hopper, or ticket-out, or cancel credit, or ECT, must not assume anywhere that the amount being paid is equal to the credit meter (especially from the aesthetic point of view). [CP:712]

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.4.6 Progressive Configuration Poll

New for QPv1.6. This poll has two functions: It allows a SC to change a game's current SAP Configuration (provided the EGM supports this feature) and it is used to inform the EGM of the current parameters being used for its LP levels (if any). [CP:713] (Sending LP level information in this poll is primarily for display purposes in audit mode of the EGM and MINRTP/MAXRTP threshold validation by the EGM. Refer 15.4.2). Support for reconfigurable SAPs (CustomSAP) is optional for the EGM manufacturer and this is indicated by the EGM via the Progressive Configuration Response (15.6.4) for each game with a SAP component. This poll must not be required by the EGM in order to configure or enable a progressive game initially after a RAM clear.

If the game's variation is locked (refer 15.6.11) then this must also lock the SAP configuration and it cannot be changed [CP:714].

This poll's SAP 'message data' must be ignored by the EGM if; the game doesn't support CustomSAPs, or if the EGM is not in idle mode, or it does not have zero credit on the credit meter upon the processing of this poll message [CP:715], or if the game has not yet been configured via the EGM Game Configuration Poll (15.4.3) [CP:716].

This poll's LP 'message data' must be ignored by the EGM only if the game doesn't support LPs, or if the game has not yet been configured via the EGM Game Configuration Poll (15.4.3) [CP:717].

If any data in the poll is accepted, changes must be immediately reflected in the protocol. [CP:718]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x2C, 1 byte hex
GVN   Game Version number (*15.1.5*), 2 bytes hex. Identifies the game being re-configured.
        Validity Check: GVN must exist (on fail ignore poll message data) and is a progressive game. [CP:719] (*RTEXT (7.10.1.16) = "GVN Unknown").*
RES   Reserved 2 bytes hex.
FLG   4 bytes hex.
        Bits 0...32 reserved = 0, mask out these bits when reading any data from this byte.
NUM   Number of repeated entries in this poll below equal to the total number of progressive levels in the game GVN. 1 byte hex. 0…8
        Validity Check: The message must be considered invalid if NUM is > 8 or incorrect for the game (GVN) [CP:720] (*RTEXT (7.10.1.16) = "NUM Invalid").*
SIZ   1 byte hex. Repeated entry size in bytes = 17 [CP:721]

The following data is repeated as indicated by the NUM field. There may be up to eight progressive levels in this poll and the levels must be sent in order from lowest level number ID (0), to highest. [CP:722]

PLF   1 byte hex. Level Flags
            Bits 0..2 Level number ID to apply changes to re the following parameters. 0…7
            Bits 3..7 Reserved = 0
SUP   New Jackpot level start-up amount. 4 bytes hex in cents.
            SUP is the amount that the jackpot level's current amount resets to after each hit of the level (provided there was no overflow to carry over).
PINC   New Jackpot level percentage increment x 10000. 4 bytes hex.
            E.g. 2.5385% is reported as 25385
            PINC is applied to applicable game turnover and is used to increment the jackpot level's current amount or overflow meters. Be advised that PINC = 0 will require special consideration in source code to avoid for example 'divide by zero' errors, or other anomalies.
CEIL   New Jackpot Level Ceiling. 4 bytes hex in cents.
            CEIL denotes the maximum value of the jackpot level's current amount. Any contribution in excess of this current amount must go into an overflow meter.
AUXRTP   4 bytes hex. New Auxiliary RTP for the level x 10000

Some games may have an additional component to the basic, or traditional progressive jackpot, whose RTP is represented by this field. (E.g., a hidden increment of the jackpot level's overflow meter). For the basic, or the traditional progressive jackpots, AUXRTP is zero. *(NB most systems do not currently support LP totalisation including a hidden increment. EGM manufacturers should check before implementing a LP game with a hidden increment)*

So the total percentage RTP of a vanilla* progressive jackpot level is

$$RTP_{total} = (PINC / 10000) + SUPRTP + (AUXRTP / 10000)$$

where SUPRTP is the RTP of the jackpot startup amount with respect to the jackpots hit rate, i.e. SUPRTP = HRATE * SUP * 100.0. *To verify an implementation of the* SUPRTP *formula is working, set HRATE = 1.0 & SUP = 1cent, then SUPRTP should equal 100 which equates to 100%. *This formula is a typical example but may vary depending on game design so should not be taken as a rule.*

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Before accepting a new progressive configuration, the following additional validity checks must be performed by the EGM:

- ~~No SUP amount must be equal to zero. (*RTEXT (7.10.1.16) = "Lev x SUP=0" where "x" is substituted with the level ID*) CP:723~~
- ~~SUP level n >= SUP level n+1 for each level (where a next level exists). (*RTEXT (7.10.1.16) = "SUPn < SUPn+1" ('n' is not required to be resolved for this RTEXT message*)). CP:724~~
- SUPRTP must be <= MAXRTP / 100. (*RTEXT (7.10.1.16) = "SUPRTP>MAXRTP"*). CP:725
- PINC / 10000 must be <= MAXRTP / 100. (*RTEXT (7.10.1.16) = "PINC > MAXRTP"*). CP:726
- CEIL must be set at least greater than or equal to the 95% Confidence Interval w.r.t. to the ceiling hit rate (Refer OLGR's Jackpot System Minimum Requirements or the QCOM SDK for formulas/functions. AUXRTP does not have to be taken into consideration for this calculation at this time). Apart from MAXPWIN for SAP levels, the EGM must not place an upper limit on the ceiling value (as this is better for multi-jurisdiction support). (*RTEXT (7.10.1.16) = "lev x CEIL95%%"* where "x" is substituted with the level ID). CP:727 CP:728
- The EGM must verify the new total theoretical RTP for the game with the above progressive configuration ~~must be~~is in the range ~~of~~ MINRTP ~~&~~to MAXRTP inclusive. (~~Refer~~ Related: 15.4.2 & 7.10.3.34~~)~~. ~~(Refer section 7.10.3.34, the OLGR's Jackpot System Minimum Requirements, or the QCOM SDK for formulas)~~ (*RTEXT (7.10.1.16) = "TRTP OutOfRange"*). CP:729 ~~The total game RTP as reported by QSIM should be used as a benchmark here.~~

Upon failure of any validity checks, the EGM must reject the poll message data and log the Invalid Progressive Configuration event (7.10.3.23). CP:730

If the new progressive configuration is accepted and actually is a change, then the EGM must log the "Progressive Configuration Changed" event (7.10.3.34). CP:731

Also, for each changed SAP level the EGM must:

- Carry over total contribution (including any overflow) of each changed SAP level to the corresponding new SAP jackpot level. CP:732 (note that a change in PINC for a SAP level must not affect its current contribution CP:733, but depending on the new SAP jackpot parameters, a backward movement in the current jackpot amounts is possible. (see example below)
- Calculate a new 'Initial Contribution' using the SAP reconciliation formula applicable to the game. (This to ensure SAP level reconciliation is not broken by the change of SAP level parameters. Refer 10.4.1 for more information.)

E.g. for a single SAP level:

	Old Parameters	New Parameters
SUP	$50	$20
PINC	1%	0.5%
CEIL	$100	$100
Current Contribution	*$75*	*$75 (must not change)*
Current Amount	*$100 ($25 in overflow)*	*$95 ($0 in overflow)*

If the EGM is a shared progressive (refer 10.9) then any change to progressive configuration successfully made by this poll to one game, must be automatically applied by the EGM to all other progressive games. CP:734

Changes may be confirmed by a SC by requesting the Progressive Configuration Response (15.6.4).

Note that a SAP level's trigger rate is not re-configurable at this time and must be locked to the game's variation or game. Also refer 9.2.

15.4.7   External Jackpot Information Poll (EXTJIP)

QPv1.6. This poll informs the EGM regarding details of any external jackpot systems it has been placed in. External jackpot systems have nothing to do with the resident games or jackpots triggered by the EGM, but they are an additional feature and RTP made available by playing the EGM, external to the EGM. These types of jackpot systems usually make use of the System Lockup Request Poll (15.4.9) to lockup the EGM upon a win and often use QCOM ECT feature (section 16) to transfer small wins to the EGMs credit meter.

The reason for informing the EGM of the jackpot details is that some jurisdictions require external jackpot details (e.g. Total combined RTP of all external jackpots) to be displayed on the EGMs PID. Refer to local jurisdictional PID requirements for more information on PIDs. Some jurisdictions also require the EGM to be able to display current external jackpot details and values on demand.

The EGM must not use this information with regard to any MINRTP / MAXRTP threshold validation (15.4.2). [CP:735]

The EGM must also display the information received from the last External Jackpot Information Poll in audit mode and store the last received details from this poll in NVRAM. [CP:736]

This poll may be sent to the EGM with information of up to a maximum of eight external jackpot progressive group/level combinations. This poll replaces any previously sent data regarding external jackpots. [CP:737]

Message Format and Order:

DLL	Data Link Layer, Refer Section 14.2
FC	Function Code = 0x2A, 1 bytes hex
RES	4 bytes hex. Reserved.
RTP	Total Percentage RTP of the external Jackpots on this EGM.
	2 bytes hex. %RTP x 100. i.e. 12.34% is represented by 1234 (0x04D2)
	An RTP of 0 here denotes that the EGM is no longer participating in an external jackpot and all other fields in this poll must be ignored except for the SIZ field. The EGM must reset all data pertaining to external jackpots (refer section 8.1). [CP:738]
FLG	Flag Byte. 2 bytes hex.

       Bits 0…3   Levels.
                               Total number of levels in the poll and the external jackpot (0…8)
                               The message must be considered invalid if 'Levels' is > 8 [CP:739]
                               It is possible for the RTP field to be non-zero and the number of levels to be zero. This is still valid and just means that the SC won't be providing the EGM with specific level details for display purposes. Refer Display Flag below.

       Bites 4…6   Reserved
       Bit 7   Display Flag. See below. [CP:740]
       Bits 8…11   Icon Display ID.
                               Support for this field is optional in QLD.
                               This field allows for the display or non display of an Icon on the EGM screen when participating in an external jackpot (Refer RTP field above). This flag is independent of the Display Flag (above) which controls the showing of the external level amounts, this way it is possible to have none, one or both on the screen at the same time. [CP:741]

                               An Icon Display ID of zero must disable the Icon Display. [CP:742]
                               A non-zero Icon Display ID enables the Icon Display and the value is an index into a range of possible hard-coded icons in the EGM. If there is no associated icon for the ID provided, then this is equivalent to an Icon Display ID of zero. [CP:743]
                               Every time a specific icon display is enabled (when it was not already enabled and provided one exists for the given ID), the EGM must log the "EGM EXTJIP Icon Display Enabled" event (7.10.3.38) with

'*ICONDESCR*' = the text name or abbreviation of the enabled icon. [CP:744]

When displaying both an icon and EXTJIP meters, the two must be displayed in close proximity to each other. [CP:745]
A 50% transparent display is preferred for the EXTJIP data/icon in order to make the display less obtrusive the EGM's own game. Acceptance of the overall legibility of the display is at the discretion of the CEO [CP:746].
Default at EGM RAM clear = 0. [CP:747]

Bits 12…15   Reserved

SIZ   1 byte hex
Size in bytes of each repeated entry below = 20. [CP:748]

The following data is repeated for each external jackpot level by the number of times indicated in the FLG field.

EPGID   External Jackpot Progressive Group ID. 2 bytes hex.
A SC may elect to broadcast external jackpot information current amounts out via the LP Broadcast Messages (refer 15.5.2). This field denotes the PGID that it will use for the level denoted by LFLG and LNAME below. Each level may have a unique PGID. [CP:749] This field has nothing to do with any PGID field appearing in any Game/Progressive Configuration polls/responses and only is applicable to external jackpots that appear in broadcast messages for the EGM's information.
Validity check: PGID must not be 0 or 0xFFFF. The EGM must ignore the level if a given PGID is invalid. [CP:750]
When a PGID is set initially or is subsequently changed for a given level, the EGM must stop displaying any associated current amounts in idle or audit modes (refer Display Flag below) until an update for the PGID is received via the LP Broadcast Messages (refer 15.5.2). [CP:751]

LFLG   2 bytes hex
Bits 0…2 level ID. 0…7 [CP:752]
Bits 3…14 reserved [CP:753]
Bit 15   Unit Modifier Flag.
Support for this field is optional in QLD.
If this flag is set (1), then the EGMit must modify the units of the current amounts associated with the display of EXTJIP data for the above level ID, from the default currency display (i.e. "$xx,xxx,xxx.xx") to unit-less (i.e. "x,xxx,xxx,xxx"). [CP:754]
(Commas are required in both types of meter displays)
This flag must control the display of all current amounts associated with the EXTJIP level, i.e. display of EXTJIP level data in idle mode and audit mode. [CP:755]
Default at EGM RAM clear = 0 i.e. display format is in $.c. [CP:756]
Any change in state to the EGM's current value for this flag must result in the Manuf. Spec. Event being logged by the EGM (7.10.1.16) with RTEXT = "EXTJIP:UMF:%u%s" where %u = level ID (0…7) and %s = "$" or "nil" accordingly. [CP:757]

LNAME   Name of the external jackpot group level (e.g. "Mega", "Maxi", "Minor" etc)
16 bytes. Comprised of up to 15 ASCII printable characters (*2.3.8*). [CP:758] The field must include at least oneing the null terminating character (0x00) in the last position. [CP:759] The field string, if less than max width, must be null terminated and right padded with 0x00's.
Validity Checks: EGM must verify in the following order: [CP:760]
1. There at least one is a null terminating character in the field (i.e. the last character)
2. All characters are ASCII printable (refer *2.3.8*), except for NULL characters
On fail the EGM assumes no name for the external jackpot. [CP:761]
This field may be all zero indicating it has no name. [CP:762]

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

<u>Display Flag (refer FLG field above)</u>

If this flag is set and the EGM has also been provided with level details (refer PGID, level ID and level name fields above), the EGM must display the last received current amounts (via LP broadcasts – section 15.5.2) and jackpot level names (LNAME) to the player at least during idle mode (only mandatory for the current game and game selection idle mode displays) and during game play (double up & second screen features may be exempted). Refer Appendix A [CP:763]. The overall display must be prominent and cannot be cycled with other messages [CP:764]. The display line/box must be of sufficient size to display at least one level at a time [CP:765]. The display may be scrolled as required to display each level in turn. Each level must be displayed for at least 4 secs before continuing/scrolling on to the next level, if any [CP:766]. The colour of the text must be suitable for the background it is placed on. A transparent display is preferred in order to make the display less obtrusive the EGM's game. Acceptance of the overall legibility of the display is at the discretion of the CEO [CP:767].

After any EGM power up, or upon receipt of a poll of this type, the EGM must not display to the player a current jackpot amount configured for display by the External Jackpot Information Poll until after the next LP Broadcast update (section 15.5.2) for the applicable level amount has been received. [CP:768] (This is on a per level basis.) This prevents the EGM from displaying aged amounts to a player. Also refer to section 15.5.2 for more information regarding LP broadcast processing.

Sample jackpot level displays:

<div align="center">

Mega Dollars $12,345,678.90

or

Lots of Money
$12,345,678.90

(if UMF=1) then,

Mega Points 1,234,567,890

or

Lots of Points
1,234,567,890

</div>

~~NB: It was not the original intent for the EXTJIP display feature to be the sole display of a jackpot system's current amounts, this is because the display of EXTJIP current amounts are allowed to be cycled on an EGM's display which means it does not meet OLGR requirements pertaining to the display of jackpot current amounts. If this is not the case for a particular EGM and the EXTJIP display meets all other requirements then there is no issue in the EXTJIP display being a sole display of jackpot current amounts. In addition, Monitoring Systems also have additionals options: refer to the OLGR QCOM Site Controller Procedures document for more information in this regard.~~

15.4.8   Program Hash Request Poll

This poll commands the EGM to initiate a program hash calculation using the given seed [CP:769] or last known seed [CP:770] depending on RES:Bit 6 field below. Refer Section 11 on Program Hashes for more general information on program hashes.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x29, 1 byte hex
RES   1 byte hex.
       Bits 0..5   Reserved = 0, mask out these bits when reading any data from this byte.
       Bit 6   if set, SEED field is present
             if not set, SEED field is not present in the message
       Bit 7   Machine Enable Flag (MEF) (15.1.2) [CP:771]
             1 = EGM enable, 0 = EGM disable
SEED Hash Algorithm Seed, LSB first, 20 bytes hex [CP:772]
       (QPv1.5 was 8 bytes hex and only the first 4 LSBs were used)
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.4.9   System Lockup Request Poll

This poll is used by the SC when the system wishes to award a system triggered prize through the EGM or lockup the EGM for some other reason as specified in the TEXT field (e.g. to indicate an important monitoring system event or condition). It can also be used to pose a question to the user and retrieve a response.

Upon receipt of this message, the EGM will enter the System Lockup condition if the EGM is in idle mode [CP:773] otherwise it must enter the System Lockup upon next return to idle mode (refer 3.3.2) [CP:774].

Upon entry into the lockup, the EGM will set the System Lockup state in the General Status Response (15.6.1) [CP:775], prominently display [CP:776] the TEXT field message data to the player and possibly (depending on FLG field bit 7) play a jackpot fanfare. The message data and lockup state must be stored in NV-RAM [CP:777]. The EGM must not display a lockup title (i.e. "System Lockup") [(QPv 1.6.4)], or "Call Attendant" or the equivalent for this lockup [CP:778]. However, in a low prominence position (and minimum permissible font size) the EGM must display the text "QCOMSL" for the duration of the System Lockup *(the text must not be cycled with other messages)* [CP:779].

The EGM will remain locked up until reset by the EGM General Reset Poll (15.4.19) [CP:780] or manual key-switch [CP:781]. In addition, having either of the 'Continue' and 'Question' bits set (see below) will also allow the user to reset the EGM from this lockup.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x27, 1 byte hex
FLG   Flag, 1 byte hex, format.
     bits 0..2 reserved = 0, mask out these bits when reading any data from this byte [CP:782]
       bit 3   Reset Key Disable Flag. [CP:783] *QPv1.6*
           If set, the EGM must disable the lockup reset key-switch's effect on the System Lockup for the duration of the lockup. I.e. turning the reset key on the EGM will have no effect on the System Lockup. However, if a fault condition occurred while in the System Lockup, then the Reset key-switch would still allow the fault condition to be cleared as normal and then disable again upon return to the lockup. [CP:784]
           If not set, then the EGM will allow the System Lockup to be cleared by either the EGM General Reset Poll or manual key-switch as normal. [CP:785]
       bit 4   Continue. *QPv1.6*
           If set, once in the lockup the EGM will (in addition to displaying the TEXT field message below) also prominently display the words "Press XXXX to continue", where "XXXX" in a non-touch screen EGM is a reference to a suitably lit button on the EGM available to the player. Once the button is pressed, the EGM must exit the system lockup condition. [CP:786]
       bit 5   Question. *QPv1.6*
           If set, once in the lockup the EGM will (in addition to displaying the TEXT field message below) also prominently display the words "YES" and "NO" in close proximity to the TEXT field message data and prompt for a response via the button panel (lit appropriately) or touch screen. Once an input is received the EGM will log the system lockup response unnumbered event (7.10.4) which indicates which choice was chosen and exit the lockup. At all times the lockup may still be reset-able in the usual fashion without a response being given. [CP:787]
           If both the 'question' and 'continue' bits are set, then the EGM must behave as per 'question', but with the addition of a 'cancel' button which just exits the System Lockup if pressed.
       bit 6   Lamp Test. *QPv1.6*
           If set, then upon entry into the lockup and for the duration of the system lockup the EGM must turn on all its lamps. This is used for lamp testing. The lamp test feature may be implemented at the discretion of the EGM manufacturer but implementation is recommended for manufacturers wishing to supply EGMs to Casinos. [CP:788]
       bit 7   Fanfare.

> If set, the EGM must also trigger a short jackpot fanfare upon entry into the lockup [CP:789]. The fanfare preferably should be no more than 2-3 seconds duration. Acceptance of the fanfare sound used is at the discretion of the CEO (refer 3.6.3) [CP:790].

LEN   1 byte hex.
Length of text field in bytes
Maximum length is 80 bytes [CP:791].
Before processing, the EGM must verify LEN <= 80, if false the System Lockup request must be ignored [CP:792]

TEXT   LEN bytes of ASCII printable characters. Refer Section (*2.3.8*). [CP:793] Before processing the EGM must verify all characters are printable (refer *2.3.8*), on fail, the System Lockup request must be ignored [CP:794]

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

If the TEXT message is too long to display all at once the EGM should scroll it across the display. As a last resort, the EGM may break the message at space characters and display consecutively (separated with trailing and leading "...") [CP:795]. All required messages to be displayed by System Lockup, must be displayed constantly (i.e. they must not be cycled on the display with other messages) *(QPv1.6).* [CP:796] To make this easier, it is acceptable for the system lockup display to overwrite the play outcome display area (e.g. reels display), but it must not overwrite any meter displays or other text messages. SPAM (15.4.20) must still be displayed during the SL as per section 15.4.20 [CP:797].

A note regarding the prominence of this feature:

Ideally the lockup should appear as a box overwriting the game result window (e.g. the reel display) with the required information displayed clearly inside. If the display box can be made transparent then this would be ideal. All meters on the game display must remain visible during the lockup. See examples below.

The display must be legible. Acceptance of the overall legibility of the System Lockup text messages on the EGM display is at the discretion of the CEO [CP:798].

The EGM must ignore further System Lockup Requests polls until the current lockup has been carried out and cleared [CP:799]. *(If a SC wishes to generate multiple consecutive system lockups, then it is recommended it also disable the EGM to prevent games being played between lockups.)*

(QPv1.6) If a System Lockup is delayed because the EGM is not currently in idle mode (e.g. an EGM with a long feature such as retriggerable free games may take a while to return), the EGM must display the message "LOCKUP PENDING…" to the patron/attendant under the same display requirements as SPAM (15.4.20), until such time as the EGM returns to idle mode and enters the System Lockup as per the requirements in this section. [CP:800]

System Lockup audit log. *(QPv 1.6.2)*

The EGM must provide in audit mode an audit log of at least the last 10 System Lockups (SL's). The log must contain the following data for each SL:

- **Time.** System Lockup Poll received time (date and time-stamp, TZADJ adjusted).
- **Delay.** (seconds) The amount of time before the EGM actually entered the SL after receipt of the SL poll (i.e. as the lockup may be delayed if the EGM was not in idle mode)
- **TEXT** field data (see above).

Examples: Using TEXT =

"System Lockup!\"#$%&\'()*+,-./0123456789:;<=>?@ABCXYZ[\\]_`abcxyz SOI ilo %s:;34END"

Figure 4 Example of a System Lockup implementation*

Screen dump courtesy of IGT (Australia) PTY LTD.

Figure 5 Example of a System Lockup with the Yes/ No feature*

Screen dump courtesy of IGT (Australia) PTY LTD.

Figure 6 Example of a System Lockup with the 'continue' feature*

Screen dump courtesy of IGT (Australia) PTY LTD.

- **NB the examples above have not been updated to confirm to QPv1.6.4 which does not allow a title on the lockup, or show the required display of the text "QCOMSL".*

15.4.10 Cash Ticket Out Request Acknowledgement Poll

New for QPv1.6.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x21, 1 byte hex
FLG   1 byte hex.

> Bits 0…5   reserved [CP:801]
> Bit 6   On-Failure Flag [CP:802]
>> This bit is ignored unless Bit 7 below is set to the fail state (= 0).
>> If set, then upon exit from the TO Lockup the EGM will automatically enter a Cancel Credit Lockup.
>> If not set, then upon exit from the TO Lockup the EGM will simply return to the previous state/mode.
> Bit 7   Success/Fail Flag [CP:803]
>> If set, it denotes that the cash ticket out request has been approved, the EGM must proceed **in the following order** to: process the remaining data in this poll; then if all ok, deduct the ticket value from the EGM's credit meter, update and queue the Cash Ticket Out group meter (refer 12.1.1), display the message "**Printing Ticket for $xx.xx…Please Wait**", update the STATE field (refer 15.6.1) and then finally instigate the ticket print with the required details (refer 22.3). [CP:804] After printing the ticket, the EGM must exit the TO Lockup. This completes the current cash ticket out transaction. After "successfully" printing a ticket (refer section 22.1.3 for the definition of a successful cash ticket print) the EGM must log the "EGM Cash Ticket Out Print Successful" event (7.10.4.12). If problems occur during printing a ticket refer section 22.1.1 for the procedure.
>>
>> If not set, this indicates that the system is not approving the ticket out, the EGM must ignore all other fields in this poll, pause and display for 5 seconds the message "**Cash Ticket Out Denied**" and then exit/return from the TO Lockup condition with respect to the "On-Failure Flag" above. [CP:805]

RES   4 bytes hex. Reserved.
TSER   Ticket Serial Number. 2 bytes hex, from last Cash Ticket Out Request Event. Refer section (7.10.4.11) for more information regarding TSER (e.g. when it is incremented). The TSER in this poll must not overwrite the EGM's TSER variable, it is for cross-checking purposes only. See following requirement.

> If TSER is incorrect with respect to the current value of the EGM's TSER variable and the success bit (FLG bit) was set, the EGM must ignore the acknowledgement poll and log the Invalid Ticket Out Acknowledgement Event (7.10.3.36), and remain in the lockup and await another TO Ack Poll. [CP:806]

TAMT   Ticket Out Amount

> 4 bytes hex in cents, amount of ticket out, LSB first, display as decimal currency of up to 11 characters. If TAMT is incorrect with respect to the instigating "Ticket Out Request" unnumbered event (7.10.4.11) and the success bit (FLG bit) was set, the EGM must ignore the acknowledgement poll and log the Invalid Ticket Out Acknowledgement Event (7.10.3.36), and remain in the lockup and await another TO Ack Poll. [CP:807]

TIME   Transaction time. Seconds, Minutes, Hours, 1 byte each, BCD, 24 hour format.
DATE   Transaction date. Day, Month, Year, 1 byte each, BCD

> System Time of Transaction in QCOM date and time format which must be printed on ticket by the EGM (this time must not be adjusted by TZADJ by the EGM). [CP:808]
> It is expected that the TITO system will populate DATE / TIME with the local gaming venue time of the TO transaction AUTHNO denotes.
> The EGM must ensure that the date and time fields are BCD and within valid ranges (i.e. 1…60, 0…23 etc). [CP:809] On failure the EGM will must use it the current local time (this is the last rx broadcast message date and time adjusted by TZADJ by the EGM)

^{CP:810}. No other ~~reasonableness~~ sanity check~~s~~s ~~is~~ are to be ~~placed~~ performed on these two fields by the EGM, nor is the EGM to set its clock via this timestamp. ^{CP:811}

AUTHNO   Ticket Authorisation Number, 16 bytes hex, LSB first.

A proportion of this number (refer 22.3 for specifics) ~~(depending on the number of digits supported by the EGM's adopted barcode format)~~ must be encoded into the ticket's barcode and utilised digits (matching the barcode) must also displayed on the ticket in a suitable location in decimal ~~(refer 22.3 for specifics)~~. ^{CP:812}

E.g. if the AUTHNO received was 0x00000000000000008000000000000000 (shown here as an MSB first hex number) the corresponding barcode value encoded and displayed on the ticket is the decimal equivalent, i.e. "57-6460-7523-0342-3488".

TITO system developers please refer to section 22.5 for information regarding AUTHNO generation.

RES   4 bytes hex. Reserved

CLEN   1 byte hex. Length of CTEXT field in bytes

Maximum length is 80 bytes ^{CP:813}.

Before processing, the EGM must verify CLEN <= 80, if false then CTEXT must be ignored ^{CP:814}

CTEXT   Arbitrary Certification String or Message.

CLEN bytes of ASCII printable characters. ^{CP:815} Refer Section *(2.3.8)*.

This text message if present must be clearly displayed on the printed ticket.

An example message could be one related to responsible gaming

Before processing the EGM must verify all characters are printable (refer *2.3.8*), on fail, the CTEXT field stored by the EGM ~~is~~must be ~~ignored~~set to an empty string ^{CP:816}

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

A SC will send this poll in response to each new valid Cash Ticket Out Request Event (7.10.4.11) received from the EGM. This poll ACKs or NAKs the EGM's last Cash Ticket Out request.

EGMs must ignore this poll if it is not currently in a Cash Ticket Out lockup condition at the time. ^{CP:817}

If the EGM is in a higher priority condition, such as a door open or fault condition then it will store the acknowledgement in NV RAM and action it upon return to the Cash Ticket Out lockup condition. ^{CP:818}

~~Duplicate Cash Ticket Out Acknowledgement Polls must be ignored by the EGM.~~If a poll of this type is currently stored / pending or being actioned, then receiving another poll of this type must have no effect ^{CP:819}

The EGM must be sure to only accept one valid Cash Ticket Out Ack Poll per Cash Ticket Out lockup condition. ^{CP:820}

Refer section 22 for more information and requirements on TITO.

15.4.11   Cash Ticket In Request Acknowledgement Poll

New for QPv1.6.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x20, 1 byte hex
FLG   4 bytes hex, reserved.
FCODE Success/Fail. 1 byte hex.

> If zero, the last cash ticket in request was successful, the EGM must, in the following order, validate TAMT (see TAMT below) and ensure AUTHNO below agrees with expected, if ok, the EGM must reset the appropriate flag in the "General Status Response" (15.6.1) and retain/stack the ticket. Once the EGM has successfully stacked the ticket, it must then add the amount of the ticket (TAMT) to the credit meter, update and queue the "Cash Ticket In" QCOM Group meter (12.1.1) and. [CP:821] The EGM must make a short cash-in sound upon adding any ticket in amount to the credit meter. [CP:822] This completes the ticket in transaction.
> Also refer sections 22.2 Ticket In Process and 22.2.2 Interruptions to the TI Process.
>
> If non-zero, this indicates a denial of the EGM's last Ticket In request, the EGM must ignore the remaining fields in this poll, reset the appropriate flag in the "General Status Response" (15.6.1) and reject back to the player the Cash Ticket-In currently waiting for acknowledgement. [CP:823]
> The reason for the failure is denoted by the value of FCODE. Refer 22.2 for the table of failure codes.

TAMT   Ticket In Amount.
> 4 bytes hex in cents, LSB first, display as decimal currency of up to 11 characters.
> TAMT must not be zero.
> TAMT must be less than MAXECT.
> TAMT plus the EGM's current credit meter value must not overflow (or rollover) the EGM's credit meter, or exceed the EGM's credit meter display capabilities, or any implemented EGM credit meter limit.
> (No other limits may be applied (e.g. CRLIMIT); the TITO system will manage any type of limits relating to TITO.)
> Upon failure of any of the above, the EGM must **abort** the transaction as per section 22.2.2. [CP:824]
> If the reason for the abort is due to the credit meter display capabilities being exceeded, or any arbitrary EGM implemented credit meter limit, then the EGM must also log the 'Transaction Denied - Credit Limit Reached' event (7.10.3.44)[CP:825]

AUTHNO   Ticket Authorisation Number, 16 bytes hex, LSB first.
> This number is from the last Ticket-In Request Event (7.10.4.13) echoed back. The AUTHNO field must only be validated by the EGM when FCODE = 0 (success). The value received in this poll must not be saved (it is only to be compared against). [CP:826]
> If this number does not equal the AUTHNO for the ticket currently being authenticated, the EGM must simply ignore the poll message data and continue waiting for a new Ticket Ack. poll for the remainder of the TI timeout period (22.2.1) [CP:827]

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

A SC will send this poll in response to each Cash Ticket In Request Event (7.10.4.13) received from the EGM. This poll ACKs or NAKs the EGM's last Cash Ticket In request.

EGM must ignore this poll if it is not currently awaiting acknowledgement for a new cash ticket in at the time. [CP:828]

If a poll of this type is currently stored / pending or being actioned, then receiving another poll of this type must have no effect Duplicate Cash Ticket In Acknowledgement Polls must be automatically ignored by the EGM. [CP:829]

The EGM must be sure to only accept one valid Cash Ticket In Ack Poll per cash ticket in. [CP:830]

Refer section 22 for more information on TITO.

15.4.12 Cash-Out Request Poll

(QPv1.5: This poll was previously named the 'Cancel Credit Lockup Request Poll')

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x1E, 1 byte hex
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

QPv1.5: Upon receipt of this poll, if the EGM is currently in idle mode and the credit meter is not zero, the EGM must immediately enter a cancel credit lockup condition [CP:831]. If the EGM was not currently in idle mode (e.g. in a fault/lockup/door open/audit mode/test mode/play [CP:832]), it will store the request in NV-RAM (so long as there is currently credit on the credit meter) and enter the cancel credit lockup condition immediately upon next return to idle mode (refer 3.3.2) if there is credit on the credit meter [CP:833]. If the EGM was already in a cancel credit lockup, it must ignore the Cancel Credit Lockup Request Poll [CP:834]. If the EGM was in cashless mode (16.2) the EGM shall instigate a cashless transfer as per ECT requirements instead of a Cancel Credit [CP:835].

QPv1.6: This poll must simply queue a 'collect' (as if the collect button was pressed) upon next return to idle mode (refer 3.3.2) if the credit meter is not zero at that time. [CP:836] If the EGM is currently in idle mode with credit, then the EGM must behave as if the collect button was just pressed. [CP:837]

NB. With the new version of this command, to force a Cancel Credit on an EGM, a system must temporality set COLLIM to zero, send the cash-out poll, then restore COLLIM to it previous value after.

15.4.13   EGM General Maintenance Poll

This multi-purpose poll can enable/disable the EGM or an individual game in an EGM. This poll is also used to request a number of EGM response types such as the EGM meter groups, progressive meters or multi-game/variation meters.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x1B, 1 byte hex
BLK   Meter group/s to report via the EGM Meter Group/Contribution Response (15.6.8), 1 byte hex
        Meter groups are defined in section 12.1.
        Bit 0   = 1 Meter group 0 request, if = 0 no action
        Bit 1   = 1 Meter group 1 request, if = 0 No action
        Bit 2   = 1 Meter group 2 request, if = 0 No action [CP:838]
        Bits 3...7   Groups 3...7 Reserved for future expansion [CP:839]
        The EGM must always report requested group meters regardless of any incorrect data (i.e. GVN or VAR) in this poll [CP:840].
        The EGM must report group meters even before EGM configuration [CP:841].
        For an EGM with configurable SAP/LP games, the EGM must not report SAP/LP win group meters until at least one game is actually configured as a SAP or LP respectively. [CP:842]

        More than one group may be requested at a time [CP:843]. If a SC requests a meter group from an EGM that has either no meters defined in it, or no meters are in use within the group (due to current EGM/game configuration), then the EGM must ignore the request for that group [CP:844].

FLG   Flags, 1 byte hex
        Bits 0...5   Reserved = 0, mask out these bits when reading any data from this byte [CP:845].
        Bit 6   Note Acceptor Status Response Request Bit. *QPv1.6*
                           If set, commands the EGM to queue the Note Acceptor Status Response (15.6.2) [CP:846]
        Bit 7   Machine Enable Flag (MEF) (15.1.2) [CP:847]
                           Bit 7 = 1 EGM enable. Bit 7 = 0 EGM disable
NUM   1 byte hex   Reserved, ignore [CP:848]
SIZ   1 byte hex   Reserved, ignore [CP:849]
        Bit 7   Reserved (formerly used re the NZ QCOM disable feature)
GVN   Game Version Number (*15.1.5*), 2 bytes hex
        If GVN was incorrect (i.e. equal to zero or does not exist) then the EGM must ignore the GFLG field below [CP:850]
VAR   Game Variation Number (*15.1.6*), 1 byte BCD
        Validity check: VAR must be BCD [CP:851]
GFLG   1 byte hex. [CP:852]
        bit 0   Player Choice Meter Response request bit. *QPv1.6*
                 If set, this commands the EGM to queue a Player Choice Meter Response (15.6.7) for the above game denoted by GVN.
                 If the bit is not set, or the GVN is invalid (i.e. does not correspond to a game in the EGM), or the game is not yet configured via the Game Configuration Poll (15.4.3), then the request must be ignored. [CP:853]
        bit 1   Bet Meters Response Request Bit. *QPv1.6*
                 If set, this commands the EGM to queue a Bet Meters Response (15.6.5) for the above game denoted by GVN.
                 If the bit is not set, or the GVN is invalid (i.e. does not correspond to a game in the EGM), or the game is not yet configured via the Game Configuration Poll (15.4.3), then the request must be ignored. [CP:854]
        bit 2   Progressive Configuration Response Request Bit. *QPv1.6*
                 If set, this commands the EGM to queue a Progressive Configuration Response (15.6.4) for the game/variation denoted by the GVN.
                 If the bit is not set, or the GVN was invalid (i.e. does not correspond to a game in the EGM), or the game has no progressive component, or if the game has not been

configured via the Game Configuration Poll (15.4.3) then the request for the EGM's Progressive Configuration Response must be ignored [CP:855]

bit 3   EGM Game Configuration Response Request Bit.

If set, this commands the EGM to queue an EGM Game Configuration Response (15.6.11) for the game denoted by GVN.

If the bit is not set, or the GVN is invalid (i.e. does not correspond to a game in the EGM), or if the EGM has not been configured[(QPv1.6)] via the EGM Configuration Poll (15.4.2), then the request is ignored. [CP:856] *(A QPv1.5 EGM reported the Game Configuration Response at all times when requested)*

bit 4&5 Progressive Meters Response Request Bit.

If either or both bits are set, this commands the EGM to queue a Progressive Meters Response (15.6.3) with all progressive level meters for the game denoted by the GVN. [CP:857] *QPv1.6*

If the bits are not set, or the GVN is invalid (i.e. does not correspond to a game in the EGM), or the game has no progressive component, or if the game has not been configured via the Game Configuration Poll (15.4.3) then the request is ignored [CP:858]

If progressive meters are requested for a game which has no progressive component, the EGM ignores the request for the progressive meters [CP:859]

(In QPv1.5, Bit 4 requested only LP Meters & Bit 5 Requested only SAP Meters in the Progressive Meter Response)

bit 6   Multi-Game/Variation Meters Response Request Bit.

If set, commands the EGM to queue a Multi-Game/Variation Meters Response (15.6.6) for the game denoted by GVN & VAR.

If the bit was not set, or the GVN or VAR was invalid (i.e. does not correspond to a game/variation in the EGM), or the game was not yet configured via the Game Configuration Poll (15.4.3), the EGM must ignore the request [CP:860].

bit 7   Game Enable Flag (GEF) (15.1.3)

= 1, enable game identified by GVN

= 0, disable game identified by GVN

(QPv1.6) It must not be possible to change the state of the GEF for a game or enable a game, via this poll for a game that has not been configured yet via the Game Configuration Poll (15.4.3). [CP:861]

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

If the GVN was correct but the VAR invalid or incorrect (i.e. does not correspond to a variation in the game), then the EGM must ignore multi-game/variation meter requests, but it must still report any other response type as requested via GFLG and set the GEF flag accordingly. [CP:862]

15.4.14   Request All Logged Events Poll

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x17, 1 byte hex
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

The SC will primarily use this poll when it is recovering after a SC RAM clear.

The EGM will re-queue for transmission all currently logged events in the primary event queue that have not been previously denoted as purged by the purge events poll. [CP:863]

Note. While the response to this poll may be a new event, the EGM must ensure that the first re-sent event is delayed to at least the response after the response to this poll (this is so the SC can correctly resync its next expected event sequence number) [CP:864]. (This behaviour should occur by default when the mandated response methodology is used. Refer to the figure in section 14.1).

15.4.15   Purge Events Poll

Instructs the EGM to 'purge' a range of logged events from its primary event queue. Refer section 7.3 "The meaning of "Purged" Events in QCOM Event Queues" to understand what the word 'purge' means in the protocol. This is the only way short of a RAM clear to purge events from this queue.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x16, 1 byte hex
PSN   Purge Events Poll - Poll Sequence Number, 1 byte hex
          Refer to Section 15.1.9 on Common Field Formats.
FLG   1 byte hex.
          bits 0..7 reserved = 0, mask out these bits when reading any data from this byte [CP:865]
EVTNO 1 byte hex. Event Sequence Number to purge to. Inclusive of EVTNO (0...255)
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

If the PSN is correct (15.1.9), the EGM must; update the next expected Purge Events Poll PSN and purge events from the primary event queue, starting from the first logged event in the event queue, until the queue is either empty, or until the event with sequence number EVTNO (inclusive), or until the first unacknowledged (by the SC) event, whichever comes first. An EVTNO of zero is a special case and commands the EGM to purge all events (except any events queued to be sent or an event awaiting acknowledgement). [CP:866]

In response to this poll the EGM must always queue a Purge Event Poll Acknowledgement Response regardless of the PSN being correct or not (refer section 15.6.10) *QPv1.6 (QPv1.5 EGMs only queued the response if the PSN was valid)* [CP:867]

15.4.16   Note Acceptor Maintenance Poll

This poll is only applicable to EGMs which support note acceptors.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x10, 1 byte hex
NAFLG 2 bytes hex
       LSB
              bits 0..7 reserved = 0, mask out these bits when reading any data from this byte [CP:868].
       MSB
              bit 0 $5
              bit 1 $10
              bit 2 $20
              bit 3 $50
              bit 4 $100
              A value of 1 enables the specified denomination banknote for acceptance by the banknote acceptor, 0 disables acceptance [CP:869].
              Disabling for acceptance any specific or all banknotes must not affect the EGM's ability to accept Cash Tickets [CP:870].
              bits 5..7   reserved = 0, mask out these bits when reading any data from this byte [CP:871]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

EGMs must also provide a display function in either test or audit mode which displays the settings controlled in this poll (but must not allow these settings to be altered. Refer 3.6.1). This poll must be the only way on the EGM to change the above settings [CP:872] *(QPv1.6)*.

The settings in this poll may be verified by the SC via the Note Acceptor Status Response (15.6.2) which may be requested via the General Maintenance Poll (15.4.13). *QPv1.6*

15.4.17   Hopper/Ticket Printer Maintenance Poll

These parameters may be changed at any time by the SC. This poll must be the only way to change these parameters on the EGM.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x0F, 1 byte hex
REFILL   Default Hopper Refill Amount. 4 bytes hex, in cents. This field changes the hopper refill default amount that is offered during a hopper empty condition; however the actual refill amount recorded may still be overridden by the attendant at the time of the refill. Refer Section on Hopper Maintenance [CP:873].
COLLIM   Hopper Collect Limit. 4 bytes hex, in cents. Upon a collect of current credit or a win payout, if the amount to be paid is less than COLLIM, then a hopper pay will occur, otherwise a cash ticket print or cancel credit will result (depending on TICKET and whether a ticket printer is present in the EGM) [CP:874]. Also refer to section 3.5 on credit redemption.
TICKET   Ticket Out Limit. 4 bytes hex, in cents. Upon a collect of current credit or a win payout, check for a hopper pay first (if applicable), then the amount to be paid is greater than or equal to COLLIM and less than TICKET, then a cash out ticket may be printed (if a ticket printer is present). If the amount to be paid is greater than or equal to TICKET then a cancel credit will result [CP:875]. Refer section 22. Also refer to section 3.5 on credit redemption.
FLG   2 bytes hex. *QPv1.6*
                Bits 0…14 reserved
                Bit 15   Print Test Ticket.
                          If set, this commands the EGM to print out a test ticket at the next opportunity.
                          [CP:876] This is for preventative maintenance purposes.
RES   1 byte hex. *QPv1.6*
DOREFILL   4 bytes hex in cents. *QPv1.6* If the EGM has a hopper and this amount is not zero then the EGM must add this amount to its hopper refill and hopper level meters, log the refill recorded event (7.10.3.9) and queue the meter (even if the EGM's main door is closed at the time). [CP:877]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Note, cashless mode (16.2) takes priority over hopper/ticket collects unless stated otherwise [CP:878].

15.4.18   Linked Progressive Award Acknowledged Poll

Receipt of this poll by an EGM denotes that the current linked progressive award on the EGM has been verified by the system and the EGM is now authorised to acknowledge the LP win to the player (e.g. perform a win show). Refer to section 10.6.1 on LP award lockups for information. It also denotes that the EGM is now authorised to clear the associated LP lockup via either the next EGM General Reset Poll or manual key-switch [CP:879]. (However also refer to section 10.6.1 regarding General Reset Polls during a LP win show.)

Upon receipt of this poll, if no linked progressive award was current on the EGM, or the current LP award had already been previously acknowledged via this poll, then the LP Acknowledgement Poll must be ignored (it must not be queued for the next LP Award) [CP:880]. However, if the EGM has a LP award current, but was in another state when this poll is received (eg. in a fault condition, door open mode or test/audit mode), then this poll must be queued in EGM NV-RAM for action upon return to the current LP Award [CP:881].

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x0E, 1 byte hex
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.4.19   EGM General Reset Poll

On receipt of this poll, the EGM will attempt to clear the current lock-up or fault condition as directed by the poll message data. This poll message type is the protocol equivalent of physically turning the reset key on the EGM, but with the added feature (*QPv1.6)* of also being able to choose the type of condition to try and reset (see FLG field below). The ability to act on just fault conditions or just lockups prevents accidental clearing of the wrong type of condition. However fault conditions are the higher priority and must be cleared off before lockup conditions may be acted upon.

Fault and lock-up conditions must still be alternatively clearable via an appropriate manual key-switch combination on the EGM in the usual manner [CP:882].

Note, linked progressive jackpots cannot be cleared by any means until they are first acknowledged via the linked progressive award acknowledged poll (15.4.18) refer section 10.6.1. In addition, EGMs are permitted to ignore requests to clear a LP lockup via this poll *(QPv1.6)* until completion of any associated LP win show, refer 10.6.1 (This prevents systems auto-clearing LP lockups from cutting LP win shows short).

A list of fault and lock-up conditions appears in the Event section (7.10) and General Status response section (15.6.1).

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x0C, 1 byte hex
FLG   1 byte hex
         Bits 0...1 indicate what type of condition/s to clear.

> bit 0   If set, commands the EGM to attempt resetting of the current fault condition.
> bit 1   If set, commands the EGM to clear the current lock-up condition [CP:883].
> bits 2-7 Reserved = 0, mask out these bits when reading any data from this byte [CP:884].
>
> Note, although this poll could potentially operate on both fault and lockup conditions simultaneously, if both types of conditions were current, only the fault condition must be acted upon if both bits were set in the poll [CP:885].

STATE 1 byte hex. *QPv1.6*
         This field may be used by the SC to ensure the poll clears only the lockup condition for which it was intended. Refer * below.
         This field is only applicable to the EGM if FLG:bit 1 above is also set. If STATE is non-zero and FLG:bit 1 = 1, then STATE denotes the lockup condition that the General Reset Poll must apply to. (Refer General Status Response 15.6.1; or see the table below for the list of applicable STATE code values.) [CP:886] If the EGM was not in the lockup condition denoted by STATE at the time it processed this poll, then the EGM must not reset the current lockup condition (if any). [CP:887]
         If STATE is zero and FLG:bit 1 is set, then the poll is applied to the current lock-up condition regardless. [CP:888]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Upon receipt of this poll, if no fault/lockup condition existed on the EGM then this poll must be ignored. If the EGM was in higher priority mode (e.g. door open, test or audit modes), then the request to reset a lower priority fault or lockup via this poll must be ignored [CP:889]. If the EGM was in both a lockup condition and one or more fault conditions, then the fault conditions must be reset first before the lockup conditions may be acted upon. [CP:890] The priority in this regard would be door opens (highest & reset by respective door closure), fault conditions** and then lockups** [CP:891]. (**Refer glossary for definitions of faults and lockups) (NB: It may also be acceptable to have door opens and faults as equal priority)

The reset command must never be queued [CP:892].

This poll cannot clear all possible types of lockups on an EGM. For example:

- This poll must not clear / reset the EGM from a Cash Ticket Out Lockup.[CP:893] Only the "Cash Ticket Out Request Acknowledgement Poll" (15.4.10) can clear this type of lockup, refer 22.1.

- This poll must not clear / reset the EGM from an ECT-from-EGM Lockup.[CP:894] Only the "ECT Lockup Reset Poll" (16.4) can clear this type of lockup.

*The list of lockups that this poll may clear are: [CP:895]

StateID
(15.6.1)   **Lockup**

StateID (15.6.1)	Lockup
0x06	RCRF Cancel Credit Lockup
0x07	Cancel Credit Lockup
0x0D	(An acknowledged) CRanE Lockup
0x08	Large Win Lockup
0x09	(An acknowledged) Linked Progressive Award Lockup [CP:896]
0x0B	System Lockup

15.4.20   Specific Promotional/Advisory Message (SPAM) Poll

This poll is used to display various text messages to patrons or attendants. Up to two messages at once (denoted A & B) must be maintained by the EGM via these polls.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Either: Function Code = 0x08 Denotes Message A, 1 byte hex
                 Function Code = 0x07 Denotes Message B, 1 byte hex
FMT   1 byte hex.
        bits 0..5 reserved = 0, mask out these bits when reading any data from this byte [CP:897].
        Bit 6   Prominence flag modifier. Applies to SPAM A only. If set, then while the EGM is in idle mode, it must display any SPAM A message text by overlaying a transparent text box dedicated to the SPAM A text on the game results display only. [CP:898] The SPAM text must be at least **10mm** high when in the prominent position. [CP:899] On a reel based game, the location is preferably between the top two rows of game symbols. If the EGM is not in, or exits idle mode, or enters another display (such as rules or the PID), then the SPAM A text must automatically revert to its default display location. As SPAM A in the prominent position may be utilised a great deal, to avoid detracting from the game as much as possible, the height of transparent text box referred to above should be as small as possible while still meeting the minimum text height for SPAM (see below). Refer to the figure below for an example.
        bit 7   Fanfare Flag.
                 If set, the EGM must also make an short attention sound (that is not a fanfare or emotive in nature) approximately 1 second duration, upon the initial display of message text [CP:900]. Acceptance of the attention sound used is at the discretion of the CEO (refer 3.6.3) [CP:901].
LEN   1 byte hex.
        Length of text field in bytes
        Maximum length is 80 bytes [CP:902]. (*QPv1.5 was only 40 bytes max)*
        0x00 means erase previous message [CP:903]
        Before processing the EGM must verify LEN <= 80, if false then the request for the SPAM must be ignored [CP:904].
TEXT   LEN bytes of ASCII printable characters. Refer Section (*2.3.8*). [CP:905] This replaces and erases any previously sent TEXT
        Before processing the EGM must verify all characters are printable (refer *2.3.8*), on fail then the request for the SPAM must be ignored [CP:906].
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Upon receipt of this message, the EGM will **immediately** display the TEXT message to the patron/attendant, except if in door open, test or audit modes (whereby SPAM TEXT display is optional) [CP:907]. If the EGM is in test or audit mode, then the message is displayed immediately upon next return to any other mode [CP:908]. The message data must be stored in NV-RAM and displayed on the EGM until told to erase or overwrite it [CP:909]. *QPv1.6* The EGM must be able to display at least 16 characters of the message at a time [CP:910]. SPAM text must be at least **5~~10~~mm** high when measured directly off a properly adjusted EGM VDU *(QPv1.6).* [CP:911] If the message is too long to display at once, the EGM should scroll it across the display. As a last resort, the EGM may break the message at space characters and display consecutively (separated with trailing and leading "...") [CP:912]. After the initial display, the EGM may also alternate the TEXT message with other text messages on the display provided there is sufficient time[*] to read the message each time it is displayed [CP:913].

[*] When any part of the message is visible, it must be displayed at a rate no faster than 16 characters per second *QPv1.6.*

The display must be legible. Acceptance of the overall legibility of the promotional text message on the EGM display is at the discretion of the CEO (refer 3.6.3) [CP:914].

Text messages sent by these polls must be displayed in addition to any text sent by the broadcast promotional message in idle mode [CP:915].

Figure 7 Example of SPAM A in the Prominent Position

Screen dump courtesy of IGT (Australia) PTY LTD

15.4.21 EGM Tower Light Maintenance Poll

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x05, 1 byte hex
FLG   2 bytes hex [CP:916]
        1st byte, 1 = Light on, 0 = Light off
                bit 0   Yellow Light (usually indicates change required)
                bit 1   Blue Light (usually indicates drinks)
bit 2   Red Light (indicates fault conditions)
                                The red light is usually controlled by the EGM to indicate faults but
                                may also be controlled by the SC. See below for an explanation of
                                priority.
                bits 3-7   Reserved = 0, mask out these bits when reading any data from this
                                byte.
        2nd byte, 1 = Light Flash
                bit 0   Yellow Light (usually indicates change required)
                bit 1   Blue Light (usually indicates drinks)
                bit 2   Red Light (indicates fault conditions)
                                The red light is usually controlled by the EGM to indicate faults but
                                may also be controlled by the SC. See below for an explanation of
                                priority.
                bits 3-7   Reserved = 0, mask out these bits when reading any data from this
                                byte.
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

The physical order of the tower lights on the EGM is not important and beyond the scope of this document.

The EGM will physically control its tower light (if present) with respect to the operator's requirements, if any. It is standard practise that the EGM automatically flashes its red light during fault conditions. However, the SC may also control the tower light using this poll. Therefore the EGM must share control of the tower light between the EGM and SC using a logical OR operation with the following priority [CP:917].

Flashing has the highest priority, OFF is the lowest priority.

There are 9 possible scenarios for each light.

EGM wants light			Protocol wants light			Light Result
Off	On	Flash	Off	On	Flash	
✔			✔			Off
	✔			✔		On
		✔			✔	Flash
✔				✔		On
	✔				✔	Flash
	✔		✔			Flash
✔					✔	Flash
	✔		✔			On
		✔		✔		Flash

15.4.22 General Status Poll

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x01, 1 byte hex
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

This message is simply to solicit a response from the EGM, to check if the status of the EGM has changed and to ensure the EGM is still responding.

15.5 Broadcast Messages

Refer to the glossary (section 2.2) for an explanation of the "Message Format and Order:" string.

15.5.1   Current Date and Time

This message is sent by the SC with every broadcast message via the broadcast address (255) only, once towards the end of every poll cycle [CP:918].

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0xFF, 1 byte hex [CP:919]
FLG   1 byte hex. Global Flags
        Bits 0..5   Reserved = 0, mask out these bits when reading any data from this byte [CP:920].
        Bit 6   Clock Display Flag. *QPv1.6*
            If set, the EGM must constantly display during idle mode and during play, a time zone adjusted clock (15.4.5) to the player (Refer to Appendix A for the exact states of when the clock must be displayed) [CP:921]. This is the only place the EGM has to display or use a time zone adjusted time (15.4.5).
            The display must be numeric and in 12 hour time format; e.g. "XX:XX" with "AM" or "PM". Do not display seconds. Clock text must be at least **7mm high** when measured directly off a properly adjusted EGM VDU. [CP:922] The display must be legible (refer 3.6.3). Acceptance of the overall legibility of the clock display is at the discretion of the CEO [CP:923].
            The clock display preferably must be located in the top right hand side of the EGM's primary display. Otherwise the clock display must be located in the bottom left hand corner of EGM's primary display. [CP:924]
            If not set, then no clock may be displayed on any player accessible EGM displays. [CP:925]
        Bit 7   Site Enable Flag (SEF) (15.1.1)
            1 = Site enabled (enable EGM), 0 = Site disabled (disable EGM) [CP:926]

TIME Current system time. Seconds, Minutes, Hours, 1 byte each, BCD, 24 hour format.
DATE Current system date. Day, Month, Year , 1 byte each, BCD
        This is the current system date and time.
        The EGM must synchronise its clocks using only this date and timestamp.
        This is the date and time the EGMs must use all for time-stamping purposes such as events and refreshing their RTC as necessary.
        Validity Check: Date and Time must be BCD and all individual field values within range else the date and time fields must be ignored. [CP:927]
        WARNING: A SC may require the EGM to update its system clock at any time (i.e. either the NV-RTC or a volatile hardware or software driven RTC, or any device the EGM uses to track the system time). Refer 7.8.1. As a consequence of this, it is recommended that an EGM does not use its system clock for any purpose other than for the player clock display (above) and the time stamping of events (7.1.5). Software timing routines and delays routines must not use the system clock unless those routines are programmed to compensate for a time change that can occur at any time. Possible side effects in QCOM of uncompensated timing routines upon a system time update by the SC are; possible premature timeout of screensave, power-save or communications timeouts (6.1.2 & 6.1.8). [CP:928] It is recommended that EGMs implement a system time independent counter for use with timing and delays routines. (E.g. a simple interrupt driven counter.) This will ensure that these routines cannot be affected by SC time changes and will not require any on-the-fly adjustment if the system time does change. [CP:929]
ESIZ 1 byte hex. Size in bytes of the EXTD field below. [CP:930]
        A value of 0x00 denotes no EXTD data field (below) is present
EXTD Extended broadcast data field. ESIZ bytes.
        EXTD data fields with an unknown function code (EFUNC) must be ignored by the EGM [CP:931]
        If present, EXTD takes on multiple formats depending on the EFUNC field which is always the first byte of the EXTD field. The formats are defined in the following sections.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.5.2   EXTD: Linked Progressive Jackpot Current Amounts

This broadcast is used to send out the current progressive amount to all linked progressive games.

EXTD field Message Format and Order:

EFUNC 1 byte hex. Extended broadcast data function code = 0x02
NUM   1 byte hex.
       Bits 0...2   The number of progressive levels in this message minus one. i.e. 0...7.
                          Up to 8 levels maximum.
       Bits 3…7   Reserved = 0, mask out these bits when reading any data from this byte [CP:932].

The following message data is repeated by the number of times as indicated in the NUM byte [CP:933].

PGID   Linked Progressive Group I.D. (*15.1.10*), 2 bytes hex
       0x0001 ... 0xFFFE
PLVL   Linked Progressive level I.D., 1 byte hex
       Bits 0…2   Progressive Level number (0-7)
       Bits 3…7   Reserved = 0, mask out these bits when reading any data from this byte.
LPAMT
       Linked Progressive group/level (PGID/PLVL) jackpot current amount, 4 bytes hex in cents, LSB first.
       Validity Check: PGID & PLVL must be correct for a level, before accepting the corresponding value in LPAMT. [CP:934]

End of repeated entry data.

The last received current progressive amount in the broadcast messages is the amount awarded to the player when a linked progressive award event is logged (7.10.2.1). Refer LP section (10.6).

Note, a LP EGM, or jackpot display system processing this broadcast, must not make any assumptions about the data ordering in any LP broadcast! For example, a SC may send out multiple PGIDs in a single LP broadcast [CP:935], in any order with respect to levels [CP:936], including sending out PGIDs and levels that are not applicable to the EGM [CP:937], or mix these up in the same broadcast with applicable PGIDs and levels [CP:938]. A SC may also send more [CP:939] or less [CP:940] levels than expected for a given PGID, or split levels [CP:941] across multiple broadcasts (this must **not** be considered an error). A LP EGM (or Jackpot display system) must check **every** entry's PGID & Level ID in the broadcast and accept only the applicable LP current amounts and ignore non-applicable levels. [CP:942]

15.5.3   EXTD: General Promotional Message (GPM)

EXTD field Message Format and Order:

EFUNC 1 byte hex. Extended broadcast data function code = 0x03
FMT   1 byte hex.
        bits 0..6 reserved = 0, mask out these bits when reading any data from this byte [CP:943].
        bit 7   Fanfare Flag. *QPv1.6*
                If set, the EGM must also make a short attention sound (that is not a fanfare or emotive in nature) approximately 2 seconds duration, upon the initial display of message text [CP:944]. Acceptance of the attention sound used is at the discretion of the CEO (refer 3.6.3) [CP:945].
LEN   1 byte hex.
        Length of text field in bytes
        Maximum length is 80 bytes [CP:946].
        0x00 means erase previous message [CP:947].
        Before processing, the EGM must verify LEN <= 80, if false the GPM request must be ignored [CP:948].
TEXT   LEN bytes of ASCII printable characters. Refer Section (*2.3.8*). [CP:949]
        TEXT replaces and erases any previously sent GPM [CP:950]. Before processing the EGM must verify all characters are printable (refer *2.3.8*), on fail the poll must be ignored [CP:951].

This extended broadcast type requests the EGM to display an arbitrary text message to the player while it is in idle mode.

Upon receipt of this message, the EGM will display the TEXT message to patrons at least while in idle mode (refer Appendix A) [CP:952]. The message data must be stored in NV-RAM and displayed until the EGM is told to erase or overwrite it [CP:953]. *QPv1.6* The EGM must be able to display at least 16 characters of the message at a time [CP:954]. If the message is too long to display at once, the EGM should scroll it across the display. As a last resort, the EGM may break the message at space characters and display consecutively (separated with trailing and leading "...") [CP:955]. The EGM may also alternate the TEXT message with other text messages on the display provided there is sufficient time[*] to read the entire message each time it is displayed [CP:956].

[*] When any part of the message is visible, it must be displayed at a rate no faster than 16 characters per second *QPv1.6.*

The display must be legible. Acceptance of the overall legibility of the promotional text message on the EGM display is at the discretion of the CEO (refer 3.6.3) [CP:957].

General promotional message text is displayed in addition to, or alternating with any text sent by the specific promotional text message polls in idle mode.

15.5.4   EXTD: EGM Poll Address Configuration

EXTD field Message Format and Order:

EFUNC 1 byte hex. Extended broadcast data function code = 0x04
NUM   1 byte hex. Number of EGMs entries in this message, 1…16
          Before processing, the EGM must verify NUM >= 1[(QPv1.6)] and NUM <= 16, if false the Extended
          Broadcast Message Data must be ignored. [CP:958]
ESIZ   1 byte hex. Size in bytes of each entry = 0x05 [CP:959]

The following message data is repeated by the number of times as indicated in the NUM byte.

SER   EGM Serial number (15.1.8), 3 bytes BCD, LSB first
MID   EGM Manufacturer I.D. (15.1.7), 1 byte BCD
          Validity check: SER & MID must be BCD [CP:960]
PADR   1 byte hex. EGM Poll Address 1..250 inclusive (*QPv1.6: was 254 in QPv1.5*).
          The EGM must only accept a poll address that is in the above range [CP:961].
          Some poll addresses (such as poll address 255) are reserved for broadcast messages which
          all EGMs must process. Address zero is never used as a corrupt all zero message would yield
          a valid CRC.

End of repeated entry data

This broadcast message is sent by the SC to ~~assign~~ designate an EGM a poll address and start the
EGM responding. This message is the only way to make an EGM resume responding to polls. An
EGM must only accept a poll address if the corresponding serial number and manufacturer ID
matches its own. This broadcast message's EXTD must be ignored if the EGM has not been setup
with a serial number (3.1.1).

Only once the EGM has been configured with a poll address by this broadcast poll type may it
commence responding to the SC after a communication time-out, EGM power up or EGM RAM clear.

QPv1.6 Once configured with a valid poll address via the broadcast message, the EGM must
commence responding within two subsequent polls to the EGM's address [CP:962].

Once configured by this broadcast, the EGM must no longer process the EGM Poll Address
Configuration message data until the next communication session (refer glossary section 2.2) [CP:963].

The poll address designated to an EGM via this message broadcast lasts until the EGM assumes
communications defaults again. Refer section 8.3 for more information.

15.5.5   EXTD: Site Details

EXTD field Message Format and Order:

EFUNC 1 byte hex. Extended broadcast data function code = 0xFE
FMT   1 byte hex.
       bits 0..7 reserved = 0, mask out these bits when reading any data from this byte.
SLEN   1 byte hex.
       Length of STEXT text field in bytes.
       Maximum length is 40 bytes.
       Before processing, the EGM must verify SLEN <= 40, if false then all the Site Details message
       data must be ignored [CP:964].
LLEN   1 byte hex.
       Length of LTEXT text field in bytes.
       Maximum length is 80 bytes. *(QPv1.6, QPv1.5 was 15)*
       Before processing, the EGM must verify LLEN <= 80, if false then all the Site Details message
       data must be ignored [CP:965].
STEXT SLEN bytes of ASCII printable characters. Refer Section (*2.3.8*). [CP:966]
       Name of licensed venue. SLEN bytes. This replaces any previous sent text data.
LTEXT LLEN bytes of ASCII printable characters. Refer Section (*2.3.8*). [CP:967]
       QPv1.6 Address / contact details of licensed venue. *(QPv1.5, used to be only the Name of
       licensed venue's City, Town or Shire).* LLEN bytes. This replaces any previous sent text data.

       Before processing the EGM must verify all characters are printable (refer *2.3.8*), on fail then all
       the Site Details message data must be ignored [CP:968].

This message is in support of cash out ticket printing (22.3). If the EGM has a ticket printer, the message
data must be stored in NV-RAM [CP:969].

The EGM may also use this information for any other purpose if desired.

Acceptance of the overall legibility of the text message on printed tickets is at the discretion of the CEO
[CP:970].

15.6 Response Message Data

Refer to the glossary (section 2.2) for an explanation of the "Message Format and Order:" string and meaning of references to "message data".

15.6.1   General Status Response

This is the lowest priority response. It is sent in response to all polls to the EGMs address if the EGM has no higher priority message to report. The data in this response must reflect the EGMs current status on the next GSR response built by the EGM [CP:971].

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x01, 1 byte hex
FLGA   1 byte hex. EGM current state - doors [CP:972]
    bit 0   EGM main door [CP:973]
    bit 1   EGM Cash Box door [CP:974]
    bit 2   EGM Processor door [CP:975]
    bit 3   EGM Belly panel door [CP:976]
    bit 4   EGM Note acceptor door [CP:977]
    bit 5   EGM Note stacker [CP:978]
    bit 6   EGM Mechanical Meter Door [CP:979] *(QPv1.6)*
    bit 7   EGM Top Box/Aux Door [CP:980] *(QPv1.6)*
        1 = open/removed, 0 = closed/NA
FLGB   1 byte hex. EGM concurrent states (they can interrupt main-line code STATE's below) [CP:981]
    bit 0   Fault Condition Active (i.e. Events defined as fault conditions)
    bit 1   EGM Audit mode Active
    bit 2   EGM Test/Service Mode Active
    bit 3   Cashless Mode (16.2) Active/Pending. *(QPv1.6)*
        Set if Cashless Mode is either active or pending (i.e. the EGM is going to enter Cashless Mode upon next return to Idle Mode. Refer 16.1.1). [CP:982]
    bit 4   Awaiting Cash Ticket In Request Acknowledgement. [CP:983] *(QPv1.6))*
        (The EGM is expecting a Cash Ticket In authorisation via the Ticket In Request Acknowledgement Poll 15.4.11. This bit must remain set until the poll either approving or rejecting the request is received or a TI time-out occurs. Refer section 22.2)
    bits 5-7 Reserved = 0
        1 = Active, 0 = inactive
STATE 1 byte hex, EGM main-line code current state [CP:984]
    0x01   Idle Mode (refer defn 3.3.1)
    0x02   Play in Progress
        Refer definition of a "play", section 2.2.
        A RCRF gamble must not be considered a "play".
    0x03   Play in Progress – Feature
        E.g. free games. (QPv1.5 included double up/Gamble in this state but is now a separate state, see below). NA to RCRF.
    0x04   Hopper Collect [CP:985]
        A hopper collect state is defined to commence when the hopper motor is energised for the purpose of paying out a coin and ends when the hopper pay is completed or aborted (see 3.4). If a fault occurs during the pay (e.g. jam/empty) then the EGM must still report a hopper collect state via this field until after the fault is reset and the hopper pay is then aborted (3.4).
    0x05   Residual Credit Removal Feature (RCRF)
        This state commences when the player is presented with their RCRF options and finishes when a hopper pay commences as a result of a RCRF win, or immediately after a RCRF gamble is lost, or the EGM enters a RCRF Cancel Credit Lockup (see below).
    0x06   ~~Residual~~ RCRF C~~c~~ancel ~~c~~Credit Lockup
        This state applies only to cancel credits entered via the RCRF.
    0x07   Cancel Credit Lockup

0x08   Large Win Lockup
0x09   Linked Progressive Award Lockup [CP:986]
0x0A   Unused. (*Was SAP Award Lockup in QPv1.5 EGMs but never used*)
0x0B   System Lockup
            (refer System Lockup Request Poll (15.4.9))
0x0C   ECT from EGM Lockup (refer section 16.3.1)
0x0D   CRanE Lockup (refer section 17.7)
0x0F   Printing Cash Ticket Out [CP:987]
            (Printing of a cash ticket out has been authorized and the EGM is in the process of printing a ticket)
0x10   Cash Ticket Out Lockup [CP:988] *QPv1.6*
            (The EGM is waiting Cash Ticket Out authorisation (if applicable) via the Ticket Out Request Acknowledgement Poll 15.4.10)
0x12   Play in Progress – Double Up/Gamble [CP:989] *(QPv1.6)*
0x13   Reserved. (NZ - Download Jackpot Lockup)
0x14   Reserved. (NZ - PID Display Lockup)

All unused values are reserved for future expansion.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.
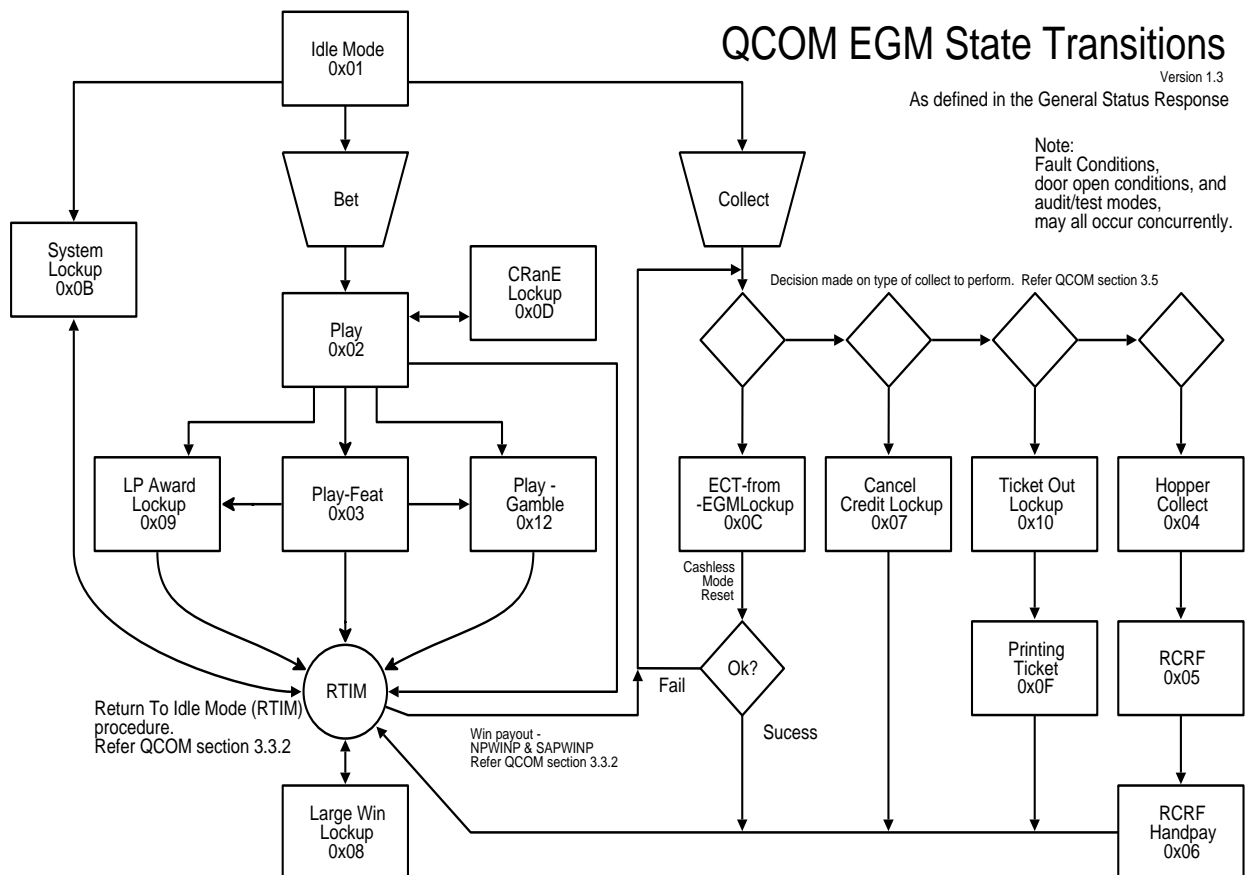
Figure 8 Typical – for reference purposes only. This diagram is not a requirement.

15.6.2   Note Acceptor Status Response

New Response type for QPv1.6

This response is mandatory for EGMs with a Note Acceptor installed. This response is requested via the EGM General Maintenance Poll (15.4.13).

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x04, 1 byte hex
FLGA   2 bytes hex.
             Bits 0…14 reserved = 0
             Bit 15   Stacker Full Flag [CP:990]
             0= stacker not full, 1 = stacker full
NAFLG   2 bytes hex.
             The EGM must report back this field here as set by the Note Acceptor Maintenance
             Poll (refer section 15.4.16) [CP:991]
RES   4 bytes hex = 0
NADS   Note Acceptor Descriptor String. 40 bytes. Comprised of up to 39 ASCII printable
             characters (*2.3.8*). The field must includeing the at least one null terminating
             character (0x00) in the last position. Refer Section (*2.3.8*). The field string, if less
             than max width, must be null terminated and right padded with 0x00's. [CP:992]
             This string should contain the following comma separated information concerning the
             note acceptor: [CP:993]
             Manufacturer (Abbreviated), Model, Firmware ID, Note Acceptor Validation Set
                        Version[*]

             E.g. "ABC,Aardvark,ABC-AU-L-1234,AU001234"

             * The 'Note Acceptor Validation Set Version' must uniquely identify the current version of the information or
             data set that the note validator is using to authenticate notes (if not already uniquely specified by the
             Firmware ID string).

             If any NADS sub-field cannot be ascertained by the EGM, then leave it empty. E.g.
             "ABC,,ABC-AU-L-1234,AU001234" (if model not known), or "ABC,Aardvark" (if only
             make and model known) [CP:994]

             If for example (worse case) the Note Acceptor firmware ID received from the Note
             Acceptor is too long or inconsistent, e.g.:

             "ID003 - U(AUS)-10-SS ID003-03V120-07 18JAN05 D358"

             then the EGM may hash the above string with a CRC16 so it can be truncated to fit
             into the NADS field and report the following e.g.:

             NADS= ",,ID003 - U(AUS)-10-SS ID003-03V(FAD4)," (= 40 bytes incl NULL)

             Where (FAD4) above is the CRC16 of the original firmware string. This may be the
             safest thing to do in some cases in order to avoid the situation where the EGM's
             software must to be upgraded just because the Note Acceptor manufacturer changes
             the format of their firmware ID string. Accordingly the EGM manufacturer must be
             wary of hard-coding a methodology to process a Note Acceptor firmware string
             whose format is outside their control. [CP:995]

             Check with the OLGR before formalising a version/hashing methodology here, as it is
             desirable where possible to ensure that the same version NA software/data set
             results in the same version/hash value across all EGM manufacturers. Acceptance of
             the NADS field is at the discretion of the CEO [CP:996]

The NADS field must be refreshed every time the EGM powers up and when the Note Acceptor resumes responding to the EGM so the data reflects the current values next time this response is requested. ^{CP:997}

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.6.3　Progressive Meters Response

This response is queued when requested via the EGM General Maintenance poll (15.4.13). Both SAP and LP jackpot current amounts can be reported simultaneously in this response [CP:998].

Each progressive game in the EGM must have only one set of progressive meters. Also refer 9.2.

Message Format and Order:

DLL　　Data Link Layer, Refer Section 14.3
FC　　　Function Code = 0x07, 1 byte hex
GVN　　Game Version Number (*15.1.5*) of the meters in this response, 2 bytes hex
PGID　　Progressive Group I.D. number of GVN (*15.1.10*), 2 bytes hex
PAMT　LP Turnover Meter. 4 bytes hex in cents, LSB first.
　　　　The total game turnover amount applicable to PGID, In a LP game this is the same field as reported in the Meter/Group Contribution Response. Refer to the glossary (2.2) for more information. This field may be zero if the game is SAP only [CP:999].
ONUM　Reserved. 1 byte hex. Must be zero. *(Used to be NUM in QPv1.5)* [CP:1000]
　　　　QPv1.5 EGMs followed this with a repeated entry (0...8) comprising of PLEV (1 byte) & CAMT (4 bytes)

New for QPv1.6:

VARES　　　　Reserved = 0, 1 byte hex.
PMRRES　　　Reserved = 0, 4 bytes hex.
NUM　　　　　1 byte hex, number of current progressive level amounts (CAMT) in response. 0...8
SIZ　　　　　1 byte hex. Size in bytes of repeated entry below = 15.

The following data is repeated for each progressive level in the game as indicated by the NUM field.

PLEV　1 byte hex.
　　　　bits 0..2　Progressive level I.D. (0...7) of the following CAMT
　　　　bits 3..6　Reserved = 0
　　　　bit 7　　Level Type.
　　　　　　　　A value of one denotes the level is a LP, zero denotes SAP [CP:1001]
CAMT　4 bytes hex in cents, LSB first.
　　　　Current Progressive Level Contribution for the game identified by the GVN and level number above.

　　　　If the level is a SAP, CAMT is the applicable contribution towards the level since the level was last won [CP:1002].
　　　　I.e. CAMT = Current jackpot display amount* – Reset amount + Overflow **
　　　　　　* capped at jackpot ceiling
　　　　　　** include amounts from a hidden increment if applicable

　　　　(NB: If there was an initial contribution received via EGM Game Configuration Poll (15.4.3), then this amount will be also included on this meter until the first jackpot is won)
　　　　This amount would be the exact liability of the SAP jackpot amount if the EGM were decommissioned.
　　　　The intent behind CAMT for SAPs is that if the EGM was RAM cleared and then reconfigured, sending the last received CAMT back to the EGM via the EGM Game Configuration Poll (15.4.3) would restore the SAP component to exactly the same state is was in immediately prior the RAM clear.

　　　　If the level is a LP, CAMT is simply the last received broadcast jackpot current amount for the LP level [CP:1003].
HITS　　2 bytes hex. Total hits on the progressive level denoted by PLEV. [CP:1004]
WINS　4 bytes hex in cents. Total wins for the progressive level denoted by PLEV. [CP:1005]
HRATE
　　　　Theoretical Hit Rate. 32 bit (4 byte) floating-point number (refer 2.3.7).

NB This field being only 32 bits is fairly low precision and this should be taken into consideration ~~when~~ with respect to any values ~~are~~ derived from it.

This number ~~is~~ must ~~a constant* and~~ represent~~s~~ the theoretical probability (p) **for a one cent bet** that the jackpot will be won for a given game variation / progressive parameters. [CP:1006] The EGM may change the reported value of HRATE in response to a variation change on the EGM (15.4.4), or a new set of progressive parameters is accepted by the EGM (15.4.6). The reported value HRATE must remain constant at all other times; however the actual hit rate of the level may vary (for example, increase in proportion to bet) as a part of the game's design and with respect to any local regulatory requirements. Related: event 7.10.3.34.

HRATE must be $\geq$ 0.0 and < 1.0.

Rounding issues on the least significant digit are acceptable however the full precision of 32 bit floating point numbers must be utilised to represent the hit-rate ~~in order to provide less than a +/- 0.1% error in any RTP formula calculation with respect to actual theorectial RTP (refer 7.10.3.34 for formulas). Four significant digits are typically not enough and around six or seven significant digits should be used.~~ [CP:1007]

When displayed on the EGM (e.g. in audit mode), HRATE must be displayed in scientific notation (at the maximum available precision) with units: "/ cent" (or "/ ¢"). E.g. "1.234565e-8 / ¢". [CP:1008]

Typically*, the Average Trigger Value in cents (ATV)
    = ((PINC / 10000 / 100.0) / HRATE + SUP)

E.g. SUP = 5000 ¢, HRATE = 0.00001 & PINC = 10000 (1%), ATV = 6000 ¢

 (*This formula may vary depending on the design of the game)*~~For games where the true value of HRATE varies (for example HRATE varies depending on the bet per play for payline symbol triggered - multiplier progressive games), the theoretical HRATE for the game's minimum bet must be reported here (or other best value given systems historically only use HRATE to verify that the game with within MINRTP and MAXRTP).~~

~~*Future systems won't use HRATE anymore. I.e. HRATE eventually be depreciated in a future version of QCOM.*~~

~~*(NB: When referring to HRATE in a verbal conversation, avoid saying "per cent bet" as this is easily misinterpreted as "% bet", instead, say "for each cent bet" or just "cents")*~~

End of repeated entry data

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

- The EGM is only required to be able to queue a single response of this type at a time (QPv1.6). *QPv1.5 EGMs were required to be able to queue multiple PMR responses.* [CP:1009]

15.6.4   Progressive Configuration Response

New for QPv1.6. This response contains jackpot parameter information for all progressive jackpot levels for a single game in the EGM. This response is queued when requested via the EGM General Maintenance poll (15.4.13). This response is only applicable to progressive games with one or more progressive levels.

Each game with one or more progressive levels must have only one corresponding set of progressive configuration parameters in the EGM (i.e. a set of progressive parameters per variation is not allowed unless the game is not variation hot-switchable, regardless in either case, QCOM is only currently supporting the reporting of progressive configuration for the currently selected VAR). Also refer 9.2.
CP:1010

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x08, 1 byte hex
GVN   Game Version Number (*15.1.5*) of the parameters in this response, 2 bytes hex
VAR   Currently Selected Game Variation Number (*15.1.6*) for the game denoted by GVN, 1 byte BCD
RES   Reserved = 0, 4 bytes hex.
NUM   1 byte hex,
       Bits 0…3 The number of progressive levels in this response. Max 8. Min 1.
       Bits 4…6 Reserved = 0
       Bit 7   CustomSAP Capability Flag. CP:1011
                If set and the level is a SAP, it denotes that this game's SAP levels are reconfigurable after initial game configuration has been completed. Refer Progressive Configuration Poll (15.4.6).
SIZ   1 byte hex. Repeated entry size in bytes = 17

The following data is repeated for the number of progressive levels in the game denoted by GVN (as indicated by the NUM field). There may be up to eight levels in one response. Levels must be in reported in order starting from level ID 0 (i.e. highest jackpot first). CP:1012

PLF   1 byte hex. Level Flags
             Bits 0..2   Level number 0..7
             bits 3..6 Reserved = 0
             Bit 7   Level type 0 = SAP, 1 = LP *(QPV1.6.1)*
SUP   Jackpot level start-up amount. 4 bytes hex in cents.
             SUP is the amount that the jackpot level's current amount resets to after each hit of the level (provided there was no overflow to carry over).
PINC   Jackpot level percentage increment x 10000. 4 bytes hex.
             E.g. 2.5385% is reported as 25385.
             PINC is applied to applicable game turnover and is used to increment the jackpot level's current amount or overflow meters.
CEIL   Jackpot Level Ceiling. 4 bytes hex in cents.
             CEIL denotes the maximum value of the jackpot level's current amount. Any contribution in excess of this current amount must go into an overflow meter.
AUXRTP   4 bytes hex. Auxiliary RTP for the jackpot level x 10000 (if any), else zero.
             Some games may have an additional component to the basic, or traditional progressive jackpot, whose RTP is represented by this field. (E.g, a hidden increment of the jackpot level's overflow meter). For the basic, or the traditional progressive jackpots, AUXRTP is zero. *NB most systems do not currently support LP totalisation including a hidden increment. EGM manufacturers should check before implementing a LP game with a hidden increment.*

             Refer to the Progressive Configuration Poll (15.4.6) for more information on these fields.
CP:1013

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

- The EGM is only required to be able to queue a single response of this type at a time. ^{CP:1014}
- If this response has been requested and the EGM has LP levels that have not been configured yet (via 15.4.6), then the EGM must report zero for all the parameters for those LP levels in this response. ^{CP:1015}
- If this response has been requested and the game has SAP levels or customSAP levels that have not been configured yet (via 15.4.6), then the EGM must report reasonable[1] default hard-coded SAP parameters for those SAP levels in this response and applied upon game configuration (refer 15.4.3). ^{CP:1016}

---

[1] "reasonable": i.e return an RTP within MINRTP & MAXRTP, ceilings within 95% CI, etc. Parameters must also abide by all jackpot level validity checks listed in section 15.4.6

15.6.5 Bet Meters Response

New in QPv1.6. This response allows the collection of EGM line/credit bet per line statistics on a **per game** basis. This response is queued upon request via the EGM Maintenance Poll (15.4.13). The EGM must maintain one set of bet meters per game. [CP:1017]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x09, 1 byte hex
GVN   Game Version Number (*15.1.5*), 2 bytes hex
RESV Reserved, 1 byte hex = 0
RESD 4 bytes hex. = 0
FLG   Flag. 2 bytes hex.
      Reserved = 0
MAXL Max lines for the game. 2 bytes hex
      Must equal 1 if lines are not applicable to the game.
MAXB Max possible bet per line for the game. 2 bytes hex in credits. [CP:1018]
      If not applicable, make MAXB = Max-bet for the game.
GBFA Game Bet Factor A. 2 bytes hex.
      In reel based games, this field is the number of possible ways (combinations) of selecting total lines bet for a play for the game (e.g. 5). (Additional permutations of the same number of bet lines are not counted.)
      In other game types with no secondary bet factor, GBFA = 1
GBFB Game Bet Factor B. 2 bytes hex.
      In reel based games this field is the number of possible ways to make a bet on a line (commonly 5).
      In other game types (e.g. draw poker, blackjack) this is just the total number of possible ways or combinations to make a bet. [CP:1019]
      Roulette & Keno are exempt and may set GBFB to 1 and report a value of 0 for the single CMET field.
RESE Reserved 4 bytes hex. = 0
SIZ   1 byte hex.
      Size in bytes of each repeated entry below = 4.

For example, if a reel game had two rows of bet buttons labelled 1, 5, 10, 20 & 25 lines and 1, 2, 5, 10 & 20 credits per line respectively, then GBFA= 5 & GBFB = 5 corresponding to the 25 total possible bet combinations.

The product of GBFA x GBFB must be <= 50. If a game has more bet categories than this, then the EGM must group them together in order to make <= 50 bet categories for the purpose to reporting bet meters via this response. [CP:1020] E.g. if a reel game had 25 individually selectable lines and it was possible to bet 1, 2, 5, 10 & 15 credits per line, then GBFA x GBFB = 125. This is too many, to reduce, group selectable lines into groups of 5 (i.e. 1-5, 6-10, 11-15, 16-20, 21-25) giving a total number of bet categories of 25.

The following data is repeated GBFA x GBFB times. Each entry corresponds to specific bet category. The order of entries in the response is GBFA meters first for a given GBFB and lowest bet categories first. For example, in a reel based game, the first 5 meters in the response would correspond to bets made for a 1 credit bet per line, for lines 1…n respectively. Then the next 5 meters in the response would correspond to 2 credits bet per line, for lines 1…n respectively. In the C programming language, this could be represented by a 2D array of the form [GBFB][GBFA].

CMET   4 bytes hex.
          The total number of bets (count) made in this category for the game. [CP:1021]

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

- The EGM is only required to be able to queue a single response of this type at a time.
- The EGM must reset bet meters to zero upon a denomination hot-switch (3.1.3.1) and it is also acceptable for the EGM to change the values of GBFA & GBFB upon a denomination hot-switch but not at any other times. [CP:1022]

15.6.6 Multi-Game/Variation Meters Response

Queued upon request via the EGM Maintenance Poll (15.4.13). Each game in an EGM must each have an exclusive set of these meters (including single game EGMs). However if the EGM allows on-the-fly switching of game variations, then the EGM must have an exclusive set of these meters for each variation of the game.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x0A, 1 byte hex
GVN   Game Version number (*15.1.5*), 2 bytes hex
VAR   Game Variation Number (*15.1.6*), 1 byte BCD

       GVN & VAR denote the game and variation of the meters in this response.

STR   Total Game Stroke (not including free spins or games), 4 bytes hex, LSB first
TURN   Total Game Turnover, 4 bytes hex in cents, LSB first
WIN   Total Game Wins added to credit meter, 4 bytes hex in cents, LSB first.
       This meter includes all SAP wins but excludes all LP wins [CP:1023]
PWIN   Total Game Linked Progressive Wins, 4 bytes hex in cents, LSB first [CP:1024]
       This meter does not include SAP wins.
GWIN   Total Games Won, 4 bytes hex. *(New for QPv1.6)*
       Total number of games for which the win amount was not zero, incremented at the end of each play if the play (including all features and gambles) results in a win. [CP:1025].
       GWIN must not be incremented for each win from gamble/double-up, or RCRF wins or for each win on a free game series or game feature. [CP:1026]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

Note, residual credit removal feature turnover and wins must not be added to these meters. [CP:1027]

The EGM is only required to be able to queue a single response of this type at a time *(QPv1.6)*. *(QPv1.5 EGMs, in a multi-game EGM, if a request for game A and game B's (etc) multi-game/variation meters is received before the EGM has responded with either, both games meter responses must still be sent [CP:1028]. Or in a multi-variation EGM, if a request for game A's variation 01 and variation 02 multi-game/variation meters is received before the EGM has responded with either, both game/variation meter responses must still be sent [CP:1029].)*

Base Game RTP & Standard Deviation Display. *(QPv 1.6.2)*

In EGM audit mode on the page where the above meters are to be displayed for a given game variation, the EGM must also display the base game RTP (WIN/TURNx100) & standard deviation of the percentage RTP of the games played since last RAM clear for the displayed game. Labelled "W/T %RTP" & "Std. Dev." (decimal numbers to 2 decimal places). [CP:1030] The formula is (in C notation): (NB: PWIN is excluded from this calculation)

$$SD = sqrt(fabs(R / STR – (WIN/TURN)\char`^2))$$

Where: **R** = per play sum of ((ΔWIN/ΔTURN) ^ 2);
("Δ" denotes "delta" or change in value)
Meaning at the end of each play:
R = R + (total win for the play excl LP wins / bet for the play) ^ 2
Store R in NV memory of the EGM
"**sqrt**" denotes the "square root" function
"**fabs**" denotes an "absolute" value function and
"**^**" denotes "to the power of"

- Divide by zero errors must be handled (i.e. when TURN == 0). [CP:1031]
- It is accepted that this formula will break upon a TURN meter rollover, however rollovers are unlikely to occur in this protocol.
- The R term and the variables used in the calculation must be double precision floating point numbers (i.e. minimum 64 bits).

Display example:

W/T %RTP:	85.68%
Std. Dev.:	4.34

15.6.7　Player Choice Meters Response

New in QPv1.6. This response allows the collection of player choice statistics on a **per game** basis (where player choices are applicable for a game). This response is queued upon request via the EGM Maintenance Poll (15.4.13).

Each meter represents one possible choice presented to the player during play; each meter is simply incremented if that choice is selected.

The EGM must maintain one set of player choice meters per game in the EGM when applicable. The only player choices that must be tallied by the EGM for a game via these meters, are the choices which affect the player's standard deviation of any feature or prize and where there is no apparent best strategy (e.g. such as in blackjack for example).

An example of a choice that must be metered is in the case of a game feature asking the player to pick between a feature of either say 5 free games, a random bonus or other feature. Typically the RTP is the same for each, but the standard deviation that results for each option may be very different.

Types of player choices that are not required to be tallied are for games such as Blackjack and Draw Poker, as the game's RTP is directly affected by the choice as well as the standard deviation, the player choices won't be tallied in these cases because there is only one best strategy and the player is also being prompted by the EGM of the best strategy to pick anyway. Other examples of player choices that would not be required to be metered are in the case where the player's choice has no consequence, e.g. when the player would obtain the same RTP & Standard Deviation no matter which option they choose. E.g. a pick a box feature where the same outcome can be obtained from choosing any box (i.e. it doesn't matter which box was actually chosen).

Message Format and Order:

DLL　　Data Link Layer, Refer Section 14.3
FC　　　Function Code = 0x0B, 1 byte hex
GVN　　Game Version Number (*15.1.5*), 2 bytes hex
FLG　　Reserved = 0. 2 bytes hex.
NUM　　1 byte hex. Number of repeated entries below. 0…16 (Maximum) [CP:1032]
SIZ　　1 byte hex.
　　　　Size in bytes of each repeated entry below = 4.

The following data field is repeated NUM times.

PCMET　　　　4 bytes hex.
　　　　　　　The total number of times this choice was made for the game.
　　　　　　　Player choice meters must be ordered in the message with choices resulting from the most probable features first, then ordered from top to bottom, left to right, as those choices were presented to the player. [CP:1033]

End of repeated entry data.

CRC　　Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Notes:

The EGM is only required to be able to queue a single response of this type at a time. [CP:1034]

15.6.8   Meter Group/Contribution Response

This message type is queued by the EGM for transmission whenever any EGM group meter (refer section 12.1) changes value containing the changed/updated meter amounts (also see 12.2.2), or whenever group meters are requested via the EGM Maintenance Poll (15.4.13) containing the requested meter groups.

Each response must contain the latest value of the meters within it at the time the response was built. For example, this response would be queued at the beginning of a play (containing the updated stroke and turnover), at the end of a play if a win resulted, any credit input, etc. Refer section (12.2) for more information when to queue meters via this response.

This response may also contain a LP contribution if the game is a LP. See below.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x0C, 1 byte hex
FLG   1 byte.
        bits 0...2   Number of group meters in this response **minus one** (0x00...0x07) i.e. 1...8 Max
        bits 3...6   reserved = 0
        bit 7   If set, it denotes the response contains a linked progressive contribution. See below.
LGVN   2 bytes hex. The Game Version Number (*15.1.5*) of the last initiated game on the EGM, else zero if no game has ever been played since the last EGM RAM clear [CP:1035]. Update LGVN upon each play irreversibly commencing (i.e. so the LGVN always corresponds to any turnover and progressive contributions in this message) [CP:1036]. In a single game EGM, this field may always be set to the only GVN of the EGM.

The following message data is repeated by the number of times as indicated by the FLG field.

MGID   EGM Meter ID, 1 byte hex, (Refer Section 12.1)
MET   4 bytes hex. EGM group meter current value as denoted by MGID

End of repeated entry data.

    Different group meters must be able to be mixed in a response (refer to section 12.2 for examples). If the EGM has meters queued to send via this response, then the response when built, must be filled with all pending meters (regardless of group ID) while there is space still available in the response (max 8 meters per response). [CP:1037] When building each response of this type, the priority of reporting meters is lowest queued meter ID first (this is within a single response and between multiple responses) [CP:1038]. (This priority means that group meters with the higher ID numbers, when queued for transmission, their transmission can be delayed by lower numbered pending group meters.)

    Suggestion for implementation of this response type for EGMs: The EGM should have one 'need-to-send' flag per meter and one overall flag that indicate that one or more group meters need sending. Then when a meter changes, set the meter flag and the overall flag. For ACK/NAK purposes, the EGM will also need to store the last transmitted meter ID numbers in the last meters response.

LP Contribution Message data: (Presence optional; indicated via FLG: bit 7)

    The EGM must send a linked progressive contribution along with the total EGM turnover group meter whenever a game is instigated which has a linked progressive component. Specifically, Meter Group/Contribution Responses containing a progressive contribution must be queued for transmission immediately after the start of a play and also must contain the latest stroke meter and turnover meter. [CP:1039] A LP contribution ~~will~~ must only be present in a meter group

response when a new game has been instigated or when a previous group meter response with a LP contribution was NAK'ed. [CP:1040]

PGID   Linked Progressive Group I.D. number of this contribution (*15.1.10*) for the game GVN, 2 bytes hex

PLVL   1 byte hex. Reserved, must be set to 0xff

PAMT   LP Turnover Meter [CP:1041]. The total game turnover applicable to PGID from the game denoted by LGVN above, 4 bytes hex in cents, LSB first. Refer to the glossary (2.2) for more information on the LP Turnover Meter.

End of optional data fields (LP Contribution Message data).

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Other requirements:

In a multi-game EGM, after any play, upon return to idle mode, any currently queued group meters, or any group meters awaiting acknowledgement, or any pending LP contribution, must be sent and acknowledged by the SC before the EGM may allow a different game to be selected and played. [CP:1042] (I.e. This requirement is not applicable when the same game is continuously played.) To implement this requirement, it is recommended that after each play, that the EGM disables the selection of other games in the game select display (e.g. grey-out) while any group meters are either pending for transmission, or awaiting transmission acknowledgement from the SC. Once there are no outstanding group meters, the EGM can then enable new game selection until after the next play. [CP:1043] The intent of this requirement is to allow a SC to pre-emptively increment multi-game meters without having to continually request the Multi-Game/Variation Meters Response (15.6.6) and to help ensure any LP contribution for a game is not missed.

15.6.9   Event Response

Queued by the EGM whenever the EGM has new or unacknowledged events to report. Refer to section 7 and section 7.10 for the list of defined events.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x10, 1 byte hex
SEQ   Event sequence number, 1 byte hex. (Modulo 256)+1, (i.e. 1..255)
            0x00 is reserved for unnumbered events which are stored and sent from the secondary event
            queue [CP:1044].
ECOD Event code, 2 bytes hex, LSB first.
ESIZ   1 byte hex
            bits 0...4   Size of EXTD field in bytes. Zero if no EXTD field present.
            bits 5...7   reserved = 0

TIME   Seconds, Minutes, Hours, 1 byte each, BCD, 24 hour format.
DATE   Current system date. Day, Month, Year, 1 byte each, BCD

            TIME & DATE is the date and time stamp (not TZADJ adjusted) of when the event was created.

EXTD   ESIZ bytes. Length of EXTD may be zero.
            An event may have additional event data, such as a cash amount attached (e.g. lockup events
            e.g. cancel credit). See event code descriptions for more information (7.10).

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

15.6.10   Purge Events Poll Acknowledgement Response

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x0F, 1 byte hex
FLG   Reserved, 1 byte hex. *QPv1.6*
PPSN   1 byte hex. *QPv1.6*
        The current Purge Poll, Poll Sequence Number (refer section 15.1.9 for an explanation of PSNs). [CP:1045]
        Immediately after EGM RAM clear this field must default to 0x01 (refer 8.1.15). After the first successful Purge Events Poll (15.4.15) this field would be reported for the first time in this response with a value of 0x02. [CP:1046]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

This response is queued by the EGM whenever it receives a Purge Events poll (15.4.15) *QPv1.6.* [CP:1047]

(QPv1.5 EGMS only queued this response if the purge was successful i.e. the Purge Poll PSN was valid. NB: this meant SCs had to handle two different purge ack methodologies. This had a potential problem with event purging in the case when a purge poll is sent and the EGM stopped responding. If the EGM still didn't respond to the purge poll when resent later on, then it wasn't possible to tell if the EGM ever got the poll or whether its purge poll PSN was incorrect.)

Refer to the Purge Events Poll (15.4.15) for more information on purging.

This response allows the SC to easily detect out of sync Purge Event poll PSN numbers in all cases.

15.6.11   EGM Game Configuration Response

This message is queued for transmission for each game (not variation) in the EGM, in response to the EGM Configuration Request Poll (15.4.1) or a game at a time when requested via the EGM General Maintenance Poll (15.4.13). *QPv1.6:* Because the list of available variations may change up until EGM configuration has been completed (9.1), this response must not be sent if EGM configuration has not been completed via the EGM Configuration Poll (15.4.2). [CP:1048]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x13, 1 byte hex
GVN   Game Version Number (*15.1.5*) of the game data in this response, 2 bytes hex
VAR   Game Variation Number (*15.1.6*), 1 byte BCD. Denotes currently selected variation of the game [CP:1049].
       The EGM must set VAR to zero if the game is not yet configured via EGM Game Configuration Poll (15.4.3) [CP:1050].
FLG   Game flags. 1 byte hex.
       Bits 0…3   reserved = 0
       Bit 4   LP Only Flag. *QPv1.6*
           This flag is applicable only to LP games. If set it denotes that all the game's progressive levels are only available to be setup as LP and the EGM will not allow any levels to be setup as SAP. [CP:1051]
       Bit 5   Variation Lock. *(QPv1.6)*
           If set, this denotes that the selected variation denoted by VAR above has been locked in this game (this only possible by the SC via the EGM Game Configuration Poll, refer section 15.4.3) and therefore the variation cannot be changed by a hot switch (15.4.4). [CP:1052] Once locked, only a RAM clear must be able to change the game's variation. The default value before game configuration has been completed for the game via 15.4.3, must always be zero [CP:1053]. Each game in an EGM must have one variation lock bit. [CP:1054]
       Bit 6   Variation Hot-Switching (HS) Capability Flag.
           If set, denotes this game supports on-the-fly variation switching [CP:1055]
       Bit 7   Game Enable Flag (15.1.3), current value for the game denoted by GVN
           1 = enabled, 0 = disabled [CP:1056]
PNUM   Total number of progressive levels in the game GVN. 0…8, 1 byte hex
PLBM   Progressive Level type Bit Mask, 1 bit per level, up to eight levels. 1 byte hex.
       Bit 0 corresponds to level 0, bit 1 = level 1, ..., bit 7 = level 7.
       0 denotes a SAP configured progressive level.
       1 denotes a LP configured progressive level.
       Note, 0 is the default value to send before configuration of the game if the level is only available as SAP, if the level is configurable as either LP or SAP, or LP only, then the default value before configuration is 1 [CP:1057]. Unused bits must be set to 1 [CP:1058]
PGID   Progressive Group I.D. of GVN (Refer to section 15.1.10), 2 bytes hex
       A value in the range 0x0001-0xFFFE indicates the game has a progressive component and has already been configured with one or more LP levels and PGID denotes the LP group I.D. of the game [CP:1059].
       0xFFFF indicates the game is a non-progressive game or was configured as, or is a SAP only game [CP:1060].
       0x0000 indicates the game requires configuration (as a SAP or LP) via the EGM Game Configuration Poll (15.4.3) [CP:1061].

NUM   1 byte hex. Total number of variations available for selection[QPv1.6] in this game. (1...16 max)[QPv1.6] *(QPv1.5: Maximum was 8 variations per game and all were selectable)* QPv1.6 EGMs may have more variations resident in the game, but only offer a maximum of 16 variations for selection at any time, based on EGM Configuration Poll (15.4.2) settings.
       The list of offered variations must not change unless the EGM is RAM cleared. Also see section 9.
SIZ   1 byte hex. Size of each individual repeated entry = 0x03

The following data is repeated NUM times in order of variation number lowest to highest.

VAR   Game Variation Number (*15.1.6*), 1 byte BCD

PRET   Theoretical Percentage Return to Player of the non-progressive component of the variation *(QPv1.6)* denoted by GVN & VAR. 2 bytes hex x 100.
e.g. for a 85.07% return game, the value would be 0x213B (i.e. 0x3B tx first then 0x21)
(NB: PRET was BCD in QPv1.5 but not used by the monitoring systems. Also in QPv1.5 PRET included progressive component except for some LPs)
For games where PRET varies (for example depending on betting options) then the PRET for the minimum bet must be reported here.
For games that have a strategy (e.g. blackjack) then PRET should be the RTP for the default/time-out strategy (typically the optimal strategy) implemented by the EGM for the game.
It should be noted that the methodology to use for reducing the number of decimal places on a game's theoretical PRTP (TPRTP) value to two decimal places for use with the PRET field has **not** been specified in any prior version of QCOM. This means that there are multiple methods currently used by EGM manufacturers making **the accuracy of any reported PRET value up to +/- 0.01%**. At this time, systems are allowed (refer SCP document) to use this field for < > comparisons with respect to MINRTP and MAXRTP boundaries. This means that EGM manufacturers have to be mindful when specifying a PRET value to ensure they don't inadvertently round over a limit they would otherwise which their game to be eligible for. E.g. In QLD systems typically perform PRET < 92.0, ergo rounding TPRTP up to give PRET is a bad idea.
~~For future consistency, all future i~~lmplementations of TPRTP to PRET conversions in QCOM EGMs must round down.^QPv1.6.5 I.e. Any significant digits dropped due to precision limits must truncated (i.e. rounded down) NB Rounding down means a game trying to meet a MINRTP value must be at least >= MINRTP (where as before a game could get close and relay on a round up to make the MINRTP.

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Note, in a multi-game EGM, if a request for game A and game B game configuration responses is received before the EGM has responded with either, both games Game Configuration responses must be sent [CP:1062].

15.6.12   EGM Configuration Response

This message is queued for transmission in response to an EGM Configuration Request Poll (15.4.1).

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x16, 1 byte hex
SER   EGM Serial number (15.1.8), 3 bytes BCD, LSB first
      This field when combined with MID below, yields a 4 byte unique ID number for each EGM.
MID   EGM Manufacturer I.D., 1 byte BCD. Refer section 15.1.7
NPRV Network Protocol Version 1 byte hex
      Bits 0&1   QPV [CP:1063]

                              QCOM Protocol Version = 0x00 for QPv1.5.x EGMs
                              QCOM Protocol Version = 0x01 for QPv1.6.x EGMs
                              For full information about this field see QPV in section 14.3.1
      Bits 2…7   Reserved = 0
FLGA Flags byte, 1 byte hex. The status of these flags is set once by the EGM, or manually by an installer upon every RAM clear and then must not change. (i.e. once "expected", a device is always expected by the EGM until the EGM is next RAM cleared. This is for security reasons, so if the device is not subsequently 'detected', then this would probably generate a fault on the EGM).   Auto-detection by the EGM is ~~good, especially~~recommended for defaults, however there ~~is~~ must also ~~a need for~~ be a manual override, in order to force a device into being "expected" (e.g. in the case an EGM is being setup after a RAM clear with a currently offline or faulty device that ~~should~~ is intended to be ~~there~~present).

      These flags have nothing to do whether a device is ~~disabled,~~ physically present, or in a fault. They simply indicate that the device "expected". Indicating that the device is "expected" by the EGM must mean that the EGM is monitoring it and that it has the required support in QCOM implemented for the device (i.e. an EGM must not indicate a banknote acceptor is "expected" if it doesn't also have the required driver to send the appropriate QCOM events and meters (12.1) for it). [CP:1064]

      Unless a peripheral device is critical to game play, machine operation or specifically stated as mandatory under these requirements, then the EGM must be able to be comissioned at RAM clear and subsequently enabled for play without the device present. [CP:1065]

      A zero bit value means the device not expected. [CP:1066]

      bit 0   1 = Mechanical Meters expected *(QPv1.6)*. [CP:1067]
      bit 1   1 = EGM NV-RTC detected and is in use for QCOM event time-stamping. [CP:1068]
      bit 2   1 = Progressive Display device expected. [CP:1069]
      bit 3   1 = Touch Screen expected. [CP:1070]
      bit 4   1 = Cash Ticket Printer (or ticket out device) expected. [CP:1071]
      bit 5   1 = Coin/Token Validator expected. [CP:1072]
      bit 6   1 = Hopper expected. [CP:1073]
      bit 7   1 = Note Acceptor expected. [CP:1074]
FLGB 1 byte hex.

      Bits   0…5   Reserved = 0
      Bit 6   1 = Auxiliary Display Device (second screen) expected *(QPv1.6)* [CP:1075]
      Bit 7   1 = Ticket In Device expected (*QPv1.6*) [CP:1076]

ODEN Reserved = 0, 2 bytes BCD. EGM Credit Denomination. *Used only by QPv1.5 EGMs.*
OTOK Reserved = 0, 2 bytes BCD. EGM Token Denomination. *Used only by QPv1.5 EGMs.*

BSVN Base Software Version Number for the EGM (refer 15.1.4). 2 bytes hex.
NUMG Total number of available games (GVN) in the EGM (= 1 if single game machine), 1 byte hex. (Refer section 9) [CP:1077] i.e. NUMG must equal the total number of Game Configuration Responses (15.6.11) reported when requested via 15.4.1->FLG->bit 6. *QPv1.6:*This figure may

decrease after EGM configuration is completed (via 15.4.2) as some games may be made ineligible ~~illegal~~ by certain EGM configuration settings (such as MINRTP/MAXRTP). Games made ineligible ~~illegal~~ by EGM configuration settings must never be reported via any response types. Once EGM configuration is complete NUMG must not change again unless the EGM is RAM cleared.

NUME 1 byte hex. Total number of games that can be enabled at any one time on the EGM. Some EGMs may have more games resident in the EGM than can actually be enabled for play at any one time. Depends on the EGM, but usually this is a limit imposed by the game select screen. [CP:1078] (Refer section 9)

LGVN Game Version number (*15.1.5*) of the last initiated game in the EGM, 2 bytes hex
Refer to the Meter Group/Contribution Response (15.6.8) for more information on this field.

LVAR Game Variation Number (*15.1.6*), as above, 1 byte BCD
If no game has ever been played since the last EGM RAM clear, then LGVN & LVAR fields must equal zero [CP:1079]

New fields for QPv1.6 EGMs:

FLGS 2 bytes hex.
       Bits 0…13   Reserved = 0
       Bit 14   Shared Progressive Component Flag. *(QPv1.6)*
                This flag is only applicable in multi-game EGMs that have a progressive component. Refer to section 10.9 for more information.
       Bit 15   Denomination Hot-Switching [CP:1080]
                If set, denotes that the EGM supports denomination hot-switching (refer 3.1.3.1).

DEN EGM Credit Denomination, 4 bytes hex in cents, LSB first

TOK EGM Coin/Token Denomination, 4 bytes hex in cents, LSB first
The EGM will assume default DEN & TOK denominations of zero in this response after a RAM clear until configured by the EGM Configuration Poll (15.4.2) [CP:1081].

The following fields are set via the EGM Configuration Poll (15.4.2) and they are reported back here for verification by a SC. (Refer 8.1.13 for default values at EGM RAM clear before EGM configuration has taken place):

MAXDEN	4 bytes hex.
RES	4 bytes hex reserved = 0
MINRTP, MAXRTP, MAXSD, MAXLINES, MAXBET, MAXNPWIN, MAXPWIN,	Each field is 2 bytes hex, LSB first
MAXECT	Each field is 4 bytes hex in cents.

Refer to the EGM Configuration Poll (15.4.2) for the definition of these fields.

CRC Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

(QPv1.6) Except for the following fields: NUMG, DEN, TOK & CRC, all the data fields in this response must be static (i.e. they must not change once reported by the EGM in the first EGM Configuration response after a RAM clear). *(In QPV1.5 EGMs the entire response was completely static)*

15.6.13   Program Hash Response

This response is queued upon completion of an EGM program hash calculation. Refer to Section 11 on Program Hashes for more information.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x19, 1 byte hex
RES   1 byte hex. Reserved = 0x00
SIGN   EGM Program Hash, 20 bytes hex *(QPv1.6)*, LSB first, zero padded if necessary [CP:1082].
        (QPv1.5 was 8 bytes hex and only the first 4 bytes was used)
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

16 Electronic Credit Transfers and Cashless Operation (ECT)

This section details the EGM cashless operational and protocol specification.

Electronic Credit Transfers (ECTs) are used to transfer credit between the EGM and SC for the purposes of cashless gaming and the transfer small frequent progressive or system awards to the EGMs credit meter.

The protocol deals strictly with the transfer of credit only, and does not have any concern for, or need to know for example, account information such as account numbers and PIN numbers etc. These are matters for the monitoring system operator.

~~The last amount transferred (if not zero [CP:1083]) via ECT, either to or from the EGM, must be displayed on the EGM (in $.¢) while in idle mode in real time (suitably labelled: i.e. "cashless in" [CP:1084] or "cashless out" [CP:1085]) for no more than 10 seconds, or until commencement of the next play or ECT in/out [CP:1086], whichever is sooner. Acceptance of the overall legibility of the display is at the discretion of the CEO [CP:1087].~~

The cumulative [CP:1088] transferred amounts during and between plays must also be displayed in last play information recall for all stored plays [CP:1089]. (Refer to GMNS last play recall requirements for play start/end boundary definitions here; please ensure coins, banknotes and ECT have all the same timing in this regard.)

The ECT meters in last play recall must be labelled "ECT/JACKPOT IN" [CP:1090] and "ECT OUT" [CP:1091]. (NB: the ECT transaction log (below) will primarily be used to audit recent cashless transfers and timing there-of) [QPv1.6.4].

(QPv1.6) The EGM must display in a suitable page in audit mode, a log of at least the last 20 attempted[*] ECT-in/out transactions. Display: Type (In/Out), date & time (TZADJ adjusted based on the ECT event time), Exp. PSN, Rxd. PSN (*NB: PSN entries are 'NA' or '—' for ECT-from-EGM*), cashless mode bit (*NB: 'NA' or '—' for ECT-from-EGM*), amount ($.c) & status (Success/Fail/Waiting Ack). ([*] transactions that fail due to an incorrect PSN, or >= MAXECT, must be included in the log) [CP:1092]

E.g. of an ECT Transaction Log:

```
                          Exp   Rxd   CMde
Date & Time       Type    PSN   PSN   Bit      Amount
                  Status

21/11/2006 14:21:00 In(ECT)   12h   12h   1        $123.45        Success
21/11/2006 14:21:23 In(LP)    12h   12h   1        $123.45        Success
21/11/2006 14:21:55 Out       NA    NA    NA       $432.10        Success
…
```

() above denotes the ECT Source ID Descriptor Field ~~is implemented by the EGM (see below) then it must also be shown in the above log.~~ It may be either the ECT Source ID raw numeric value or descriptor.

16.1 Electronic Credit Transfers to the EGM

16.1.1 'ECT to EGM' Poll

This poll is used to send credit to the EGM's credit meter and/or to direct the EGM in or out of cashless mode (16.2).

Message Format and Order:

DLL Data Link Layer, Refer Section 14.2
FC Function Code = 0x22, 1 byte hex
PSN Credit Transfer - Poll Sequence Number (15.1.9), 1 byte hex
FLG 1 byte hex.
       Bits 0…2   ECT Source ID. See below.
       bits 3...6   reserved = 0, mask out these bits when reading any data from this byte[CP:1093].
       bit 7   Cashless mode bit.
           1 = Enter cashless mode (16.2) [CP:1094]
           0 = Do not enter cashless mode [CP:1095], or exit cashless mode if already in [CP:1096].
OCAMT 3 bytes BCD, in cents.
     This field is processed only by QPv1.5 EGMs and must not be processed by QPv1.6 EGMs. [CP:1097]

     Validity check: must be BCD [CP:1098]
     Amount to add to current credit meter.
     Maximum $9,999.99. [CP:1099]
EAMT 4 bytes hex, in cents. *This field is processed only by QPv1.6 and newer EGMs.*
     Amount to add to current credit meter. [CP:1100]
CRC Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

All ECT requests must be ignored if EGM configuration has not been successfully completed via the EGM Configuration Poll (15.4.2). [CP:1101]

The following conditions must be met in order for the ECT-to-EGM to take place:

- The EGM must have accepted an EGM Configuration Poll (15.4.2)[1] before it will process any part of the ECT-To_EGM Poll message data (apart from the DLL and FC which must always be processed). [CP:1102]
- the PSN must be valid (see 15.1.9). [CP:1103]
  If the PSN was valid, the EGM must immediately update its next expected ECT PSN at this time [CP:1104]
- EAMT must be less than MAXECT (*QPv1.6*) [CP:1105],
- the credit meter display (on all screens) must be able to handle the display of it current amount plus EAMT (*QPv1.6*) [CP:1106] Any EAMT that would wrap the credit meter must also be considered unable to be displayed. [CP:1107]

If all the above conditions are met, the EGM will immediately add EAMT if not zero [CP:1108] to the Total Cashless In and the Total Cents In meters and queue an EGM Meter Group / Contribution Response (15.6.8) [CP:1109]. Then, if the EGM is in idle mode[2], or a play is in progress[3] (and the play is not suspended for any reason), it must add EAMT to the credit meter and set/reset cashless mode (16.2) with respect to the cashless mode bit [CP:1110]. Otherwise if the EGM was not in idle mode [or a play is in progress] (e.g. fault/lockup, test/audit mode, door open condition, etc – refer to Appendix A), it must

---

[1] It is at the discretion of the EGM manufacturer as to whether or not it will allow an ECT-to-EGM to take place before any games are configured.
[2] Note, since Communications Disabling Conditions are defined as a part of idle mode in QCOM, Cashless-In's must be added to the credit meter during one or more CDCs while in idle mode. Refer 6.1
[3] it is not mandatory to support crediting cashless-in to the credit meter while an EGM is in play[*QPv1.6.4*], the EGM may wait until the EGM returns to idle mode if desired.

queue in critical NV memory [CP:1111] a command to add the amount to the credit meter and set/reset cashless mode accordingly upon next return to idle mode [CP:1112] (refer 3.3.2 RTIM).

(The reason for changing the state of the cashless mode bit only in idle mode [or during play], is because changing it at other times could cause unexpected results, e.g. if the EGM was already in an 'ECT From EGM' Lockup condition at the time. The reason for updating the credit meter in idle mode for an 'ECT-in' is for similar reasons, e.g. a credit meter update during a hopper payout or gamble could also have unexpected results)

If any of the above listed conditions are not met then the transfer must not take place and EAMT and the Cashless Mode bit must be ignored *(QPv1.6)*. [CP:1113] (*QPv1.5 EGMs would always process the Cashless Mode bit.)* If the reason for the failure is due to the credit meter display width being exceeded (refer last dot point in the list above), or any arbitrary EGM implemented credit meter limit, then the EGM must log the 'Transaction Denied - Credit Limit Reached' event (7.10.3.44)[CP:1114]

If multiple valid ECT-to-EGM polls were received before the EGM had added any prior EAMT to the credit meter, then the EGM must accumulate the EAMT field amounts to be added to the EGM's credit meter upon the next return to idle mode [CP:1115] (see 3.3.2). A 'pending cashless mode flag' (NB: only the last received cashless mode bit must be stored) and a 'Queued ECT-in' meter will be required in order to implement an 'ECT to EGM'. The 'Queued ECT-in' meter should also be taken into account when the EGM performs a self audit and (*QPv1.6)* this meter must also be displayed in audit mode on an appropriate page [CP:1116].

Any residual amount not of a whole credit value will be stored by the EGM until it is either transferred back to the SC [CP:1117], rounded off (due to subsequent ECT transfers) [CP:1118], gambled (via the residual credit removal feature) [CP:1119] or cancelled [CP:1120] or cashed out via a ticket print [CP:1121].

If the currently displayed credit meter is zero but some residual credit (credit not of a whole credit value) remain, the EGM must also display to the player the residual credit amount in $.¢ if not already displayed. [CP:1122]

This poll may be used with a zero EAMT value to command the EGM in or out of cashless mode (16.2) without transferring credits [CP:1123]. *(FYI QPv1.5 EGMs would always process the Cashless Mode bit e.g. regardless of the value of the ECT PSN.)*

When credit is actually added to the EGM's credit meter as a result of this poll, the EGM must emit a short cash-in sound effect [CP:1124].

ECT Source ID Field

Refer FLG field above. (Mandatory from since QPv1.6.5)

This field is a FYI field for the EGM, it provides the EGM with more information with regards to the source of the ECT which can be displayed in audit mode. The EGM must display a table of ECT in amounts by Source ID or Descriptor on a suitable screen in EGM audit mode. [CP:1125] and tThe 'ECT Source ID number or Descriptor Ffield' must also be displayed in the audit mode 'ECT Transaction Log' (see above) for each ECT-to-EGM entry. [CP:1126]

For display purposes in EGM audit mode, the possible IDs currently predefined are:

Source ID	Descriptor	Description
0	LP	QCOM LP Prizes (i.e. EGM Triggered)
1	ECT	Cashless Gaming / Account Based Gaming
2	JPA	External Jackpot A
3	JPB	External Jackpot B
4	JPC	External Jackpot C
5	TITO	Ticket-In (TITO)

| 6 | RES6 | Reserved6 |
| 7 | RES7 | Reserved7 |

16.1.2   ECT-To-EGM Acknowledgement Response (QPv1.6)

This response (new for *QPv1.6*) is queued by the EGM whenever it initially processes (i.e. upon receipt of) an 'ECT to EGM' poll (Refer section 16.1.1), regardless of whether the ECT-to-EGM is accepted or not, for whatever reason. [CP:1127] This allows the SC to more easily detect out of sync 'ECT to EGM' Poll PSN numbers especially when transferring zero credit in order to toggle the EGM's cashless mode flag. The only time the ECT-to-EGM Ack. Response is not queued in response to an 'ECT to EGM' poll, is when EGM Configuration has not been completed via the EGM Configuration Poll (15.4.2). [CP:1128]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x0E, 1 byte hex
FLG   1 byte hex. Reserved = 0
ECTPSN   Next ECT PSN. 1 byte hex
        The next expected 'ECT to EGM' Poll Sequence Number. (Refer section 15.1.9 for an explanation of PSNs). [CP:1129]
        Immediately after EGM RAM clear this field would default to 0x01 (8.1.15) and after the first ever successful ECT to EGM Poll this field would be reported for the first time in this response with a value of 0x02.
TCCIM Total Cashless Credit In Meter. 4 bytes hex in cents.
        This is the same meter as the group meter ID = 0x14 (refer 12.1.2) [CP:1130]
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Note to SCs. The EGM may increment its ECT PSN indicating that it received an ECT to EGM poll with a valid PSN, however the EGM could still ignore the ECT, if the amount was too large. This shouldn't happen but SCs should still also verify that the EGM total cashless-in meter above reflects the last ECT-in amount.

16.2 Cashless Mode

Cashless mode is defined as follows:

The EGM will place itself in cashless mode depending on bit 7 of the FLG field in the 'ECT to EGM' poll. While in this mode, pressing of collect on the EGM, instead of initiating a hopper pay, ticket voucher printing or cancel credit, the EGM will initiate an "ECT from EGM lockup" [CP:1131]. Also while in cashless mode, all cancel credits will initiate an "ECT from EGM" lockup instead.

The EGM will remain in cashless mode until either the EGM next exits from an 'ECT from EGM' lockup condition (regardless of success [CP:1132] or fail [CP:1133], see below), at which time the EGM automatically returns from cashless mode [CP:1134], or until directed out of cashless mode via another 'ECT to EGM' poll [CP:1135].

16.3 Credit Transfers from the EGM

Listed below are the only ways to initiate an ECT of EGM credit to the SC (the EGM must be in cashless mode (16.2) first and not have zero credits [CP:1136] and the amount to be transferred is less than MAXECT[QPv1.6]):

1. The EGM receives an 'ECT from EGM' Lockup Request Poll (refer 16.3.2) [CP:1137]
2. Pressing collect on the EGM while in idle mode [CP:1138]
3. A Cash-Out Request Poll is received (0) [CP:1139]
4. An applicable win exceeding either NPWINP [CP:1140], or SAPWINP [CP:1141] win payout thresholds (refer 15.4.5).

(*Most QPv1.5 EGMs also initiated a ECT out on zero credit if the EGM was in cashless mode* [CP:1142]*, but some exemptions were granted*)

In all cases above, except for case 4, the amount to be transferred must be the whole current EGM credit meter value. In case 4, the amount to be transferred is the win resulting from the play.

(QPv1.6) If the amount to be transferred is >= MAXECT, then the EGM must immediately reset itself from cashless mode [CP:1143] and a hopper payout [CP:1144], cancel credit lockup [CP:1145], or ticket out [CP:1146] must result instead (which one depends on the current hardware configuration of the EGM and the thresholds set in the Hopper/Ticket Printer Maintenance Poll (15.4.17)).

Also refer to section 3.5 on credit redemption.

16.3.1 'ECT from EGM' Lockup

If any of events occur listed in section 16.3, the EGM will lockup in an 'ECT from EGM' lockup condition at the next opportunity via idle mode [CP:1147] so long as the amount to be transferred is less than MAXECT.

Upon entry into this condition, the EGM will log an ECT Event [CP:1148] (16.3.3) and constantly display to the player, the static message:

"Transferring $xx.xx Please wait…",

the lockup must have no other title. [CP:1149]

Where "xx.xx" is the amount to be transferred in $.¢ plus any residual [CP:1150]. It is preferred if the lockup has no other title or label. See the figure below for an example. The EGM will remain locked up in this condition displaying the above message until reset by the ECT Lockup Reset Poll [CP:1151].

The EGM must not display "call attendant" or the equivalent during the lockup as this condition is automatically cleared by the SC. [CP:1152]

Meters are updated upon exit from the lockup depending on the command received in the ECT Lockup Reset Poll. [CP:1153] Refer to section 12.2 Meter Updating for more information.

Figure 9 Example of an ECT-from EGM Lockup

Screen dump courtesy of IGT (Australia) PTY LTD.

16.3.2 'ECT from EGM' Lockup Request Poll

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x23, 1 byte hex
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Upon receipt of this poll, if the EGM is currently in cashless mode (16.2) with non-zero credit[QPv1.6] and idle mode, it will enter an 'ECT from EGM' lockup condition [CP:1154]. If the EGM was not currently in idle mode and in cashless mode (and not already in an 'ECT from EGM' lockup condition), it will store the request in NV-RAM (regardless of the credit meter value) and then enter the lockup immediately upon next return to idle mode as long as the credit meter is not equal to zero and the cashless mode flag is still set at that time [CP:1155] (see previous checkpoint for special case. Also refer to section 3.3.2).

If the EGM was not in cashless mode upon receipt of this poll, then the message must be ignored [CP:1156].

Implementation note: Bugs in EGM implementations of ECT are commonly found when a simultaneous ECT in/out occur on an EGM. There are two cases; either an 'ECT to EGM' occurs when the EGM was already in an ECT lockup, or both an 'ECT to EGM' and ECT Lockup Request is received when the EGM was not in idle mode. In the second case it is required the EGM carries out the ECT lockup first. To avoid any problems, the EGM should only write to its cashless mode flag in idle mode and keep any 'ECT from EGM' state variables independent of 'ECT to EGM' state variables. [CP:1157]

SCs note: Upon sending this poll when the EGM is not in idle mode, SCs should also send MEF=0 and CRLIMIT=0 to disable physical credit input once an ECT-from-EGM has been queued, this avoids the potential for another player's credit from inadvertently being transferred to that player's account.

16.3.3 ECT from EGM Event

This event is logged by the EGM upon entry into an 'ECT from EGM' lockup condition [CP:1158].

Event Code 0x3009   *QPv1.6* This is a secondary or unnumbered advisory event type.
                           (Event code was 0x3004 in QPv1.5)

Display Format: **"EGM ECT From EGM $<CAMT>"**

Extended event data: **<CAMT>**

CAMT   4 bytes hex in cents, display up to 14 characters of unsigned currency (e.g. $xx,xxx,xxx.xx). CAMT is the amount to transfer to the SC [CP:1159].
        This field was 4 bytes BCD in QPv1.5.

This event has no follow up event type (i.e. 0x01 Lockup clear) [CP:1160].

16.4 ECT Lockup Reset Poll

If the EGM was in an 'ECT-from-EGM' lockup, then upon receipt of this poll, the EGM will perform the function indicated by the CFUNC field, exit the current 'ECT from EGM' lockup condition [CP:1161] and return from cashless mode (16.2) [CP:1162].

If the EGM was in an 'ECT from EGM' lockup condition, but in a higher priority state (such as a fault condition [CP:1163], or a door open condition [CP:1164], or audit mode [CP:1165]), then upon receipt of this poll, the required action must be held pending (in NV RAM) and carried out immediately upon return to the ECT lockup condition [CP:1166].

Upon receipt of this poll, if no 'ECT from EGM' lockup condition was current, then the ECT Lockup Reset request must be ignored [CP:1167].

This poll must be the only way on the EGM to clear an 'ECT from EGM' lockup condition.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x24, 1 byte hex
CFUNC   1 byte hex

       bit 0   If set, the credit transfer was successful, at the next opportunity the EGM must deduct the amount successfully transferred from the current credit meter, update Total EGM Cents Out [CP:1168] and Total EGM Cashless Out [CP:1169] Meters. Exit the lockup.
                 If not set, the credit transfer from the EGM is denied. *(QPv1.6)* The EGM must pause and display the message "Transfer Denied" for 5 seconds, then exit the lockup, then automatically initiate either a hopper payout [CP:1170], or a cancel credit [CP:1171] or a ticket out [CP:1172] for amount of the failed transfer attempt. (Which type of payout will depend on the current hardware configuration and the thresholds set in the Hopper/Ticket Printer Maintenance Poll (15.4.17)).
                 (On a fail, QPv1.5 EGMs would initiate a CC, even for an amount of 0$) [CP:1173]

       bits 1..7   reserved = 0, mask out these bits when reading any data from this byte [CP:1174].

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

16.5 General Status Response, Addendum

Refer to section 15.6.1. Additional state code:

STATE

0x0C   'ECT from EGM' Lockup Condition [CP:1175]

and

16.5.1.1   FLGB Bit 3 Cashless Mode (16.2) Active/Pending. *(QPv1.6)*

17 Configurable Random Events (CRanE)

~~Pronounced; "C R & E", or alternatively pronounced "crane".~~

17.1 General

It is recommended EGM manufacturers should implement CREs last when implementing QCOM.

CRanE allows the monitoring system to configure an EGM with an event that occurs (i.e. logged) randomly during game play.

17.1.1 Each EGM will provide provision for up to eight CRanE. Each event is configured with a hit probability and optionally may also lockup the EGM upon its occurrence. Each event may also be enabled or disabled at any time by the system (except while the EGM is in a CRanE Lockup).

17.1.2 Once per play, the EGM will evaluate whether a hit has occurred with respect to each enabled event (refer 17.3 RATE field). [CP:1176] The CRanE hit evaluation (and possible resulting hit lockup/s & Hit Event logging) must be performed immediately upon a bet at the beginning of each play on the EGM, after the Stroke and Turnover meters have been updated [CP:1177] but before the play outcome is determined (via the RNG) and before the play-show commences (i.e. the reels start spinning). [CP:1178] This is to ensure that a possible CRanE lockup cannot adversely interfere with the normal game's "play-show" (for example, by freezing the play-show in a random and possibly winning combination). The evaluation must be performed one level at a time, in order from lowest numbered event to highest [CP:1179]. If more than one event is enabled then it is possible that more than one event can be hit per play and therefore it may also be possible to hit all eight events (if all enabled) on a single play on the EGM [CP:1180].

17.1.3 The EGM must be capable of generating 32 bit random numbers to implement CRanE with a cycle of at least four times 2^{32} [(QPv1.6)]. [CP:1181] If the module implementing CRanE uses a separate RNG from the game RNG, then the CRanE RNG must also be continuously cycled, seeded and maintained as per GMNS requirements for RNGs. [CP:1182]

17.1.4 If an enabled event is hit, the EGM must immediately log the CRanE Hit Event (0) [CP:1183] and possibly enter the CRanE lockup state depending if the event is also set to lockup (17.3) [CP:1184]. After the event is logged and the lockup cleared, the EGM will automatically resume evaluating any remaining configured CRanE. [CP:1185] If no other trigger occurs, the EGM then continues the current play as normal [CP:1186].

17.2 The CRanE Lockup Condition

If the EGM is configured to lockup upon a CRanE hit (refer 17.3), then the EGM will lockup [CP:1187] and display the message "VERIFYING PLEASE WAIT…" *(QPv1.6)* [CP:1188] and wait for the CRanE Lockup Acknowledgement Poll (17.5) [CP:1189]. The EGM must not display "call attendant" (or the equivalent) at any stage for this lockup (unless interrupted a fault condition for example). A Lockup title is not required to be displayed; if a title is displayed then it must just be the text "Lockup". *(FYI QPv1.5 EGMs displayed "Call Attendant" and "Verifying Prize".)*

~~Then u~~Upon receipt of the CRanE Lockup Acknowledgement Poll (17.5), the previous verification message ~~is~~ must be removed [CP:1190] and replaced with the text message contained in the acknowledgement poll [CP:1191]. *(QPv1.5 EGMs also displayed the text "CONGRATULATIONS YOU HAVE WON" [CP:1192])* This message then remains on display in the CRanE lockup condition until the EGM is finally reset via the EGM General Reset Poll or key-switch [CP:1193].

Acceptance of the overall legibility of the CRanE text messages on the EGM display is at the discretion of the CEO [CP:1194]. Ideally, the CRanE lockup display should be very similar in appearance and prominence to the System Lockup display (refer section 15.4.9).

The EGM will not allow itself to be reset via the EGM General Reset Poll or key-switch until the CRanE lockup condition is first acknowledged via the CRanE Lockup Acknowledgement Poll (17.5) [CP:1195].

When CRanE lockup is finally terminated via the EGM General Reset Poll, the EGM must continue evaluating remaining enabled CRanE's, or if the are no CRanE's remaining then continue the play as normal [CP:1196].

17.2.1   CRanE Authentication Code

The EGM will also display (discretely but clearly readable) for the duration of the CRanE lockup condition, an authentication code comprised of the following information:

$$(QPv1.6)\ \text{"CRanE: } ID\text{-}HITS\text{-}RATE\text{"} \quad [CP:1197]$$

Where:
- *ID*   is the ID number of the current CRanE (1 digit 0...7)
- *HITS*   is the total number of hits for the CRanE with ID. 4 digits decimal (minimum). Zero padded. Defaults to zero once per RAM clear.
- *RATE*   is the hit rate of the event (zero padded) as configured via the last CRanE Configuration Poll (17.3). 8 digits hex.

The CRanE Authentication Code must also be displayed in the EGMs last play recall display in audit mode for all stored plays [CP:1198]. Because more than one CRanE can be hit per play, it is acceptable to only display the least probable CRanE event that hit per play in last play recall display [CP:1199].

Operational Note: If CRanE is being used for large awards or other high risk applications, then upon a hit/win, an attendant, or authorised person, or trusted 3rd party should manually verify within the CRanE lockup:

- That the authentication code RATE field is as expected for the given ID
- It is a real CRanE lockup (and not a System Lockup faking a CRanE for example). (Suggest crosschecking the Authentication Code in last play recall after lockup clear if the EGM's System Lockup and CRanE lockups are not distinctly different)
- Record and sign off on the authentication code.

CRanE Lockup Display Examples:

Figure 10 CRanE lockup example pre CRanE Ack. Poll.

Screen dump courtesy of IGT (Australia) PTY LTD.
Note the text "CRE Lockup" above is depreciated, refer 17.2.

Figure 11 CRanE Lockup example post CRanE Ack. Poll.

Screen dump courtesy of IGT (Australia) PTY LTD.
Note the text "CRE Lockup" above is depreciated, refer 17.2.

17.3 CRanE Configuration Poll

Function Code = 0x11

This poll configures or re-configures the EGMs CRanE. This poll may be sent at any time by the SC.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x11, 1 byte hex
NUM   1 byte hex
        bits 0..3 Number of CRanE in this message poll 0...8. If this field is greater than 8 then the EGM must ignore the poll message data [CP:1200].
        bits 4..6 Reserved [CP:1201]
        bit 7   If set, the EGM will queue for future response, the CRanE Status Response (17.6). If not set, no action. [CP:1202]
SIZ   Size of each repeated entry. See below. 1 byte hex. All entries are the same size. = 0x05 [CP:1203]

The following data is repeated the number of times as indicated by the NUM field.

LEV   CRanE ID. 1 byte hex.
        bits 0...2 CRanE ID. This identifies the event being configured. (0...7)
        bits 3...4 Reserved [CP:1204]
        bit 5   = 1 This event when hit will lock-up EGM in a CRanE Lockup
                = 0 This event when hit will not lock-up EGM [CP:1205]
            bit 6   Reserved [CP:1206]
        bit 7   = 1 Enable this event
                = 0 Disable this event [CP:1207]
RATE   CRanE Hit Rate. 4 bytes hex
        This describes the probability (for a one cent bet) of triggering the CRanE with the ID number above.
        Specifically, the EGM must give a RATE in 2^32 chance for every cent bet of triggering the event. [CP:1208]
        Accordingly, the probability 'p' (for a one cent bet) of a hit for a play is equal to **RATE/2^32 * bet**
        EGMs must handle the special case where RATE*bet >= 2^32 (overflow). p must be forced to 1 (i.e. 100% hit rate) [CP:1209].
        The EGM must also ensure that a RATE of zero has no side effects (i.e. Divide by zero error) [CP:1210].

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

The EGM must not allow any CREs to be configured or reconfigured via this poll if the EGM is currently locked up in a CRanE lockup condition [CP:1211].

QPv1.6.1d: The EGM's current CRanE configuration must be displayed in EGM audit mode as required re QCOM section 4.1.11. [CP:1212] *(Prior QCOM v161d there existed a requirement stating specifically not to display CRanE information in audit mode. This ~~paragragh~~paragraph and checkpoint has been added to ensure that the information was actually added back into EGM audit mode and not overlooked as a part of a QCOM evaluation)*

17.4 CRanE Hit Event

Event Code = 0x3005 This is an unnumbered advisory event type [CP:1213].

Name: **"EGM CRanE Hit   Lev: <LEV> Rate: <RATE>"** *(QPv1.5 has different descriptor)*
This event is logged for every occurrence of a CRanE Trigger on the EGM.

Extended Event Data: <LEV><RATE>

LEV   CRanE ID or level. 1 byte hex.
        bits 0...2   CRanE ID or level (0...7), display as one character of decimal
        bits 3...4   Reserved = 0
        bit 5   Boolean, display as either 'L' or '-' respectively
                    = 1 This event ID is configured to lock-up EGM
                    = 0 This event ID is not configured lock-up EGM [CP:1214]
        bits 6...7   Reserved = 0
RATE   Hit Rate. 4 bytes hex, display as 8 characters of unsigned hex. Described previously [CP:1215].

17.5 CRanE Lockup Acknowledgement Poll

Function Code = 0x12

The poll type acknowledges an EGM locked up in a CRanE lock-up condition (if the event was configured to lockup). It does not clear the lockup condition, but it signifies the current CRanE has been verified by the system and the EGM is now authorised to clear the lockup via the EGM General Reset Poll or manual Key-Switch combination [CP:1216]

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.2
FC   Function Code = 0x12, 1 byte hex
LEN   Length of TEXT field. 1 byte hex.
        Before processing the EGM must verify LEN <= 80 *(QPv1.5 was only 40 bytes)*, if fail, the poll must be ignored [CP:1217]
TEXT   LEN bytes of ASCII printable characters. Refer Section *2.3.8*. [CP:1218] The EGM must prominently display the TEXT message for the duration of the CRanE lockup [CP:1219]. The TEXT contains arbitrary message.
        The EGM must verify all characters are printable (refer *2.3.8*), on failure, the poll message data and acknowledgement must be ignored [CP:1220]. Acceptance of the overall legibility of the text message on the EGM display is at the discretion of the CEO [CP:1221].
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Upon receipt of this poll, if no CRanE lockup condition existed on the EGM, the poll is ignored [CP:1222].

17.6 CRanE Status Response

Function Code 0x06

This message is queued if requested from a CRanE Configuration Poll (17.3).

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code = 0x06, 1 byte hex
NUM   1 byte hex
       bits 0..3 Number of enabled CRanE's 0...8
       bits 4..7 Reserved
SIZ   Size of each repeated entry. See below. 1 byte hex. All entries are the same size. = 0x05

The following data is repeated the number of times as indicated by the NUM field for every enabled CRanE. Details for CRanE's that are disabled must not be sent.

LEV   CRanE ID or level. 1 byte hex.
       bits 0...2   CRanE ID or level (0...7)
       bits 3...4   Reserved = 0
       bit 5   = 1 This CRanE when hit will lock-up EGM
                         = 0 This CRanE when hit will not lock-up EGM
       bit 6 [CP:1223]   = 1 indicates the EGM is currently locked up with a CRanE of this ID
                         = 0 otherwise
       bit 7   Reserved = 0
RATE   Hit Rate. 4 bytes hex. Described previously.

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

17.7 General Status Response, Addendum

Additional State Code to be included in the EGM General Status Response (15.6.1):

0x0D   CRanE Lockup Condition [CP:1224]

18 QCOM and RUGMs

This section was added in QPv1.6. This section is no longer required reading as there is no plan to add support for this feature to QCOM v1.x. The section remains published to preserve any intellectual property for possible future use.

There will be an extension to the QCOM Protocol that will specify the minimum requirements for remote downloads of files, or 'packages' to EGMs. The files may be program, upgrades, game upgrades, new games or variations, or peripheral upgrades (such as for note acceptors). It will also specify the minimum requirements for the levels of integrity and security.

There will be a mandatory components for EGM manufacturers operating on the QCOM protocol to implement in this area. However due to the size of this feature, its implementation will be a staged approach over time. Accordingly, EGM manufacturers will be granted additional time to allow integration the required new hardware, software and protocols into their EGMs.

For more information refer to the OLGR publication "Principles for Remotely Upgradeable EGMs".

Possible applications of QCOM enabled RUGMs:

- Game downloads.
- Variation downloads.
- EGM program patches/upgrades.
    This feature represents potentially huge saving in the current cost of retrofitting EGM software. (NB: An EGM manufacturer will not be allowed to provide game or variation downloads until EGM program patching and upgrades have been successfully implemented and once the requirements are finalised.)
- Peripheral Software Upgrades.
    E.g. Note Acceptor upgrades to support a new note type or to fix any identified susceptibilities.
    Potentially a huge saving in the existing cost of physically upgrading note acceptor software in the field.

18.1 Stage 1 Requirements

As a part of the first stage QCOM - RUGM implementation, EGM Manufacturers are required to start integrating the following hardware and protocols into their future EGM products at the earliest available opportunity. EGM manufacturers are requested to submit a projected schedule for implementation of stage 1 requirements as soon as possible.

Hardware:

- 1 x UTP Ethernet Port. (The port must have extra fast transient protection to at least 4.5kv). Minimum supported speed is 100 Mbit/sec. (However, for security reasons, the EGM's Ethernet port must remain disabled in all production EGMs, until such time as EGM program patching and upgrades have also been successfully implemented once the requirements have been finalised)
- The EGM must utilise re-writable PSDs for all its programs and software. Note, the EGM must also contain enough free space on its PSDs to store its current software set plus a newer version during a download. The PSDs must also still be able to be easily removed from the EGM for isolated manual verification of the current contents when required for when higher security is mandated.
- The EGM must implement the OLGR's 'Electronic Seal Minimum Requirements' hardware requirements.
- Note, any downloads must be performed in background by the EGM while it maintains all other operations, including game-play.

Protocols:

- Ethernet, TCP/IP & UDP

  The above list of protocols is not complete. The actual protocol to be used for the file transfer is at the discretion of the EGM manufacturer but must run on top of TCP/IP on a re-assignable range of ports.

  All protocol drivers utilised must be demonstrated to be immune from all known malformed message attacks.

  (The whole of QCOM will eventually move to these protocols but not before QCOM RUGM support is finalised and up and running)

Software:

- An encryption library. (The exact algorithms to be supported are still to be finalised, however RSA & SHA-1 such as "RSASS PKCS #1 v1.5 with SHA-1" are strong candidates).
- A compression library.

18.2 Other requirements under consideration for later stages

The following section outlines possible future requirements for later stages for the purposes of discussion and feedback.

18.2.1   Compression.

All downloads must be compressed.

Software upgrades greater than 10 Mbytes must be implemented as differential patches (i.e. patch files that modify the existing code) when this results in a smaller patch file.

18.2.2   Security.

Downloaded file authentication facilitated by Digital Signatures that the EGM will use to authenticate downloaded files before activation, will be a mandatory requirement on all EGM software type downloads. A standardised method of authentication will be mandated by the OLGR. TBA.

Individual files may be encrypted at the discretion of the EGM manufacturer. The encryption algorithm is also at the discretion of the EGM manufacturer, but any encryption utilised must be a published and publicly available algorithm with a demonstrated track record. (However the file transfer itself may yet end up being encrypted if for example SSL is used. TBA)

18.2.3   Package and Version Control.

Downloaded files must contain the necessary information and instructions indicating to the EGM what the file is and what it must do with the file once activated.

For patch files that must be applied to a specific version of software, if the wrong version patch is downloaded for the wrong version EGM software, the patch must automatically not take if activated. I.e. the patch must contain embedded information as to what version software it must be applied to. This is a requirement for all file types and is the manufacturer's responsibility to implement a reliable method.

The EGM must not allow its software to be downgraded to a previous version. I.e. The EGM must not accept any download with an older version number than it is currently using.

18.2.4   File type specific requirements.

The format of the files is at the discretion of the EGM manufacturer, but must meet certain criteria depending of the file type:

- Game software or patch.

  File/package authentication requirements. These types of files must contain or have attached at least two Digital Signatures (DS) generated by an acceptable public key encryption algorithm over the file contents. One DS is generated by the EGM manufacturer the other by the Regulatory Authority. Just prior file activation, if either DS does not validate, then the file is must not be activated by the EGM. The file must authenticate before any other data within the file is used by the EGM. Only if both signatures authenticate, may the EGM proceed and activate the file.

  Game software files must also have system of preventing previously approved but no longer acceptable software (aged software) from being re-used. For example an expiry date, or serial number, or an incrementing version number, embedded in the authenticated portion. If the file version is too old relative to its current software, then the file is automatically deleted by the EGM upon any activation attempt. If it was required to revert back to an expired version of software, then this can be done as a new upgrade with a new version number.

- Peripheral software (e.g. software for either a note acceptor, coin acceptor, I/O controller, etc.)

  These files may contain a manufacturer password at the discretion of the EGM manufacturer, otherwise no other security or Digital Signature is mandatory.

- Support variable length, "do nothing when activated" arbitrary files, for development and testing purposes and space filler files, must be supported by all EGMs implementing QCOM RUGM support. Space filler files are non-algorithmic compressed white noise (random data) from a trusted source that may be used on occasion to ensure a device, such as an EGM, has erased all available free space by getting the EGM to return a seeded hash value of the file once it has been sent to the EGM.

18.3 Areas to be finalised.

- Network topology.
- Minimum requirements for Encryption & other algorithms.

At this stage RSA encryption and SHA1 are strong candidates (both are patent free and source is freely available).

- Encryption/Authentication key management protocols
- File and package management protocols.
- Encryption key management.

Other possible requirements and software (this list is not finalised):

- EGM may be required to run an FTP client. TBA. All downloads must be resumable if interrupted.
- A Telnet server may be required. TBA.
- Once a download is complete, the file is stored by the EGM until it is authenticated and then either activated, overwritten or deleted.
- All downloads must be digitally signed, compressed and packaged. Details TBA.
- An encryption API. Requirements TBA.

19 EGM to EGM Communications

This section was added in QPv1.6. ***This section is no longer required reading as there is no plan to add support for this feature to QCOM v1.x. The section remains published to preserve any intellectual property for possible future use.***

This section represents an **optional** feature for use by EGMs manufacturers. Do not implement without first checking with the OLGR.

This feature allows EGMs of the same manufacturer to communicate with each other by using the SC to relay messages. This eliminates the need for the EGM manufacturer to run a separate physical network when wishing to implement EGM to EGM communications. Possible applications of this feature are but not limited to; multi-terminal games and the synchronisation of events between EGMs such as idle mode attract animations.

Because of the current network topology, only the SC is physically in a position to send messages to all EGMs networked to it, it makes sense for the EGM to utilise the SC to pass on any desired message data. However, the data echoed by the SC on behalf of the EGMs is an unconfirmed broadcast; if the EGM wishes to send a confirmed packet then it must implement its own method where the EGMs send an acknowledgement in response.

Bandwidth is also very limited; each message sent for broadcast by the EGM may be no longer than a specified length (see below) and once the EGM has a send a message it may not send another message until after the next EGM broadcast for its manufacturer. Refer to the EGM Broadcast Message below for more information.

One condition of use of this feature is that all communication methods used to communicate between EGMs using this feature are first approved by the OLGR.

The first EGM manufacturer interested in using this feature must give the OLGR 6 months notice in advance to allow requirements to be finalised.

19.1 EGM Broadcast Request Response

An EGM may queue this message in response to a poll to its address (with respect to higher priority responses) whenever it wants the SC to broadcast a message to all other EGMs on its behalf. The SC will broadcast the requested message at the next opportunity (refer EGM Broadcast Message below).

A SC will ignore any further broadcast requests from an EGM until it has broadcast the EGM's last broadcast request, unless the high priority bit is set, in which case the SC will queue the broadcast for the EGM on that poll cycle with the high priority message within, overwriting any previously queued message from that EGM.

An EGM can confirm transmission of its request broadcast by monitoring subsequent EGM Broadcast Messages.

Message Format and Order:

DLL   Data Link Layer. Refer Section 14.3
FC   Function Code = 0x05. 1 byte hex.
FLG   Reserved. 2 bytes hex.
        Bits 0…6 Reserved.
        Bit 7   High Priority bit.
             If set the SC will send out the EGM broadcast for the EGM on the current poll cycle. EGMs may only set this bit under special circumstances as approved, such as a jackpot or feature win.
LEN   2 bytes hex. Length of Data field below.
        The length must be <= 16 bytes.

DATA   LEN bytes hex.
        This is the data that the SC will broadcast on behalf of the EGM.
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

19.2 EGM Broadcast Message

The SC will send out EGM Broadcast Messages to all EGMs under it on a periodic basis when messages have been queued by one or more EGMs via the EGM Broadcast Request Response. The broadcast is separate from the general data and time broadcast and is sent out on a special reserved poll address for each manufacturer using this feature (TBA). EGM broadcast messages will be sent out by the SC no more than once every four poll cycles per EGM manufacturer. During a poll cycle, the SC will send out EGM broadcast messages after polling each EGM in turn as normal, just before the Current Data and Time Broadcast is sent. If the SC has not sent out an EGM broadcast for a particular manufacturer for more than four poll cycles and a new EGM broadcast request is received, then the SC must send out the corresponding EGM Broadcast at the end of the current poll cycle.

An EGM using this feature must be capable of processing 3 messages or better per second from a SC.

Message Format and Order:

DLL   Data Link Layer. Refer Section 14.2
FC   Function Code = 0xFD. 1 byte hex.
NUM   Number of EGM messages in this broadcast. 1 byte hex.
        Must not be zero.
        As SC may receive more than one broadcast request from a number of EGMs per broadcast, they are combined into a single broadcast. If more EGM broadcast requests are received that can be fit into a single broadcast (limited by the maximum message length, see 14.2) then the SC ignores further broadcast requests from EGMs until after the current broadcast has been sent. It is the EGM's responsibility to ensure that they do not flood the SC with too many EGM broadcast requests at once.

The following message data is repeated NUM times.

SER   Serial number of EGM. 3 bytes BCD.
MID   EGM Manufacturer I.D., 1 byte BCD. Refer section 15.1.7
LEN   Length of DATA field below. 2 bytes hex.
        Before processing each repeated entry, the EGM must ensure that it will not process past the end of the message CRC field in the event of a corrupt NUM or LEN field. [CP:1225]
DATA   LEN bytes.
        The DATA field from the last received Broadcast Request Poll for the EGM denoted by SER&MID.

End of repeated entry data.

CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

19.3 EGM Broadcast Timeout Event

"EGM Broadcast Timeout"

Event Code: TBA
Extended Event Data: None
Description: This event logged by the EGM when it detects that a requested EGM broadcast was not send by the SC. The EGM must allow at least 4 poll cycles for the any requested broadcast to be sent out.

20   QCOM Command Prompt

This section was added in QPv1.6.

Implementation of QCOM Command Prompt is **optional** for EGM manufacturers. Do not implement without first checking with the OLGR. *** This section is no longer required reading as there is no plan to add support for this feature to QCOM v1.x. The section remains published to preserve any intellectual property for possible future use.***

QCOM Command Prompt is a method of remotely interfacing with an EGM via a simple text interface to perform custom operations not already covered by QCOM or the EGMs audit/test mode.

This feature has a number of possible applications including:

- Possible support of down loadable EGM software.
- A convenient text interface for built in custom debugging or configuration tools.
- Access to custom reports generated by the EGM. Such as:
    - Line/Credit bet per line frequency reports.
    - Combination hit reports.
    - Symbol frequency reports.
- Remote diagnostics. (E.g. CPU/System temperatures, free memory, free NV memory, etc)
- Remote access to EGM audit mode.
- Remote access to last play recall information

In production EGM software, services offered via QCOM Command Prompt must not be able to affect the security and integrity of the EGM and its games. However, some services that could affect security and integrity may be allowed remain in production software if access to the EGM's processor door is required in order to enable the service. Final approval of what services may remain in production software is at the discretion of the Regulatory Authority.

Examples of restricted services in EGM production software:

- General Operating System write-access.
- Operating System read-access when that access could be used to compromise the security of the EGM.
- Peek/Poke operations.

20.1 Command Prompt Poll.

This poll is used to send text commands to the EGM. The EGM may respond, when results are ready with one or more Command Prompt Responses.

Message Format and Order:

DLL	Data Link Layer. Refer Section 14.2
FC	Function Code = 0x02, 1 byte hex.
FLG	Reserved. 1 byte hex.
LEN	1 byte hex. Length in bytes of CTEXT field below.
	Before processing CTEXT, the EGM must verify LEN < 200 and LEN is not equal to 0.

CP:1226

CTEXT	A text string of LEN bytes consisting of ASCII printable characters only (*2.3.8*).
	EGM must verify all characters are printable (refer *2.3.8*).
CRC	Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

20.2 Command Prompt Response.

This poll is sent by the EGM in response to the Command Prompt Poll. The data in the response depends on the data in the Command Prompt Poll.

Message Format and Order:

DLL   Data Link Layer. Refer Section 14.3
FC   Function Code = 0x02, 1 byte hex.
FLG   1 byte hex.
        Bits 0..6 Reserved = 0
        Bit 7 Last flag.
                If set, this indicates that the EGM has no more EGM Command Prompt Responses to
                send after this one.
                If not set, this indicates that more Command Prompt Responses are pending after
                this one.
LEN   1 byte hex. Length in bytes of RTEXT field below.
        LEN must be less than 200 and not equal to zero.
RTEXT
        A text string of LEN bytes. The string consists primarily of printable characters (*2.3.8*) but
        may also contain carriage returns (0x0D), the escape character (0x1B) and tab characters
        (0x09). (Line feeds (0x0A) will be appended automatically to all carriage returns). A special
        repeat character code is available (TBA). TAB size is 8 characters. Use of TABs is
        recommended to reduce the number of characters transmitted.

        Only the carriage return denotes a new line, multiple Command Prompt Responses are
        concatenated together.
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

21   Finding EGMs on the LAN (Seek EGM)

This section/feature was added in QPv1.6.

Typically in QCOM, a SC requires prior knowledge of the EGM's MID & Serial number in order to establish communications with it (see Poll Address Configuration Broadcast Message 15.5.4). This feature enables a SC with the additional ability to find any number of unknown EGMs sitting on its LAN, without any prior knowledge or information about the EGMs, such as their manufacturer IDs or serial numbers.

In monitoring systems that already authenticate serial numbers; this feature will enable detection of incorrectly entered serial numbers in EGMs.

While implementation of this feature is mandatory for EGM manufacturers, this feature is not required to be implemented by SCs in order to establish communication with an EGM if its serial number is already known.

It is recommended EGM manufacturers implement this feature only towards the end of the QCOM implementation and only after the successful implementation and testing of the standard QCOM poll address configuration methodology (15.5.4), also refer to the implementation note at the end of this section.

21.1 Methodology and Requirements

A SC may send a new broadcast message (called the "Seek EGM Broadcast Poll") to a special new reserved poll address called the 'seek EGM poll address'. (The seek EGM poll address value is defined in section 21.2.)

EGMs must only respond to the Seek EGM Broadcast Poll during a communications timeout (6.1.2) [CP:1227] with only Seek EGM Broadcast Response types (refer 21.3) and with respect to the following:

> Upon initial entry into a communications timeout or after a power up, an EGM must respond to the **very next** Seek EGM Broadcast Poll received on the seek EGM poll address. [CP:1228] Then after this initial response, the EGM must only subsequently respond to the Seek EGM Broadcast Poll, at random, once every 32 received poll cycles on average, for the duration of the Communications Timeout (6.1.2). [CP:1229]

Polls to the seek EGM poll address must not bring an EGM out of a communications timeout. [CP:1230]

Acknowledgements are not applicable to polls/responses on the seek EGM poll address. [CP:1231]

When responding to the seek EGM poll address, the EGM will always respond with the same message, the message simply contains the EGM's current MID & SERIAL number (refer 21.3).

Obviously collisions are expected and are handled by the random retry. A worse case scenario with 32 EGMs on a LAN and all EGMs in a Communications Timeout, it takes the Site Controller on average ~100 secs to retrieve all the EGMs serial numbers.

It is not the intent for this feature to be used every time an EGM stops responding. Once a SC finds an EGM on a specific serial number, it must continue to try to use that serial number until directed otherwise by the host system. Otherwise this could be a security risk.

21.2 Seek EGM Broadcast Poll

This poll is only sent by the SC to the seek EGM poll address. The value of the seek EGM poll address is 0xFC (252).

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
FC   Function Code, 1 byte hex = 0xFC. ^{CP:1232}
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

21.3 Seek EGM Broadcast Response

This is the only message the EGMs may use to respond to the Seek EGM Broadcast Poll. [CP:1233] The data in this message must never change unless the EGM is RAM cleared and a new serial number is assigned to it.

Message Format and Order:

DLL   Data Link Layer, Refer Section 14.3
       The poll address the EGM must use for this response type is the seek EGM poll address (refer previous section for it value). Bit 7 of the CNTL field is not applicable in this response, but bits 5&6 of the CNTL field must be set accordingly.
FC   Function Code = 0x20, 1 byte hex
SER   EGM Serial number (15.1.8), 3 bytes BCD, LSB first
       This field when combined with MID below, yields a 4 byte unique ID number for each EGM.
MID   EGM Manufacturer I.D., 1 byte BCD. Refer section 15.1.7
CRC   Data Link Layer - Cyclic Redundancy Check. 2 bytes hex. Refer section 14.2.

Implementation note for EGM manufacturers:

It is ~~strongly~~ recommended that the EGM manufacturer implement the Seek Broadcast Poll/Response at the lowest possible level (e.g. under interrupt) and completely separated from the rest of the QCOM code. Trying to implement this response in the EGM as a part of the normal QCOM response methodology (refer 14.1) is actually very difficult and will probably lead to many unwanted side effects. To minimise this risk, treat section 21 in the EGM software like a separate, standalone protocol within QCOM.

22 Ticket In / Ticket Out (TITO)

A Cash Ticket In/Out solution is supported by QCOM (since QPv1.6). Please note that the TITO solution proposed in this document is just one possible solution of several (e.g. refer 'NB' in paragraph below).

Implementation of QCOM TITO is not mandatory. However as of July 2012 the OLGR is now accepting TITO system submissions for evaluation.

The TITO solution described in this section is applicable to a TITO system in which the EGM is interfacing directly to the TITO devices. I.e. the EGM owns the TITO devices. (NB: it has always been possible in QCOM to add a TITO system on top of QCOM EGMs where the TITO devices interface directly to the TITO system (not the EGMs) and the TITO system adds and removes credit via the QCOM ECT in/out working in cooperation with the monitoring system or be a part of it.)

The methods provided here may be utilised as only a Ticket-Out (TO) system, or both a Ticket-In and Ticket-Out (TITO) system. Ticket In may be facilitated by the EGM by either a dedicated ticket in device or a banknote acceptor that can read tickets; QCOM neither differentiates nor cares either way.

Quick index to relevant TITO sections of QCOM:

"General Status Response" - TI/TO states (15.6.1)
"EGM Configuration Response" - TI/TO device present indicators (15.6.12)

"Ticket Out Request" unnumbered event (7.10.4.11)
"Cash Ticket Out Request Acknowledgement Poll" (15.4.10)
"EGM Cash Ticket Out Print Successful" unnumbered event (7.10.4.12)
"EGM Cash Ticket Out Print Failure" fault event (7.10.1.32)
"Cash Ticket Out Meter" to group 0 meters (12.1.1)

"Ticket In Request" unnumbered event (7.10.4.13)
"Cash Ticket In Request Acknowledgement Poll" (15.4.11)
"Cash Ticket In Meter" to group 0 meters (12.1.1)
"Ticket-In Timeout" event (7.10.3.42)
"Ticket-In Aborted" event (7.10.3.43)

"Invalid Ticket Out Acknowledgement" Event (7.10.3.36)
"EGM Ticket Printer Paper Out" Event (7.10.1.29)
"EGM Ticket Printer Paper Jam" Event (7.10.1.30)
"EGM Ticket Printer General Fault" Event (7.10.1.31)
"EGM Ticket Printer Ink Low" Event (7.10.3.17)
"EGM Ticket Printer Paper Low" Event (7.10.3.20)

Meters - TITO related (12.1)

"TICKET" field.   Refer Hopper/Ticket Printer Maintenance Poll (15.4.17)
Ticket print test.   Refer Hopper/Ticket Printer Maintenance Poll (15.4.17)

22.1 Ticket Out Process

If the EGM has a ticket out printer, a cash ticket out may be instigated when a player presses collect on the EGM (provided the EGM has credit, no fault conditions, is in idle mode and all doors are closed), or for a win payout (15.4.5). Whether a Ticket Out is instigated or not is with respect to the thresholds set via the Hopper/Ticket Printer Maintenance Poll (15.4.17). Also refer to section 3.5.
CP:1234

If a cash ticket out is instigated, the EGM must:

- log the "Ticket Out Request" unnumbered event (7.10.4.11),
- enter a Cash Ticket Out Lockup condition (this sets the relevant TO lockup STATE in the General Status Response (15.6.1),
- display the message "**Printing Cash Out Ticket - Please Wait**" and

Receipt of the above event by the system will instigate an approval or otherwise for the transaction via the Ticket Out Request Acknowledgement Poll" (15.4.10). [CP:1235]

The EGM must remain in this lockup condition until it receives the "Cash Ticket Out Request Acknowledgement Poll" (15.4.10) which either approves or denies the ticket out request. [CP:1236]

Refer to 15.4.10 for the remainder of this procedure.

All forms of physical credit input on the EGM (e.g. coins/notes/TI) must be disabled for the duration of the Cash Ticket Out Lockup condition. [CP:1237]

The EGM must not display "call attendant" or the equivalent during the Cash Ticket Out Lockup as this condition is automatically cleared by the SC. [CP:1238]

~~The EGM must display details of the last successful Cash Ticket Out while in idle mode for no more than 10 seconds, or until the start of the next play or the next ticket in/out event, whichever is sooner. Eg. "Ticket Out $xxx.xx".~~ [CP:1239]

Unless actually printing a ticket which is handled in the next section, the EGM must recover and resume from where it left off after any interruption to the ticket out process. [CP:1240]

No interruption to the TO process at any stage on the EGM may cause a financial loss to any party (EGM, TITO system or player) or EGM RAM error (self audit error). [CP:1241]

22.1.1   Difficulties in Printing a Cash Out Ticket.

If a ticket print is detected as incomplete[1] by the EGM (e.g. an EGM ~~reset~~ restart / power fail), or fails in general (e.g. a fault with the ticket printer such as a paper feed jam), the EGM must log any applicable events including the "EGM Cash Ticket Out Print Failure" fault event (7.10.1.32) and enter a fault condition. [CP:1242] *In this case, the EGM's meters will have already been updated as section 15.4.10 requires that the ticket print is performed last.* [CP:1243]

For any fault condition on the EGM concerning the ticket printer that occurs during a ticket print that prevents successful print[1] (e.g. events 7.10.1.30 and possibly 7.10.1.31), because the EGM's credit meter will be already updated by this time, the EGM must also display the ticket's serial number and amount during the fault condition display for easy reference. [CP:1244]

After a fault condition in the above scenario is cleared, the EGM must not resume or retry the ticket print. [CP:1245]

If any other fault (i.e. a fault other than a fault with the ticket printer) or door open condition occurs during a ticket print, the EGM must endeavour to complete the print in progress without delay.

It should be noted that QCOM operates a little differently with respect to when TO meters are updated when compared to other TITO systems. In QCOM TITO, TO meters are updated the instant the EGM receives the TO Request Ack Poll (refer 15.4.10). In other TITO systems, the EGM waits until after it "believes" a ticket has been "successfully" printed before updating meters. QCOM's approach is intentional and must be implemented. QCOM's approach requires less work to implement in a TITO system (as there is no 3rd handshake stage where the TITO system as to wait for confirmation that the EGM has successfully printed the ticket – which may be a false positive anyway) and it lowers the possibility of a double payment, because in QCOM, once a ticket print starts it is a constant that a ticket record exists in the TITO host and is only redeemable from there.

---

[1] ~~Once the AUTHNO is printed on the ticket in any format, then it may be considered a full and successful print.~~ Refer section 22.1.3 for the definition of a successful cash ticket print.

22.1.2   Cash Ticket Out Log

The EGM must maintain a dedicated Cash Ticket Out log accessible via audit mode containing at least the last 35 attempted ticket-out transactions containing all the information from the Cash Ticket Out Request event and corresponding "Cash Ticket Out Request Acknowledgement Poll" (15.4.10). [CP:1246] The Cash Ticket Out log display must **exclude** any display of the barcode numbers / AUTHNO fields [CP:1247]. The inclusion of CLEN & CTEXT fields in the log is optional (*an option here is to just display CTEXT for a single entry under a user controlled cursor, but not at all is fine*).

A new entry must be added to the log upon the logging of each "Ticket Out Request" unnumbered event (7.10.4.11) [CP:1248]. The log must indicate for each transaction whether the ticket-out is awaiting approval ("sysWait"), denied/failed by the system ("denied"), approved & printing ("printing"), approved & successfully printed[1] ("success"), or approved & print failure ("printFail"). [CP:1249] Acceptance of the overall legibility of the log on the EGM display is at the discretion of the CEO [CP:1250].

22.1.3   Definition of a Successful Cash Ticket Print

In QCOM version QPv166 a successful cash ticket print was defined as:

"Once the AUTHNO is printed on the ticket in any format, then it may be considered a full and successful print"

(Note that as AUTHNO is the first item printed on a ticket, a successful print occurs almost immediately in time after the print commences. So if the EGM cant go on and print the barcode, it doesn't matter, it's still logged as a successful print.)

However in reality, this definition of a successful cash ticket print is not readily detectable and therefore realistically able to be implemented by EGM / ticket printer hardware. So to avoid confusion (particularly with TITO system developers and operators), the definition of a successful cash ticket print is changed[QPV1.6.7] to be more realistic, see below. This change in definition does not require a change to existing EGM implementations of QCOM TITO as the two definitions are complementary. Regardless, any EGM that can actually achieve the old definition may continue to use it, it makes no difference.

A successful cash ticket print is defined as [(QPv167)]:

> **Once the EGM believes the ticket print has commenced to the extent of its capabilities in hardware, then the cash ticket print is assumed to be a success for the purpose of logging the "EGM Cash Ticket Out Print Successful" event (7.10.4.12).**

TITO system developers should note:

- It is possible to get a ticket jam event or other ticker print error event even after a ticket print success event is logged.
- When paper out errors do occur, they will always occur immediately after a ticket print. With ticket printers the last printed ticket prior a paper out will be perfectly readable (unless some other issue occurred).
- There will always exist a range of possible print failures the EGM / ticket printer hardware cannot ever detect or "see". So getting a success event from the EGM, but an unreadable ticket output will occur on occasion. These instances can only be dealt with manually.

---

[1] Once the AUTHNO is printed on the ticket in any format, then it may be considered a full and successful print.

22.2 Ticket In Process

If the EGM has a ticket in device (such as a banknote acceptor) then a cash ticket in may be inserted at any time while the EGM is enabled to accept credit as per banknote insertion requirements.

If a cash ticket in with a readable barcode* is inserted into the EGM at any time while the EGM is enabled to accept credit, in the following order CP:1251 the EGM must:

- log the "Ticket In Request" unnumbered event (7.10.4.13) to instigate authorisation for the amount on the ticket for the correct barcode number as represented by the barcode on the ticket, and
- set the appropriate TI flag in the "General Status Response" (15.6.1) and

*The EGM must be able to read barcodes of 16 and 18 digits (minimum) CP:1252. All barcode numbers readable by the EGM must all be displayed correctly in the EGM audit mode where required to be displayed (e.g. Ticket In Log). CP:1253

*Support for reading barcode formats other than Interleave 2 of 5 is at the discretion of the EGM manufacturer. However as QCOM (i.e. AUTHNO) does not support anything other than numeric barcodes, tickets not comprised of fully numeric based barcodes must be rejected by the EGM.

If the EGM is unable to read the barcode on the ticket or the EGM is not currently enabled to receive credit, then the ticket must be ejected back to the player. CP:1254 (Also referRelated: EGM Excessive Note/Ticket Acceptor Rejects Fault (7.10.1.23)).

The EGM must reject all other ticket insertions until the current ticket has been either approved or rejected. CP:1255

Once the system receives the "Ticket In Request" unnumbered event, it will approve or reject the request based onvia the "Cash Ticket In Request Acknowledgement Poll" (15.4.11).

Refer to 15.4.11 for the procedure relating to the processing of the above poll.

The EGM must display details of the last approved Cash Ticket In while in idle mode until the start of the next play or the next ticket in/out event, whichever is sooner. Eg. "Ticket In $xxx.xx". CP:1256

If the ticket in request is rejected by the system, then the EGM must eject the physical ticket back to the player and concurrently** and prominently display the message "**TICKET REJECTED – XXXX - SEE CASHIER**" where "XXXX" is the text representation of the reason for the failure as specified by the FCODE field in the "Cash Ticket In Request Acknowledgement Poll" (15.4.11). Refer table below for the text. The message must be displayed during play and idle modes for no more than 5 seconds or until the start of the next play or the next attempted banknote/ticket in/out operation;event, whichever is sooner CP:1257 and must be legible. Acceptance of the overall legibility of the text message on the EGM display is at the discretion of the CEO CP:1258.

(**i.e. while continuing the current task)

FCODE (Refer 15.4.11) CP:1259	Reason
0x01	"Ticket System Unavailable"
0x02	"Ticket Expired"
0x03	"Ticket Amount Too Large"
0x04	"Ticket Invalid"
0x05	"Ticket Not Found"
0x06	"Ticket Already Redeemed"
0xXX	*EGM must display the raw FCODE hex value (e.g. if FCODE = 0xAA the text to display is "0xAA")*

22.2.1   TI Acknowledgement Time-out

When the EGM has a Cash Ticket in escrow and is waiting for a "Cash Ticket In Request Acknowledgement Poll" (15.4.11); if the EGM does not receive this poll with a FCODE = Success (and a valid AUTHNO), or a FCODE = Fail code, within **10 seconds** of logging the original "Ticket In Request" unnumbered event (7.10.4.13), then the EGM must abort the TI process [CP:1260] by:

- resetting the appropriate TI flag in the "General Status Response" (15.6.1),
- log the "Ticket-In Timeout" event (7.10.3.42),
- eject the ticket back to the player
[CP:1261]

The EGM is not required to display any error message upon a TI time-out as the ticket rejection itself is considered sufficient user feedback in this case.

Following a time-out, all subsequent "Cash Ticket In Request Acknowledgement Poll" (15.4.11) poll message data received by the EGM must be ignored until the next ticket insertion. [CP:1262]

22.2.2   Interruptions to the TI Process

Interruptions refer to EGM ~~resets~~restarts, power fails, door opens, credit-in disable, or state change including audit mode access and fault conditions.

The EGM must behave to the same requirements as per banknote acceptance in regards to interruptions to the TI process and may abort the process as it sees fit and according to those requirements.

No interruption to the TI process at any stage on the EGM may cause a financial loss to any party (EGM, TITO system or player) or result in an EGM RAM error (e.g. an EGM self audit error). [CP:1263]

If the EGM aborts a Ticket In Process for any reason the EGM must:

- reset the appropriate TI flag in the "General Status Response" (15.6.1),
- log the "Ticket-In Aborted" event (7.10.3.43)*
- eject the ticket back to the player
[CP:1264]

(*excluding "fail" acknowledgements re section 15.4.11 and timeouts re section 22.2.1)

Except for FCODE aborts (15.4.11), the EGM is not required to display any error message upon a TI abort as the ticket rejection itself is considered sufficient user feedback in this case. (For FCODE aborts refer to the previous section for required display messaging.)

Following an abort, all subsequent "Cash Ticket In Request Acknowledgement Poll" (15.4.11) poll message data received by the EGM must be ignored until the next ticket insertion. [CP:1265]

Ticket Retention / Stacking Issues

In the event that the EGM cannot stack a ticket (stack jam, full, or general error), then as per banknote acceptance, the following GMNS version 10.1 requirements must apply to the TI process:

"A gaming machine must not register credits as the result of banknote input until the banknote has passed the point where it is possible to be rejected by the acceptor or be withdrawn."

It should be noted (system developers) this means it is possible for an EGM to receive and acknowledge receipt of the "Cash Ticket In Request Acknowledgement Poll" (15.4.11) but for the ticket still not to be credited. I.e. the meters will never update.

22.2.3   Cash Ticket In Log

The EGM must maintain a dedicated Cash Ticket In log accessible via audit mode containing at least the last 35 attempted ticket-in transactions containing all the information from the "Cash Ticket In Request Acknowledgement Poll" (15.4.11) and corresponding Cash Ticket In Request event (7.10.4.13). I.e. Date, Time and AUTHNO (display AUTHNO as per the current ticket barcode numeric format e.g. "00-2224-2867-5154-7568") CP:1266

A new entry must be added to the log upon the logging of each "Ticket In Request" unnumbered event (7.10.4.13) CP:1267. The log must indicate for each transaction whether the ticket-in request is/was awaiting approval ("sysWait"), denied by the system ("denied") i.e. FCODE non-zero (also include an abbreviated display of the FCODE decoded text descriptor if non-zero along with this (see 22.2 above), or approved ("success"), or timed-out ("timeout"), or retention/stacking error ("stackErr"), or "tooLarge" (re any EGM implemented credit meter limit),or aborted for any other reason ("aborted"). CP:1268 Acceptance of the overall legibility of the log on the EGM display is at the discretion of the CEO CP:1269.

22.3 Ticket Details / Layout

This section details the required cashout ticket print layout for QCOM TITO [QPv165] :
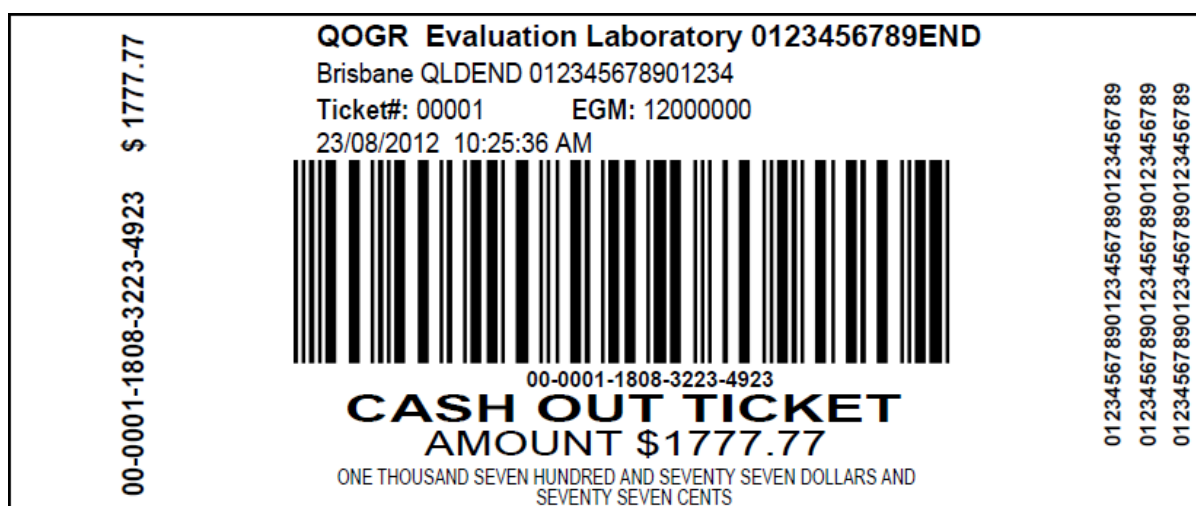
Where:

- The **barcode** must be centred in the ticket and must have a minimum length of 8~~6~~5mm (wrt an **18 digit US standard Interleave 2 of 5 ITF barcode)** and a minimum ~~width~~ height of 26mm. *NB (FYI only) it has been reported that some banknote validators request 15mm of whitespace at either end of the barcode in order to maximise the acceptance ratio of tickets.*
- The overall dimension of the ticket must be 156 ±1mm by 65 ± 1mm.

- **Venue Name.** The top horizontal line of text shown in the above example must be substituted with STEXT field data as received from the last Site Details Broadcast (15.5.5). [CP:1270] The EGM must be able to fully display a text message of ~~maximumn~~maximum length comprised of only the widest proportion character (e.g. "WWW…") for the given font used. [CP:1271]

- **Address Line.** The next line down shown in the above example must be substituted with LTEXT field data as received from the last Site Details Broadcast (15.5.5). [CP:1272] The EGM must be able to fully display a text message of ~~maximumn~~maximum length comprised of only the widest proportion character (e.g. "WWW…") for the given font used. [CP:1273] It is acceptable for the EGM to break the address line over two lines in order to fit it.
- **"Ticket#".** This field is an EGM assigned ticket serial number from the "Ticket Out Request" unnumbered event (7.10.4.11) 'TSER' field. [CP:1274] 1…65535
- **"EGM:"** Refer section 15.6.12 MID and SER fields. 8 digits in the form *mmssssss* where *mm* = QCOM Manufacturer ID (Refer 15.1.7) and *ssssss* = QCOM Serial Number (Refer 15.1.8). [CP:1275]
- The next line is the Date and Time as received from equivalent fields in the applicable "Cash Ticket Out Request Acknowledgement Poll" refer (15.4.10). 24 hour "**dd/mm/yyyy hh:mm:ss**" or 12 hour formats are both acceptable so long as so long as am/pm is indicated when using the 12 hour format. [CP:1276]
- The barcode must be an **18 digit US standard Interleave 2 of 5 ITF barcode**. [CP:1277]
- The number encoded by the barcode must be taken from the first 18 least significant decimal digits of the Cash Ticket Out Request Acknowledgement Poll* (15.4.10) **AUTHNO** field. E.g. ~~in the above example~~ 00-2224-2867-5154-7568 represents an AUTHNO field value of 0x00000000000000000007E6FA1DB460B0 (shown left here as an MSB first number; in the actual QCOM message packet it is tx'd LSB first) [CP:1278] The maximum possible value able to be represented by the barcode is 99-9999-9999-9999-9999 which equates to an AUTHNO field value of 0x0000000000000000DE0B6B3A763FFFF [CP:1279] *I.e the barcode ~~–~~-number-to-print ~~=~~must equal AUTHNO mod (10^18) [CP:1280]

- The barcode number must appear in the two places shown appear roughly in the size and location shown in the above example image. CP:1281
- The words "CASH OUT TICKET" must appear roughly in the size and location shown in the above example image. CP:1282
- The words "CASH AMOUNT" or just "AMOUNT" followed by the amount in $.c and in words must appear must appear roughly in the size and location shown in the above example image. CP:1283
- Another copy of the amount in $.c must appear to the RHS of the barcode that is written vertically on the LHS of the ticket image example above. CP:1284
- The Certification String text (CTEXT) from the applicable "Cash Ticket Out Request Acknowledgement Poll" (15.4.10) must appear as shown in the above image example vertically written on the RHS. CP:1285 A line break must be inserted by the EGM upon printing every 30 characters. CP:1286 The EGM must be able to fully display a text message of maximumn maximum length comprised of only the widest proportion character (e.g. "WWW…") for the given font used. CP:1287
- The back of the ticket must have the OLGR prescribed gambling helpline information text. Contact the OLGR for details. It is noted and understood that the information printed on the back of a ticket is **pre-printed** by the ticket paper supplier. CP:1288
- The only other **pre-printed** text (not shown in the above image) is the "INSERT THIS SIDE UP" message.

Acceptance of the overall legibility and layout of a Cash Out Ticket printed by an EGM is at the discretion of the CEO CP:1289.

FYI an online barcode generators (one of many) may be found at:

http://www.bcgen.com/linear-barcode-creator.html http://www.idautomation.com/java/linearservlet.html

22.4 Other

The state of the current Ticket-in or Ticket out operation on the EGM must be stored in EGM critical NV memory and by fully fault tolerant against power and other interruptions. CP:1290

If the EGM hosts a ticket printer then the firmware version of the ticket printer must be displayed in EGM audit mode in a suitable location provided the ticket printer is also capable of sending this information to the EGM. CP:1291 (For those ticket printers not currently capable the requirement is not mandatory, however it is recommended future functionality as it will allow auditing of ticket printer firmware versions over a network in future versions of QCOM.)

22.5 AUTHNO Generation

Notes to TITO system developers

An AUTHO representing a barcode of all 0's or all 9's is reserved for EGM test ticket prints.

In QCOM TITO, AUTHNO values are generated by the TITO system. It should be noted that the AUTHNO field in QCOM TITO message packets have some spare capacity. Accordingly, the system should be mindful of the actual number of digits encoded onto a printed ticket for the given EGM and not generate an AUTHNO utilising more digits than will be actually encoded by the EGM into the printed ticket barcode.

The following paragraphs assume a ticket database is utilised (at this time printed barcodes would be challenged to encode the entire output of a strong Digital Signature so its a database is really the only option unless another an alternative technology is used. If tickets could encode a Digital Signature then it could be feasible to implement a secure TITO system without the need for a ticket database.).

The intent of the following discussion is to raise awareness that a weak AUTHNO generator algorithm is potentially a security risk regarding theft of unclaimed tickets.

Each generated AUTHNO must uniquely identify the ticket among all currently active tickets in the system.

The AUTHNO generator algorithm which determines AUTHNO must be approved and must be secure given the level of operating risk. This means the AUTHNO generator algorithm must ensure that the probably of a collision (or duplicate) in the ticket database is virtually zero.

For the best integrity and security, the AUTHNO generator algorithm would also ensure that the probably that an attacker may either guess or reverse engineer an active AUTHNO is virtually zero. For example, given an attacker has full knowledge of the AUTHNO generator algorithm and all the details of a given printed ticket excluding the barcode information, an attacker should still not be able to determine the correct AUTHNO (i.e. barcode ID) of the ticket.

The above requirement infers that the ticket database has a secret e.g. a PRNG state or HMAC seed (whether the secret is per ticket or global or how often it changes should be balanced against operating risk).

Acceptance of any AUTHNO generator algorithm is at the discretion of the CEO.

Appendix A - EGM State Test Pattern

QCOM v1.6 - EGM State, Test Pattern

Legend:

Y = Yes,
n = No,
. = Don't care,
NA = Not Applicable

Communicate[1]	Physical Credit Input Enabled[2]	Allow ECT In[3]	Display CDCs	Display GPM	Display SPAM	Player Clock Display (QPV1)	Ext. Jackpot Inf. Poll Display(QPV1)	
								EGM States
Y	Y	Y	Y	Y	Y	Y	Y	Idle Mode - General/Betting/Game Display
Y	Y	Y	Y	Y	Y	Y	Y	Idle Mode - Game Selection
Y	.	Y	Y	Y	Y	.	.	Idle Mode - Full (primary) Screen Reserve Feature
Y	.	Y	Y	Y	Y	.	.	Idle Mode - Full (primary) Screen Attract Mode
Y	.	Y	Y	Y	Y	Y	.	Idle Mode - Other 2nd Screen Displays[A]
Y	n	Y	NA	NA	NA	NA	NA	Idle Mode – Power-save
Y	n	Y	n	.	Y	.	.	Hopper Collect
Y	n	Y	n	.	Y	.	.	Ticket Out Lockup
Y	n	Y	n	.	Y	.	.	Cancel Credit
Y	n	Y	n	.	Y	.	.	RCRF
Y	n	Y	n	.	Y	.	.	RCRF - Cancel Credit
Y	n	Y	n	.	Y	.	.	ECT Lockup (ECT out)
Y	n	Y	n	.	Y	.	.	System Lockup
Y	.	Y	n	.	Y	Y	Y	In Play
Y	.	Y	n	.	Y	Y	.	In Play - Game Features
Y	.	Y	n	.	Y	Y	.	Gamble
Y	n	Y	n	.	Y	.	.	Large Win Lockup
Y	n	Y	n	.	Y	.	.	LP Award Lockup
Y	n	Y	n	.	Y	.	.	CRanE Lockup
n	n	n	n	n	n	.	n	Power Up - Self Tests
								EGM Concurrent States
Y	n	Y	.	.	.	.	.	Audit & Test Modes
Y	n	Y	.	.	.	.	.	Door Open
Y	n	Y	.	.	.	.	.	Fault Conditions
n	n	n	.	.	.	.	.	Fault Condition - RAM Error

Notes:

1. Assume manual setup is completed
2. Assume config complete, no CDCs & credit meter < limit
3. ECT In's are accepted but CM is typically not updated until return to Idle Mode (16.1.1)
A. This refers to game rules displays, PIDs.

Figure 12 EGM State Test Pattern

Appendix B – ANZ Gaming Machine National Standards (GMNS) and QCOM

Generally speaking, Iif a conflict is identified between GMNS and QCOM, then where it is possible to satisfy both requirements, then this must be done. Where conflicting requirements cannot both be satisfied, then generally the QCOM Protocol takes priority unless specified otherwise in this document. Please notify the OLGR of any conflicts that cannot be resolved by satisfying both requirements documents or are not listed here.

Known conflicts with GMNS v10:

- GMNS uses CRECANLIM & MAXHOPPER (<=) to decide whether a hopper pay or CC should occur. QCOM uses only one parameter COLLIM (<).
- Differing definitions of idle mode should be noted (3.3.1). NB: the QCOM definition does not extend beyond the scope of this document.
- GMNS v9 3.13.8 states "No changes to the set of games offered to the patron for selection (or to the paytable) are permitted while there are credits on the player's credit meter or while a game is in progress. Except for the reference to changes to the paytable, for QCOM EGM's this requirement is implemented by the monitoring system rather than the EGM. With respect to changes to the paytable, QCOM agrees with GMNS on this point.
- GMNS v10 section 3.16 - table 3.6 states "External Peripheral Controller Fault / Disconnect" are basically fault conditions however QCOM treats LP time-outs only as CDCs.
- QCOM mandates the RTP of the RCRF must equal MINTRP, where as GMNS states the RTP of the RCRF may be in the range MINRTP to 100%. (It should be noted that making the RCRF RTP greater than or equal to the RTP of any game in an EGM may make the RCRF more attractive to play than the game itself, thus detracting from playing the gaming machine as intended)
- In QCOM v1.6.6 "CRLIMIT" was changed to not apply to cash ticket in. Reason: suspected possible future scenarios whereby it may be desirable to have a different banknote limit with respect to cash ticket-in limit. Contention with GMNS 10.3 section 5.3.9 & potentially affects the wording of 5.3.10
- Meters. Some meters have different definitions between QCOM and GMNS (e.g. Cancel Credit (12.1.1), Progressive and Money Out meters). This only affects EGM audit mode meter display. Refer to section 4.1.6 for how this must be managed.

Appendix C – Multi-jurisdictional Support

Multi-jurisdictional support in QCOM is provided by offering a wide range of parameters, allowing each EGM to be aware of and configured as per the local requirements of each jurisdiction. EGMs wishing to achieve better multi-jurisdiction support on the QCOM protocol should also be aware of the following issues:

- Some Casinos/jurisdictions require tower lights, so QCOM tower light support is recommended (15.4.21).
- Some jurisdictions do not require a player reserve function, it is recommended that the reserve function be able to be toggled on or off from within test/audit mode of the EGM, or upon EGM setup, or by some other method.

As a general rule for better multi-jurisdictional support, it is recommended that the EGM supports as many optional parameters and features of QCOM as possible.

Appendix D – Gaming Machine Manufacturer ID Numbers

Refer section 15.1.7 for more information on the MID field.

This list is current as of the date of this document.

(*The MID master list is located within the OLGR internal document with reference FM35. This list includes reserved values. The OLGR licensing database should also be consulted*)

EGM Manufacturer	MID
Aristocrat Leisure Industries	0x00
I.G.T	0x01
Milwell…No longer in use	*0x02*
Olympic	0x03
S.G.T…No longer in use	*0x04*
ARUZE GAMING AUSTRALIA[1]	0x05
E.L.I…No longer in use	*0x06*
~~SHFL ENTERTAINMENT~~Bally Technologies[2]	0x07
~~Bally~~Not in use	*0x08*
Konami	0x09
Atronic	0x10
Not in use	0x11
A.G.T.	0x12
Not in use	0x13
Wymac Gaming Solutions	0x14
Not in use	0x15
Not in use	0x16
Voyager Gaming Technologies (VGT)	0x17

---

[1] *formerly Pacific Gaming*

[2] *Formerly SHFL Entertainment, formerly Shufflemaster, formerly Stargames, formerly Vidco*

Appendix E – List of Subjective QCOM Requirements

"Subjective" means; that acceptability of a particular requirement may be subject to the opinion of the evaluator, typically due to the aesthetic nature of the requirement. Refer / search for abbreviation: "CEO", also see Display of Text Data – General Requirements. (section 3.6.3). Testers must record on file screenshots of all subjective requirement implementations for the record.

3.1.2   EGM Serial Number Setup Function

3.4.2   Hopper Refills

3.6.3   Display of Text Data – General Requirements.

4.1   Audit Mode General Items

6.2   Display of Communication Disabling Conditions

7.10.3.12   EGM Door Open/Close and Stacker Removed/Returned Events

10.1   Progressives General

10.5   QCOM Stand Alone Progressive Jackpots

10.6.1   LP Lockups **x 2**

15.4.5   EGM Parameters Poll

15.4.7   External Jackpot Information Poll

15.4.9   System Lockup Request Poll **x 2**

15.4.20 Specific Promotional/Advisory Message (SPAM) Poll **x 2**

15.5.3   EXTD: General Promotional Message (GPM) **x 2**

15.5.5   EXTD: Site Details

15.6.2   Note Acceptor Status Response

17.2   The CRanE Lockup Condition

17.5   CRanE Lockup Acknowledgement Poll

22.2   Ticket In Process

22.2.3   Cash Ticket In Log

22.1.2   Cash Ticket Out Log

22.3   Ticket Details

Appendix F – Additional Submission Requirements for QCOM Evaluations

This appendix specifies the QCOM specific submission items (ancillary to all GMNS & OLGR Submission Requirements) for a QCOM EGM software evaluation. All required submission material must be in electronic format unless not applicable. The most recent published version of the list is effective immediately on all future submissions.

1. Source code reference table (refer section "QCOM Checklist – Full"). A table detailing for each QCOM Checklist checkpoint with "PC: Verify in source code" as a precondition, a cross-reference to the module name and line number in source code with a short explanatory note (where not obvious) of how the requirement of the checkpoint was implemented. (This may be submitted after the first pass of the QCOM evaluation has been completed.)

2. A 'how-to' procedure for the reconciliation of the EGM's reported QCOM program hash, with the binary images produced from the EGM software build process for the submitted game.

3. A complete list of all possible faults (& manufacturer specific custom lockups if any) on the EGM, cross-referenced against the applicable QCOM event code.

4. A complete source code reference list of all calculations & comparisons involving dates or time (other than simple TZADJ adjustments for display or logging purposes). Refer 15.5.1

5. RTEXT Event Reference Table. A list of all uses of the RTEXT event message (7.10.1.16) implemented by the EGM with source code references. E.g. "RTEXT", Description (may be a QCOM section reference where it is a QCOM required RTEXT), plus a reference to the source module name and line number.

6. A precise list of all QCOM requirements / functionality that are implemented by EGM **game** code (as opposed to being implemented by shell/base code which does not change for every game). E.g. Typically: Fonts & colours used by QCOM messages, position of messages, etc. (This is used for creating EGM model/manufacturer specific QCOM game checklists)

7. A fast-boot version of the EGM (preferred but optional). (The time in which it takes an EGM to boot-up, if long, can blow out evaluation times due the large number of times an EGM must be rebooted during a full QCOM evaluation.)

8. Toward the end of the evaluation, the manufacturer will be directed to submit an EGM and software to various monitoring operators for system integration testing. (When submitting a new QCOM compliant base to monitoring operators for system integration testing, please ensure there is an explanatory covering letter detailing: what the submission is for, any specific configuration issues, manual & instructions, any areas of significance that should be examined in detail, e.g. first progressive game etc.)

QCOM Protocol EGM Checklist - GAME
QCOM Protocol Version 1.6.76. Game Checklist Version j̶i̶

The following quick checklist ~~or similar (if not covered elsewhere)~~that must be completed for every new game developed for QCOM. *(NB: The document must be protected in order to activate the controls in this checklist)*

Manufacturer:	
Game/Multi-game name & software version:	
Manufacturer base/shell software version:	
Evaluating Officer/s:	
All ok. (reviewer) Date & Signature:	

This checklist is motivated by EGMs which utilise different fonts, colours & graphics between games and ~~any other items~~ implement some QCOM functionality on "game-side" that may ~~change~~ require change on a ~~per~~ game per game basis.

Tick if ok, or mark as NA if not applicable.

1. Bring the EGM on-line, add credit and play some games and note any problems appearing on QSIM or on the EGM. Ideally try to get a non-zero value on every meter the EGM reports and check the meters balance (EGM->Request->More->More->Reconcile). (Any red coloured events appearing in the event window must be investigated as they could indicate an issue) ....☐

2. Ensure the EGM returns its Program Hash response within the current OLGR specified time-out period...☐

3. Ensure for all Game Configuration Responses (F4) (15.6.11) that the message data (I.e. No. of Games, GVNs, PNUM, VARs & PRET incl. the LPOnly flag & HS flag) is all correct for each game ....☐

4. Ensure full length SPAM A & B messages (15.4.20) are still displayed correctly and are still clearly legible and 5mm high (min) & (10mm for prominent). (~~c~~Check all printable characters (*2.3.8*) as fonts may change between games). (NB: If the display font is proportional, make sure the string is filled with "FATTER" characters, i.e. capitals) (Rationale: some games use a new font, screen layout & colours for every game). Also note that in QPv1.6 SPAM A has two display positions that must be checked ....☐

5. Ensure a full length General Promotional Message (15.5.3) (with "FATTER" chars) is still displayed correctly and are still clearly legible and 5mm high (min). (Also verify a range of each character type…use the 'default' GPM string for this)...☐

6. Check display of Communications Disabling Conditions (6.2) is still acceptable and are still clearly legible ....☐

7. Ensure a full length System Lockup (15.4.9) still functions (check all modes), displays correctly and is still clearly legible and 5mm high (min). Check "QCOMSL" text is present and 5mm........☐

8. Ensure a System Lockup is correctly queued if sent during any game mode (e.g. play, double-up, game features) ....☐

9. Ensure the ECT to/from EGM Lockup still functions and displays correctly (re last transferred amount & ECT lockup) ....☐

10. If the game is a LP, then ensure LP Award Events are logged the instant the win is revealed to the player and with the last received current amount for the level ....☐

11. If the game is a progressive, ensure that the Progressive Meters Response (especially HRATE's value and precision, refer 15.6.3) and Progressive Configuration Responses (15.6.4) are correct for the game...☐

12. Ensure the Bet Meters response is correct for the game (use debug window) and is correctly incrementing for a range of bets (15.6.5). QPv1.6 only..☐

13. Ensure the Player Choice Meters response is correct for the game (use debug window) and is correctly incrementing for all player choice options (15.6.7). QPv1.6 only. ....☐

    How many PC meters were there for each game:

14. Verify the game display of the External Jackpot Information poll (15.4.7) is acceptable. (Check 8 levels linked to a PGID). QPv1.6 only ....☐

15. Verify CRanE Lockups (17.2) operate and display correctly (NB use a RATE value of -1 to force a lockup) ....☐

16. Auto-play the actual production version of the game for a number of hours at normal game speed to ensure no serious events are generated by the protocol simulator or other observed issues..☐

17. ~~After play-testing, ensure the Top NP Prize Event (7.10.3.35) 0x2054 has not been logged more frequently than expected (95% CI for example) or desired (re spamming) for the given amount of turnover. (NB: this event is no longer logged in QCOM v1.6.3 EGMs and above)~~......................☐

~~18.~~ 17. ....Complete the manufacturer specific QCOM game checklist created from Appendix F Item 6 ....☐

~~19.~~ 18. ....If the game has a feature (other than vanilla free games), especially linked progressives, then record a video of the feature to completion (RTIM) and store on file ....☐

19. Verify cancel credit lockups operate and display correctly...☐

20. Verify the Residual Credit Removal Feature operates and displays correctly (check all four outcomes; win, lose, cancel credit, abort) ....☐

21. Verify the clock display is still clearly legible (15.5.1) and 7mm high (min).................................☐

Notes:

QCOM Protocol EGM Checklist – FULLFULL* / PARTIAL

For Protocol Version 1.6.76, Checklist Version ji

A "FULL" evaluation (required for a new implementation of QCOM) denotes an evaluation where at a minimum, all checkpoints were completed that were not marked with "PC:". I.e. "FULL" is not literal; the checklist is rarely if ever fully completed for any given evaluation.

Manufacturer:	
Game/Multi-game name & software version:	
Manufacturer base/shell software version:	
Evaluation Type: *(If partial, then incl. ref to FULL eval checklist)*	FULL / PARTIAL
Reason for Evaluation: *(or what was evaluated)*	_____
Start Date:	_____
Evaluating Officer/s:	
Complete: *(Evaluating officer/s please sign and date)*	_____
Reviewer Date & Signature:	
System Integration Testing Required?	

Correspondence History: (i.e. letter date, reference, subject: rejection/approval. If rejection, sign and date when all issues in letter have been addressed. Also reference any QCOM PEP bugs here. If this is documented elsewhere then include a reference)

Checklist Sections ~~Tested~~Completed:

Date, Initial and Tick (Initial and Date when complete)

FULL: The checklist sections below must all be completed before QCOM approval may be granted where no prior QCOM approval has been granted for the given implementation of QCOM ....☐

"Emulator" checks completed...☐

"Ram clear" checks completed ....☐

"Verify in source code" checks completed ....☐

Program Hashing methodology and content acceptable (Principal Technical Officer (QCOM) ....☐

"CEO" checks completed (photo recorded and signed of~~f~~ by OLGR CEO or ~~delgated~~delegated ~~OLGR~~

Officer)..☐

All remaining checks as applicable ....☐

Live system check (EGM must be test on at least two monitoring systems before approval).............☐

Principal Technical Officer (QCOM) must review the implementation of subjective requirements ....☐

PARTIAL: The sections below indicate optional sections of the QCOM protocol which may have ~~or have not~~ already been tested or yet to be tested. All applicable sections below must be completed before QCOM approval will be granted or when a new feature is added to the EGM software not previously tested with respect to QCOM to maintain an existing QCOM approval. (Quoted text below indicates the reference text used within the checklist body pertaining to the given functionality)

All ("ALL"). "**PC: ALL**:" checks must be performed for all ~~possible~~ partial QCOM evaluations ~~(see below)~~ ....☐

Optional checkpoints are marked "PC:" + one or more <QuotedAbbrevNameBelow>:

Variation Hot Switching ("**HS**") ....☐

Denomination Hot Switching ("**DHS**") ....☐

"**Multi-Game**" support ....☐

"**Extended Multi-Game Support**" ....☐

"**Extended Variation Support**" ....☐

"**Multi-Jurisdiction**"..☐

Banknote acceptor support "**Note Acceptor**" ....☐

Real Time Clock "**RTC**"..☐

"**Progressive**" & "**LP**" ....☐

"**Progressive**" & "**SAP**"...☐

"**CustomSAP**" ....☐

Shared Progressive "**SP**" ....☐

Ticket Out ("**TO**")..☐

Ticket In ("**TI**")...☐

"**Tower Light**" ....☐

Games with "**Free Games**" (incl games with features)..☐

EGMs with Mechanical Meters ("**Mech Meters**")..☐

EGMs with Hoppers ("**Hopper")** ....☐

Notes and recommendations on completing the checklist.

The checklist which follows is to be used in conjunction with the protocol document. The checkpoint numbers correspond to superscript endnote numbers that appear throughout the protocol document (i.e. "CP:". The checklist does not introduce any new requirements.

Disclaimer for external users of the QCOM checklist: This checklist is issued without any warranty whatsoever and shall be used completely at the user's risk.

For every new QCOM implementation for an EGM, **it is mandatory** that the EGM manufacturer to complete the checklist to the best of their ability and provide it along with their next QCOM EGM software submission.

The checklist assumes a thorough working understanding of the QCOM protocol and QSIM Protocol Simulator.

- Work through the protocol document locating the endnotes. Each endnote corresponds to a check in the checklist. To work out what needs to be checked, carefully read the sentence/section of the protocol containing the endnote. *(Suggestion: If jumping from each checkpoint to the relevant QCOM section is found to be a hassle, then open the QCOM document in Word and select Insert->Reference->Footnote, then select 'Convert', then select 'Swap footnotes with endnotes' and press 'ok'. Also ensure that footnotes are set to 'continuous' and 'numbers' under the footnote format display in Word. Now the entire list of checkpoints should have been converted to footnotes and will be distributed throughout the document on their appropriate page and section.)*
- **Recommended order of completing the QCOM checklist.** Do not complete the checklist sequentially starting from point 1. The QCOM checklist is best completed working from the low level up. The following order for completion of the QCOM checklist is recommended:
    - Physical Layer (section 13),
    - Section 11 on Program Hashes, especially 11.1.9. (problems in this area can take some time to address and therefore need to be checked early)
    - the Data Link Layer (section 14)
    - Poll Address Configuration (refer sections 15.5.4, 8.3 & 21)
    - The remaining section EGM defaults (Section 8)
    - The Event Queues (Sections 7 except 7.10, 15.4.15 & 15.4.14.)
    - Then complete section 3 and anything to do with EGM/Game configuration.
    The remainder of the checklist may be completed in any order except for source code checks which are best saved until last.

Checkpoints labelled 'verify in source code' should not to be completed until all problems found via empirical testing have been addressed. At this time the EGM manufacturer will then be requested to submit a list detailing for each checkpoint with a "PC: Verify in source code" precondition, a cross-reference to the module name and line number with a short explanatory note of how the requirement of the checkpoint was implemented.

- For all points that the EGM fails, place a reference to the checklist number on the notes page provided on the next page. Also include the protocol document section it came from plus details of exactly how the EGM failed the test. This will make report writing later much quicker.
- If you cannot work out how to perform a test to complete a check, do not waste too much time on it, but just leave it blank and continue. The difficult tests are in Polling and Response section of the protocol (section 14.1).
- During the evaluation the test person must be able to explain the occurrence of every event QSIM reports. Any unexplained events may indicate an EGM software deficiency.
- The checklist (excluding emulation and source code checks and EGM boot-up time*) takes approximately 120 hours to complete by a test engineer experienced in QCOM and assuming no non-conformances are found. The time to process a single non-conformance is very roughly estimated to be ~4-8 hours on average (this includes analysis & reporting, side effect testing, re-testing & regression testing). *NB: Also, it should be noted that the time in which it takes an EGM to boot-up can affect the evaluation time quite significantly due the large number of times an EGM must be rebooted during a QCOM evaluation. In some cases a fast boot version of the software may be desirable.*

- Checkpoints labelled "RAM clear tests". These checkpoints require the tester to perform tests at various stages of EGM configuration. To facilitate this, the auto-queuing of polls by the protocol simulator may be turned off under the poll cycle menu in QSIM.
  Stages of configuration after an EGM RAM clear (for performing RAM clear tests):
    0   EGM RAM cleared and powered up but serial number not entered.
    1   Serial number entered, but no communications received yet.
    2   EGM Configured with Poll Address via broadcasts and resumed responding on designated poll address.
    3   Configured via EGM Configuration Poll (15.4.2).
    4   Configured via EGM Game Configuration Poll (15.4.3).
    5   Configuration completed but before first play.
- A quick way of performing an EGM RAM clear (~~eg.~~e.g. Button on main p.c.b.) will expedite testing.
- Please notify OLGR of any discovered non-conformances found which were not covered by a checkpoint.
- "CEO" requirements and checks. These all refer to requirements that have an aesthetic aspect to them such visual legibility or intuitiveness. I.e. it is not strictly black and white as to whether the requirement is met. They are kept to a minimum as much as possible. However to ensure some sort of consistency with the implementation of these requirements, they can only signed off by a central authority at this time, namely the CEO - OLGR. To ensure a record of acceptable behaviour is kept, the evaluator is required to record a screenshot of the implementation once it is deemed acceptable.
- The following macros have been pre-defined in this document to assist in completing the checklist:
  - Alt-X   =   Substitutes the character to the left with red "X" symbol.
  - Alt-T   =   Substitutes the character to the left with Green "√" symbol.

Checklist Abbreviations:

PC: "xxx".

> PC means 'Pre-Condition/s' (this refers to the applicability of a checkpoint in the checklist following) where "xxx" corresponds to one of the "quoted" text labels on checklist ~~title~~ summary page. For example, say the first SAP game is being evaluated under on a ~~previsouly~~previously approved QCOM base. A QCOM checklist would be partially completed inclusive of all the checkpoints labelled with any combination of "PC: SAP/Progressive/ALL"

List Here:
- ***Exemptions / Acceptable differing behaviour to note*** (Approved exemptions must be approved in writing or listed on the QCOM protocol approval letter)
- ***Supplementary Checks*** (to be incorporated as check points into the next protocol release)

If this is documented elsewhere then site a reference.

Section   Check - Reason for exemption/Description of differing behaviour

Notes: (Reference all failed points here, including details; checkpoint, section, description of non-conformance, how to duplicate problem if not obvious. If this is documented elsewhere then site a reference)

(If cross-referencing each checkpoint to the relevant QCOM section is found to be a hassle, then open the QCOM document in Microsoft Word and select Insert->Reference->Footnote, then select 'Convert', then select 'Convert all endnotes to footnotes' and press 'ok'. The entire list of checkpoints will have been converted to footnotes and distributed throughout the document on their appropriate page and section.)

Checklist Begins:

1   Is an Ethernet Port present to the required specifications .... Pass ☐
2   PC: Hopper. RCRF Implemented as per GMNS .... Pass ☐
3   Verify door opens are not considered fault conditions (they must be self clearing and no fault flag set in default response) .... Pass ☐
4   PC: LP* .... N/A ☐ Pass ☐
    PC: Hopper & LP* .... N/A ☐ Pass ☐
    *PAMT Meter is implemented as defined. (ensure all EGM meters are non-zero (incl. RCRFT) when checking the above two checkpoints)
5   For each font used by the EGM for QCOM messages, verify all printable characters (i.e. @!#$aZ... etc) are displayed correctly (QSIM has a default broadcast message with all printable symbol characters in it) .... Pass ☐
6   PC: TO .... N/A ☐ Pass ☐
7   PC: TO .... N/A ☐ Pass ☐
8   PC: RAM clear (stage 0) .... Pass ☐
9   PC: RAM clear (stage 1) & Demonstration Mode .... N/A ☐ Pass ☐
10 PC: RAM clear (stages 0…5) .... Pass ☐
11 PC: RAM clear (stage 0) .... Pass ☐
12 PC: RAM clear (stage 0) .... Pass ☐
13 PC: RAM clear (stage 0) .... Pass ☐
    Also confirm the following serial numbers all work correctly:
    987654 .... Pass ☐
    999999 .... Pass ☐
    000000 .... Pass ☐
14 PC: RAM clear (stage 0) .... Pass ☐
15   .... Pass ☐
16 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
17 PC: DHS .... Pass ☐
18 PC: DHS .... Pass ☐
19 PC: DHS .... Pass ☐
    Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
20 PC: DHS .... Pass ☐
21 PC: DHS & Verify in source code .... Pass ☐
22 PC: Emulator .... Pass ☐
23 PC: Emulator .... Pass ☐
24 PC: Emulator .... Pass ☐
25 PC: Emulator .... Pass ☐
26 PC: Emulator
    Check all other ways of generating a RAM Error .... Pass ☐
27 PC: Emulator .... Pass ☐
28   .... Pass ☐
29   .... Pass ☐
30   .... Pass ☐
31 For each RTIM item, queue the event during a play and ensure that the queued event activates at the end of the current play in the case when the play button is being held down (ensure autoplay is on, ref:Parameter Poll) .... Pass ☐
32 It is very important to check all possible permutations (up to 15 tests may need to be performed) .... Pass ☐
33 PC: Hopper .... N/A ☐ Pass ☐
34 PC: Hopper .... Pass ☐
35 PC: Hopper .... Pass ☐
36 PC: Hopper .... Pass ☐

37   PC: Hopper .... Pass ☐
38   PC: Hopper .... Pass ☐
39   PC: Hopper .... Pass ☐
40   PC: Hopper .... Pass ☐
41   PC: Hopper .... Pass ☐
42   PC: Hopper .... Pass ☐
43   PC: Hopper .... Pass ☐
44   PC: Hopper .... Pass ☐
45   PC: Hopper & CEO (Store a screenshot on file if ok) .... Pass ☐
46   PC: Hopper .... Pass ☐
47   .... Pass ☐
48   PC: Hopper .... Pass ☐
49   PC: TO .... Pass ☐
50   .... Pass ☐
51   .... Pass ☐
52   .... Pass ☐
53   PC: TO & Hopper .... N/A ☐ Pass ☐
54   .... Pass ☐
55   PC: CEO (Store a screenshot on file if ok) .... Pass ☐
56   Check high priority states such as faults/lockup/CDCs .... Pass ☐
57   PC: ALL Ensure for all audit mode tests that the field label is clear and units are clearly
     indicated (eg. $.¢ or "0x" for hex numbers)
58   .... Pass ☐
59   PC: ALL. Is the correct PVERSION being reported wrt QCOM incept dates and the evaluation
     being conducted? .... Pass ☐
60   .... Pass ☐
61   .... Pass ☐
62   Verify labels and units are all displayed correctly for the following:
63   .... Pass ☐
64   .... Pass ☐
65   PC: LP or SAP .... N/A ☐ Pass ☐
66   .... N/A ☐ Pass ☐
67   .... N/A ☐ Pass ☐
68   .... Pass ☐
69   PC: RAM clear (stages 0..3) .... Pass ☐
     Re RAM clear (stage 2): Also ensure the display of denomination is not misleading in either
     audit mode or on the main game display before EGM configuration has been completed Pass ☐
70   PC: RAM clear (stages 3…4) .... Pass ☐
71   PC: Progressive .... Pass ☐
72   PC: ALL .... Pass ☐
73   PC: CEO (Store a screenshot on file if ok) .... Pass ☐
74   .... Pass ☐
75   .... Pass ☐
76   In real time .... Pass ☐
77   Stop sending or corrupt (do both) broadcast msgs and this time display should freeze .... Pass ☐
78   Ensure system time display is correct .... Pass ☐
79   PC: RTC. Real time and correct label .... N/A ☐ Pass ☐
     Changes wrt TZADJ .... N/A ☐ Pass ☐
80   PC: RTC. Keeps updating if broadcasts are not sent .... N/A ☐ Pass ☐
81   Should increment once every poll cycle and have the correct label .... Pass ☐
82   Invalid CRC polls to the EGMs adrs (QSIM special function) and global messages are not
     counted ....
      .... Pass ☐
83   .... Pass ☐
84   Increments only on CRC error to the EGMs adrs and not global CRC errors .... Pass ☐
85   .... Pass ☐
86   .... Pass ☐
87   .... Pass ☐
88   PC: Verify in source code. Overrun error implementation (Note, you can't generate an overrun
     error) .... Pass ☐

89 .... Pass ☐

90 Increments only upon a negative ack sent to the EGM (QSIM Special Function) .... Pass ☐

91 .... N/A ☐ Pass ☐

92 .... Pass ☐

93 Verify it increases as events are generated and is zero when queue is empty .... Pass ☐

94 .... Pass ☐

95 Normally zero. Stop poll cycle, generate some events on the EGM and this number must increase by the number of events generated .... Pass ☐

Restart polling and this number should slowly start decreasing to zero as events are sent .. Pass ☐

96 …and are updated in real time, try purging some events... Pass ☐

97 .... Pass ☐

Changing TZADJ should not alter the values of logged events .... Pass ☐

98 .... Pass ☐

99 .... Pass ☐

100 .... Pass ☐

101 .... Pass ☐

102 Verify display is correct for a couple of different variations and games .... Pass ☐

103 PC: SAP.. Pass ☐

104 PC: LP... Pass ☐

105 .... Pass ☐

Changing TZADJ should not alter this value after the fact .... Pass ☐

106 .... Pass ☐

107 .... Pass ☐

108 PC: DHS .... N/A ☐ Pass ☐

109 .... Pass ☐

Ensure a game is not queued.. Pass ☐

110 .... Pass ☐

111 .... Pass ☐

112 PC: Free games. Speically check the below for free games... Pass ☐

For everything else: Specifically check a play completes upon a CDC and when the play is a win it is paid correctly.. Pass ☐

113 For EGMs which allow credit input during a play ensure that upon a CDC during a play, the EGM immediately disables credit input.. Pass ☐

114 Check methods of collecting... Pass ☐

115 .... Pass ☐

116 .... Pass ☐

117 .... Pass ☐

118 .... Pass ☐

119 .... Pass ☐

120 .... Pass ☐

121 .... Pass ☐

122 .... Pass ☐

123 .... Pass ☐

124 .... Pass ☐

125 Verify while in all disabling conditions simultaneously... Pass ☐

126 PC: RAM clear (stages 0...5). Confirm disabling condition operates as per definition (disabled throughout stages 0..2, enabled 2+).. N/A ☐ Pass ☐

127 Ensure invalid CRC polls to the EGMs address result in a communication timeout disable after 10 seconds.. Pass ☐

Ensure no communication to the EGM results in a communication timeout disable after 10 seconds (QSIM No Poll Function) .... Pass ☐

Ensure the EGM still detects a communication timeout (& resets its PA) during audit mode and double up, etc. .... Pass ☐

128 Specifically check that the EGM re-enables on the PA broadcast which contains the EGM serial nos & pol address .... Pass ☐

E.g. Stop polling the EGM in QSIM ('NoPoll). Wait for EGM to enter CT. Restart polling until QSIM logs an 'EGM not Responding' event for the EGM then immediately set 'NoPoll' again. (Now that QSIM has declared the EGM as not responding, it will start adding the EGM into PA broadcasts). Result: The EGM should come in & out of CT ~ every 8-10 secs depending on how frequently QSIM sends out a PA Broadcast with the EGM in it. .... Pass ☐
Leave in this state for a minute and then restore EGM polling. Only one communications timeout event should be logged by the EGM .... Pass ☐

Ensure the EGM does not get stuck in limbo (an issue with QCOM v1.5 where the EGM wont accept a new poll address if it had never been polled on the prev assigned PA) if it the EGM is assigned a poll address but never gets a poll to that address (the EGM should come out of CT for 10 secs then return to CT) .... Pass ☐

129	....	Pass ☐
130	....	Pass ☐
131	....	Pass ☐

QPV v1.5 only. Ensure the EGM does not inadvertently consider 10 secs worth of NAKs to be a communications time-out .... Pass ☐

132	....	Pass ☐
133	....	Pass ☐
134	....	Pass ☐
135	....	Pass ☐
136	....	Pass ☐

Ensure SEF disable activates at the end of the current play in the case when the play button is being held down (ensure autoplay is on, ref:Parameter Poll) .... Pass ☐

137 Verify operation with all of the polls (EGM Cfg, Progr. Sig, Req. & Maint Poll) which send this flag

.... Pass ☐

Ensure MEF disable activates at the end of the current play in the case when the play button is being held down (ensure autoplay is on, ref:Parameter Poll) .... Pass ☐

138	PC: LP..	Pass ☐
139	PC: LP..	Pass ☐
140	PC: LP..	Pass ☐
141	PC: LP. But may be implemented regardless. Check operation of the CDC .... N/A ☐	Pass ☐
142	Refer Figure 12	Pass ☐

Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐

143	....	Pass ☐
144	....	Pass ☐
145	....	Pass ☐
146	Verify the EGM considers Comms. Disable conditions to be a part of idle mode. Ie. A Cashless-in when in door open mode when the EGM is also disabled via GEF. The EGM should add to credit meter upon door closure and should not wait for GEF enable	Pass ☐
147	PC: CEO (Store a screenshot on file if ok)	Pass ☐
148	PC: RAM clear (stages 0..2). Correct text is displayed	Pass ☐
149	....	Pass ☐
150	....	Pass ☐
151	....	Pass ☐
152	....	Pass ☐
153	....	Pass ☐
154	....	Pass ☐
155	PC: LP..	Pass ☐
156	.... N/A ☐	Pass ☐
157	Confirm correct letters are used for each condition .... N/A ☐	Pass ☐
158	Verify when the EGM is not being polled all new events are queued and are sent once polling recommences in the correct order	Pass ☐
159	....	Pass ☐

Events logged must not be affected by TZADJ .... Pass ☐

Changing TZADJ should not alter the values of already logged events............................. Pass ☐

160	....	Pass ☐
161	....	Pass ☐

Using the QSIM debug window with time stamping switched on, generate an event on the EGM, compare the event time-stamp with the received time stamp displayed in the debug window and ensure the event time-stamp is not older than 3 seconds .... Pass ☐

162 Verify all buffered events in both queues are still present after an EGM power down .... Pass ☐

163 .... Pass ☐

164 .... Pass ☐

165 .... Pass ☐

Note the current EVTCRC and then request all events, send some NAKs & CRC errors during the resend including a power fail as well as other poll requests (eg. Meters, config) and ensure the same EVTCRC is received .... Pass ☐

166 Using the Purge Event Poll (Request events after each test to ensure the correct number of events have been purged) :
Purge one event at a time, this should have no effect until the 2nd event is purged at a minimum (check it goes full again on the next generated event) .... Pass ☐
Fill queue again then retest by purging half .... Pass ☐
Fill again then test by purging all .... Pass ☐

167 By generating many numbered events on the EGM, check the entire range is used and rolls over correctly (no zero SEQ should appear) .... Pass ☐

Purge up to 0xFE & request to confirm results .... Pass ☐
Purge up to 0x02 (ensure SEQ > 0x02) & confirm .... Pass ☐

168 Verify the PSN is preserved after an EGM power down .... Pass ☐

169 .... Pass ☐

170 .... Pass ☐

171 Ensure after sending a request all logged events that temporary event queue events are not reported again .... Pass ☐

172 Stop polling the EGM, queue events in both queues, eg. Door close & hopper collect, then verify the oldest is sent first .... Pass ☐

Then check vice versa (ie hopper collect then door open) .... Pass ☐

~~Note, be aware of potential year 2000 problems when comparing dates!~~

~~Also verify when date > year 2000~~..~~Pass ☐~~

~~PC: Verify in source code. Verify EGM source code for year 2000 bugs in this area of source code~~
~~...~~~~Pass ☐~~

173 Stop polling the EGM, overrun the event buffers on the EGM by generating many events, restart polling, ensure no events were lost before both queues became full (look for queue full events in correct positions) .... Pass ☐

Repeat this test from an empty queue .... Pass ☐

174 Check both numbered & unnumbered events logged to the temp. event queue .... Pass ☐

NAK an unnumbered event and ensure it is resent .... Pass ☐
NAK an unnumbered extended event and ensure it is resent .... Pass ☐
Repeat the above tests for numbered events the general event queue .... Pass ☐

175 QSIM will auto detect this .... Pass ☐

176 PC: RTC. Test for RTC drift. Synchronise QSIM's & the EGM's RTC with a known accurate clock. Suppress sending broadcasts for at least 24 hours and compare the time difference between the EGM and clock. A difference approximately greater than 4 sec/day is of concern .... N/A ☐ Pass ☐

177 PC: RTC. Use QSIM custom function to adjust time and use EGM RTC display .. N/A ☐ Pass ☐

178 PC: RTC.. N/A ☐ Pass ☐

179 PC: RTC.
Confirm the EGMs event queue display is correct for year 2000 & year 2030 .... N/A ☐ Pass ☐

180 PC: RTC.

Confirm the EGMs event queue display is correct for year 2000 & year 2030 .... N/A ☐ Pass ☐
181 PC: Verify in source code .... Pass ☐
182 PC: Verify in source code .... Pass ☐
183 PC: Verify in source code .... Pass ☐
184 PC: Emulator .... Pass ☐
185 PC: Verify in source code .... Pass ☐
186 For each fault/lockup condition below verify the following (there is a box after each FC to tick):

- The fault/lockup flag is set in the default response as applicable for fault/lockup events
- It is generated according to its definition (note there may be more than one way to generate the fault)
- Any extended data is correct (check all possible variants)
- No problems occur if the EGM is powered down during the condition eg. Relogging the event/lockup
- An 'All Faults Cleared' event is logged after it has been cleared and no other fault condition exist. Or in the case of a lockup the 'lockup cleared' event is logged for each lockup as applicable (Refer the lockup cleared event for a description of exactly when)
- The EGM manufacturer must supply with their submission a complete list of fault/lockup conditions and how they are detected and generated.

187 .... Pass ☐
188 .... N/A ☐ Pass ☐
189 .... Pass ☐
190 .... Pass ☐
191 .... Pass ☐
192 .... N/A ☐ Pass ☐
193 .... N/A ☐ Pass ☐
194 PC: Hopper .... N/A ☐ Pass ☐
195 PC: Hopper ~~present~~ .... Pass ☐
196 PC: Hopper ~~present~~ .... Pass ☐
197 PC: Hopper ~~present~~ .... Pass ☐
198 PC: Hopper ~~present~~ .... Pass ☐
199 PC: Hopper ~~present~~ .... N/A ☐ Pass ☐
200 PC: Mech Meters .... N/A ☐ Pass ☐
201 .... N/A ☐ Pass ☐
202 .... N/A ☐ Pass ☐
203 PC: Touch Screen .... N/A ☐ Pass ☐
204 PC: Verify in source code .... N/A ☐ Pass ☐
205 PC: Verify in source code & CEO (Store a screenshot on file if ok) .... N/A ☐ Pass ☐
206 Check All .... N/A ☐ Pass ☐
207 .... N/A ☐ Pass ☐
208 PC: Stepper .... N/A ☐ Pass ☐
209 PC: Bonus Device .... N/A ☐ Pass ☐
210 PC: NA .... N/A ☐ Pass ☐
211 PC: NA .... N/A ☐ Pass ☐
212 PC: NA .... N/A ☐ Pass ☐
213 PC: NA .... N/A ☐ Pass ☐
214 PC: NA .... N/A ☐ Pass ☐
215 PC: TI .... N/A ☐ Pass ☐
216 PC: NA .... N/A ☐ Pass ☐
217 PC: Emulator .... N/A ☐ Pass ☐
218 PC: Emulator .... N/A ☐ Pass ☐
219 PC: Emulator & EEPROM .... N/A ☐ Pass ☐
220 PC: RAM clear (stage 2) .... Pass ☐

The default date and time should not appear .... Pass ☐
221 PC: TO .... N/A ☐ Pass ☐
222 PC: TO .... N/A ☐ Pass ☐
223 PC: TO .... N/A ☐ Pass ☐
224 PC: TO .... N/A ☐ Pass ☐
225 PC: Emulator .... N/A ☐ Pass ☐

226 .... N/A ☐ Pass ☐
227 .... N/A ☐ Pass ☐
228 .... N/A ☐ Pass ☐
229 .... N/A ☐ Pass ☐
230 .... Pass ☐
231 PC: LP... N/A ☐ Pass ☐
232 PC: LP... N/A ☐ Pass ☐
233 PC: LP. Send a new LP current amount broadcast just prior the LP Award event is logged and
ensure the EGM logs the event with the new value .... N/A ☐ Pass ☐
234 PC: SAP.. N/A ☐ Pass ☐
235 .... N/A ☐ Pass ☐
236 .... Pass ☐
237 .... Pass ☐
238 This should have been checked pretty thoroughly by now.. Pass ☐
239 .... Pass ☐
240 .... Pass ☐
241 .... N/A ☐ Pass ☐
242 PC: LP... N/A ☐ Pass ☐
243 PC: Hopper .... Pass ☐
244 .... Pass ☐
245 .... Pass ☐
246 PC: SAP.. N/A ☐ Pass ☐
247 .... Pass ☐
248 .... Pass ☐
249 PC Hopper .... N/A ☐ Pass ☐
250 .... N/A ☐ Pass ☐
251 .... Pass ☐
252 .... Pass ☐
253 Do not poll the EGM for approximately one minute, ensure the EGM only reports one CT event
for this period .... Pass ☐

Constantly NAK the EGM's responses for over a minute; afterwards confirm the EGM did not
log any CT events .... Pass ☐
254 .... Pass ☐
255 Stop polling the EGM in QSIM (NoPoll). Wait for the EGM to enter CT. Restart polling until
QSIM logs an 'EGM not Responding' event for the EGM then immediately set 'NoPoll' again.
(Now that QSIM has declared the EGM as not responding, it will start adding the EGM into PA
broadcasts). Result: The EGM should come in & out of CT ~ every 8-10 secs............... Pass ☐
Leave for at least 30 secs, then once polling is resumed ensure that the EGM logged only the
one CT event .... Pass ☐
256 .... Pass ☐
257 PC: HS .... N/A ☐ Pass ☐
258 PC: RAM clear (stage 4) .... Pass ☐
259 PC: RTC... N/A ☐ Pass ☐
260 .... Pass ☐
261 PC: RTC.. Pass ☐
262 .... Pass ☐
263 PC: RTC.. Pass ☐
PC: RTC. After a period with no communications, say 30 secs, power the EGM off and on and
ensure the power off event time is exactly correct +/- 1sec .... Pass ☐
264 .... Pass ☐
265 .... Pass ☐
266 .... Pass ☐
267 .... Pass ☐
268 .... Pass ☐
269 .... Pass ☐
270 .... Pass ☐
271 .... Pass ☐
272 .... N/A ☐ Pass ☐
273 .... N/A ☐ Pass ☐

274 PC: NA .... N/A ☐ Pass ☐
275 PC: NA .... N/A ☐ Pass ☐
276 PC: NA .... N/A ☐ Pass ☐
277 PC: NA .... N/A ☐ Pass ☐
278 PC: Mech Meters .... N/A ☐ Pass ☐
279 PC: Mech Meters .... N/A ☐ Pass ☐
280 .... N/A ☐ Pass ☐
281 .... N/A ☐ Pass ☐
282 PC: CEO (Record in the space below the approach used) .... Pass ☐

283 .... N/A ☐ Pass ☐
284 PC: NA .... N/A ☐ Pass ☐
285 PC: NA .... N/A ☐ Pass ☐
286 PC: NA .... N/A ☐ Pass ☐
287 PC: NA .... N/A ☐ Pass ☐
288 4 cases per door. Initial door open/close then open/close during power fail, the event is only
logged for the closed open case .... Pass ☐
289 .... Pass ☐
290 .... N/A ☐ Pass ☐
291 .... N/A ☐ Pass ☐
292 PC: NA .... N/A ☐ Pass ☐
293 .... N/A ☐ Pass ☐
294 PC: TO .... N/A ☐ Pass ☐
295 PC: TO .... N/A ☐ Pass ☐
296 PC: NA .... N/A ☐ Pass ☐
297 PC: NA .... N/A ☐ Pass ☐
298 PC: NA .... N/A ☐ Pass ☐
299 PC: RAM clear (stage 2) .... Pass ☐
300 PC: RAM clear (stage 3) .... Pass ☐
.... Pass ☐
301 PC: LP or CustomSAP... Pass ☐
.... Pass ☐
302 PC: LP or CustomSAP... Pass ☐
.... Pass ☐
303 .... N/A ☐ Pass ☐
304 .... N/A ☐ Pass ☐
305 .... N/A ☐ Pass ☐
306 .... N/A ☐ Pass ☐
307 .... N/A ☐ Pass ☐
308 .... N/A ☐ Pass ☐
309 PC: DHS .... N/A ☐ Pass ☐
310 PC: RAM clear (stage 3) .... Pass ☐
311 PC: RAM clear (stage 3) .... Pass ☐
312 PC: DHS .... Pass ☐
.... Pass ☐
313 PC: Hopper .... N/A ☐ Pass ☐
314 PC: LP & CustomSAP... N/A ☐ Pass ☐
315 PC: LP & CustomSAP... N/A ☐ Pass ☐
316 PC: LP & CustomSAP... N/A ☐ Pass ☐
317 .... N/A ☐ Pass ☐
318 PC: TO .... N/A ☐ Pass ☐
319 .... N/A ☐ Pass ☐
320 PC: Hopper .... N/A ☐ Pass ☐
321 .... Pass ☐
322 .... Pass ☐
323 .... Pass ☐

324 .... Pass ☐

325 PC: TI.. N/A ☐ Pass ☐

326 PC: TI... Pass ☐

327 PC: TI.. N/A ☐ Pass ☐

328 PC: TI... Pass ☐

329 .... Pass ☐

330 PC: TI... Pass ☐

331 An arbitrary limit should be present. See section 12 .... Pass ☐

332 .... Pass ☐

Some EGMs abort a payout if a fault occurs during. Ensure the event is still logged with the correct amount ....
.... N/A ☐ Pass ☐

333 PC: Hopper .... N/A ☐ Pass ☐

334 A good way to overflow the temp. event queue is to use 8 DLSA levels at 100% with no lockup for about five or so games .... Pass ☐

335 PC: Multi-game .... N/A ☐ Pass ☐

336 PC: Multi-game .... N/A ☐ Pass ☐

337 .... N/A ☐ Pass ☐

338 PC: SAP.. N/A ☐ Pass ☐

339 PC: Hopper .... N/A ☐ Pass ☐

340 PC: TO .... Pass ☐

341 PC: TO .... Pass ☐

342 PC: TO .... Pass ☐

343 PC: TO .... Pass ☐

344 PC: TI .... Pass ☐

345 PC: TI .... Pass ☐

346 PC: Verify in source code .... Pass ☐

347 PC: RAM clear (stages 2...3). Verify for coins, notes & cash-less in................................. Pass ☐

348 PC: RAM clear (stage 1). Use EGM real time display to verify .... Pass ☐

349 PC: RAM clear (stage 2) .... Pass ☐

350 PC: RAM clear (stage 2) .... Pass ☐

351 PC: RAM clear (stage 2) .... Pass ☐

NB. Stacker door initial state is indicated via the Stacker removed/returned events.

352 .... Pass ☐

353 PC: RAM clear (stage 2). This will be the NV-RAM clear event .... Pass ☐

354 PC: RAM clear (stage 5) .... Pass ☐

355 PC: RAM clear (stage 5) .... Pass ☐

356 PC: RAM clear (stage 5) & Progressive .... N/A ☐ Pass ☐

357 PC: RAM clear (stage 2). Turn off QSIM auto-queuing of polls and verify first received EGM Cfg Response has zero DEN & TOK .... Pass ☐

358 PC: RAM clear (stage 3) & Progressive .... N/A ☐ Pass ☐

359 PC: RAM clear (stage 3) & Non-progressive... Pass ☐

360 PC: RAM clear (stage 5). Verify for all PSNs (event purge & cashless in) by sending these polls
.... Pass ☐

361 PC: RAM clear (stage 5) & NA. Test EGM accepts all notes by default .... N/A ☐ Pass ☐

362 PC: RAM clear (stage 5). Verify all parameters via audit mode display............................ Pass ☐

363 PC: RAM clear (stage 5). Verify empirically .... Pass ☐

364 PC: RAM clear (stage 5) .... Pass ☐

365 PC: RAM clear (stage 5) & Hopper. Verify in audit mode display .... N/A ☐ Pass ☐

366 PC: RAM clear (stage 5) & TO. Verify Empirically.................................... N/A ☐ Pass ☐

367 PC: Verify in source code .... Pass ☐

368 .... Pass ☐

369 .... Pass ☐

370 PC: Verify in source code .... Pass ☐

371 Start with the door open initially, power down the EGM, close the door, power up the EGM and expect to see logged a door close event and no power off door access event (if applicable)
.... Pass ☐

(Make careful observation of door state flags during this test)

Start with the door closed initially, power down the EGM, open the door, power up the EGM and expect to see logged a door open event and a power off door access event (if applicable) .... Pass ☐

Check all doors individually .... Pass ☐
The above test also applies to note stacker removal .... N/A ☐ Pass ☐
372   PC: Verify in source code .... Pass ☐
373   PC: LP .... N/A ☐ Pass ☐
374   For all check both power up & CT & 3 NAKS .... Pass ☐
For exactly 3 NAKS on a QPV1.6 EGM it is normal (acceptable) to see one additional response before the machine stops responding. (this is because the EGM typically queues the response before processing the NAK)
375   Ensure QSIM's debug window is on and it is also configured to display General Status Polls/Resp for these tests below!

Stop sending broadcasts & power off/on the EGM and ensure the EGM does not resume responding
.... Pass ☐

While the EGM is powered off change the poll address, power on the EGM and ensure it responds only to the new pa. Look for poll address mismatch events or a resumed responding on the old poll address.
.... Pass ☐

Repeat the above two tests but for a communication time-out instead of a power down .. Pass ☐
Repeat the above two test again for a Comms. time-out but while the EGM is in test/audit/FC/double up .... Pass ☐
376   .... Pass ☐
377   *QPv1.5* Power the EGM off immediately after it sent an event response and observe no dupl. events (QPv1.5) otherwise a dup event is expected (QPv1.6) and that no pending events are not eventually transmitted .... Pass ☐

Repeat test for a communications timeout & 3 NAKs .... Pass ☐
378   .... Pass ☐
379   .... Pass ☐
380   PC: Multi-game & Extended Multi-game Support .... N/A ☐ Pass ☐
381   PC: Multi-game & Extended Multi-game Support .... N/A ☐ Pass ☐
382   PC: Multi-game .... N/A ☐ Pass ☐
383   PC: Multi-game & Extended Variation Support .... N/A ☐ Pass ☐
384   PC: HS. Ensure credit meter is zero and ensure the EGM Game configuration poll cannot be used to change variations .... N/A ☐ Pass ☐
385   PC: HS .... N/A ☐ Pass ☐
386   .... Pass ☐
387   PC: SAP .... N/A ☐ Pass ☐
388   PC: progressive .... N/A ☐ Pass ☐
389   PC: progressive .... N/A ☐ Pass ☐
390   PC: LP .... N/A ☐ Pass ☐
391   PC: LP & SAP .... N/A ☐ Pass ☐
392   PC: Progressive. For each item in the table, check labelling and that the displayed value actually represents the defined data .... N/A ☐ Pass ☐
393   PC: progressive .... N/A ☐ Pass ☐
394   PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
395   PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
396   PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
397   PC: CustomSAP .... N/A ☐ Pass ☐
398   PC: CustomSAP .... N/A ☐ Pass ☐
399   PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
400   PC: progressive .... N/A ☐ Pass ☐

401 PC: SAP. Current progressive amounts must always be reconcilable since RAM clear from data displayed in audit mode .... N/A ☐ Pass ☐
402 PC: Progressive .... N/A ☐ Pass ☐
403 PC: CustomSAP .... N/A ☐ Pass ☐
404 PC: SAP .... N/A ☐ Pass ☐
405 PC: SAP .... N/A ☐ Pass ☐
406 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
407 PC: SAP .... N/A ☐ Pass ☐
408 PC: SAP .... N/A ☐ Pass ☐
409 PC: SAP .... N/A ☐ Pass ☐
410 PC: SAP .... N/A ☐ Pass ☐
411 PC: SAP .... N/A ☐ Pass ☐
412 PC: LP .... N/A ☐ Pass ☐
413 PC: LP .... N/A ☐ Pass ☐
414 PC: LP .... N/A ☐ Pass ☐
415 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
416 PC: LP .... N/A ☐ Pass ☐
417 PC: LP .... N/A ☐ Pass ☐
418 PC: LP. Check after power fail, general reset & key .... N/A ☐ Pass ☐
419 PC: LP .... N/A ☐ Pass ☐
420 PC: LP .... N/A ☐ Pass ☐
421 PC: LP .... N/A ☐ Pass ☐
Setup QSIM's so there are LP broadcasts for the group under test at least every second (i.e. increase the poll cycle rate & disable other LP groups). Turn on auto-increments for the LP group under test so there exists some level of increment in the current amounts each LP broadcast. Then after the next LP hit, ensure the EGM is consistent with the amount of the award during a LP win wrt the LP Award Event and what is displayed during the LP lockup .... N/A ☐ Pass ☐
422 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
423 PC: LP Test functionality re ignoring General Reset Polls during LP win shows .... N/A ☐ Pass ☐
Also ensure that a fault during a LP win show is still clearable by a General Reset Poll .... N/A ☐ Pass ☐
424 PC: LP .... N/A ☐ Pass ☐
425 PC: LP .... N/A ☐ Pass ☐
426 PC: LP .... N/A ☐ Pass ☐
427 PC: Multi-game & Progressive .... N/A ☐ Pass ☐
428 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
429 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
430 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
431 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
432 PC: Multi-game & Progressive & SP. Check for each poll type mentioned that the configuration is carried over to all games. .... N/A ☐ Pass ☐
Specifically test with non-zero CAMT's (try a different value for each game and ensure the EGM only processes the first one) care should be taken by the EGM to ensure it does not process an initial SAP contribution (CAMT) more than once per game to avoid doubling up on initial contributions
.... N/A ☐ Pass ☐
433 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
434 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
435 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
436 Queue a program sig. request poll while the EGM was in the process of calculating a signature and ensure the correct result is still returned .... Pass ☐
Repeat this test a number of times .... Pass ☐

Repeat the above tests but also change the seed each time. To ensure the correct result was obtained re-request it to confirm the same result .... Pass ☐

Repeat the above tests but do not send a seed in the sig. Request polls .... Pass ☐
437 .... Pass ☐

438 Verify this while the EGM is under heavy load e.g. playing games, hopper collect or the EGM real time display in audit mode. Check all EGM modes .... Pass ☐
439 While the EGM is continually calculating a signature verify:

Coin in/out handling is unaffected (1000+ coin in/out test) .... Pass ☐
Communication is unaffected (check response times & time-outs) .... Pass ☐
Doors are still monitored, ... Pass ☐
All peripherals are still monitored for faults, ... Pass ☐
Check faults, lockup and audit mode (esp. real time displays) cause no side effects .... Pass ☐
Play the EGM for a number of hours an note any problems .... N/A ☐ Pass ☐
Repeat the above tests while constantly requesting EGM group meters & configuration (QSIM special function) .... Pass ☐
440 .... N/A ☐ Pass ☐
441 Confirm size and contents (refer separate std here) of program image used in the calculation are acceptable. **This checkpoint must be signed off by a Principal Technical Officer (QCOM)**. .... Pass ☐
442 XOR is utilised for combining images. .... Pass ☐
443 Confirm an acceptable signature algorithm and resolution was used .... Pass ☐
444 Every play commencement .... Pass ☐
445 PC: Free games... Pass ☐
446 PC: Hopper .... N/A ☐ Pass ☐
447 Every play commencement & residual gamble .... Pass ☐
448 PC: Hopper .... N/A ☐ Pass ☐
449 Upon confirmed winnings (i.e. after double-up/gamble), residual gamble wins & other wins depending on the game .... Pass ☐
450 Upon exit from CC lockup & residual CC lockup .... Pass ☐
451 PC: TO .... Pass ☐
452 PC: TO .... Pass ☐
453 PC: TO .... N/A ☐ Pass ☐
454 .... Pass ☐
455 PC: TI.. Pass ☐
456 .... Pass ☐
457 PC: TO .... Pass ☐
458 PC: TO .... Pass ☐
459 PC: SAP... Pass ☐
460 PC: LP.. N/A ☐ Pass ☐
461 PC: TI... N/A ☐ Pass ☐
462 .... N/A ☐ Pass ☐
463 Queued for every winning play (also ensure winning games whose wins were completely lost in gamble are not counted) .... Pass ☐
464 PC: Hopper .... N/A ☐ Pass ☐
465 PC: SAP... Pass ☐
466 When all meters are non-zero reconcile group 0 meters with the EGM current credit meter (QSIM display function provided)... Pass ☐

Confirm an auto-queued movement in each of the following meters as defined:
467 PC: Hopper. Upon a refill being performed .... Pass ☐
468 .... Pass ☐
469 .... Pass ☐
470 .... Pass ☐
471 .... Pass ☐
472 .... Pass ☐
473 PC: NA .... Pass ☐
474 PC: NA & TI .... Pass ☐
475 PC: NA .... Pass ☐
476 PC: NA & TI .... Pass ☐
477 PC: Hopper .... Pass ☐
478 PC: Hopper .... Pass ☐
479 PC: NA .... Pass ☐
480 PC: NA & TI .... Pass ☐

481 .... Pass ☐

482 .... Pass ☐

483 PC: NA & TI .... Pass ☐

484 PC: NA .... Pass ☐

485 PC: NA .... Pass ☐

486 PC: NA .... Pass ☐

487 PC: NA .... Pass ☐

488 PC: NA .... Pass ☐

489 .... Pass ☐

When all in use group meters have a non-zero value, reconcile with EGM audit mode display
(QSIM special function).. Pass ☐

490 PC: Progressive .... Pass ☐

491 .... Pass ☐

492 Check wrt the PID meter... Pass ☐

493 .... Pass ☐

494 With no other activity on the EGM or communications issues, just one meter packet should
result for this example and it must contain just these meters in order .... Pass ☐

495 As above .... Pass ☐

496 As above .... Pass ☐

497 As above .... Pass ☐

498 As above .... Pass ☐

499 PC: Hopper. As above .... Pass ☐

500 As above .... Pass ☐

501 PC: Hopper. As above .... Pass ☐

502 As above .... Pass ☐

503 As above .... Pass ☐

504 As above .... Pass ☐

505 As above .... Pass ☐

506 As above .... Pass ☐

507 As above .... Pass ☐

508 Check all remaining group meters for movement and queuing at the correct times .... Pass ☐

509 As above .... Pass ☐

510 PC: Verify in source code. Confirm for all QCOM group meter updates in EGM code. Refer
meter table for all update relationships.. Pass ☐

511 PC: Emulator.. Pass ☐

Option: setting MAXECT to -1 at EGM configuration (no limit) allows easy rollover testing of a
number of ECT-to-EGM related meters.. ☐

512 PC: End of Eval. During the course of the evaluation look for invalid meter increment or
backwards meter events.. Pass ☐

513 Using QCOM ECT, run up the EGM's credit meter to its highest possible value (RAM clear and
increasing MAXECT will make this a lot quicker, but even at multiples of $9999.99 it only takes
approximately 30 mins). Record the limit below:

Ensure the EGM logs an appropriate events (formerly RTEXT but as of QPV1.6.6 there is
dedicated event 0x205D) when it reaches its limit and denies an ECT (similar to ECT-to-EGM
credit meter display limit denials – refer 16.1.1) .... Pass ☐

514 PC: Verify in source code or Emulator.. Pass ☐

515 PC: SAP & Verify in source code or Emulator... Pass ☐

516 Transfer some cashless to the EGM while is in a lockup or fault condition (the EGM should not
add the amount to the credit meter...until return to idle mode)) and power the EGM down/up a
self audit error does not occur .... Pass ☐

517 PC: Emulator.. Pass ☐

518 Ensure the EGM does not go into RAM error during any CM limit related test at any stage.
Large or many incremental transfers of ECT or TI must never result in a RAM error ....
.... Pass ☐

519 If a characteristic of the EGM's production physical layer is that when one EGM transmits out of
turn it could clobber communications of another device/EGM on the network, the following test
must be performed (Applicable to QLD & NZ):

Ensure the EGM does not prevent or clobber communications during power up or down for a any significant period*... N/A ☐ Pass ☐

*If any clobbering is observed it must be reported as an issue, if unable to be improved or eliminated, dispensations may be considered so as long as the maximum amount of time an EGM may clobber a loop is well less than the maximum possible QCOM poll cycle rate x 2 (i.e. if an EGM disrupts three poll cycles in a row, then all EGM will get disabled until they return a new program hash)

To use QSIM to very roughly measure a period of clobbering: Set QSIM to loop mode (if not already), set "No Poll" ON for all EGMs, turn auto-queue OFF and set the poll cycle to 60msec. Then observe any burst increments of the "Echo Bad" meter on QSIM's F11 display during EGM power up/down sequences. Each burst increment of the meter potentially represents an clobber of 60msec +/- 60msec per increment. If a more accurate measurement is required (e.g. for borderline cases) a Digital-CRO should be used.

520 CharErr field on QSIM under normal operation* must remain at zero (* at least one test in this checklist may inflate this figure)... Pass ☐

521 PC: Verify in source code .... Pass ☐

522 Ensure QSIM is also polling IGT Protocol EGMs (QSIM will perform an LCOMSIM poll cycle by default, hit CTRL-TAB to switch to the LCOMSIM user interface and hit the plus key to add some machines to the poll cycle) and this will generate many framing errors on QCOM EGMs, the EGM should be unaffected (worse case you may see the EGM audit mode CRC error increasing)
  .... Pass ☐

523 Verify in EGM audit mode OVERRUN is always zero .... Pass ☐

PC: Verify in source code. Overrun error counting is implemented .... Pass ☐

524 PC: DSO, CRO or Protocol Analyser required .... Pass ☐
Also verify bits are being transmitted with a 50% duty cycle (easily done if the EGMs poll address is 0x01). Un-calibrated measurement.. Pass ☐

525 When a poll requests a specific response from an EGM, that response must not be received until the response to the following poll... Pass ☐
With an enabled game, send a Gen. Maint Poll requesting the game cfg. Then the very next poll to the EGM, send another Gen. Maint Poll but with GEF= 0, the response to that poll should be an EGM Game Cfg. Response with GEF = 1 not 0 .... Pass ☐
With auto-queue off, send exactly three consecutive NAKs to the EGM while watching the debug window (ensure the display of GSR's is turned on in the debug window). Three corrupt response should be observed (this is normal as this is how QSIM forces a NAK to be sent). Then for a QCOM v1.6 EGM you should see once more response then the EGM will stop responding, for a QCOM v1.5 EGM, the EGM will just start responding again.................. Pass ☐

526 A good way to pickup an issue here is to test the machine's global NAK implementation at this point – re if the machine one (incorrect) or two (correct) global NAKs for a single missed or corrupt broadcast message.

527 .... Pass ☐

528 PC: Verify in source code .... Pass ☐

529 Each of the following tests should be run over a period of at least a couple of hours while the EGM is in play:

Verify the EGM does not miss responses when it is the first polled EGM in the poll cycle and the poll cycle is running at 100% (i.e. Increase the total no. of EGMs being polled until the poll status window says the poll cycle duty is > 100% and disable the LCOMSIM poll cycle). Look for any error counts increasing over time on the Data Link Layer display.......................... Pass ☐

Verify the EGM does not miss processing broadcast messages or polls when it is the last EGM in the poll cycle (look for the global NAK counter on QSIM increasing, provided the EGM is correctly NAKing broadcast messages).. Pass ☐
PC: QPv1.6 Ensure the EGM can handle the max poll cycle rate .... Pass ☐

530   Set QSIM PC time to just under 250msec and ensure the EGM does not miss responses, especially polls & responses requiring significant processing such as meter responses or Purge Polls when the event queue was full (excemption may be granted if no more than 2 responses in a row are missed upon purging a full event queue).. Pass ☐
With a cycle of 250msec, the game enabled (GEF=1), send a General Maintenance Poll with GEF=0 & Game Cfg. Request bit = 1 and ensure the game cfg. response reflects the GEF=0 .... Pass ☐
Also with a cycle of 250msec and zero credit & Cmode = 0 in idle mode, send an ECT amount with Cmode =1 immediately followed by a ECT Lockup request poll, the EGM should end up in an ECT-from-EGM lockup .... Pass ☐

531   Add an EGM to QSIM and try setting it to a poll address that is a multiple of 128 of the EGM under test's current address (eg. If EGM pa=2 then try 130).
Ensure the EGM does not try to respond .... Pass ☐
Add 64 EGMs to QSIM and ensure the EGM only responds to its own pa .... Pass ☐

532   Refer QSIM corrupt CRC special function .... Pass ☐
533   Look for response timeout events or the QSIM NONRESP counter increasing................. Pass ☐
534   The EGM must not respond to polls with invalid CRCs... Pass ☐
535   Verify each responses priority wrt the next lower priority response (or vice versa) .... Pass ☐

For just two responses verify their respective priorities when NAKs are constantly being sent (eg. Send an EGM config request with bit 6 set then immediately constantly NAK the EGM, the EGM should alternate responses between the EGM config response and the EGM Game Config response. Ensure debug mode display with garbage is turned on) .... Pass ☐

Kick the EGM (QSIM special function) while is playing games and ensure no error events are logged .... Pass ☐

536   Test all response types .... Pass ☐
537   Test for group meters responses... Pass ☐
538   Send a poll msg which causes the EGM to carry out some obvious function (such as a meter request) but with an invalid CRC (QSIM 'corrupt poll' special function), ensure the EGM does not process the poll message (note however, QSIM will automatically resend the message so just look at the response timings) .... Pass ☐

Also check the EGM does not process global messages which have an invalid CRC ...... Pass ☐
539   .... Pass ☐
540   Look for message frame timeout events and observe LGAP. If the time is borderline use a DSO or Protocol analyser. Enter average gap _____ .... Pass ☐

Ensure the EGM does not apply an inter-character timeout QSIM special function. .... Pass ☐
541   There is a special QSIM function which will interrupt an EGM response part way through by sending an 0xff (try with and without parity set). While watching the hex data responses in debug mode ensure the response is aborted at least when the parity bit is set (do it on group 1 meter responses as they are longer). .... Pass ☐
Make the EGM last in the poll cycle and then use the Send Broadcast Early function which interrupts the next EGMs response with a broadcast message. Ensure the EGM receives the broadcast and a broadcast NAK should not result .... Pass ☐
Also, there is another QSIM special function which appends additional bytes to a poll. As above .... Pass ☐
542   (Use QSIM custom poll function) .... Pass ☐
543   .... Pass ☐
544   No ill side effects observed .... Pass ☐
545   Ensure responses such as a meter or event response is not resent if a custom poll FC is sent following its response.. Pass ☐
546   Use QSIM special function (QSIM v161.5+ special function EGM->More->Polling->Extend/Crop) to add bytes to a poll before the CRC, check up to a LEN field of 0x9E for QPv1.5 EGM and 0xFF for QPv1.6 EMGs .... Pass ☐

Check up to maximum poll message length (160 bytes minimum should be supported on QPv1.5 EGMs or as high as the EGM will go with no adverse side effects such as RAM corruption. No responses over 160 bytes is normal for QPv1.5 EGMs)............................. Pass ☐

Verify adding bytes after the CRC is handled by EGM…
PC: Verify in source code (EGM low level message rx state machine) .... Pass ☐

547   Test how the EGM handles purposely shortened poll messages. I.e. valid FC & CRC but overall poll message data is too short wrt the FC. ~~The~~ It must not corrupt memory in the EGM (e.g. causing a RAM error or crash), however at this time it is not a fail for the EGM to accept garbage for fields that were cropped as a result of the shortened message. E.g. shorten a Parameter Poll by 4 bytes and the EGM will most likely still accept a garbage value for the last field in the poll as there are no other validation checks on this field.
(QSIM v161.5+ special function EGM->More->Polling->Extend/Crop) .... Pass ☐

548   QSIM v161.5+ special function EGM->More->Polling->Extend/Crop, to extend a poll message after the CRC, be careful however, as the results can be confusing as it depends on whether QSIM is in P2P mode, or loop mode as what will appear to occur in QSIM. The EGM may either stop resonding or simply ignore the extra bytes depending how many there are. Both are fine so long as nothing internal to the EGM can be corrupted by adding any number of bytes after the CRC (e.g. buffer overrun, corruption, comms process hanging).......................... Pass ☐

549   Rapid Polling.
Configure QSIM for only the EGM under test and disable lcomsim polling, lower the poll cycle time to 0 msec and observe. RAM errors, corrupt responses, or CRC Errors (also check for CRC errors increasing in EGM audit mode display) are a fail, missed responses is normal behaviour. Run this test for at least 5 minutes, play some games .... Pass ☐

Request the event buffer a number of times and ensure the same EVTCRC is obtained for each request
 .... Pass ☐
Retest the above & send CRC errors & NAKs .... Pass ☐
Request events then send a purge all almost straight away, verify the EGM purged only ACKed events (by requesting all events again) .... Pass ☐

What is the lowest poll cycle time the EGM can handle before problems occur? _____msec

550   EGM must not respond to cropped messages as they do not have a valid CRC. (QSIM v161.5+ special function EGM->More->Polling->Extend/Crop) Start with -1, **send one cropped message only,** quickly return poll to uncropped and verify EGM's comms process does not hang. Work backwards until the poll address is all that being sent .... Pass ☐
The EGM must ignore all cropped polls and automatically resume responding once they are back to normal length.. Pass ☐

551   PC: Verify in source code .... Pass ☐
552   PC: Verify in source code .... Pass ☐
553   PC: Verify in source code .... Pass ☐
554   PC: Verify in source code .... Pass ☐
555   PC: Verify in source code .... Pass ☐
556   Ensure the EGM responds on PAs 1, 128 & 254 .... Pass ☐
Once set with a valid PA ensure the EGM does also try to not respond on PAs 0 or 255 . Pass ☐
557   PC: Verify in source code .... Pass ☐
Also verify by shortening/extending polls to the minimum/maximum length (QSIM special function) and verifying that the EGM still responds ok .... Pass ☐

558   PC: Verify in source code .... Pass ☐
559   .... Pass ☐
560   .... Pass ☐
561   PC: Verify in source code .... Pass ☐
562   Play games while NAKing meter responses, observing the debug window the meter data should change
 .... Pass ☐
563   Send a forced NAK for every response type and ensure it is re-transmitted .... Pass ☐

564 To test, slow QSIM's poll cycle down to 3 secs to allow time to complete action required. Select a poll such as General Maint. Poll. With MEF = enabled, send the poll and use the QSIM special function to make QSIM force NAK the response. QSIM will resend the poll, but before it does change the MEF to disable. If the EGM is processing the resent poll, the EGM will disable (watch for comms disable display on EGM) = *fail*, if it isn't, the EGM wont disable upon receipt of the resent poll = *pass*...Pass ☐

565 Ensure that the EGM always processes the 1st poll message data in a session regardless of the state of the ACK/NAK bit ....Pass ☐

566 Ensure if the first response was an event response that if NAK'd it is resent ....Pass ☐

567 PC: Verify in source code ....Pass ☐

568 Test by powering down the EGM immediately after an event response and ensure it is not inadvertently re-transmitted. Test when the last value of bit 7 was a zero and a one.......Pass ☐

569 A broadcast NAK must result from a suppress broadcast, corrupt broadcast....................Pass ☐
(QCOM's DLL requirements mean QSIM should receive two global NAKs for a single corrupt or missing global message. This is a correct implementation. If the machine only returns one, then the machine likely has an issue either with its global NAK implementation or its DLL implementation.)
An invalid broadcast function code should not result in a broadcast NAK ....Pass ☐
Two consecutive broadcast messages should not result in a broadcast NAK (Use QSIM function 'No Poll' when EGM under test is the only EGM on QSIM) ....Pass ☐

570 ....Pass ☐

Ensure if any response with the global NAK bit set is NAKed by QSIM, that the EGM keeps the bit set until the response is ACKed...Pass ☐
(To test the above, corrupt a global for a single poll cycle, then NAK the response with the global NAK twice in a row, stop NAKing and the broadcast NAK counter should then increase by two counts.)
Ensure the global NAK counter does not increment over a number of days......................Pass ☐

571 ....Pass ☐

Additionally perform the following procedure (and all possible variants):

1) Corrupt Global Messages (QSIM special function, bind this command to a hot key)
2) Wait for the EGM to start sending global NAKs
3) Start NAKing the EGM's responses (QSIM special function, bind this command to a hot key)
4) While still NAKing the EGM stop corrupting broadcasts
5) Wait a few seconds then stop NAKing EGM

QSIM should then receive at least one more global NAK ....Pass ☐

572 ....Pass ☐

573 PC: multi-game .... N/A ☐ Pass ☐

574 PC: ALL. BSVN unique (as applicable)? A common mistake is for the manufacturer to update the base and forget to update the BSVN...Pass ☐

575 ....Pass ☐
PC: Multi-game. Ensure the BSVN is not equal to any GVN...Pass ☐

576 ....Pass ☐

577 PC: Verify in source code ....Pass ☐

578 Separate & NV for poll type ....Pass ☐

579 For all poll message types using PSNs check:

The EGM ignores the data if the PSN is out of sequence (specifically check duplicate PSN) ....Pass ☐

Go back to the correct next expected PSN and verify the EGM accepts the poll message...Pass ☐

The EGM accepts all messages over the PSN rollover..Pass ☐

580 Check both poll types...Pass ☐

581 ....Pass ☐

582 ....Pass ☐

583 Ensure if set these bits have no effect.. Pass ☐
584 Confirm operation .... Pass ☐
585 .... Pass ☐
586 .... Pass ☐
587 Ensure all PSNs were not at their default values first, then test operation......................... Pass ☐
588 .... Pass ☐
589 PC: RAM clear (stage 2) .... Pass ☐
590 PC: RAM clear (stage 2) .... Pass ☐
591 PC: RAM clear (stage 2) .... Pass ☐
592 Check that the EGM wont accept a change to each field individually .... Pass ☐
593 PC: Verify in source code .... Pass ☐
594 PC: Verify in source code .... Pass ☐
595 PC: Verify in source code. Get the manufacturer to reference all uses and record in the QCOM
    status spreadsheet... Pass ☐
596 PC: Verify in source code .... Pass ☐
597 test unreasonable values .... Pass ☐
    Test the EGM cash operations (eg. ECT in/out, coin in/out, CRanE & RCRF) for at least one
    other DEN/TOK settings other than 1c & 1$. E.g. 2c & $5 .... Pass ☐
598 .... Pass ☐
599 .... Pass ☐
600 .... Pass ☐
601 PC: Hopper .... Pass ☐
602 PC: RAM clear (stage 2) .... Pass ☐
603 test unreasonable values .... Pass ☐
    Test the EGM cash operations (eg. ECT in/out, coin in/out, CRanE & RCRF) for at least one
    other DEN/TOK settings other than 1c & 1$. E.g. 2c & $5 .... Pass ☐
604 .... Pass ☐
605 .... Pass ☐
606 .... Pass ☐
607 PC: Hopper. Verify empirically the RTP of the RCRF .... N/A ☐ Pass ☐
608 .... Pass ☐
609 .... Pass ☐
610 .... Pass ☐
611 Lower and confirm no play exceeds MAXBET .... Pass ☐
612 .... Pass ☐
613 .... Pass ☐
614 .... Pass ☐
615 .... Pass ☐
616 PC: RAM clear (stage 3) .... Pass ☐
617 .... Pass ☐
618 PC: Verify in source code .... Pass ☐
619 PC: RAM clear (stage 3) & HS .... Pass ☐
620 PC: RAM clear (stage 3) .... Pass ☐
621 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
622 PC: RAM clear (stage 3) & LP .... N/A ☐ Pass ☐
623 PC: RAM clear (stage 3) .... Pass ☐
624 PC: RAM clear (stage 3) .... Pass ☐
625 PC: Verify in source code .... Pass ☐
626 PC: RAM clear (stage 3) .... N/A ☐ Pass ☐
627 PC: Verify in source code. If the EGM uses this field to process repeated entries in the
    message, ensure that the EGM firstly validates this field to ensure that a malformed message
    cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with
    total message length).. N/A ☐ Pass ☐
628 PC: Verify in source code .... Pass ☐
629 PC: RAM clear (stage 3) & LP .... N/A ☐ Pass ☐
630 PC: RAM clear (stage 3) & SAP .... N/A ☐ Pass ☐
631 PC: RAM clear (stage 3) & Progressive .... N/A ☐ Pass ☐
632 PC: RAM clear (stage 3) & SAP .... N/A ☐ Pass ☐
633 PC: RAM clear (stage 3) &SAP .... N/A ☐ Pass ☐
    Check (-1) i.e. very large initial SAP contribution .... N/A ☐ Pass ☐

634   PC: RAM clear (stage 3) & SAP .... N/A ☐ Pass ☐
635   PC: RAM clear (stage 3) & SAP .... N/A ☐ Pass ☐
636   PC: RAM clear (stage 3) & LP .... N/A ☐ Pass ☐
637   PC: RAM clear (stage 3) .... Pass ☐
638   PC: RAM clear (stage 2) .... Pass ☐
639   PC: RAM clear (stage 3) .... Pass ☐
640   PC: RAM clear (stage 3) .... Pass ☐
641   PC: RAM clear (stage 3) .... Pass ☐
642   PC: RAM clear (stage 3) .... Pass ☐
643   PC: RAM clear (stage 3) .... Pass ☐
644   .... Pass ☐
645   PC: HS. Check basic operation .... N/A ☐ Pass ☐
646   PC: LP. Check basic operation... N/A ☐ Pass ☐
647   .... N/A ☐ Pass ☐
648   .... N/A ☐ Pass ☐
649   PC: RAM clear (stage 3), HS or LP .... N/A ☐ Pass ☐
650   PC: HS or LP .... N/A ☐ Pass ☐
      Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
651   PC: HS or LP .... N/A ☐ Pass ☐
652   PC: HS or LP .... N/A ☐ Pass ☐
653   .... Pass ☐
654   PC: HS .... N/A ☐ Pass ☐
655   PC: HS .... N/A ☐ Pass ☐
656   PC: HS .... N/A ☐ Pass ☐
      Also ensure there are no ill side effects if a VAR change is sent while the machine is in the PID
      or other idle mode variants .... N/A ☐ Pass ☐
657   PC: HS .... N/A ☐ Pass ☐
658   PC: HS .... N/A ☐ Pass ☐
659   PC: HS .... N/A ☐ Pass ☐
660   Verify operation .... N/A ☐ Pass ☐
661   PC: LP. Verify operation .... N/A ☐ Pass ☐
662   .... N/A ☐ Pass ☐
663   PC: LP.. N/A ☐ Pass ☐
664   PC: LP.. N/A ☐ Pass ☐
665   PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐
666   PC: HS or LP. Ensure the event is logged whenever the poll is ignored for any listed reason
      .... N/A ☐ Pass ☐
667   .... Pass ☐
      Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
668   .... N/A ☐ Pass ☐
669   .... N/A ☐ Pass ☐
670   .... N/A ☐ Pass ☐
671   Check features autoplay .... N/A ☐ Pass ☐
      PC: LP specifically test autoplay continues after a LP award .... N/A ☐ Pass ☐
672   .... Pass ☐
673   Lower and confirm operation .... Pass ☐

      Ensure 2^32-1 (enter -1 in QSIM) is supported and has no ill side effects such as causing
      inadvertent lockup for small wins... Pass ☐
      PC: Free Games .... Pass ☐
674   PC: SAP.. Pass ☐
675   .... Pass ☐
676   .... Pass ☐
677   .... Pass ☐
678   PC: CEO (Store a screenshot on file if ok) .... Pass ☐
679   Lower and confirm operation (a lockout message should be displayed as per other OLGR
      requirements)
      .... Pass ☐
      Specifically test a value of 0, as this amount may be used to disable further credit input near the
      end of licensed gaming hours .... Pass ☐

680 .... Pass ☐
681 .... Pass ☐
682 PC: TI... Pass ☐
    If peforming this checkpoint as a part of a PARTIAL QCOM evaluation, then to be safe, also
    test "CRLIMIT" functionality in full .... Pass ☐
683 Lower and confirm operation .... Pass ☐
684 Confirm maximum value works correctly .... Pass ☐
685 .... Pass ☐
686 Set DUMAX to zero & reserved all to 1 & confirm DU is disabled (enter 0xf0) .... Pass ☐
687 Lower and confirm operation .... Pass ☐
    Check a value of zero disables double up .... Pass ☐

    Ensure 2^32-1 (enter -1 in QSIM) is supported and has no ill side effects .... Pass ☐
688 PC: QPv1.6. Ensure all displayed times are correctly adjusted by TZADJ check +/- .... Pass ☐
689 PC: QPv1.6 Ensure a subsequent change of TZADJ does not affect all previsouly logged time-
    stamps
    .... Pass ☐
690 PC: QPv1.6 .... Pass ☐
691 PC: QPv1.6 .... Pass ☐
692 PC: QPv1.6 .... Pass ☐
693 .... Pass ☐
694 .... Pass ☐
695 .... Pass ☐
696 .... Pass ☐
697 When PWRTIME=0 in power-save:
    Also ensure pressing a button on the EGM or a SEF 0 to 1 transition doesn't cause a split
    second exit from power-save... Pass ☐
698 When PWRTIME=0 in power-save test each mode for temporary power-save exit .... Pass ☐
699 .... Pass ☐
700 Specifically check when SEF=1 in power-save (now possible in QPv1.6.2)...................... Pass ☐
701 PC: PID .... N/A ☐ Pass ☐
702 .... Pass ☐
703 .... Pass ☐
704 Lower and confirm operation (try 0, 10c, $1, $5, $75, $125…*)...................................... Pass ☐
    *Ensure every type of payment method is tested, from ECT to TO (as applicable)

    Ensure 2^32-1 (enter '-1' in QSIM) is supported and has no ill side effects such as causing
    inadvertent payouts of small wins.. Pass ☐
    PC: Free Games .... Pass ☐
705 .... Pass ☐
706 PC: SAP Lower and confirm operation (try 0, 10c, $1, $5, $75, $125…*) .... N/A ☐ Pass ☐
    *Ensure every type of payment method is tested, from ECT to TO (as applicable)

    Ensure 2^32-1 (enter '-1' in QSIM) is supported and has no ill side effects such as causing
    inadvertent payouts of small SAP wins.. N/A ☐ Pass ☐
707 PC: SAP.. N/A ☐ Pass ☐
708 PC: SAP.. N/A ☐ Pass ☐
709 PC: SAP.. N/A ☐ Pass ☐
710 .... Pass ☐
711 .... Pass ☐
712 .... Pass ☐
713 PC: Progressive .... N/A ☐ Pass ☐
714 PC: SAP.. N/A ☐ Pass ☐
715 PC: CustomSAP .... N/A ☐ Pass ☐
716 PC: CustomSAP & RAM clear (stage 3), HS... N/A ☐ Pass ☐
717 PC: LP & RAM clear (stage 3), HS .... N/A ☐ Pass ☐
718 PC: Progressive .... N/A ☐ Pass ☐
719 PC: Progressive .... N/A ☐ Pass ☐
720 PC: Verify in source code .... Pass ☐

721 PC: Verify in source code. If the EGM uses this field to process repeated entries in the message, ensure that the EGM firstly validates this field to ensure that a malformed message cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with total message length).. N/A ☐ Pass ☐

722 PC: Progressive .... Pass ☐

723 PC: Progressive .... N/A ☐ Pass ☐

724 PC: Progressive .... N/A ☐ Pass ☐

725 PC: Progressive .... N/A ☐ Pass ☐

726 PC: Progressive .... N/A ☐ Pass ☐

727 PC: Progressive .... N/A ☐ Pass ☐

728 PC: Progressive. Ensure PINC = 0 is handled & does not cause a Divide by Zero Error
.... N/A ☐ Pass ☐

729 PC: Progressive .... N/A ☐ Pass ☐

730 PC: Progressive .... N/A ☐ Pass ☐

731 PC: LP or CustomSAP.. N/A ☐ Pass ☐

732 PC: CustomSAP .... N/A ☐ Pass ☐

733 PC: CustomSAP .... N/A ☐ Pass ☐

734 PC: Multi-game & Progressive & SP .... N/A ☐ Pass ☐

735 PC: Verify in source code .... Pass ☐

736 .... Pass ☐

737 .... Pass ☐

738 .... Pass ☐

739 PC: Verify in source code .... Pass ☐

740 .... Pass ☐

741 .... N/A ☐ Pass ☐

742 .... N/A ☐ Pass ☐

743 .... N/A ☐ Pass ☐

744 .... N/A ☐ Pass ☐

745 .... N/A ☐ Pass ☐

746 PC: CEO (Store a screenshot on file if ok) .... N/A ☐ Pass ☐

747 .... N/A ☐ Pass ☐

748 PC: Verify in source code. If the EGM uses this field to process repeated entries in the message, ensure that the EGM firstly validates this field to ensure that a malformed message cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with total message length).. N/A ☐ Pass ☐

749 Verify with display bit = 1 in idle mode and in audit mode... Pass ☐

750 .... Pass ☐

751 .... Pass ☐

752 Test 0,1,2,3 & 8 levels .... Pass ☐
Note there is no lev ordering specified, verify the EGM can handle any type or level ordering and non-consecutive lev numbers .... Pass ☐
Test duplicate lev handling. no special handling of duplicate entries is required and the EGM is not required to correctly display all duplicate entries, however the EGM must not crash, corrupt memory, or overrun any buffers if one or more duplicate entries are sent to the EGM ..... Pass ☐

753 PC: Verify in source code, ie masked out... Pass ☐

754 .... N/A ☐ Pass ☐

755 .... N/A ☐ Pass ☐

756 .... N/A ☐ Pass ☐

757 .... N/A ☐ Pass ☐

758 Ensure the machine can correctly display the full range of printable chars .... Pass ☐

759 Verify max length string is displayed correctly (NB. If the duisplay font is proportional, make sure the string is filled with FATTER characters) .... Pass ☐

760 PC: Verify in source code .... Pass ☐

761 .... Pass ☐

762 If told to display the level (via the display flag), the EGM should just display the amount only on the main display .... Pass ☐

763 .... Pass ☐

764 .... Pass ☐

765 .... Pass ☐

766 .... Pass ☐

767 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
768 .... Pass ☐
769 .... Pass ☐
770 .... Pass ☐
771 Confirm operation .... Pass ☐
772 Ensure setting unused bits to 1 on the unused bits and sending does not affect the signature
result
.... Pass ☐
773 .... Pass ☐
Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
774 Verify it is queued when sent to the EGM while it is in all EGM states and a number of other
modes including all game modes (play, feature double up, rules display etc) .... Pass ☐

Even after a power fail .... Pass ☐
Ensure any CDC doesn't delay a SL .... Pass ☐
775 .... Pass ☐
776 .... Pass ☐
777 Ensure the EGM returns to the lock up after a power fail.................................... Pass ☐
778 .... Pass ☐
779 .... Pass ☐
780 .... Pass ☐
781 .... Pass ☐
782 Confirm setting these bits (enter 0x7f in QSIM) and sending has no ill effect.................... Pass ☐
783 .... Pass ☐
784 .... Pass ☐
785 .... Pass ☐
786 .... Pass ☐
787 .... Pass ☐
788 .... N/A ☐ Pass ☐
789 Confirm operation .... Pass ☐
790 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
791 Check lengths of 0,1,20,80 .... Pass ☐
792 .... Pass ☐
793 Ensure the machine can correctly display the full range of printable chars .... Pass ☐
794 PC: Verify in source code .... Pass ☐
795 .... N/A ☐ Pass ☐

Ensure a full length message with no spaces displays correctly (NB. If the duisplay font is
proportional, make sure the string is filled with FATTER characters) .... Pass ☐
796 .... Pass ☐
797 .... Pass ☐
798 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
799 .... Pass ☐
800 .... Pass ☐
801 PC: TO .... N/A ☐ Pass ☐
802 PC: TO .... N/A ☐ Pass ☐
803 PC: TO .... N/A ☐ Pass ☐
804 PC: TO .... N/A ☐ Pass ☐
805 PC: TO .... N/A ☐ Pass ☐
806 PC: TO .... N/A ☐ Pass ☐
807 PC: TO .... N/A ☐ Pass ☐
808 PC: TO .... N/A ☐ Pass ☐
809 PC: Verify in source code & TO.. N/A ☐ Pass ☐
810 PC: TO .... N/A ☐ Pass ☐
811 PC: TO .... N/A ☐ Pass ☐
812 PC: TO .... N/A ☐ Pass ☐
Try an AUTHNO too large for the given EGM's supported barcode format and ensure the EGM
correctly extracts least significant number of decimal digits it supports for its given ticket
barcoide format.. N/A ☐ Pass ☐

813 PC: TO Check lengths of 0,1,20,40,80 (NB. If the duisplay font is proportional, make sure the string is filled with FATTER characters)... N/A ☐ Pass ☐

814 PC: TO, Verify in source code .... N/A ☐ Pass ☐

815 PC: TO Ensure the machine can correctly display/print the full range of printable chars .. Pass ☐

816 PC: TO
PC: Verify in source code .... N/A ☐ Pass ☐

817 PC: TO .... N/A ☐ Pass ☐

818 PC: TO .... N/A ☐ Pass ☐

819 PC: TO .... N/A ☐ Pass ☐

820 PC: TO .... N/A ☐ Pass ☐

821 PC: TI... N/A ☐ Pass ☐

822 PC: TI... N/A ☐ Pass ☐

823 PC: TI... N/A ☐ Pass ☐

824 PC: TI Check TAMT of zero; TAMT > MAXECT; CM overflow (i.e. raise MAXECT) and use a very large TAMT   .... N/A ☐ Pass ☐

825 PC: TI... Pass ☐

826 PC: TI PC: Verify in source code .... N/A ☐ Pass ☐

827 PC: TI... N/A ☐ Pass ☐

828 PC: TI... N/A ☐ Pass ☐

829 PC: TI... N/A ☐ Pass ☐

830 PC: TI... N/A ☐ Pass ☐

831 Check with the Credit Meter at amounts of $0, $9999.99, $100, 1c, CM < 1 token value, 1 token < CM < hopper pay limit, etc .... Pass ☐

832 Check each .... Pass ☐

833 .... Pass ☐

Check after a power fail .... Pass ☐
With CM = 0 during play send CC lockup and ensure the EGM locks up if the play is a win
.... Pass ☐

834 .... Pass ☐

835 .... Pass ☐

836 .... Pass ☐

837 .... Pass ☐
Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐

838 Do a binary check on the above three bits and ensure the correct meters are received for each General Maintenance poll sent .... Pass ☐

839 Ensure these bits have no effect if set and sent... Pass ☐

840 .... Pass ☐

841 PC: RAM clear (stage 2). Ensure the EGM reports group meters even before EGM Configuration has taken place .... Pass ☐

842 PC: SAP or LP .... N/A ☐ Pass ☐

843 Request all meter groups in a single GMP (expected to see a meter group response every 2nd response with 8 meters in each except for the last response. There should be no timeout events logged by QSIM) .... Pass ☐

With the poll cycle @ 1000msec, then once per second for three* seconds, request a different meter group via the GMP (* three possible groups when the EGM has a note acceptor), ensure that all meters are received as expected and there are no timeout events logged by QSIM (should see 8 meters per response except for possibly the 1st and last group meters responses. Result should be very similar to the first test except the 1st meters response may not have 8 meters in it if group 0 also does not have 8 meters in use within it) .... Pass ☐

844 .... Pass ☐

845 .... Pass ☐

846 .... Pass ☐

847 .... Pass ☐

848 .... Pass ☐

849 .... Pass ☐

850 .... Pass ☐

851 PC: Verify in source code .... Pass ☐

852 Confirm basic operation of the following bits .... Pass ☐

853 .... Pass ☐

854 .... Pass ☐

855 PC: RAM clear (stage 3) & Progressive .... N/A ☐ Pass ☐

856 PC: RAM clear (stage 3) .... Pass ☐

857 PC: Progressive .... N/A ☐ Pass ☐

858 PC: RAM clear (stage 3) & Progressive .... N/A ☐ Pass ☐

859 .... N/A ☐ Pass ☐

860 PC: RAM clear (stage 3) .... Pass ☐

861 PC: RAM clear (stage 2 & 3) .... Pass ☐

862 .... Pass ☐

863 Use the EvtCRC to confirm the same event log is received .... Pass ☐

While receiving events send corrupt polls and force NAKs and ensure the same event log is received

.... Pass ☐

Also interrupt a resend with another request and ensure the same log is received .... Pass ☐

864 Perform this test twice once by sending the request when the EGMs response would be an event response and once when it would be a default response .... Pass ☐

Specifically send a request poll when the last received event was an extended event and ensure the correct log is received again .... Pass ☐

Send two RALE's in a row and observe for correct behaviour (Should get sequence: >RALE <GSR, >RALE <1st Event in Q, >GSP <GSR, >GSP <1st Evt in Q.) .... Pass ☐

865 Confirm setting this byte to 0xff has no effect... Pass ☐

866 For each test below start with a queue of about 10 events (the more the better) and use a request all logged events poll to confirm the results. You may wish to speed up the poll cycle while performing these tests.

With EVTNO at zero ensure all events are purged .... Pass ☐

With EVTNO set at the last event ensure all events are purged .... Pass ☐

Also using EVTNO...

Purge one event only .... Pass ☐

Purge half... Pass ☐

Purge all but one .... Pass ☐

Request events, part way through purge all and ensure the unsent events are still received and not purged

.... Pass ☐

Purge to a specific EVTNO over the EVTNO rollover and ensure the correct events are purged

.... Pass ☐

If the first event in the EGM's Q. is X then send a purge to X-1 and ensure all events are purged... Pass ☐

Send a purge all while the EGM was in the process of sending a number of queued events. Use a request all events poll to confirm that the EGM only purges events that have been sent and received an acknowledgement from QSIM... Pass ☐

Repeat the above purging to a specific event number within the unacked event range .... Pass ☐

867 .... Pass ☐

868 PC: NA. Confirm no effect if set... N/A ☐ Pass ☐

869 PC: NA. Confirm operation .... N/A ☐ Pass ☐

870 PC: TI & TO .... N/A ☐ Pass ☐

871 PC: NA. Confirm no effect if set... N/A ☐ Pass ☐

872 PC: NA .... N/A ☐ Pass ☐

873 PC: Hopper. Confirm correct operation with at least two other values .... N/A ☐ Pass ☐

874 PC: Hopper. Confirm correct operation with at least two other values .... N/A ☐ Pass ☐

Ensure a value of 0 disables all hopper payout including the RCRF .... N/A ☐ Pass ☐

Ensure a value of < = TOKDEN works correctly. I.e collect of < TOKDEN should instigate a RCRF initially, if won, a CC for the value of 1 token should result .... N/A ☐ Pass ☐

Ensure the EGM does not round this value... N/A ☐ Pass ☐

Ensure 2^32-1 (enter -1 in QSIM) works correctly & has no ill side effects .... N/A ☐ Pass ☐

875 PC: TO. As above .... N/A ☐ Pass ☐
876 PC: TO .... N/A ☐ Pass ☐
877 PC: Hopper. Confirm correct operation with at least two other values .... Pass ☐
878 .... Pass ☐

879 PC: LP. Ensure the condition cannot be cleared before receipt of this poll (check power fail)
    .... N/A ☐ Pass ☐
880 PC: LP. Ensure the command is not queued .... N/A ☐ Pass ☐
881 PC: LP.. Pass ☐

882 General Reset Poll... Pass ☐
883 Do binary test using two types of lockup conditions and two ways of clearing .... Pass ☐
884 Ensure this bits if set have no effect on both lockup types.. Pass ☐
885 Ensure this bits if set have no effect on both lockup types.. Pass ☐
886 .... Pass ☐
887 .... Pass ☐
888 .... Pass ☐
889 .... Pass ☐
890 .... Pass ☐
891 .... Pass ☐
892 .... Pass ☐

893 PC: TO .... N/A ☐ Pass ☐
894 PC: ECT.. Pass ☐
895 Check all can be cleared via this poll; either generally or via STATE. .... Pass ☐
896 PC: LP.. Pass ☐
897 SPAM Poll. Ensure these bits have no effect if set (ie. Inadvertent fanfare)..................... Pass ☐
898 Prominance Flag. Check functionality .... Pass ☐
    Ensure it is prominent only the game results display (ie check rules & PID displays for
    corruption. .... Pass ☐
    Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
899 .... Pass ☐
900 .... Pass ☐
901 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
902 Ensure the EGM can handle 0,1,20 & max length messages (NB. If the duisplay font is
    proportional, make sure the string is filled with FATTER characters) .... Pass ☐
903 .... Pass ☐
904 PC: Verify in source code .... Pass ☐
905 Ensure the machine can correctly display the full range of printable chars .... Pass ☐
906 PC: Verify in source code .... Pass ☐
907 .... Pass ☐
    Messages must be displayed during play, and especially features and free games .... Pass ☐
    Message must be displayed in all faults & lockups. Check only a couple of FCs, but specifically
    check all lockups, especially the System Lockup (refer Appendix A)................................. Pass ☐
908 .... Pass ☐
909 .... Pass ☐
910 .... Pass ☐
911 .... Pass ☐
912 .... N/A ☐ Pass ☐
913 .... Pass ☐
914 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
915 Queue all three messages at max length and ensure all are displayed............................. Pass ☐

916 PC: Tower light. Confirm operation.. N/A ☐ Pass ☐
917 PC: Tower light .... N/A ☐ Pass ☐

918 Broadcasts.
    Check QSIM custom broadcasts .... Pass ☐

Change the broadcast pa to the EGM under test's pa and observe. The EGM should just see this as an additional poll to its address but with an invalid FC (it should cause no side effects except that the EGM will think its is receiving NAKs) .... Pass ☐

919 Change the broadcast message's function code and ensure the EGM no longer processes the global messages .... Pass ☐

920 .... Pass ☐

921 .... Pass ☐

Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐

922 .... Pass ☐

923 PC: CEO (Store a screenshot on file if ok) .... Pass ☐

924 .... Pass ☐

925 .... Pass ☐

926 Confirm Operation... Pass ☐

927 PC: Verify in source code .... Pass ☐

928 (NB QCOM submission requirements reference all use of date and time) .... N/A ☐ Pass ☐

929 Ensure that a significant +/- adjustment of the system time does not cause the EGM to inadvertently jump into a communications timeout, power-save or unwanted other side effect .... Pass ☐

PC: Verify in source code. Verify the above for all timers used and timed periods programmed into the EGM software) .... Pass ☐

930 PC: Verify in source code. If the EGM uses this field to process repeated entries in the message, ensure that the EGM firstly validates this field to ensure that a malformed message cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with total message length).. N/A ☐ Pass ☐

931 .... Pass ☐

932 PC: Verify in source code & LP .... N/A ☐ Pass ☐

933 PC: LP. Ensure the EGM can correctly retrieve is current its LP amounts from this broadcast (ensure there are at least one other NA group being broadcast) .... N/A ☐ Pass ☐

PC: LP. Ensure the EGM can still process all applicable levels when they are not all present, too many, or are mixed with other groups in a broadcast.. Pass ☐

934 PC: LP. Check CAMT=0, check CAMT >= $12,345,67,78 all levels. Note how the EGM behaves when the number is too big for display (refer jackpot display system min reqs.) Pass ☐

935 PC: LP.. N/A ☐ Pass ☐

936 PC: LP. PC: Verify in source code... N/A ☐ Pass ☐

937 PC: LP.. N/A ☐ Pass ☐

938 PC: LP.. N/A ☐ Pass ☐

939 PC: LP.. N/A ☐ Pass ☐

940 PC: LP.. N/A ☐ Pass ☐

941 PC: LP. Split level test. It's a bit of a manual process at this time but....

1. Supress broadcasts for all the levels of the PGID of interest with qsim command:'Linked Progr->More->Supress Levels'.
2. Then manually, one level at a time (every 15 secs or so), unsupress a level & queue it (Linked Progr->Q LP Broadcast) then supress it again.
3. Goto next level in group & repeat. (this will send a single level each time)

Repeat the above for > 60 secs. Expected EGM behaviour is that it should not go into a LP comms timeout.

.... N/A ☐ Pass ☐

942 PC: LP PC: Verify in source code.. N/A ☐ Pass ☐

943 Ensure these bits have no effect if set (ie. Inadvertent fanfare) .... Pass ☐

944 .... Pass ☐

945 PC: CEO (Store a screenshot on file if ok) .... Pass ☐

946 Check lengths of 1,40 & 80 (NB. If the duisplay font is proportional, make sure the string is filled with "FATTER" characters) .... Pass ☐

947 .... Pass ☐

948 PC: Verify in source code .... Pass ☐

949 Ensure the machine can correctly display the full range of printable chars .... Pass ☐

950 .... Pass ☐

951 PC: Verify in source code .... Pass ☐
952 .... Pass ☐

Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
953 .... Pass ☐
954 .... Pass ☐
955 .... N/A ☐ Pass ☐

Ensure the EGM handles the case when there are no white-space characters in a full length message
.... Pass ☐
956 .... Pass ☐
957 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
958 PC: Verify in source code .... Pass ☐
959 PC: Verify in source code. If the EGM uses this field to process repeated entries in the message, ensure that the EGM firstly validates this field to ensure that a malformed message cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with total message length).. N/A ☐ Pass ☐
960 PC: Verify in source code .... Pass ☐
961 When the EGM is responding already on another valid pa, configure two other EGMs on QSIM to have PAs of 0 & 255 and ensure the EGM does not respond to either of these addresses
.... Pass ☐

Repeat this test but simply change the EGMs under test's pa to 0 & 255 and ensure the EGM does not try to respond to these addresses.. Pass ☐

Also ensure the QSIM garbage byte count did not increase when performing these tests Pass ☐

Change the EGMs MID and ensure it does not resume responding.................................. Pass ☐
962 *(apply this test to all implemented versions of QCOM, not just QPv1.6)*
Turn off auto-queue under the poll cycle menus.
Stop polling just the EGM under test (under the EGM->More->Polling menu)
Wait 10 secs
Change the EGMs poll address to a new unused value.
Restart polling the EGM.
Queue a PA Broacast under the Broadcast menu.
EGM must start responding within 2 poll cycles.
When u are finished dont forget to turn auto-queue back on .... Pass ☐
963 Add another EGM to QSIM's poll cycle (don't physically add another EGM) with the same serial no. but a different pa. And ensure the EGM under test does not try to respond or change its pa. while currently responding to its existing pa .... Pass ☐

Powering the EGM off/on should result in the EGM resuming responding to only one of the two EGMs on QSIM with the correct serial no. with no other side effects .... Pass ☐
964 PC: Verify in source code .... N/A ☐ Pass ☐
965 PC: Verify in source code .... N/A ☐ Pass ☐
966 PC: TO Ensure the machine can correctly display/print the full range of printable chars .. Pass ☐
967 PC: TO Ensure the machine can correctly display/print the full range of printable chars .. Pass ☐
968 PC: Verify in source code .... N/A ☐ Pass ☐

Check max & zero length messages work correctly for both fields (NB. If the display font is proportional, make sure the string is filled with FATTER characters) .... N/A ☐ Pass ☐
969 PC: TO .... N/A ☐ Pass ☐
970 PC: TO & CEO (Store a screenshot on file if ok).. N/A ☐ Pass ☐
971 If audit mode is accessed, or a STATE change occurs (check both) just after a poll / resp on the EGMs address, then it is expected that the change be reflcted in the Gen. Stat. response to the 2nd poll after the change occured .... Pass ☐
972 Check individually then all at once.. Pass ☐
973 .... Pass ☐
974 .... Pass ☐
975 .... Pass ☐

976 .... N/A ☐ Pass ☐
977 PC: NA .... N/A ☐ Pass ☐
978 PC: NA .... N/A ☐ Pass ☐
979 PC: Mech Meters .... N/A ☐ Pass ☐
980 .... N/A ☐ Pass ☐
981 Confirm operation .... Pass ☐
982 Confirm operation wrt 'active' operation .... Pass ☐
      Verify CM 'pending' situation .... Pass ☐
      Verify when the EGM is in cashless mode and an exit of CM is pending that the flag here still
      indicates CM .... Pass ☐
983 PC: TI. Confirm operation .... Pass ☐
984 Put the EGM into each of the above lockups and ensure the correct code is returned within 2
      responses of lockup entry provided no higher priority responses delay this .... Pass ☐

      Specifically check the correct code is displayed after a power down .... Pass ☐
      Interrupt a state with a fault and ensure the state is preserved and the fault bit indicated Pass ☐
      Ensure STATE is updated as soon the each lockup is exited (within 2 responses of lockup exit
      provided no higher priority responses delay this) .... Pass ☐
985 PC: Hopper .... N/A ☐ Pass ☐
986 PC: LP .... N/A ☐ Pass ☐
987 PC: TO .... N/A ☐ Pass ☐
988 PC: TO. Confirm operation .... N/A ☐ Pass ☐
989 .... Pass ☐
990 PC: NA .... N/A ☐ Pass ☐
991 PC: NA .... N/A ☐ Pass ☐
992 PC: NA. Null Term .... N/A ☐ Pass ☐
993 PC: NA .... N/A ☐ Pass ☐
994 PC: NA .... N/A ☐ Pass ☐
995 PC: NA. Confirm with EGM manufacturer if they are manipulating a NA firmware string that the
      situation in this paragraph cannot occur .... N/A ☐ Pass ☐
996 PC: NA Is the hashing methodology acceptable? .... N/A ☐ Pass ☐
      Record hashing methodology for NADS field in the QCOM evaluation report/spreadsheet
            N/A ☐ Pass ☐
997 PC: NA change NA type and confirm NADS is update in both cases .... N/A ☐ Pass ☐
998 PC: SAP or LP .... N/A ☐ Pass ☐
999 PC: SAP .... N/A ☐ Pass ☐
1000 PC: Progressive .... N/A ☐ Pass ☐
1001 PC: Progressive .... N/A ☐ Pass ☐
1002 PC: Progressive .... N/A ☐ Pass ☐
1003 PC: Progressive .... N/A ☐ Pass ☐
1004 PC: Progressive .... N/A ☐ Pass ☐
1005 PC: Progressive .... N/A ☐ Pass ☐
1006 PC: Progressive .... N/A ☐ Pass ☐
1007 PC: Progressive This can be really only checked by cross-checking with the game maths,
      however if 6 or 7 non-zero significant digits are being reported, then this will be fine. N/A ☐ Pass
      ☐
1008 PC: Progressive .... N/A ☐ Pass ☐
1009 PC: Progressive .... N/A ☐ Pass ☐
1010 PC: Progressive .... N/A ☐ Pass ☐
1011 PC: CustomSAP .... N/A ☐ Pass ☐
1012 PC: Progressive. Check highest (least prob) jackpot is first. .... N/A ☐ Pass ☐
1013 PC: Progressive. Check re content is correct wrt Poll data .... N/A ☐ Pass ☐
      Also ensure that specifically the PINC field does not incur a rounding error. Set PINC to 0.3%
      in the Progr. Cfg. Poll. Send to the EGM and ensure the EGM reports 0.3% back in the Progr,
      Cfg. Response and not 0.299 or other value .... N/A ☐ Pass ☐
1014 PC: Progressive .... N/A ☐ Pass ☐
1015 PC: LP .... N/A ☐ Pass ☐
1016 PC: CustomSAP .... N/A ☐ Pass ☐
1017 .... Pass ☐
1018 Ensure correct values for MAXL & MAXB for game .... Pass ☐

1019	Ensure correct values for GBFA & GBFB for game .... Pass ☐
1020	.... Pass ☐
1021	Ensure correct values for all possible bets .... Pass ☐
1022	PC: DHS .... Pass ☐
1023	.... Pass ☐
1024	PC: LP... N/A ☐ Pass ☐
1025	.... Pass ☐
1026	PC: Free Games. Check each type, especially free games .... Pass ☐

PC: Hopper re RCRF .... Pass ☐

| 1027 | .... Pass ☐ |

Ensure all multi-game variation meter sets have a non-zero value and perform a reconciliation against the total EGM group meters (QSIM function provided)... Pass ☐

As above but with variation meters in EGM audit mode.. Pass ☐

Ensure residual gamble and wins do no ever get added to these meters .... Pass ☐

1028	PC: Multi-game .... N/A ☐ Pass ☐
1029	PC: HS .... N/A ☐ Pass ☐
1030	.... Pass ☐

NB re SD formula implementation testing: To reconcile EGM displayed SD with QSIM SD for a game in order to verify the SD forumula implementation in the EGM; QSIM must have seen a individual meter packet for every game played. Typically, this wont be the case during test and the tester will have to; RAM clear the EGM, increase QSIM PC time to 2-3 sec, ensure clean comms (no breaks in comms during play)…play games for at least 5 mins or more and pause, at this time the EGM SD display should reconcile with QSIM's SD display exactly.

1031	.... Pass ☐
1032	Ensure is correct for game.. Pass ☐
1033	Ensure are being incremented correctly for the game.. N/A ☐ Pass ☐
1034	.... Pass ☐
1035	RAM clear (stage 5) .... Pass ☐
1036	PC: Multi-game .... N/A ☐ Pass ☐

Change games and confirm the GVN does not change until after the next play has started .... N/A ☐ Pass ☐

| 1037 | .... Pass ☐ |
| 1038 | Verify by requesting multiple groups, playing games and inserting coins .... Pass ☐ |

Also refer checkpoint 843
An early sign of issues with meter group responses is when a play is commenced and a meter group response is received on the very next response from the EGM (instead of the response after). This is an indication of a misunderstanding of the QCOM poll/build/response cycle and will lead to overly complex code .... Pass ☐

1039	PC: LP.. N/A ☐ Pass ☐
1040	PC: LP.. N/A ☐ Pass ☐
1041	PC: LP. Check rollover... N/A ☐ Pass ☐
1042	PC: Multi-game .... N/A ☐ Pass ☐

PC: Multi-game. Ensure once new game selction is available again, that it does not **disable***
again if a coin (or equiv) is inserted during idle mode .... N/A ☐ Pass ☐

PC: Multi-game. Ensure new game selection is not still **disabled*** if a new communications session occurs or after an EGM power fail (e.g. stop polling the EGM or cut its power immediately after a play, before either pending group meters are sent, or acknowledged)…
.... N/A ☐ Pass ☐

| 1043 | PC: Multi-game. Ensure once new game selction is available again, that new game selction does not **disable*** again if e.g. group meters are requested. .... N/A ☐ Pass ☐ |

*WARNING: With respect to the above two checkpoints. Some EGMs implement the game select "**disable**" by blocking (or freezing) game selection instead of the visually obvious grey-out example cited in QCOM. This freeze approach is acceptable but the tester must be very careful not to miss a disablement under these two checkpoints (which is a fail) when the EGM

uses the freeze approach. **I.e. test for the undersired "disable" interactively as well as visually**

1044 .... Pass ☐

Confirm range for numbered events 1..255 and zero never appears .... Pass ☐
1045 .... Pass ☐
1046 PC: RAM clear (stage 2) .... Pass ☐
1047 QSIM will detect a timeout of this response.. Pass ☐
1048 PC: RAM clear (stage 2&3) .... Pass ☐
1049 Confirm current .... Pass ☐

Via HS or RAM Clear change variations and confirm result here .... N/A ☐ Pass ☐
1050 PC: RAM clear (stage 3) .... Pass ☐
1051 PC: LP .... Pass ☐
1052 PC: HS. Check operation .... Pass ☐
1053 PC: RAM clear (stage 3) & HS .... Pass ☐
1054 PC: HS .... Pass ☐
1055 Is HS flag set correctly for game? .... Pass ☐
1056 Confirm operation .... Pass ☐
1057 PC: RAM clear (stage 3) .... Pass ☐
1058 .... Pass ☐
1059 PC: Progressive. Change LP group and confirm result .... N/A ☐ Pass ☐
1060 PC: RAM clear (stage 3) & not LP .... N/A ☐ Pass ☐
1061 PC: RAM clear (stage 3) & Progressive .... N/A ☐ Pass ☐
1062 PC: Multi-game .... N/A ☐ Pass ☐
1063 .... Pass ☐
1064 Setup the EGM in turn with each optional device removed and ensure it does not send meters for a device that is not detected .... Pass ☐
1065 PC: RAM clear (stage 0) Ensure the EGM can be setup and enabled without essential devices. E.g. no coin/note acceptor, hopper, TI, TO, etc (E.g. An ECT only EGM should be possible)
.... Pass ☐
1066 Reconcile with current EGM hardware and then setup the EGM in turn with each optional device removed and ensure the flags are set/reset correctly .... Pass ☐
1067 PC: Mech Meters .... N/A ☐ Pass ☐
1068 PC: RTC .... N/A ☐ Pass ☐
1069 PC: LP & SAP .... N/A ☐ Pass ☐
1070 .... Pass ☐
1071 PC: TO .... N/A ☐ Pass ☐
1072 PC: CV .... N/A ☐ Pass ☐
1073 PC: Hopper .... N/A ☐ Pass ☐
1074 PC: NA .... N/A ☐ Pass ☐
1075 .... Pass ☐
1076 PC: TI .... N/A ☐ Pass ☐
1077 .... Pass ☐
1078 .... Pass ☐
1079 PC: RAM clear (stage 5) .... Pass ☐
Play 1st game and confirm the GVN & VAR .... Pass ☐
PC: HS. Change variations and confirm the above GVN & VAR does not change until next play
.... N/A ☐ Pass ☐
PC: Multi-game. Change games and confirm the above GVN & VAR does not change until next play
.... N/A ☐ Pass ☐
1080 PC: DHS .... N/A ☐ Pass ☐
1081 PC: RAM clear (stage 2) .... Pass ☐
1082 Make a binary image of the EGMs program, then using QSIM verify the EGM returns the correct signature wrt the disk image for the following seeds:
(QSIM has a function to do a signature over a file)

For QPv1.5

0x12345678 .... Pass ☐
0xff55aa00 .... Pass ☐
0x00000000 .... Pass ☐
0xffffffff .... Pass ☐

For QPV 1.6

00000000 00000000 00000000 00000000 00000000 (0000) .... Pass ☐
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF (4ACD) .... Pass ☐
12345678 90987654 32123456 78909876 54321234 (286E).. Pass ☐
88888888 99999999 AAAAAAAA BBBBBBBB CCCCCCCC (641C) .... Pass ☐

For each of the above seeds, the seeds and corresponding hash result must be displayed
identically in the following four areas: 1. a (LSB first*) 3rd party HMAC-SHA1 utility. 2. EGM
audit mode, 3 EGM's on-screen hash calc. utility. 4. QSIM's display .... Pass ☐

*(*To test if you are using an LSB first HMAC-SHA1 utility, right padding with zeros should not
change the hash result, where left padding will.)*

1083 .... Pass ☐
1084 .... Pass ☐
1085 .... Pass ☐

Ensure both are not displayed simultaneously (only the last) .... Pass ☐
1086 .... Pass ☐
1087 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
1088 .... Pass ☐
1089 .... Pass ☐
1090 .... Pass ☐

*(The "Jackpot In" wording add was a client request, editor not to change without a client
consult)*
1091 .... Pass ☐
1092 Check functionality and that all info logged is correct... Pass ☐
Perform two ECT-in during non-idle mode and ensure two correct entries appear .... Pass ☐
1093 Ensure if set these bits have no effect.. Pass ☐
1094 .... Pass ☐
1095 .... Pass ☐
1096 .... Pass ☐
1097 .... Pass ☐
1098 PC: Verify in source code .... Pass ☐
1099 Check maximum .... Pass ☐

Check $0.00.. Pass ☐
1100 Check MAXECT & MAXECT-1 & -1 .... Pass ☐

Check $0.00.. Pass ☐
1101 PC: RAM clear (stage 2) .... Pass ☐
1102 PC: RAM clear (stage 3) .... Pass ☐
1103 .... Pass ☐
1104 .... Pass ☐
1105 .... Pass ☐
1106 .... Pass ☐
1107 .... Pass ☐
1108 .... Pass ☐
1109 .... Pass ☐
1110 .... Pass ☐
Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1) .... Pass ☐
1111 Verify with the EGM not in idle mode that a queued ECT followed by a power fail is queued in
NVRAM .... Pass ☐

Verify while in a couple of Fault Conditions and for **all possible lockup conditions** that an ECT to EGM works correctly .... Pass ☐
With an ECT-in queued (i.e in-play/gamble) generate and clear the following: fault condition & door open – ensure the queued ECT-in remains queued throughout .... Pass ☐

1112   Verify CM & cmode updates for ECT are queued **in all states and modes** of the EGM, except idle mode ....
.... Pass ☐
Verify Cashless-In & total cents in meters are updated immediately **in all states & modes** and are never delayed by any EGM state even a fault condition .... Pass ☐
Specifically ensure ECT in (both ECT modes) during a hopper collect has no ill side effects ....
.... Pass ☐
Specifically verify cashless in during any communications disabl~~e~~ing condition (CDC) and System Lockup operates correctly (ie. Cashless meter updates must not be delayed)..... Pass ☐

1113   .... Pass ☐
1114   .... Pass ☐
1115   .... Pass ☐
1116   .... Pass ☐
1117   .... Pass ☐
1118   .... Pass ☐
1119   .... Pass ☐
1120   .... Pass ☐
1121   PC: Ticket .... Pass ☐
1122   .... Pass ☐
1123   .... Pass ☐
1124   .... Pass ☐
1125   .... Pass ☐
1126   .... N/A ☐ Pass ☐
1127   .... Pass ☐
1128   .... Pass ☐
1129   .... Pass ☐
1130   .... Pass ☐
1131   .... Pass ☐
1132   .... Pass ☐
1133   .... Pass ☐
1134   .... Pass ☐
1135   .... Pass ☐
Also ensure large win lockups upon exit don't exit the EGM from cashless mode .... Pass ☐

1136   .... Pass ☐
1137   .... Pass ☐
1138   .... Pass ☐
Also check when the EGM is in idle mode and in a CDC (E.g. MEF = 0) .... Pass ☐

1139   .... Pass ☐
1140   .... Pass ☐
1141   PC: SAP... Pass ☐
1142   PC: QPv1.5 ECT out on zero credit.. Pass ☐
1143   .... Pass ☐
1144   .... Pass ☐
1145   .... Pass ☐
1146   PC: Ticket .... N/A ☐ Pass ☐
1147   .... Pass ☐
1148   .... Pass ☐
1149   .... Pass ☐
1150   .... Pass ☐
1151   Try to reset the EGM via all other means including a power fail, it should have no effect . Pass ☐
1152   .... Pass ☐
1153   PC: Ticket-in, NA, CV. Start with the EGM cashless mode with Credit (e.g. $100.01). Insert a note (e.g. $10.00) and press collect just after the EGM accepted the note. The EGM should display the message "Transferring $100.01 Please wait" and an ECT Lockup event with $100.01 is sent to the QSIM. The credit meter should increment as a result of the note-in, but the ECT in progress must be unaffected. End result; the EGM must xfer the original $100.01

and end up in idle mode with $10 on the credit meter and it must be clear at all times as to what is occurring on the EGM. .... Pass ☐

Repeat above test with all possible credit input methods on the EGM where possible (e.g. coin-in, TI). State which methods were tested:_____ .... Pass ☐

1154 .... Pass ☐

Also specifically check this point wrt all sub-modes within idle mode (refer 3.3.1). Especially a CDC (E.g. MEF = 0)... Pass ☐

1155 There are a number of permutations to check here before ticking this checkpoint .... Pass ☐
Check special case (*as clarified in QPv1.6.3*):
Start a game so the Credit meter == 0 upon play ~~comencement~~commencement, queue an ECT Lockup Request Poll while play show is in progress. Ensure when the play is a win (and taken) that an ECT-from-EGM Lockup results upon Return to Idle mode..................................... Pass ☐
(Prior QPv1.6.3, brands of EGMs behaved differently in this regard. Some pass the above check & others simply ignored the ECT-from-EGM lockup request if the CM was == 0 upon poll recepit E.g. IGT & KON. This was acceptable prior QPV1.6.3)
Following on from the above, if the play is a loss then the EGM must end up in idle mode with zero credit and still in cashless mode .... Pass ☐

1156 Ensure it is not inadvertently queued for next time.. Pass ☐
1157 Simultaneous ECT in/out.
Start not in cashless mode. Send an ECT with cashless mode and an ECT lockup during a game. The EGM should ignore the ECT lockup and take the ECT amount and end up in cashless mode
.... Pass ☐
Start in cashless mode. Send an ECT with cashless mode and an ECT lockup during a game. The EGM should perform the ECT lockup and then take the ECT amount and end up in cashless mode
.... Pass ☐
Start not in cashless mode. Send an ECT without cashless mode set and an ECT lockup during a game. The EGM should ignore the ECT lockup and take the ECT amount and not end up in cashless mode
.... Pass ☐
Start in cashless mode. Send an ECT without cashless mode set and an ECT lockup during a game. The EGM should perform the ECT lockup and then take the ECT amount and not end up in cashless mode ....
.... Pass ☐

Enter an ECT lockup with non-zero credit. Sent an ECT in with cashless mode. Clear the lockup (try success & fail) and the EGM should end up in idle mode in cashless mode. .. Pass ☐

Enter an ECT lockup with non-zero credit. Sent an ECT in without cashless mode. Clear the lockup (try success & fail) and the EGM should end up in idle mode not in cashless mode
. ... Pass ☐

During a play, while in cashless mode with credit, send an 'ECT from EGM' Lockup Request Poll while the autoplay flag is on and a button is stuck down. Ensure the EGM enters an ECT from EGM lockup after completetion of the current play as expected .... Pass ☐
1158 .... Pass ☐
1159 Confirm all amounts to date were correct .... Pass ☐
heck a transfer out is correct for the max ECT in .... Pass ☐
1160 .... Pass ☐
1161 .... Pass ☐
1162 .... Pass ☐
1163 .... Pass ☐
1164 .... Pass ☐
1165 .... Pass ☐
1166 Also check audit mode and power fail .... Pass ☐
1167 Ensure it is not inadvertently queued.. Pass ☐
1168 .... Pass ☐
1169 .... Pass ☐

1170 .... Pass ☐
1171 .... Pass ☐
1172 PC: Ticket .... N/A ☐ Pass ☐
1173 .... Pass ☐

Test for credit of zero (QPv1.5 only).. Pass ☐
1174 Ensure no effect if set .... Pass ☐
1175 Confirm it is the current state when in the lockup.. Pass ☐
1176 Ensure the EGM doesn't inadvertently evaluate DLSA upon every free game.................. Pass ☐
1177 .... Pass ☐
1178 PC: Emulator or Verify in source code... Pass ☐
1179 .... Pass ☐
1180 Do it... Pass ☐
1181 PC: Verify in source code .... Pass ☐
E.g. Contatenating two successive calls to a 16 bit RNG to make a 32 bit number is NOT a 32 bit RNG
1182 PC: Verify in source code .... Pass ☐
1183 .... Pass ☐
1184 Check both.. Pass ☐
1185 Ensure that the EGM does not evaluate any subsequent configured CRanEs until the current CranE lockup is cleared .... Pass ☐
1186 .... Pass ☐
1187 .... Pass ☐
1188 .... Pass ☐
1189 .... Pass ☐
1190 .... Pass ☐
1191 .... Pass ☐
1192 .... Pass ☐
1193 .... Pass ☐
1194 PC: CEO (Store a screenshot on file if ok) .... Pass ☐
1195 .... Pass ☐
1196 .... Pass ☐
1197 Confirm text & details... Pass ☐
1198 .... Pass ☐
1199 .... Pass ☐
1200 PC: Verify in source code .... Pass ☐
1201 Ensure no effect if set .... Pass ☐
1202 Confirm operation .... Pass ☐
1203 PC: Verify in source code. If the EGM uses this field to process repeated entries in the message, ensure that the EGM firstly validates this field to ensure that a malformed message cannot cause the EGM to process past the end of any buffer or variable (i.e. crosscheck with total message length).. N/A ☐ Pass ☐
1204 Ensure no effect if set .... Pass ☐
1205 Confirm operation .... Pass ☐
1206 Ensure this bit has no effect if set... Pass ☐
1207 Confirm operation .... Pass ☐
1208 PC: Source Code. Verify the implementation of RATE in the source code, ensure no bias .... Pass ☐

If the RNG used for CRanE's is new or modified, get the mathematician to verify the RNG .... Pass ☐

General CRanE tests:

Configure all eight levels with a rate of -1(2^32) and ensure all eight levels are hit each play (check when locking up and not locking up) .... Pass ☐

Configure all eight levels so that on average a hit is obtained every eight plays per level, turn off CRanE event logging in QSIM (special function) and run for 100,000 plays. When finished confirm hits for all levels is within expected values .... Pass ☐

Configure all eight levels so over 100,000 plays a total of 1 hit per level is expected and repeat the above test ....Pass ☐

Test RATE is implemented correctly for at least one other denomination combination on the EGM ....

....Pass ☐

1209	....Pass ☐	
1210	....Pass ☐	
1211	....Pass ☐	
1212	....Pass ☐	
1213	....Pass ☐	
1214	Confirm ....Pass ☐	
1215	Confirm in all CRanE Hit events that have occurred while under test...............................Pass ☐	
1216	....Pass ☐	
1217	PC: Verify in source code ....Pass ☐	
1218	Ensure the machine can correctly display the full range of printable chars ....Pass ☐	
1219	Test lengths of 0,1,20 & 40 work ok (NB. If the duisplay font is proportional, make sure the string is filled with FATTER characters)..Pass ☐	
1220	PC: Verify in source code ....Pass ☐	
1221	PC: CEO (Store a screenshot on file if ok) ....Pass ☐	
1222	Ensure it is not inadvertently queued...Pass ☐	
1223	....Pass ☐	
1224	....Pass ☐	
1225	PC: Verify in source code ....Pass ☐	
1226	PC: Verify in source code ....Pass ☐	
1227	....Pass ☐	
1228	....Pass ☐	
1229	....Pass ☐	
1230	....Pass ☐	
1231	....Pass ☐	
1232	....Pass ☐	
1233	....Pass ☐	
1234	PC: TO ....N/A ☐ Pass ☐	
1235	PC: TO ....N/A ☐ Pass ☐	
1236	PC: TO ....N/A ☐ Pass ☐	
1237	PC: TO ....N/A ☐ Pass ☐	
1238	PC: TO ....Pass ☐	
1239	PC: TO ....N/A ☐ Pass ☐	
1240	PC: TO (e.g. power, fault, door open)...N/A ☐ Pass ☐	
1241	PC: TO ....N/A ☐ Pass ☐	
1242	PC: TO ....N/A ☐ Pass ☐	
1243	PC: TO ....N/A ☐ Pass ☐	
1244	PC: TO ....N/A ☐ Pass ☐	
1245	PC: TO ....N/A ☐ Pass ☐	
1246	PC: TO ....N/A ☐ Pass ☐	
1247	PC: TO I.e. do not display AUTHNO in this log ....N/A ☐ Pass ☐	
1248	PC: TO ....N/A ☐ Pass ☐	
1249	PC: TO ....N/A ☐ Pass ☐	
1250	PC: CEO (Store a screenshot on file if ok) ....N/A ☐ Pass ☐	
1251	PC: TI & Verify in source code – order: set flag first then log event ....N/A ☐ Pass ☐	
1252	PC: TI..N/A ☐ Pass ☐	
1253	PC: TI. I.e check 16, 18 digit barcode are display correctly wherever they appear in any EGM display...N/A ☐ Pass ☐ Check outside this range for possible ill effects e.g. 14 & 20 digits ....N/A ☐ Pass ☐	
1254	PC: TI..N/A ☐ Pass ☐	
1255	PC: TI..N/A ☐ Pass ☐	
1256	PC: TI..N/A ☐ Pass ☐	
1257	PC: TI..N/A ☐ Pass ☐	

1258   PC: CEO (Store a screenshot on file if ok) .... N/A ☐ Pass ☐

1259   PC: TI. Check each failure code + one unknown code .... N/A ☐ Pass ☐

1260   PC: TI.. N/A ☐ Pass ☐

1261   PC: TI.. N/A ☐ Pass ☐

1262   PC: TI.. N/A ☐ Pass ☐

1263   PC: TI (Check for interruptions occurring immediately after: physical ticket insertion, receipt of TI event, TI Ack Poll and during stacking) .... N/A ☐ Pass ☐

1264   PC: TI.. N/A ☐ Pass ☐

1265   PC: TI.. N/A ☐ Pass ☐

1266   PC: TI.. N/A ☐ Pass ☐

1267   PC: TI.. N/A ☐ Pass ☐

1268   PC: TI.. N/A ☐ Pass ☐

1269   PC: CEO (Store a screenshot on file if ok) .... N/A ☐ Pass ☐

1270   PC: TO .... N/A ☐ Pass ☐

1271   PC: TO .... N/A ☐ Pass ☐

1272   PC: TO .... N/A ☐ Pass ☐

1273   PC: TO .... N/A ☐ Pass ☐

1274   PC: TO Ensure the mandatory text and the data is displayed correctly .... N/A ☐ Pass ☐

1275   PC: TO Ensure the mandatory text and the data is displayed correctly Ensure a serial number such as 123456 or similar is tested (no zero's) .... N/A ☐ Pass ☐

1276   PC: TO .... N/A ☐ Pass ☐

1277   PC: TO .... N/A ☐ Pass ☐

1278   PC: TO .... N/A ☐ Pass ☐

1279   PC: TO .... N/A ☐ Pass ☐

1280   PC: TO For a very large values of AUTHO (i.e.much greater than what the brocade supports), ensure the EGM correctly ignores the additional decimal digits and encodes the correct barcode. I.e. when extracting the barcode number from an AUTHNO the EGM should be doing the equivalent of:

barcode_number = AUTHNO mod 10^18 .... N/A ☐ Pass ☐

1281   PC: TO .... N/A ☐ Pass ☐

1282   PC: TO .... N/A ☐ Pass ☐

1283   PC: TO .... N/A ☐ Pass ☐

1284   PC: TO .... N/A ☐ Pass ☐

1285   PC: TO .... N/A ☐ Pass ☐

1286   PC: TO .... N/A ☐ Pass ☐

1287   PC: TO .... N/A ☐ Pass ☐

1288   PC: TO .... N/A ☐ Pass ☐

1289   PC: CEO (Store a screenshot on file if ok) .... Pass ☐

1290   PC: TI & TO. Power down the EGM at every stage of the TITO process and verify the EGM recovers correctly.. N/A ☐ Pass ☐

1291   PC: TO .... N/A ☐ Pass ☐

Notes to document editors:

- *QCOM Version Numbering Convention*

  A change to the first two digits represents a major change to the protocol indicating software changes are required in EGMs and possibly SCs (this may also require the QPV & DLLVER fields in the protocol to be incremented if the change is something a SC needs to be aware of). A change to a third digit indicates a clarification release only and no software changes are required in EGMs or SCs.

- *For every change to the protocol, consider and ensure:*

  The revision history is updated (clearly highlight changes that require EGM or system changes) ....☐
  The QCOM feature list is updated as necessary (section 1) ....☐
  Add new checkpoints as necessary..☐
  In the area changed specify (if applicable) which version it applies to (e.g. QPv1.5, QPv1.6)☐
  Manually spell-check (as the document is now too large for Word's auto-spell-check to function) ....☐
  Consider for every change, is backward compatibility affected? (~~eg.~~e.g. SCs, EGMs, third party protocol sniffers & sniffing LP displays) ....☐
  Does the change affect the Protocol Simulator? (Raise SCR if yes)...................................☐
  Does the change affect any related documents listed in section 1.1? (Especially the Site Controller Procedures document) (Raise QIR if yes) ....☐

  Does the change affect EGM software? ....☐
  Does the change affect the Data Requirements? (significant project) ....☐
  Does the change affect the LMOs (SC software and CMS)? (special incept required) ....☐

- *Final release checks. Before releasing a new version ensure:*

  The front page and revision history is updated especially wrt release/incept dates (NB: even dates of release of draft version must also be recorded as individual releases)...................☐
  Update the Version Number on~~ on~~ page 1 and on the checklist title pages..........................☐
  Note, all external releases, including drafts, must be individually tracked in the revision history ....☐
  Copyright notices (front page & footer) is current with respect to the year and version.........☐
  Refresh both Table-of-Contents..☐
  Check page formatting (with 'mark-up' turned off) ....☐
  Search and remove IIS's (if any)..☐
  Search and remove comment boxes ....☐
  Search and remove TBA's where possible..☐
  Does NPRV or PVERSION fields require updating? ....☐
  Manually spell-check all changes (as doc is now too large for auto spell checking)..............☐
  Ensure references are correct (In Word, select All, then press F9)......................................☐
  Search and remove 'Error! Reference Source not found' errors (NB these are most likely to occur in the revision history) ....☐
  Check any web references are still ok (search 'www') ....☐
  Clarify in the revision history (typically) the incept date/s for new or changed requirements. ☐
  Do any new/changed requirements need to be made 'effective immediately' or other timeframe? ....☐
  Release both the changed marked ~~verson~~version as well as the final version in Word document format (Word format is used because hotlinks will not function and there are too many to reliably carry over into a pdf)..☐

- *General remarks/ design rules regarding QCOM*
  - *All variable length / repeated fields should have a count (NUM) & size (SIZ) fields associated with them.*

- *Always allow repeated entry NUM fields to be zero (better for extensibility as you can extend the protocol by superseding the repeated fields by making the NUM field zero).*
- *Optional features require a present or capability flag, or other means of detecting implementation of the feature.*
- *Create new events instead of modifying old events.*
- *Increments in Multi-game/Variation meters must be able to predicted from movements in group meters.*
- *Be mindful of existing reconciliation formulas used (especially on group 0 meters, refer 12.1.1) when adding/changing group meters.*
- *Never edit or delete from the revision history as this is very important with respect to IP.*
- *Event EXTD field size must be <= 16 (ref qsim – esp: qceq)*

- *Other*

    If the document is in Word format then there are a number of bookmarks defined to make navigating this document easier. All cross references are hyperlinks for convenience.

Revision History

Note:

- Change-marked versions of this document are available on request.
- MS Word format versions of this document are available on request to all QLD Machine Gaming / QCOM Licensees. (This enables all in-document hyper-links for easier document navigation)

QCOM Version 1.6.7j Draft for industry comment
Date: tba
Author: Robert Larkin
Publication Date & Method: – pdf - http://www.business.qld.gov.au/industry/liquor-gaming

1. Clarified implementation of COLLIM and TICKET fields with some supporting pseudo code. Refer section 3.5.1.
2. Clarified sections 7.1.3 & 7.5.4 which were arguably in contention with each other in some rare scenarios concerning time changes.
3. Changes to HRATE (refer sections 15.6.3 & 10.4). An EGM may now change the reported value of HRATE at certain times and also report HRATE as zero to signify a disabled jackpot level. Accordingly HRATE is no longer required to be displayed in EGM audit mode (10.4). (*NB reminder: as per QCOM SCP requirements v1.6.6 dated 5-Sep-2013, systems are no longer to use HRATE for any purpose*)
4. Progressive Cfg Poll (15.4.6), the sanity checks: *SUP must not be equal to zero* and *SUPn >= SUPn+1* are no longer an integrity check for EGM and systems. The approval of games that adhere to these checks are a decision for each jurisdiction at game approval. Note that games that break these former rules should not be submitted for one year from the date of publication (as systems are given a year to implement the required changes.)
5. Progressive Game Cfg Poll (15.4.3). It is no longer mandatory to divert a % of CAMT to the hidden increment pool upon processing of CAMT in the poll. I.e. an initial CAMT contribution may be all dumped onto the main prize value. So long as nothing is lost.
6. Added a requirement for the EGM to be able to scan tickets with other than 18 digits in the barcode. (This is in support of both forward and backward compatibility) (Refer section 22.2)
7. Added a requirement and checkpoint pertaining to the ability to enable an EGM for play without most peripheral devices installed (refer 15.6.12). E.g an ECT only EGM.
8. Added more checkpoints to ensure that existing meters affected by TITO are properly tested during a QCOM TITO only evaluation (12.1.1).
9. Added a new scenario in which the EGM Progressive Configuration Changed event must be logged. Also clarified the PRTP field in this event in cases where this field varies e.g. wrt bet. (Refer section 7.10.3.34)
10. Deleted all requirements pertaining to the display of the last TITO & ECT operation to the player in idle mode. Refer sections 0 and 22. This should help lower the chance of confusion, misconception and clutter on the EGM's display. Requirements like these are outside the scope of QCOM. Note: All player queries in the area are better addressed via individual logs pertaining to TITO and ECT in EGM audit mode.
11. Also reduced the time a TI error is displayed to 5 seconds max in order to avoid player confusion in thinking an error is ongoing when it isn't and still being displayed during next ticket insertion. Refer 22.2.
12. Many ticket printers treat '^' as a control code. Refer section 2.3.8 for options.

13. Clarified some "STATE" definitions in the General Status Response to help ensure consistency in implementation (15.6.1).
14. Relaxed barcode length dimension tolerance slightly and advised of banknote validator whitespace requirements (22.3).
15. New checkpoint added at the bottom of Meter Group/Contribution Response section 15.6.8. The new checkpoint ensures that game selection does not inadvertently disable (again) once all meters have been received / Acked by the SC.
16. Take into account that PIDs are no longer required for QLD and thus PIDs may or may not be implemented. Refer "PID".
17. Changed the definition of a "successful cash ticket print". Refer section 22.1.3. This new definition should be complementary with the old definition and therefore should not require any change to EGMs or systems.
18. Clarified EGM Configuration Poll (15.4.2) sanity checks by EGMs.
19. Clarified CRanE lockup title display (17.2)
20. Lowered min text size for non-prominent SPAM to the default height of 5mm high minimum (was 10mm). Clarified prominent SPAM must be 10mm min (Refer 15.4.20)
21. Section 6.2 Display of CDCs; changed display of "EGM Disable" to "Machine Disable" (i.e. more human friendly)
22. Make CDCs less noisy: No sound denoting CDC entry may be emitted if the EGM was already in a CDC. Refer section 6.
23. Clarified Communication Timeout & Communications Fault detection, related event logging and CDC display. Refer sections 6.1.2, 6.1.3, 7.10.3.6 and 8.3. *(The main thing to note for QCOM EGM developers is that CT CDC display cant be linked to poll address designation, and the 3NAKs CDC display cant be linked to the NAK counter – they must all be separate variables in order for the requirements to be implemented correctly)*
24. Added a QCOM Poll Cycle heartbeat display during CDCs. Refer section 6.2.
25. Appendix D. There has been a change of name for SHFL ENTERTAINMENT (AUSTRALASIA) PTY LIMITED to BALLY TECHNOLOGIES ANZ PTY LTD.
26. Added QCOM 8-bit mode. Refer 13.3. (QLD LMOs please consult with OLGR before use.)
27. Added a jurisdiction flag (JUR) for Tasmania. Section 15.4.2.
28. Added a note on machines with 64 bit meters. Section 12.3.1.
29. Updated to docx file format.
30. Added a PID number for Tasmania. Refer Section 15.4.5.

Incept date for the implementation of new or changed requirements unless stated otherwise is one year from the date of publication.

This document is now only published as a pdf, however alternative formats (where the internal links all function – such as its native .doc format) are available on request to QCOM licensees.

QCOM Version 1.6.6i
Date: 13 June 2013
Author: Robert Larkin
Publication Date & Method: 13 June 2013 _ Published on www.olgr.qld.gov.au

1. Minor clarification to SPAM TEXT display requirements (15.4.20) in order to make it agree with Appendix A – EGM State Pattern.
2. Manufacturer feedback resulted into a review of credit meter rollover scenarios and risks thereof. Refer section 12.3.3.
3. Removed the former self audit requirement regarding CM >= 0 check (refer 12.4) and added a new paragraph concerning credit meter validations.
4. The Cash Ticket In Ack Poll (15.4.11) / process had no event like ECT-to-EGM did (16.1.1) for when the EGM denies an amount for being too large. Created new generic event for this called 'Transaction Denied - Credit Limit Reached' (7.10.3.44) now for use with both ECT-to-EGM and Cash Ticket In. This event replaces the old ECT RTEXT event used in this scenario – refer section 16.1.1)

Incept date for the implementation of new or changed requirements unless stated otherwise is one year from the date of publication. (V1.6.6g draft change #9 (see below) is the only change that is effective immediately.)

QCOM Version 1.6.6i 2nd draft
Date: 22 May 2013
Author: Robert Larkin
Publication Date & Method: 22 May 2013 via email to industry email group

1. Minor clarification in TITO ticket barcode dimension (ie the barcode dimensions are a minimum size). Refer 22.3
2. The requirement to display ticket printer firmware in EGM audit mode (22.4) is only mandatory if the ticket printer is capable.
3. Re version 1.6.6g draft change #5, made it a generic requirement to display TSER and amount during any ticket printer related fault occurring during a ticket print. (22.1.1)
4. An EGM is **not** to resume/retry a ticket print upon an error during the printing process (22.1.1)
5. Other minor clarifications (not TITO related) refer changed marked version of document.

Incept date for the implementation of new or changed requirements unless stated otherwise is one year. V1.6.6g draft change #9 (see below) is effective immediately.

QCOM Version 1.6.6g draft
Date: 8 May 2013
Author: Robert Larkin
Publication Date & Method: 8 May 2013 via email to industry email group

1. Function code fields were missing from CRanE poll / response message format layouts. (*In QCOM by defn. there is always a FC field following the DLL field*).
2. ECT-To-EGM Poll (16.1.1) – made PSN processing & update timing a little clearer based on recent feedback by a new manufacturer.
3. Ticket layout – permit display of the address line text over two lines (refer 22.3)
4. Ticket layout (related) – removed paragraphs pertaining to a SC potentially using two consecutive space characters to indicate a suitable place for a line break in the LTEXT field (refer 15.5.5).
5. Refer "EGM Cash Ticket Out Print Failure" fault event (7.10.1.32), The fault condition display associated with this event must also display the extended event data.
6. Added notes and clarification concerning the mapping of machine monitored doors onto QCOM door events. Refer 7.10.3.12
7. Added Iamapoll and Iamaresp flags to DLL CTRL byte of all poll / broadcast messages (refer 14.2 and response message types). This doesn't affect any existing or future EGM implementations.
8. Added the requirement to display ticket printer firmware in EGM audit mode (22.4)
9. **Clarified that CRLIMIT is not to apply to Ticket In** (refer 15.4.5 & 15.4.11). The TITO system will apply any TI limit as it is likely that some jurisdictions may want to set a different limit for TI than physical credit in as controlled by CRLIMIT.

An incept period of one year is proposed bar point 9 which is intended to be effective immediately.

QCOM Version 1.6.5h
Date:10 Jan 2013 Robert Larkin
Publication Date & Method: 10th Jan 2013 - Internet Release

2nd draft to final changes:

- Added final Ticket layout diagram (section 22.3) and added three new checkpoints concerning the potential for text display field overrun wrt Venue, Address & CTEXT fields.
- Clarified a couple of minor issuses concerning jackpots with a hidden increment of the overflow meter and rounding
- Fixed incorrect event code on the "Ticket-In Aborted" event (7.10.3.43)
- Added an additional sanity checks re TAMT in the Ticket-In Ack Poll (15.4.11) i.e. TAMT != 0

Incept date: Unless stated otherwise* the incept date for which Queensland Licensed EGM manufacturers must implement all new or updated QCOM requirements up to and including this version of QCOM in all supported EGM base software and submit this for evaluation to the OLGR, is **one year** from the above QCOM version date. QLD LMO systems must be submitted and approved to the updated requirements no later than the above incept date. Existing gaming machine monitoring systems must support the reporting of new/updated EGM events within one month as per QLD Monitoring System minimum requirements.

*While the implementation of TITO in EGMs remains 'not mandatory', updated TITO requirements are effective immediately if TITO is implemented.

QCOM Version 1.6.5h 2nd DRAFT 06/12/2012 Robert Larkin
Publication Date: industry group email: 6/12/2012

- Player Selectable Denomination (3.1.3.3). The EGM must require the credit denomination to be set at one cent. (Ref NW & DR2.5)
- **Clarifications to TITO based on draft feedback**:
    - Clarified TITO – TI re meters 0x16, 0x17, 0x1A & 0x1E (refer 12.1.2)
    - Clarified TITO – TO section on difficulties in printing a ticket (22.1.1)
    - TITO – TI. The EGM no longer logs the formerly named "Invalid Ticket Acknowledgement Event" (7.10.3.36) at any stage for the TI process (refer 15.4.11). Changed the name of the event accordingly (added "Out" to event name descriptor).
    - TITO - Display requirements for the "ticket rejected" message have been made easier to implement (22.2)
    - TITO - Created a TI Acknowledgement Time-out (22.2.1) and event (7.10.3.42)
    - TITO - Capped TI to MAXECT (15.4.11)
    - TITO - Changed section on interruptions to TI process (22.2.2) and created "Ticket-In Aborted" event (7.10.3.43)

QCOM Version 1.6.5h DRAFT 23/10/2012 Publication Date: 01/11/2012
Robert Larkin

Incept Date: NA – draft release for industry feedback. Please submit feedback within 4 weeks from the release date.

- Allow low fan RPM (7.10.3.25) & CPU Overtemp (7.10.3.24) events to be operated as fault conditions.
- Addressed an issue that arose concerning the previous introduction of the requirement for SAP self audit. The issue is that SAP reconciliation formulas would be broken by a meter rollover (albeit rare) or a SAP parameter change in customSAP games (likely). Refer 12.4 & 10.4.1 for how this is must be addressed. **The incept date for this change as per the incept date for QCOM v1.6.4g.**
- Noted the SD display formulas susceptibility to TURN rollover 15.6.6.
- Increased LPB timeout to 90 seconds (6.1.8)

- Clarified QCOM's progressive level numbering convention re a new progressive game type (refer QCOM internal daily log 11/4/2011 for more information). Refer section 10.2
- Clarified the term "reasonable" wrt default SAP parameters. Refer section's 15.6.4 footnote.
- Added checkpoint concerning loop clobbering. Refer section 13.1
- Clarified reserved bit masking in all cases to help reduce confusion. Refer term "mask"
- Added further poll/response processing notes (refer section 14.1.1) as a result of new client feedback
- Created a new section for shared progressive support (10.9) and moved primary requirements to this section. Also clarified requirements to make them easier to understand
- Reviewed section QCOM Message Synchronisation (13.2.2)
- Added JUR ID 0x04 denoting South Australia and requested feedback on use in EGMs (refer section 15.4.2).
- Game Configuration change Poll (15.4.4). Deleted the requirement introduced in QPv1.6 which required the machine upon a VAR change, to not affect the result, or display of the last played game on the EGM. This requirement was deleted because 1. It was resource heavy to implement 2. It was in contention with GMNS 3. It made VAR changes confusing for games whose paytable also changed upon a VAR change (primary reason for deletion) 4. VAR changes can be better managed by the system (e.g. the system can time the VAR change appropriately and also give player notifications as required)
- Added new section discussing QCOM **TITO** AUTHNO generation (22.5)
- Assigned a **TITO** category in the ECT-To-EGM ECT Source ID table (16.1.1)
- Remove the requirement for a **TITO** (TO) reprint facility for security reasons.
- **TITO**: AUTHNO must not be displayed in the Cash Ticket Out log (refer 22.1.2). This is for better security and is related to ticket forgery attacks (tickets can still be looked-up in the ticket database given the machine ID, time-stamp & voucher ID; all available from the log). (*NB there is no significant benefit in hiding the AUTHNO on the EGM's ticket-in log as this event represents a presented ticket and thus the end-of-life of the given AUTHNO*)
- **TITO**: Adopted the GTA proposed ticket print layout. Refer 22.3.
- **TITO**: Adopted GTA wording: "Printing Cash Out Ticket - Please Wait" to be used when printing a cash out ticket
- Made the implementation of ECT Source ID mandatory (16.1.1) as ECT-in being used for multiple services is a proven certainty.

QCOM Version 1.6.4g finalised: 20/08/2010 Released: 04/11/2010* Robert Larkin

* release was delayed due to OLGR website redesign project

Incept Date: If not stated otherwise, the incept date for Queensland EGMs for which new or updated requirements in this document must be implemented, is as per the incept date for next major version release of the ANZ National Standards for Gaming Machines document. EGM manufacturers may implement earlier if desired with prior permission from the OLGR. QLD LMOs must be approved on the updated requirements no later than the above mentioned incept date. Existing gaming machine monitoring systems must support the reporting of new/updated EGM events within one month as per QLD Monitoring System minimum requirements.

- Put upper limit on message timing concerning win payout thresholds (15.4.5)
- Updated to new DEEDI report format template
- "System Lockup" display must no longer have a title. The intent is the make system lockup less 'error' looking as it is often used for awards etc. (15.4.9)
- Added the requirement for EGMs to also self audit SAP levels (12.4)
- Clarified the display on real time/ non-real time items in EGM audit mode (section 4)

QCOM Version 1.6.4g DRAFT Released 13/05/2010 - Robert Larkin

- Fixed error in formula for RTP Stndard Deviation (a bracket set was missing). Refer 15.6.6
- Added new section regarding Multiple LP Award Handling (10.8)
- Review/clarified the usage of the term 'cashless mode' throughout document.

- Addressed contention wrt last play recall display requirements for an ECT-in that occurs during a play (again). I.e removed requirement altogether as the new ECT log covers this better. Refer Section 16.
- Made a minor change to the integrity check in section 15.4.6 re $SUP_n > SUP_{n+1}$. Changed to $SUP_n >= SUP_{n+1}$ RE SAP Game <*editor: refer qcom daily log for more information*>
- Clarified PRET field re strategy games. Refer 15.6.11
- Updated front page logo to exclude "Treasury" and added reference to DEEDI.
- Clarified MAXNPWIN definition is as per GMNS.
- Reserved EGM Parameter Poll FLG Bit 3 for NZ (Ref 27/11/2009) Section 15.4.5
- Clarified CDCs with respect to physical credit disablement timing (section 6).
- Clarified requirements during LP win shows (section 10.6.1).
- Allow option of adding cashless-in's (ECT-in) to the credit meter when the EGM is in-play (16.1.1) (rather than mandating waiting until RTIM).
- Added QCOM State Diagram pertaining to the General Status Response (15.6.1)
- Deleted QCOM requirement to display progressive jackpot awards until start of next play (10.1). This was an effort to implement, rarely used and last play recall and the event logs cover this all sufficiently. (NB: GMNS v10 3.8.7 still has requirements in this area however.)
- Clarified that RCRF must not contribute to a EGM triggered QCOM progressive (10.1)
- Checklist: added requirement to record screenshots of CEO checkpoints.
- Clarified power-save exit when PWRTIME = 0 (15.4.5)
- Clarified Progressive meters in audit mode re shared progressives (10.4.1)
- Added a sanity check for GCP:CAMT (15.4.3)

QCOM Version 1.6.3f Released 17 March 2009 (published 24/3/2009) - Robert Larkin

Incept Date: If not stated otherwise, the incept date for Queensland EGMs in which new or updated requirements in this release must be implemented, is as per the QLD incept date for National Standards Version 10 (QLD incept date for this was 1/10/2009). EGM manufacturers may implement earlier if desired with prior permission from the OLGR. Monitoring Systems should support the reporting of new events within one month.

- Clarified the new licensing events are optional for an EGM manufacturer.
- Corrected all multiplers concerning PINC, AUXRTP, MINRTP, MAXRTP & SUPRTP as used in examples throughout the document.
- 10.4.1 Moved 'win log' item to last in meter list.
- Added date and time calcs/comparisions to submission requirements list. This change is effective immediately.
- Added date and time display to RAM error fault display. Refer 3.2

QCOM Version 1.6.3e Draft Released 15/01/2009 - Robert Larkin

Incept Date: If not stated otherwise, the incept date for Queensland EGMs in which new or updated requirements in this release must be implemented, is as per the incept date for National Standards Version 10. EGM manufacturers may implement earlier if desired with prior permission from the OLGR. Monitoring Systems should support the reporting of new events within one month.

- Clarified required precision for HRATE field. Refer section 15.6.3. This change is effective immediately.
- The maximum number of decimal places for a progressive game's percentage increment amount is 4. Refer Section 10.1. This change is effective immediately.
- Progressive implementations must not accumulate rounding errors. Refer Section 10.1
- Clarified an ECT-from-EGM special case when a play is in progress with CM==0 and an ECT from EGM poll is received during the play show and the play results in a win. Refer section 16.3.2
- Clarified last play recall display for an ECT-in that occurs during a play. Refer Section 16.
- New event: EGM Primary Display Device Failure 7.10.1.36

- New event: EGM Tertiary Display Device Failure 7.10.1.37
- EGM Top NP Prize Hit Event – removed 7.10.3.35
- Reserved EGMPP FLG:Bit 4 for Tattersalls use (15.4.5)
- New events: EGM License Key Detected. Refer sections 7.10.3.40 & 7.10.3.41
- New event: EGM License Key Missing/Failure 7.10.1.38
- 0x03 = NZ version PID (15.4.5)

QCOM Version 1.6.2e Released 9/7/2008 - Robert Larkin

Incept Date: If not stated otherwise, the incept date for Queensland EGMs in which new or updated requirements in this release must be implemented, is as per the incept date for National Standards Version 10.   EGM manufacturers may implement earlier if desired with prior permission from the OLGR.   The new submission requirements in Appendix F are effective immediately.

- Added RTEXT event for ECT-IN too large due to credit meter display width limit (16.1.1).
- Implemented name change to from QOGR to Office of Liquor Gaming and Racing (OLGR) (effective 1/7/2008)
- Changed section titled 'QCOM FTP' to 'QCOM and RUGMs' and a quick review of section 18.

QCOM Version 1.6.2e DRAFT Released 15/05/2008 - Robert Larkin

Incept Date: If not stated otherwise, the incept date for Queensland EGMs in which new or updated requirements in this release must be implemented, is as per the incept date for National Standards Version 10. EGM manufacturers may implement earlier if desired with prior notice to the OLGR.

- Clarified that power-save mode is a part of idle mode (3.3.1).
- Clarified that rounding issues on the LSD of the HRATE field in 15.6.3 are expected and are acceptable.
- Clarified with respect to EXTJIP player display (15.4.7), that this is typically not intended to be used as a sole display of jackpot current amount in a jackpot system (however...).
- Clarified the PID upon a variation hot-switch (15.4.4)
- Made display requirements for seeds & hashes as per GMNS (4.2.10)
- Added Appendix F – Submission Requirements for QCOM EGMs. **The new submission requirements are effective immediately**
- Clarified display of HRATE (15.6.3) in EGM audit mode for consistency.
- Arranged progressive meters in order as per GMNS (refer 10.4)
- Audit mode: Added total Game RTP display for progressive games (4.1.10)
- Clarified display of title in ECT From EGM lockups (16.3.1)
- Hashes of system components must be combined via XOR (11.1.10) to bring in line with other OLGR standards.
- Clarified Multi-game selection when GEF=0 (refer section 9)
- Added the 'EGM Hopper Overpay Amount' advisory event (7.10.3.39)
- Added requirements for a System Lockup audit log in EGM audit mode (15.4.9).
- Changed ECT "CASHLESS" IN/OUT labels as required in last play recall for ECT (0)
- Added new Total EGM Coins/Tokens Cleared & Total EGM Notes Cleared meters (12.1.2)
- (NB: Released draft to Tabcorp 25/3/2008 RLL.)
- Clarified exactly when Stroke and Turnover meters must be updated with respect to CRanE lockups (17.1.2)
- Added the requirement for an EGM to display each game's current RTP Standard Deviation in audit mode (15.6.6).
- Added the ability to force individual EGMs in/out of power-save (15.4.5) while leaving others in play.
- Added new 'EGM Cash Box Cleared' event (7.10.3.13)
- System Lockups must not display "call attendant" or the equivalent (15.4.9).

- Clarified the 'Shared Progressive Flag' (refer 15.6.12) wrt PGID changes via the EGM Game Configuration Change Poll. **This clarification is effective immediately.**
- EGM no longer may display "call attendant" during a LP lockup (10.6.1)
- MAXBET was changed from FYI to a bet limiting parameter per play (15.4.2)

QCOM Version 1.6.1d. Released 28/02/2007 - Robert Larkin

Incept Date: Where not stated otherwise, the incept date for new or changed requirements in this version of the document is six months from the (non-draft) release date in all QCOM EGM software submissions to the OLGR. (Incept date of this release was extended to coincide with NS9 on the 1/10/2007)

- Clarified that the PID access methodology must be a unique action on the EGM to ensure a correct value for PID accessed meter (12.1.1).

QCOM Version 1.6.1c. 8th draft Released 7/12/06 - Robert Larkin

Incept Date: Where not stated otherwise, the incept date for new or changed requirements in this version of the document is six months from the (non-draft) release date in all QCOM EGM software submissions to the OLGR.

- Added an icon display option to the External Jackpot Information Poll (15.4.7) as requested by Tattersalls. Support for this new feature is optional for QLD.
- Added the Unit Modifier Flag to the External Jackpot Information Poll (15.4.7) as requested by Tattersalls. Support for this new feature is optional for QLD.
- Added the ECT Source ID field to the ECT to EGM Poll (16.1.1) as requested by Tattersalls. Support for this new feature is optional for QLD.

QCOM Version 1.6.1c - 31/10/06 7th Draft. Released 2/11/06 - Robert Larkin

Incept Date: Where not stated otherwise, the incept date for new or changed requirements in this version of the document is six months from the (non-draft) release date in all QCOM EGM software submissions to the OLGR.

- Clarified requirements for multiple updates to QCOM group meters to ensure that only one group meter response will result. Refer 12.2.2
- An ESD protected UTP / Ethernet / TCP/IP capable port will be mandatory on all new EGM models submitted from 2007. Refer section 1.
- Clarified hardware flags definition in the EGM Configuration Response (15.6.12)
- The EGM must reset bet meters to zero upon a denomination hot-switch (15.6.5).
- Added two more validity checks for progressive levels (PINC & SUPRTP) in the Progressive Configuration Poll (15.4.6)
- Clarified processing of LP Broadcasts during a LP lockup (10.6.1).
- Clarified what is 'shared' exactly when an EGM sets the Shared Progressive Component Flag (15.6.12)
- Added screen dumps examples courtesy of IGT (Australia) PTY LTD.
- With the introduction of win payouts (15.4.5), the document was still assuming in a number of places that collects would always be of the whole current credit meter amount. Fixed all.
- Added RTEXT suggestions. Search for RTEXT throughout this document for all suggestions. Also refer section 7.10.1.16 which becomes mandatory in 2007.
- Clarified section 14 with respect to malformed messages (14.1.21)
- Added new optional CDC: "RTP out of range" (6.1.9).
- Added macros to assist in the electronic completion of the checklists in this document.

QCOM Version 1.6.1c 6th draft 9/03/2006 Robert Larkin

- Prevented possible of event 0x204E (7.10.3.29) causing an event runaway.

- Added a pending requirement regarding the logging of diagnostic information when the EGM invalidates any of the configuration poll types (7.10.1.16).
- Clock display text height reduced to 7mm 15.5.1
- SPAM min text height reduced to 10mm 15.4.20
- Added requirement for the logging of Manf. Spec. Fault event re invalid configs (7.10.1.16)
- Other changes to document since the last released draft are highlighted.

QCOM Version 1.6.1c 5th draft 08/11/2005 Robert Larkin

- Changed CRE abbreviation to CRanE pronounced "C R an E". Refer section 16.5.1.1.
- Added >= verification to self audit requirements (12.4)
- Other changes to document wrt last draft are highlighted.

QCOM Version 1.6.1c – 4th DRAFT Robert Larkin – 14/10/05

- Finalised QCOM checklist in preparation for the first QCOM v1.6 evaluation
- No longer refer to program 'signatures' but to program 'hashes'. This is a cosmetic change only except refer sections 4.2.10 & 11.1.8.
- Increased prominence of SPAM A Poll (15.4.20).
- Increased prominence of clock display and mandated preferred locations (15.5.1).
- Assigned new JUR, OPR & PID numbers for use by Tattersalls.
- TITO Added option for Interleave 2 of 5 barcode (22.3).

QCOM Version 1.6.1b – 3rd DRAFT Robert Larkin – 19/08/2005

- EGM Cash Ticket In Request (7.10.4.13) – increased AUTHNO field to 16 bytes and now to be displayed as MSB first hex number.
- Removed AUTHNO field from the "EGM Cash Ticket Out Print Failure" Event (7.10.1.32) to keep Extd event data under 16 byte limit (was a redundant field anyway).
- Cash Ticket Out Request Acknowledgement Poll (15.4.10) – increased AUTHNO to 16 bytes hex
- Cash Ticket In Request Acknowledgement Poll (15.4.11) – increased AUTHNO to 16 bytes hex

QCOM Version 1.6.1b – 2nd DRAFT Robert Larkin – Released 22/07/2005
QCOM Version 1.6.1b – 1st DRAFT Robert Larkin - Released 18 May 2005

- Clarified 'win payouts'. Refer 3.3.2 & 15.4.5.
- Added the 'Shared Progressive Component Flag' to the EGM Configuration Response (15.6.12).
- Added EGM Invalid Progressive Configuration Event (7.10.3.23)
- Changed the name of the 'SAP Configuration Change Poll' to 'Progressive Configuration Poll' & included LP information in this poll (15.4.6). This will allow a LP EGM to display LP information in audit mode and to validate its total game RTP wrt MINRTP/MAXRTP.
- Changed the name the 'SAP Configuration Response' to 'Progressive Configuration Response' & included LP information. (15.6.4)
- The EGM Group Meter (0x07 refer 12.1.1) has been moved to the spare group meter slot (0x0C) in order to make QPv1.6 more backwardly compatible with existing systems and QCOM v1.5. This is because, in all versions of QCOM to date, SAP wins are already added to the EGM Total Wins Meter (0x02). So by also adding SAP wins to group meter 0x07 would have required changing a number of taxable metered win & RTP formulas used in many existing systems. This minor change avoids this problem.
- Hopper level meter: the EGM default after RAM clear is now $0 (8.1.18).
- TZADJ now affects all timestamps on the EGM except the QCOM event queue (15.4.5)
- All other changes are clarifications. Refer to the changed marked version of this document for details.
- Added 'EGM Period Meters Reset' event. Refer 7.10.3.37.
- Clarified MAXECT threshold is 'less than'

- Clarified ECT transaction log and included PSN in the data (0)

QCOM Version 1.6. Robert Larkin – Released 1st February 2005

Release Notes.

QCOM version 1.6 represents the biggest change to the QCOM protocol since its initial implemented release as version 1.5. All changes to QCOM v1.5 (taking it to v1.5.5) were basically only clarifications, where QCOM v1.6 introduces new elements and methods that need to be programmed into both the EGMs and SCs. Since both v1.5.5 & v1.6 QCOM EGMs will be in use at the same time for an extended period, it was considered important to be able to clearly distinguish between the differences of the two protocol versions in this document. Accordingly requirements throughout this document that have become specific to QCOM v1.5.5 EGMs are labelled with '*QPv1.5'* and all new requirements that are specific to QCOM v1.6 EGMs are labelled '*QPv1.6'.*

QCOM v1.5.5 – v1.6 Backward Compatibility Notes

- Third party QCOM monitoring software (such as player loyalty & financial analysis software) that monitor the EGM Configuration & Meter Group/Contribution Responses should still work without requiring a software upgrade.*
- Third party LP displays that monitor LP broadcasts should also still work without requiring a software upgrade.*

* The above points assume that the value added services alluded to were initially correctly programmed, eg they masked out all reserved fields and did not assume a specific message length for any message types.

- A QPv1.6 EGM will not work on a QPv1.5 only SC (the SC must be upgraded first)
- QPv1.5 EGMs will be supported indefinitely by QPv1.6 SCs
- A SC can send any QPv1.6 poll to a QPv1.5 EGM with no side effects. (Poll message data area extension has been thoroughly tested on all QPv1.5 EGMs. I.e. The QPv1.5 EGM will only process data in the poll up to the limit of its knowledge of QPv1.5 and no further.)

Summary of changes made since QCOM v1.6 draft 1 released 20/10/2004:

- Clarified use of hoppers with weight sensors.
- Hopper level mismatch is now a fault condition (7.10.1.5).
- Added section on general requirements for the display of text data (3.6.3).
- Made PSN fields reported back by the EGM consistent (15.6.10 & 16.1.2).
- Added requirement for ECT audit trial (0)
- Site Details - removed references to commas (15.5.5)
- Display feature of External Jackpot Information Poll is now mandatory (15.4.7)
- Added Note Acceptor Yo-Yo Event (7.10.1.24)
- Added EGM Top NP Prize Hit Event (7.10.3.35)
- Added 'continue bit' and 'Reset Key Disable bit' to System Lockup Request Poll (15.4.9) and also changed pre-system lockup display requirement from displaying the TEXT field, to displaying "Lockup pending".
- Added "End of Day Time" EODT field to EGM Parameters Poll (15.4.5)
- Added EGM total games won meter to group 0 meters (12.1.1)
- Adjusted defn of new 'games won' meter in multi-game meters response (15.6.6) to include gamble result.
- New more generic method of operation of the Cancel Credit Lockup Request Poll (0) and renamed the poll to the 'Cash-Out Request Poll'.
- Clarified operation of the RCRF Cancel Credit Lockup when the EGM has a Ticket Printer (3.5.3).
- Clarified role of ticket printers in 'ECT from EGM' (16.3)
- A failed ECT-from-EGM now displays a failure message and will now result in a normal collect procedure instead of always being a CC (16.4)
- Added an 'On Failure Flag' to the 'Ticket Out Request Acknowledgement Poll' (15.4.10).

- Added ability for an attendant to print cash out tickets. (22.1.2)
- Added display of 2nd date and time to ticket out details (22.3)
- Added failure code to Ticket In Ack Poll (Refer 22.2 & 15.4.11)
- Made sure the AUTHNO field can be easily extended if required later. (Refer all use of "AUTHNO")
- Clarified LP Award handling with respect to LP award win shows (10.6.1).
- Large win lockups are now for verification purposes only (15.4.5)
- EGM Game Configuration Change Poll (15.4.4) – went back to QPv1.5 method of changes taking effect only upon return to idle mode to avoid problems caused by delaying until start of next play.
- Relaxed inter-char timing for EGMs using FIFOs (14.1.15) – removed, not needed.
- TZADJ converted to minutes and clarified display of Sytem Time (15.4.5)
- Removed EGM serial number from all TITO events. Not required as poll address denotes the EGM's serial number.
- Added AUTH ID to EGM Cash Ticket Out Print Failure Event (7.10.1.32)
- Added EGM Invalid Ticket Acknowledgement Event (7.10.3.36)
- Clarified maxmimum message display rate and minimum message display width for SPAM (15.4.20) & GPM (15.5.3)
- Removed game denomination support and stayed with existing method. Denomination Hot-switching is facilitated via the EGM Configuration Poll (15.4.2). Refer section 3.1.3.1 on Denomination Hot-Switching.
- Added MAXLINES to the EGM Configuration Poll (15.4.2).
- Added total SAP RTP to EGM SAP Configuration Changed Event (7.10.3.34)
- Removed STD DEV field from Game Configuration Response (15.6.11)
- External Jackpot Information Poll (15.4.7). Moved PGID field out of repeated entry section.
- Top NP Prize Hit Event (7.10.3.35) now also is applicable to progressive games.
- Added STATE field to the EGM General Reset Poll (15.4.19).
- Reduced allowable Poll Address range (15.5.4)
- Added propsed ticket out format to section 22.3
- Note Acceptor Status Response (15.6.2) – made more generic by using descriptor string.
- Added Win Payout Thresholds(NPWINP & SAPWINP) to the EGM Parameters Poll (15.4.5)
- LP Award Ack Poll (15.4.18) removed new fields (problem is now addressed by SC broadcasting PA's more often. This also keeps QCOM more backwardly compatible with existing Jackpot Displays)
- Clarified EGM Excessive Note/Ticket Acceptor Rejects Fault (7.10.1.23) re ticket In.

Summary of all protocol Changes from QCOM Version 1.5.5 to Version 1.6

1. **Data Link Layer (DLL) Refer section 14.**
   EGMs must now be able to sustain a poll cycle period down to 250msecs if required. (14.1.3)
   *Added a protocol version number (QPV) to the DLL response control field to help a SC immediately work out which version protocol EGM it is communicating with (14.3.1).
   Option 2 poll processing method is now mandatory. This has no effect as all EGMs process messages this way already (14.1.1)
   Broadcast NAK bit (14.3.1). Made this easier to implement for EGMs, the original implementation implied setting the bit on the very next response could be difficult.
   Increased maximum message length to be supported by EGMs to 257 bytes. (14.2)
   Fixed possibility of EGM getting stuck in limbo if it is assigned a poll address but is never polled on that address (refer 6.1.2). I.e. a QPv1.5 EGM would never go back into communications timeout until it was either polled on the assigned address at least once, or powered down.
   A Communications Fault now ends a session (6.1.3).
   *EGM now ignores the poll message data if the poll contains a NAK (14.2.1). (Helps prevent duplicated processing of polls. SCs must be careful to note this new behaviour especially the first poll in a session)
   *The EGM no longer has to store ACK/NAK bit in NV-RAM (Refer 14.2.1).
   *EGM no longer applies the next session's ACK/NAK to an event response when it was last response in a session (8.3.4).
2. ***New & Updated Events: (7.10)**
   Minimum primary event queue size is now 200 (was 100) (Ref: 7.2.1).

- EGM Power Down Incomplete 7.10.1.34
- EGM Bonus Device Fault (optional) 7.10.1.18
- LP (7.10.2.1) & SAP Award events (7.10.4.8). Extra mask now required on PLVL field
- Numbered version of the 'SAP Award event' - decommissioned (7.10.2.2)
- Unnumbered SAP Award event created (7.10.4.8)
- A manufacturer specific fault extended event with 16 character reason text field. (7.10.1.16)
- Processor Over-Temperature (7.10.3.24)
- Cooling Fan Failure (7.10.3.25)
- Call Service Technician (7.10.3.26)
- Mechanical Meter door open/close (7.10.3.12)
- Mechanical Meter door power off accessed (7.10.3.15)
- Added Ticket Paper Low event (7.10.3.20)
- Added "Note Stacker High Level Warning" event (7.10.3.19)
- Added Note Acceptor Yo-Yo Event (7.10.1.24)
- Added 0x2032 "EGM Note Stacker Full" notification only event (7.10.3.18)
- Added Top Box/Aux door open/close events (7.10.3.12)
- New ECT from EGM event (re change from BCD to hex) (16.3.3)
- New System Lockup User Response Event. (7.10.4.7)
- Updated Note Acceptor Stacker Cleared event. (More positive method of logging this event) (7.10.3.14)
- New RCRF Cancel Credit Lock Up Event. Field is now 4 bytes. (7.10.4.10)
- Updated Communications Timeout Event to avoid superfluous re limbo fix (7.10.3.6)
- Clarified EGM Power Down event (7.10.3.11). Modified requirement for power down event logging when the EGM is using its RTC. This fixes a potential problem with event time stamping.
- EGM Recoverable RAM Corruption (EC:0x204E) (7.10.3.29)
- RTC Refreshed Event (7.10.3.8). Clarified which time to use in the event data.
- Auxiliary Display Device Failure (7.10.1.35)
- Added Denomination Enabled event (7.10.3.31)
- Added Invalid Denomination event (7.10.3.32)
- Added Hopper Calibration Event (7.10.3.33)
- Added EGM Hopper Level Mismatch Event (7.10.1.5)
- EGM SAP Configuration Changed (7.10.3.34)
- New PID session started event (7.10.3.30)
- Added EGM Top NP Prize Hit Event (7.10.3.35)
- Clarified EGM Excessive Note/Ticket Acceptor Rejects Fault (7.10.1.23) re ticket In.

3. ***Updated EGM Configuration Poll. (15.4.2)**
New DEN/TOK fields now in hex and are 4 bytes each (up from 2 bytes).
Clarified hopper-less EGM behaviour wrt TOK denomination.
A range of new parameters supported: MINRTP, MAXRTP, MAXSD, MAXLINES, MAXBET, MAXPWIN, MAXNPWIN & MAXECT for better multi-jurisdiction support.

4. ***Updated EGM Configuration Response (15.6.12)**
Added Mechanical Meter, Aux display & TI device present flags
QPV field is now in hex & QPV changed to 0x01 to equal QPV in 14.3.1.
The reported number of games may now decrease after EGM Configuration is completed.
New DEN & TOK fields, now in hex and 4 bytes.
A range of new parameters reported back here. Refer to EGM Confuguration Poll for details on these parameters.
Added denomination hot-switchin capability flag.

5. ***Updated EGM Game Configuration Poll (15.4.3)**
Due to extended variation support (9.1), this poll is now ignored if EGM Configuration is not completed.
Added Variation Lock.
LP only games must reject poll if told to be SAP.
EGM to ignore poll if PNUM is invalid, simpler.

6. ***Updated EGM Game Configuration Response 15.6.11**
Added LP only flag.

Due to extended variation support (9.1), this response is no longer sent by the EGM if EGM Configuration is not completed.

Increased maximum variations to 16 (up from 8)

PRET field format changed and made hex

Added standard deviation reporting.

Added Variation Lock. (the ability to lock a variation for a game so it cannot be changed unless the EGM is RAM cleared. Called Variation Lock see (15.4.3). Some jurisdictions may require this feature.)

Progressive levels may be LP only.

7. ***ECT. (Section** 16**)**

EGM must no longer support an ECT out on zero credit.

A cashless mode bit indicator has been added to the General Status Response (15.6.1)

ECT xfer amounts are now all in hex allowing larger cashless transfers. This affected the existing ECT to EGM Poll & ECT from EGM event (7.10.4.9).

Created the MAXECT failsafe parameter (15.4.2)

Cashless mode bit not processed any more if PSN is invalid (16.1.1).

Clarified use of ticket printers in ECT.

A failed ECT-from-EGM now displays a failure message and will now result in a normal collect procedure instead of always being a CC (16.4)

8. ***New 'ECT-To-EGM' Poll Acknowledgement Response (16.1.2).**

This allows the SC to more easily detect a 'ECT to EGM' Poll Timeout, especially when zero credits were transferred to the EGM when changing cashless modes.

9. ***Updated General Status Response (15.6.1)**

New door flags for mechanical meter door & top box/auxiliary door.

Added cashless mode indicator bit.

Added Ticket In / Out states.

Added dedicated double up/gamble states

10. **Return to Idle mode. Section 3.3.**

Order of requirements now mandatory & changed order wart QPv1.5. This will make EGM behaviour on QCOM more predictable.

11. **Audit Mode. Refer section 4**

Some previously optional audit mode requirements now mandatory. Some new requirements. In fact all implemented poll/response data must now be displayed.

EGM must display last seed and hash (4.2.10).

Must show each game's configuration status (4.1)

Display 'queued cashless in' meter (16.1.1)

Display the secondary event queue (4.2.13)

Increased the frequency in which an EGM must perform a self audit. 12.4

LP, added display of last jackpot hit time (Section 10.3)

12. **Large Win Lockups**

Large Win Lockups on QCOM EGMs must never initiate a payment for the win amount (Refer 15.4.5).

13. ***New SAP Configuration Response (15.6.4).**

This allows a SC to always work out the exact hit rate and %RTP of a progressive game.

14. ***New SAP Configuration Change Poll (15.4.6)**

15. ***Updated Progressive Meters Response (15.6.3)**

Added SIZ field & some spare fields for future expansion

Added hits, wins meters.

Added hit rate info.

EGM now only has to be able to queue one response of this type at a time.

16. ***Updated Multi-game/Variation Meters Response (15.6.6)**

Added total games won meter.

Only one response of this type can be queued at a time.

17. ***Extended Multi-game/Variation support (section 9)**

An EGM may now have up to 65535 games, but only a maximum of 255 games may be offered, based on EGM configuration Poll settings.

Each game may now have up to 99 variations, but only 16 may be offered to choose from based on EGM Configuration Poll settings.

18. **Variation Hot Switching. Section 15.4.4.**

Hot-switching support is now mandatory for multi-variation games except progressives. (9.2)

19. ***Updated EGM Variation Change Poll 15.4.4.**
Changed the name of this poll to "**EGM Game Configuration Change Poll**".
Switching VARs must not affect the current game display.
20. **System Lockup Poll. Section 15.4.9**
Must now display "SYSTEM LOCKUP" in lockup condition.
System Lockup messages must be displayed constantly by the EGM.
Added 'Reset Key Disable' Bit
Added option to "press to continue"
Added option to ask the user a question.
Added option to turn on all lamps while in the lockup (this is an optional feature for EGMs).
A special System Lockup text message is displayed if the SL is delayed
21. **SPAM Poll (15.4.20)**
*Increased maximum text length from 40 to 80 bytes.
Clarified maxmimum message display rate and minimum message display width.
22. **RAM Clear Defaults. Refer Section 8.1.6**
Default date and time is now 1/1/2000.
23. **Power up procedures (8.2).**
Procedure order is now mandatory to standardise behaviour between EGM brands.
24. **CRanE Configurable Random Events (CRanE's). Section 16.5.1.1**
Renamed and changed some on screen messaging.
Clarified required RNG cycle.
25. ***Updated EGM Parameter Poll 15.4.5**
Added Reserve Feature enable/disable flag.
Added an Auto play Flag.
OPR, field is now a hex field (was BCD)
Added toggle CRLIMIT mode flag. CRLIMIT now has two modes of operation.
Clarified that changes must now take effect straight away if in idle mode. This is probably already the case with all EGMs.
Add Time Zone Adjust TZADJ.
Added PWRTIME that adjusts the EGM power-save timeout value.
Added PID version select.
Added "End of Day Time" EODT field.
Added Win Payout Thresholds (NPWINP & SAPWINP)
26. **EGM General Reset Poll (15.4.19)**
Added STATE field.
27. ***General Promotional Message 15.5.3**
Added chime flag.
Clarified maxmimum message display rate and minimum message display width.
28. **Cancel Credit Lockup Request Poll (0).**
Renamed to 'Cash-Out Request Poll'
Is now made more generic by being just like pressing the collect button on the EGM
29. ***New/Updated Group Meters (12.1)**
Total Rejected Notes (refer 12.1.2). (Allows tracking note acceptor performance)
PID Accessed Meter (12.1.1)
Cash Ticket Out Meter (12.1.1)
Clarified SAP wins meter (12.1.1)
Added EGM total games won meter to group 0 meters (12.1.1)
To allow monitoring of gamble, the following meters were added:
Gamble Turnover (12.1.2)
Gamble Wins (12.1.2)
30. **Clarified Note Acceptor Maintenance Poll 0.**
Must now be the only way to enable/disable individual notes on the EGM.
31. ***New Note Acceptor Status Response (15.6.2)**
32. ***Updated Program Signature Poll/Response (15.4.8)**
Signature field is up from 8 bytes to 20 bytes to handle SHA signatures.
33. ***Updated EGM Ganeral Maintenance Poll 15.4.13.**
Combined LP & SAP meter requests into a single progressive meter request. Makes life a bit simpler for EGMs & SCs
Added SAP Configuration Response request flag.
Added Note Acceptor Status Response request flag.

Added Bet Meters and Player Choice Meters request flags.
34. **Communications Disabling Conditions (CDCs)**
    Added "LP Broadcast Tinmeout CDC (6.1.8)
35. ***Broadcast Message 15.5.1**
    Added a clock display to players and the ability to toggle it on or off.
36. **Site Details 15.5.5**
    Increased max length of LTEXT field to allow full address & contact details.
    SC may use 2 consecutive space characters in LTEXT to indicate appropriate spots for new lines when displaying this field.
37. ***Updated Purge Ack Response. Refer 15.6.10**
    Added current Purge Poll PSN to Purge Ack. response.
    The response is now queued regardless of whether the purge PSN is valid. This fixes a potential problem with event purging in the case when a purge poll is sent and the EGM stops responding. If the EGM still didn't respond to the purge poll when resent later on, then it wasn't possible to tell if the EGM ever got the poll or whether its purge poll PSN was incorrect.
38. ***Hopper/Ticket Printer Maintenance Poll (15.4.17).**
    Added the ability for the SC to record a refill on the EGM on demand.
    Ability to request EGM to print out a test ticket.
    Hopper pay-outs must now be aborted if interrupted (3.4).
39. **Player Information Display (PID) Support**
    Clarified PID is a part of idle mode
    *Accessed Meter (12.1.1)
    *Version selection & enable/disable via EGM Parameters Poll (15.4.5)
    *New session started event (7.10.3.30)
40. ***Added the 'External Jackpot Information Poll' (15.4.7)**
41. ***Bet Meters Response (15.6.5)**
    This response allows the collection of EGM line/credit bet per line statistics on a **per game** basis.
42. **Added Section 18 on downloadable software** (first published 6/11/2001 in V2 draft 2)**.**
43. **Added Section 19 on 'EGM to EGM Communications'**
44. **Added new Section 20 'QCOM Command Prompt'**
45. **Added new Section 21'Finding EGMs on the LAN'**
46. **Ticket In / Out Support 22**
    Added "Ticket Paper Low" advisory event. (7.10.3.20)
    Added "EGM Cash Ticket Out Print Successful" event (7.10.4.12)
    Added "EGM Cash Ticket Out Print Failure" advisory event (7.10.1.32)
    Added "Ticket Out Request" unnumbered event (7.10.4.11)
    Added "Ticket Out Acknowledgement Poll" (15.4.10)
    Added "Cash Ticket In Meter" to group 0 meters (12.1.1)
    General Status Response - added TI/TO states (15.6.1)
    "EGM Configuration Response" - TI device present indicator (15.6.12)
    Added "Ticket In Request" unnumbered event (7.10.4.13)
    Added "Ticket In Acknowledgement Poll" (15.4.11)
47. ***Added Player Choice Meters Response 15.6.7**
48. **Added section on Denomination Support and Control (3.1.3)**
    *Added support for denomination hot switching.
    Clarified support for fractional game denomination.
    Clarified support for "Player Selectable Denomination" games.
49. ***Added support for hoppers with weight sensors**
    Refer 3.4, 7.10.3.33 & 7.10.1.5.

* Internal Use. Indicates changes that affect existing QLD Site Controller software.

Version 1.5.5 Checklist Version A 19/12/2001 Robert Larkin

Converted Protocol document to Microsoft Word. Some sections have been renumbered and moved as a result
Clarified Global NAKs
Method 2 response methodology is now mandatory
Clarified new program signature timeout time
Add option to auto-pay, auto-clear SAPs.

Progressive Meters Response and Game Configuration Poll SAP contribution meter units changed.

Version 1.5.3 Checklist Version E   12 May, 2000 R. Larkin

Clarification release. No programming changes should be required. (If there are any document changes requiring EGM or SC software changes then they are not required to be initiated until specified the next release of this document.)
Be advised. All "prefer..." words in redline text will become mandatory in the next release of this document.
Created Appendix A
Added Appendix B
Label changes:
        MGEF is now GEF
        GEF is now MEF
        Var. Cfg. Poll is now the Var Change poll.
Refer red-line and strike-out throughout the document for other changes.

Version 1.5.2D 26 October, 1999 R. Larkin

Updated the protocol checklist. Refer red-line & strike-out
Changed copyright notice.
No changes to protocol.

Version 1.5.2 16 October, 1998 R. Larkin

Added QCOM EGM checklist refer endnotes
Clarified display of multi-game disable in Section 6.2
Other changes to the document are minor clarifications, the main reason for the new version is the checklist..
Refer red-line and strike-out throughout the document.

Any changes requiring EGM program changes are to be made at the EGM manufacturer's earliest convenience, but no longer then 3 months from the date of this document.

Version 1.5.1 4 September, 1998 R. Larkin

Section 4.4. Added a couple of extra audit formulas so more group meters are validated.
Added "Configuration Required" Communications Disabling Condition to Section 6
Section 16. ECT. Added requirement for display of residual credit.
Clarified poll message data checks the EGM must perform.
Added more paragraph numbers.

Version 1.5 19 February, 1998, 1997 R. Larkin.

Updated copyright notice
Protocol changes:

Section 7 Put a maximum size on general event queue
Removed full duplex option.
Added RCRF turnover and RCRF win meters to meter group 1.
Deleted the Meter Group Response, superfluous wrt Meter Contribution Response
Meter Contribution Response: Changed response priority and function code

Version 1.4 17 September, 1997 by R. Larkin.

Most changes are simply minor clarifications.
Other changes are to make implementation of the protocol easier for the SC.

Refer red-line and strike-out throughout the document. A summary of the changes follows:

Incorporated Ver 1.3 Amendment 1 into the document
Section 15.7.1, removed state 0 idle/reserved
Section 17.5 DLSA response, added bit 6 to indicate if DLSA award is current
New EGM Events:
        Invalid EGM Configuration
        Invalid Game Configuration
Added a Purge Event poll Acknowledgement response
Changed and clarified the Meter Contribution Response (again)
Changed units of SAP current amount to contribution (turnover) since last jackpot. This is a more sensible unit to use. Two sections are affected:
        15.7.3 Progressive Meters Response and
        15.5.3 EGM Game Configuration Poll
Poll Address Configuration must now also be possible after a 10 second communication time-out. Refer Sections 16.6.1.3 and 6.1.1

Version 1.3 31 July, 1997 by R. Larkin.

Refer red-line and strike-out throughout the document. A summary of the changes follows:

MID field was incorrectly stated as "hex" instead of "BCD" in some places
Changed "Executive Director" to "Chief Executive"
Clarified hot-switching of progressive games (Section 9.2)
Clarified section 5.0.8 on the inter-character time-out
Added Figure 1. explaining message processing options
Added to section 5.0.16
Clarifications in sections 6 & 7
Added meters to section 10.3 - progressive audit meters
Simplified the general maintenance poll. It is now a fixed length poll which acts on a single game only.
"Cash Ticket Printed" had the wrong event code.
Clarified event 0x2001
Added optional new event "Cancel Credit Cancelled" (0x2002)
Added Section 13.2.5 baud rate tolerances
Added GEF flag to EGM Configuration Poll Section 15.5.1
Added section *2.3.8* defining ASCII printable characters
Section 15.6.1.3 Poll Address Configuration. Now can also change an EGMs poll address (ie. prevents an EGM from being RAM cleared just because you move an EGM from one SC to another).
Clarified Section 14. "When a response is rebuilt due to a NAK, it must reflect the latest EGM data (except for events which are a queue and are resent until acknowledged)."

Version 1.2 Initial Release 7 March, 1997 by R. Larkin. Checked by M. Cavalchini

Changes and clarifications have been extensive as the document was still under development. Please read the whole document carefully. A summary of the changes follows. From this point on, all future changes and clarifications will be thoroughly detailed in this section.

Added broadcast message negative acknowledgement to General Response Message Format
Rearranged EGM Meter Group meters and added Total Coins in/out meters for completeness
Appendix A - SC Operating procedures is now a separate document for operators.
Added Section 3.2 on EGM RAM Errors
Clarified section on Event Queue Integrity
Renumbered (spread out) event codes & function codes
Clarified section 8.1.5 and added MID & SER to EGM NV-RAM cleared event
Clarified display of specific and promotional text messages.
Created Section on EGM CPU Load Considerations.
Interface requirements is now a separate document.
Changed the Hopper Maintenance functions and created a section on Hopper Maintenance.
Meter Group Response. Extended. Changed group 1 meter I.D. nos. Changed bit defs in MGRP field.
Added EGM Variation Configuration Poll
Added double-up parameters to EGM parameter poll

Program Signature Request Poll. SEED field is now optional.
Added additional default response states as necessary
Added Note Accepted event.
Re-worked progressive sections, as a game may have both a SAP & LP component
Added new chapter on DLSA
Added EGM Ticket Printer support
Added progressive win meter to multi-game meters
GVN field is now a hex field instead of BCD
Added Temporary Event queue full event
Deleted GVN & VAR from meter contribution response and put them in the EGM configuration Response
Added New Game Selected Event
Increased temporary event queue to 15

Version 1.1 DRAFT (based on 1.0) 26 February, 1997. R.Larkin

General Status Response. Inserted Large Win Lockup State code
Event Response. Clarified large win lockup description.
Last Game Display. Added Date and time display.
Added sections on UART Information & Flow Control.
Physical Layer, improved interface.
Specific Promotional/Advisory Messages, added sound option.
EGM Parameters, removed MAXCR, redundant, lockup is system controlled.
Changed term "game disabled" to "play disabled".
Program Signatures. Clarified PSA section and removed PSA offset from poll.
Configuration. Clarified restoring serial no. from EEPROM upon RAM clear.
Added new section explaining Multi-game support in more detail.
Clarified EGM RTC usage and added "RTC Refreshed Event" and present bit.
Clearly defined a "play".
Completely reworked all multi-game/variation support.
Event code size increased to 2 bytes for future expansion.
Added more paragraph numbering in sections 3-14
Added full duplex option.
Further developed appendix A
Reduced EGMs/LAN from 64 to 32 as per RS-485 specs.

Version 1.0 DRAFT (based on REEF Protocol Version 1.7) 11 November 1996 by R. Larkin.
Actual first release date 10th Dec 1996

Designed: 12 July, 1996 Robert Larkin